

# Minimization of Information Loss through Neural Network Learning

M. D. Plumbley\*

## Abstract

In this article, we explore the concept of *minimization of information loss* (MIL) as a target for neural network learning. We relate MIL to supervised and unsupervised learning procedures such as the Bayesian maximum a-posteriori (MAP) discriminator, minimization of distortion measures such as mean squared error (MSE) and cross-entropy (CE), and principal component analysis (PCA). To deal with unsupervised systems where complex noise is present, we introduce the idea of the signal  $\Omega$  being *well-mixed* with the noise  $\Xi$ . If this holds, minimizing information loss about the pair  $(\Omega, Xi)$  will proportionately minimise information loss about  $\Omega$  itself. This situation may hold in early processing stages of complex sensory systems such as the retina in higher mammals.

## 1 Introduction

In recent years, a number of authors have used concepts from Information Theory to develop or explain neural network learning algorithms, particularly in sensory systems [1, 2, 3, 4, 5]. A neural network in a sensory system is thought of as part of a *communication system*, transmitting Shannon Information [6] about the outside world to further processing stages. The

---

\*Centre for Neural Networks, Department of Mathematics, King's College London, Strand, London, WC2R 2LS, UK

information-transmitting capability of such a neural network is limited both by constraints, such as the number of available units in a particular layer, and by costs, such as the average power used to transmit the information.

In this article, we explore the concept of *minimization of information loss* (MIL) [2] as a target for neural network learning in this context. MIL is closely related to Linsker’s *Infomax* principle [1]. By relating MIL to more familiar supervised and unsupervised learning procedures such as Error Back Propagation (‘BackProp’) [7] and principal component analysis (PCA) [1], we show how it can be used as a *lingua franca* for learning in all stages of a neural network sensory system.

## 2 Mutual Information and Information Loss

### 2.1 Entropy and Information

Central to Information Theory are the concepts of *entropy*  $H$  and *mutual information*  $I$  [6]. Generally speaking, the entropy  $H$  measures the amount of *uncertainty* that exists in a particular variable: in other words, how much there is that we do not yet know about the state of that particular variable. The mutual information  $I$  between two variables measures how much we reduce the entropy  $H$  of one variable, hence reducing its uncertainty, if we know the value of the other. It turns out that  $I$  is symmetrical with respect to the two variables, hence the name *mutual* information, i.e. the information shared between the two variables.

To formalize this approach, consider a discrete random variable (RV)  $X$  which takes values  $x \in \mathcal{X}$  with probabilities  $P_X(x)$ . Where clear from context, we shall simply write  $P(x)$ . The entropy  $H(X)$  of this RV is given by

$$H(X) = - \sum_{x \in \mathcal{X}} P(x) \log P(x) = -E(\log P(X)) \quad (1)$$

which is zero if  $P(x^*) = 1$  for some  $x^* \in \mathcal{X}$  or strictly positive otherwise, with  $H(X) \leq |\mathcal{X}|$ . If the logarithm in (1) is taken to base 2, the resulting entropy is in *bits*. Unless otherwise indicated, we shall normally take this to

the base  $e$ , to make the mathematics easier: the only difference is a scaling factor on the result. In a similar way, we can define *joint entropy*

$$H(X, Y) = -E(\log P(X, Y)) \quad (2)$$

and *conditional entropy*

$$H(X|Y) = -E(\log P(X|Y)) = H(X, Y) - H(Y) \quad (3)$$

which is the entropy left in  $X$  if we observe  $Y$ . If  $X$  and  $Y$  are independent,  $H(X|Y) = H(X)$ , since observing  $Y$  tells us nothing about  $X$ . From these, the mutual information between  $X$  and  $Y$  is defined to be

$$I(X, Y) = H(X) - H(X|Y) = E\left(\log \frac{P(X, Y)}{P(X)P(Y)}\right) \quad (4)$$

which is zero if  $X$  and  $Y$  are independent (i.e.  $P(x, y) = P(x)P(y)$ ) and strictly positive otherwise, with an upper limit of  $\min(H(X), H(Y))$ .

