On Information Theory and Unsupervised Neural Networks

Mark D. Plumbley

CUED/F-INFENG/TR.78

13th August 1991

Summary

In recent years connectionist models, or *neural networks*, have been used with some success in problems related to sensory perception, such as speech recognition and image processing. As these problems become more complex, they require larger networks, and this typically leads to very slow training times. Work on these has primarily involved the use of *supervised* models, networks with a 'teacher' which indicates the desired output. If it were possible to use *unsupervised* models in the early stages of systems to help with the solutions to these sensory problems, it might be possible to approach larger and more complex problems than are currently attempted. We may also gain more insight into the representation of sensory data used in such sensory systems, and this may also help with our understanding of biological sensory systems.

In contrast to supervised models, unsupervised models are not provided with any teacher input to guide them as to what they should 'learn' to perform. In this thesis, an information-theoretic approach to this problem is explored: in particular, the principle that an unsupervised model should adjust itself to minimise the *information loss*, while in some way producing a simplified representation of its input data as output.

Initially, general concepts about information theory, entropy and mutual information are reviewed, and some systems which use other information-theoretic principles are described. The concept of information loss and some of its properties are introduced, and this concept is related to Linsker's 'Infomax' principle. The information loss across supervised learning systems is briefly considered, and various conditions are described for a close match between minimisation of information loss and minimisation of various distortion measures.

Next information loss across a simple linear network with one layer of processing units is considered. In order to progress, an assumption must be made concerning the *noise* in the system instead. With the noise on the input to the network dominant, a network which performs a type of principal component analysis is optimal. A common framework for various neural network algorithms which find principal components of their input data is derived, these are shown to be equivalent in an information transmission sense.

The case of significant output noise for position and time-invariant signals is considered. Given a power cost constraint in our system, the form of the optimum linear filter required to minimise the information loss under this constraint is analysed. This filter changes in a non-trivial manner with varying noise levels, mirroring the way that the response of biological retinal systems changes as the background light level changes. When the output noise is dominant, the optimum configuration can be found by using anti-Hebbian algorithms to decorrelate the outputs. Various forms of networks of this type are considered, and an algorithm for a novel Skew-Symmetric Network which employs inhibitory interneurons is derived, which suggests a possible rôle for cortical back-projections.

In conclusion, directions for further work are suggested, including the expansion of this analysis for systems with various non-linearities; and general problems of representation of sensory information are discussed.

Acknowledgements

There are so many people whose ideas and comments have found their way into this thesis in some way or another, that it would be impossible to name them all. The Speech, Vision and Robotics Group, past and present, must get a special mention, including my supervisor Frank Fallside, and Tony Robinson, Patrick Gosling, Niranjan, Richard Prager, Dave Rainton, Tim ('Large Waster') Marsland, Mike ('Small Waster') Chong, Visakan and Maha Kadirkamanathan, Steve Young, Ron Daniel, and many, many more. Particular thanks to Patrick and Niranjan for proof-reading, and also to Mavis Barber for helping to keep everything running smoothly.

Finally, my thanks to Penny, whose support and occasional reminders meant that this thesis did get written—eventually.

This report is adapted from my Ph.D. thesis of 30th May 1991 entitled 'An Information-Theoretic Approach to Unsupervised Connectionist Models'. It covers part of my work between January 1987 and May 1991 at Cambridge University Engineering Department on Information Theory and Neural Networks.

Between January 1987 and December 1989 I was financially supported by an award from the U.K. Science and Engineering Research Council. Since January 1990 I have been employed at CUED as a Research Associate on the application of Genetic Algorithms to Neural Networks.

Contents

List of Symbols

X, Y, ...	Random variables (r.v.)	2.1.2
$\Pr(\cdot)$	Probability of event	2.1.2
$p_X(x)$, $P_X(x)$, $p(x)$, $q(x)$,...	Probability density, distribution (p.d.)	2.1.2
$E(X)$	Expected value of X	2.1.2
$H(X)$, $H(Q)$	Entropy of r.v. X