In the case of continuous random variables with probability density functions  $p(\cdot)$ , a similar entropy measure can be defined

$$H(X) = -\int_{x \in \mathcal{X}} p(x) \log p(x) dx = -E(\log p(X)) \quad (5)$$

although this is no longer guaranteed to be positive, in contrast with the discrete  $H(X)$  defined in (1). However, the continuous version of mutual information

$$I(X, Y) = E\left(\log \frac{p(X, Y)}{p(X)p(Y)}\right) \quad (6)$$

is still positive, or zero for independent  $X$  and  $Y$ , but with no upper limit.

## 2.2 Information Loss

Now we have a concept of information, we can explore what happens to this information when we process it with a neural network. Suppose that we are interested in the value of some random variable  $\Omega$ . However all we have access to at the input to our neural network is some sensory observation  $X$  which is derived from the random variable  $\Omega$ , and some other random variable  $\Phi$  which we are not interested in, i.e.

$$X = f_{\text{world}}(\Omega, \Phi). \quad (7)$$

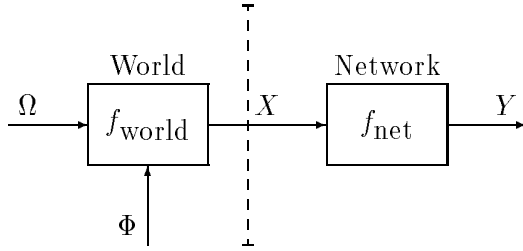


Figure 1: ‘Signal’ RV  $\Omega$  and ‘noise’ RV  $\Phi$  transformed by the world into the sensory input  $X = f_{\text{world}}(\Omega, \Phi)$  presented to the network. This is then transformed by the network into the network output  $Y = f_{\text{net}}(X)$ .

The network itself will transform that sensory input  $X$  into its output, (Fig. 1)

$$Y = f_{\text{net}}(X). \quad (8)$$

We are interested in what happens to the information  $I$  about  $\Omega$  as the signal passes through this network. At the input to the network, we have an amount of information  $I(X, \Omega)$  about  $\Omega$ , while at the output of the network, we have information  $I(Y, \Omega) = I(f_{\text{net}}(X), \Omega)$  about  $\Omega$ . We therefore define the *information loss* [2]  $\Delta I_{\Omega}(X, Y)$  about  $\Omega$  across this transform to be the difference

$$\Delta I_{\Omega}(X, Y) = I(X, \Omega) - I(Y, \Omega). \quad (9)$$

This information loss quantity turns out to have a number of useful properties. For example, it is *additive* across a chain of transformations (networks), i.e.

$$\Delta I_{\Omega}(X_1, X_3) = \Delta I_{\Omega}(X_1, X_2) + \Delta I_{\Omega}(X_2, X_3) \quad (10)$$

which can be verified by substituting the definition (9) into both sides of the equation.

It can also be shown that  $I(f(X), \omega) \leq I(X, \Omega)$  for any transformation  $f(\cdot)$ , with equality if  $f(\cdot)$  has an inverse, i.e. the transformation is reversible [8]. Therefore  $I(Y, \Omega) \leq I(X, \Omega)$  for any  $f_{\text{net}}(\cdot)$ , so information loss  $\Delta I_{\Omega}(X, Y)$  is always *positive* (or zero) across any noiseless transformation.

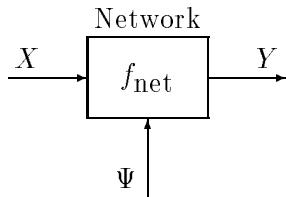


Figure 2: Network performing a noisy transformation from  $X$  to  $Y = f_{\text{net}}(X, \Psi)$ .

In fact, the positivity result above can be extended to transformations which include extra independent noise. Consider the transform

$$Y = f_{\text{net}}(X, \Psi) \quad (11)$$

for some network function  $f_{\text{net}}(\cdot, \cdot)$  of the input  $X$  and independent noise  $\Psi$  (Fig. 2). Since  $\Psi$  is independent of  $(\Omega, X)$ , it can be shown that  $I(X, \Omega) = I((X, \Psi), \Omega)$ , i.e. we get the same information about  $\Omega$  whether we include  $\Psi$  or not. Therefore substituting this into (9), the information loss about  $\Omega$  across this system is

$$\Delta I_{\Omega}(X, Y) = I((X, \Psi), \Omega) - I(f_{\text{net}}(X, \Psi), \Omega) \quad (12)$$

which again must be positive. Note that this property does not depend on the type of noise (additive, multiplicative, etc.), only that it is independent<sup>1</sup>.

To summarize, information loss is *additive* across any chain of transforms, and *positive* across any noiseless or noisy transform. If we lose information about  $\Omega$  at any stage, it is impossible to regain it without input from another information source. Consequently, if we want to retain the maximum amount of information at the output of our chain of networks, i.e. minimize the information loss across the whole system, a reasonable strategy to adopt would be the minimization of information loss (MIL) across each stage of the chain independently.

Of course, this is a *greedy* approach to the optimization problem of minimizing information loss across the whole system, and there is a chance that

---

<sup>1</sup>In a previous paper [2] this result was obtained for additive noise.

an optimization early on in a chain will upset its operation later on, making a sub-optimal solution. However, it does give a reasonable starting point, and the chance to use unsupervised learning to optimize early stages of a system without regard to the eventual output of a subsequent supervised learning stage.

MIL is closely related to Linsker’s *Infomax* principle [1], which states that the mutual information  $I(Y, X)$  transmitted across the network should be maximized. The difference is that MIL is concerned with information about something which is of interest ( $\Omega$ ) and what happens to that information across the network, while Infomax is concerned with the information capacity of the network itself. In practice, the difference between the two is primarily a technical one for unsupervised learning. Since  $\Omega$  and  $X$  are independent of changes to  $f_{\text{net}}(\cdot)$ , MIL is equivalent to maximizing  $I(Y, \Omega)$ . Consequently if we consider  $\Omega$  to be the effective input to the network for the purposes of Infomax, the two become equivalent.

For supervised learning, the emphasis on the variable of interest ( $\Omega$ ) means that we can use this directly as our desired target output. This allows us to relate MIL to more familiar supervised learning approaches such as minimizing the probability of error and Error Back Propagation.

### 3 Minimizing the Probability of Error

Suppose that the variable  $\Omega$  we are interested in has one of two states, 0 or 1, and we would like our network to predict this value on the basis of its input  $X$  and produce this estimate as its output  $Y$ . During training, i.e. while we are adjusting our network function  $f_{\text{net}}(\cdot)$ , we have access to  $\Omega$  as a *target* signal: this is *supervised* learning.

A traditional approach to this problem is to build a model for the class conditional probability densities  $p(x|\omega)$  and accumulate the *a priori* class probabilities  $P(\omega)$ . From an input value  $x$  we can use Bayes theorem to

find the class  $\hat{\omega}(x)$  which has maximum *a posteriori* (MAP) probability

$$\hat{\omega}(x) = \arg \max_{\omega} P(\omega|x) \quad (13)$$

where  $P(\omega|x) = p(x|\omega)P(\omega)/p(x)$ . The factor  $p(x) = \sum_{\omega} p(x|\omega)P(\omega)$  can be ignored in the maximization process, since it is equal for all  $\omega$ . We then choose  $\hat{\omega}$  as our output [9], so our network function is given by  $f_{\text{net}}(\cdot) = \hat{\omega}(\cdot)$ . If  $\omega$  can only be one of two values, the chosen value  $\hat{\omega}(x)$  will always have  $P(\hat{\omega}(x)|x) \geq 0.5$ . By consistently choosing the most probable class, the probability of error is minimized. Under certain circumstances, neural networks trained with BackProp will attempt to approximate this approach [10].

Now, since  $\hat{\omega}$  consistently assigns the most probable class to  $y$ , the probability distribution of  $\Omega$  for a particular value of  $y$ ,  $P(\Omega|Y = y)$ , will be as extreme as possible. Any other assignment scheme will lead to a probability distribution which is more balanced. Therefore the entropy  $H(\Omega|Y = y)$  must be minimized, since a more balanced probability distribution would increase  $H$  [11]. Consequently the conditional entropy

$$H(\Omega|Y) = \sum_y H(\Omega|Y = y)P(Y = y) \quad (14)$$

will be minimized, so the mutual information

$$I(Y, \Omega) = H(\Omega) - H(\Omega|Y) \quad (15)$$

will be maximized. The information loss (9) will therefore be minimized, since both  $\Omega$  and  $X$  are unaffected by our choice of  $f_{\text{net}}$ . So, by choosing  $f_{\text{net}}$  to be a Bayesian MAP classifier, and consequently minimize the probability of error, we have also minimized the information loss.

Of course, in the Bayesian classifier, a lot depends on the accuracy of the estimate of the probability function in (13). In the preceding analysis, we have assumed that the probability can be estimated very accurately. In practice, this accuracy may be limited by the number of parameters in our system (the number of weights in a network, for example) which must be small enough to avoid generalization problems, and the limited amount of

data available to train the system. However, even when the system is not able to be a perfect Bayesian classifier, minimizing probability of error will minimize an upper bound on information loss in a two-class system, with best fit when the errors  $((\Omega - Y) \bmod 2)$  are independent of the chosen class  $Y$  [2].

The case of minimization of error with  $N$  classes has also previously been investigated. In this case, simply minimizing the probability of error will still minimize an upper bound on information loss, but the bound is increasingly ‘loose’ for large  $N$ . There may be a mismatch of up to  $\log_2(N - 1)$  bits between the upper bound and the true information loss, depending on the type and distribution of errors [12]. This brings into question the effectiveness of systems which attempt to simply minimize error where a large number of training classes are used, for example 60 or more in a speech phoneme recognition system [13].

## 4 Minimizing Distortion through Supervised Learning

Rather than attempting to be a Bayesian classifier, most neural network models are trained to minimize some mean distortion measure

$$\epsilon = E(D(\Omega, Y)) \quad (16)$$

between the network output  $Y$  and the desired output  $\Omega$ . In the Bayesian case considered in the previous section we would have  $D(\Omega, Y)$  equal to 0 if  $Y = \Omega$  and 1 otherwise. However, this is not differentiable, so we cannot use it directly in the Error Back Propagation algorithm, which relies on the ability to differentiate  $\epsilon$  with respect to all parameters in the network [7].

### 4.1 Mean Squared Error

Probably the most popular distortion measure used is the *mean squared error* (MSE)

$$D_{\text{MSE}}(\omega, y) = |\omega - y|^2. \quad (17)$$

Minimizing MSE will minimize an upper bound on information loss, but the bound is not particularly tight [12]. There are a number of terms which have to be well-behaved for a tight bound:

- the errors ( $\Omega - Y$ ) on the outputs should be *independent of the output*  $Y$ ;
- the errors on each of the network outputs should be *independent of each other*;
- the errors on each of the network outputs should have the *same variance* as each other; and
- the errors should have a *Gaussian distribution*.

This suggests that to get minimization of MSE to effectively minimize information loss, the network should be adjusted to avoid multiple outputs, which cause correlated error terms, and oversized outputs, which will swamp the errors from other outputs and ‘grab’ all of the error  $\epsilon$ .

## 4.2 Cross-Entropy Distortion Measures

Alternatives to MSE can be used. Solla, Levin and Fleisher [14] suggested a component-wise cross-entropy distortion measure

$$D_{\text{SLF}}(\omega, y) = \sum_i \omega_i \log \frac{\omega_i}{y_i} + (1 - \omega_i) \log \frac{1 - \omega_i}{1 - y_i} \quad (18)$$

where  $\omega_i$  and  $y_i$  are the  $i$ th components of  $\omega$  and  $y$  treated as vectors. This can be used if the  $i$ th component of  $\omega$  and  $y$  represents the presence or absence of some independent feature in  $\omega$ . Each of the components  $\omega_i$  will be either 1 or 0 during training. The same measure can also be used if the outputs are used to classify into one of  $N$  classes, by choosing the largest output. In this case, the target values  $\omega_i$  will not be independent, since one and only one of the target components must be 1 at each stage. For either case, after training, the  $i$ th output  $y_i$  becomes an estimate of the probability  $P(\Omega_i = 1 | X = x)$  that  $\omega_i$  is equal to 1 for a given input value  $x$ .

For the case of independent features, the component-wise cross entropy measure is quite a close match with MIL [12]. The only mismatch arises due to the accuracy with which each output  $y_i$  represents the probability  $P(\omega_i|y_i)$ . These two terms should be identical in a perfect system.

For 1-of- $N$  classification, a closer match can be achieved using a distribution-wise cross entropy measure

$$D_{\text{CE}}(\omega, y) = \sum_i \omega_i \log \frac{\omega_i}{y_i} \quad (19)$$

which must be preceded by a differentiable normalization stage such as Bridle's *Softmax* function [15]

$$y_i = \frac{\exp(u_i)}{\sum_j \exp(u_j)} \quad (20)$$

where  $u_i$  is the output from the penultimate stage of the network, which ensures that the  $y_i$ s are positive and sum to 1, as required for a probability distribution. Again, minimization of the cross-entropy distortion (19) will minimize an upper bound on information loss, with the closest match when the output  $y$  accurately indicates the probability  $P(\omega|y)$  [12].

## 5 Unsupervised Learning and Well-Mixed Signals

In an unsupervised learning system, MIL and Linsker's Infomax are very similar. They show that principal component analysis (PCA) is optimal for a Gaussian signal containing independent equal-variance additive noise passing through a linear dimension-reducing network [1, 2]. The mutual information measure itself can also be used as a Lyapunov function to prove convergence of many PCA algorithms [16]. When corrupting output noise and a power limit is introduced, linear networks are optimal if they produce decorrelated equal-variance outputs. The information measure can also be used to derive learning algorithms to achieve this optimum [4].

However, these unsupervised learning systems have no means of observing the desired variable  $\Omega$ . They only have access to the input  $X$  and output  $Y$ , and so must make certain assumptions about the input arriving

at the system. Normally this is a simple assumption such as one of additive Gaussian noise corrupting the variable  $\Omega$  before it reaches the input. Unfortunately, this is rarely the case in a real system: the ‘noise’, which is simply the part of the input signal which we are not interested in, could have quite a complex relationship with the signal, and may take a very non-linear system to remove or reduce it. So, if the input signal contains a mixture of simple additive noise and noise which is more tightly bound up with the signal  $\Omega$ , we can only hope to reduce the additive noise (at least in a linear network), and the more complex noise will continue to be passed on to the next processing stage.

Another problem for the MIL approach is caused by the disparity between the total amount of information present at the input and that at the output of a typical chain of transforming networks. In many cases, the final objective is to provide a 1-or-0 decision, which can have at most 1 bit of information at the output of our final stage. However, a sensory input can have much more information than this. If we simply reduce the information loss at the input stage, the few bits we *have* lost may have been all the information about  $\Omega$ , and hence all the information about the eventual decision which we wanted to make.

To approach this problem, we shall propose that in some cases, the input is a function of the variable  $\Omega$  which we are interested in (the ‘signal’) and a variable  $\Xi$  which we are not interested in (part of the ‘noise’), with possibly some additional simple noise  $\Phi$ . We shall assume that  $\Omega$  and  $\Xi$  are in some sense *well-mixed* with respect to the transforms we can apply across our system, so that the proportion of information lost about the variable of interest  $\Omega$  is approximately the same as the proportion of information lost about the pair  $(\Omega, \Xi)$ .

To formalize this idea a little, the information loss about the pair  $(\Omega, \Xi)$  is given by

$$\Delta I_{(\Omega, \Xi)}(X, Y) = I(X, (\Omega, \Xi)) - I(Y, (\Omega, \Xi)) \quad (21)$$

with the information loss about  $\Omega$  given by (9) as before. We say that  $\Omega$

is *well-mixed* with  $\Xi$  with respect to the set of transforms  $\{f_{\text{net}}\}$  under consideration, if we have that

$$\frac{\Delta I_{\Omega}(X, Y)}{\Delta I_{(\Omega, \Xi)}(X, Y)} \approx \frac{I(X, \Omega)}{I(X, (\Omega, \Xi))} \quad (22)$$

for all transforms in the set  $\{f_{\text{net}}\}$ .

It is important to note that this concept is not necessarily symmetrical in  $\Omega$  and  $\Xi$ , since the information  $I(X, (\Omega, \Xi))$  is not simply the sum of the information  $I(X, \Omega)$  in  $X$  about  $\Omega$  and the information  $I(X, \Xi)$  in  $X$  about  $\Xi$ , *even if  $\Omega$  and  $\Xi$  are independent*. To see this, consider the case where  $X = (\Omega \text{ xor } \Xi)$ , with  $\Omega$  and  $\Xi$  being evenly distributed over  $\{0, 1\}$ . In this case, both  $I(X, \Omega)$  and  $I(X, \Xi)$  are zero, but  $I(X, (\Omega, \Xi))$  is 1 bit. Instead we find that

$$I(X, (\Omega, \Xi)) = I(X, \Omega) + H(\Xi|\Omega) - H(\Xi|X, \Omega) \quad (23)$$

$$= I(X, \Omega) + I((X, \Xi)|\Omega) \quad (24)$$

where we write  $I((X, \Xi)|\Omega)$  in (24) as we do to reflect the fact that it is symmetrical with respect to  $X$  and  $\Xi$ .

A trivial example occurs where  $\Xi = 0$ . In this case,  $I(X, (\Omega, \Xi)) = I(X, \Omega)$ , so this is well-mixed according to our definition. Of course, this is not really mixed with anything, but it verifies that the definition applies in the null case.

As another simple example, suppose that  $\Omega$  and  $\Xi$  are independent but identically distributed, and pass through totally separate channels. Suppose that each channel has an additive noise component  $\Phi_i$ , but that the transfer functions  $(f_{\text{net}})_i$  in the two channels is forced to be identical. In this case, the system can be physically separated into two independent parts, with

$$I(X, (\Omega, \Xi)) = 2I(X, \Omega) \quad (25)$$

and similarly for  $Y$ . Therefore  $X$  and  $\Xi$  are well-mixed with respect to the set of channel functions  $\{f_{\text{net}}\}$ . Clearly, the signals  $\Omega$  and  $\Xi$  are not mixed in together, in the sense that they could be separated by using a simple

selection on one of the channels. However, the operations which can be performed by the functions  $f_{\text{net}}$  which we are allowing operate identically on both.

For a third example, consider a two-component input  $X = [X_1, X_2]$  where  $X_1 = \Omega + \Xi$  and  $X_2 = \Omega - \Xi$ , and where these components are passed through independent but identical linear channels. Under similar conditions to that above,  $\Omega$  will again be well-mixed with  $\Xi$ , as long as the separate channels cannot be combined. Of course, as soon as the separate channels can be added together, perhaps in a later processing stage,  $\Omega$  can be separated from  $\Xi$ . This arrangement is perhaps slightly closer to (although still much simpler than) what we might expect from a well-mixed signal in reality.

As a signal passes through a neural sensory system, we would expect it to be able to separate ‘signal’ from ‘noise’ increasingly well. An initial stage, possibly linear, would only be able to cope with simple additive noise from the rest of the input. With respect to its linear transform, we would hope that the rest of the signal  $\Omega$  of interest will be well-mixed with the rest of the noise, so that it is not lost ‘by accident’ while minimizing information loss. However, later processing stages with more complex non-linear processing, may be able to separate  $\Omega$  from more of the noise with which it was initially well-mixed. Provided the signal is well-mixed at each stage (including at the final stage when we hope that  $\Xi = 0$ , thus removing all the unseparable noise), minimizing information loss about  $(\Omega, \Xi)$  will proportionately minimize the information loss about  $\Omega$ .

In biological sensory systems, frog retinas seem to have the ability to extract information about ‘moving blobs’, presumably corresponding to a fly (i.e. food), while higher mammals appear to have a much more general-purpose retina [17]. If  $\Omega$  represents information about food, while  $\Xi$  represents information about other miscellaneous images, the frog appears to demonstrate that  $\Omega$  is separable from  $\Xi$  to a large extent by its retina, while in higher mammals this is not the case. For higher mammals we would hope

that  $\Omega$  is well-mixed with  $\Xi$  with respect to the processing that the retina can perform, so any general optimization of total information as suggested by e.g. Atick and Redlich [3] will apply proportionately to  $\Omega$ .

## 6 Conclusions

We have explored the approach of minimization of information loss (MIL) as an objective for neural network learning. Comparing it with more traditional learning methods, we have seen that MIL leads to Bayesian MAP classification, and that minimizing mean squared error (MSE) and cross-entropy (CE) objective functions minimizes an upper bound on information loss. These bounds are tightest when the errors produced by the network are symmetrical and balanced.

For unsupervised learning, we have seen that MIL is very similar to Linsker's Infomax [1] principle. In order to deal with the problem of unwanted effects which are not simple additive noise, we introduced the concept of a signal which is *well-mixed* with noise, with respect to the set of possible transformations under consideration. This proposes that the information loss about the variable of interest will be in proportion to the total information loss. We hope that this will enable information-theoretic approaches to perception such as MIL to be extended to more complex processing than has been investigated so far.

## References

- [1] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, March 1988.
- [2] M. D. Plumbley and F. Fallside. An information-theoretic approach to unsupervised connectionist models. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 239–245. Morgan-Kaufmann, San Mateo, CA, 1988.

- [3] J. J. Atick and A. N. Redlich. Towards a theory of early visual processing. *Neural Computation*, 2:308–320, 1990.
- [4] M. D. Plumbley. Efficient information transfer and anti-Hebbian neural networks. *Neural Networks*, 1993. (In press).
- [5] R. Linsker. Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural Computation*, 4:691–702, 1992.
- [6] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.
- [7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, pages 318–362. Bradford Books/MIT Press, Cambridge, MA, 1986.
- [8] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.
- [9] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
- [10] J. B. Hampshire and B. A. Pearlmutter. Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function. In D. Touretsky, J. Elman, T. Sejnowski, and G. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*. Morgan Kaufmann, San Mateo, CA, 1990.
- [11] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, second edition, 1984.
- [12] M. D. Plumbley. On information theory and unsupervised neural networks. Technical Report CUED/F-INFENG/TR. 78, Cambridge University Engineering Department, UK, 1991.

- [13] R. P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989.
- [14] S. A. Solla, E. Levin, and M. Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2, 1988.
- [15] J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In F. Fogelman Soulié and J. Hérault, editors, *Neurocomputing - Algorithms, Architectures and Applications*, pages 227–236. Springer-Verlag, Berlin, 1990.
- [16] M. D. Plumbley. Lyapunov functions for convergence of principal component algorithms. Submitted to *Neural Networks*, 1993.
- [17] 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.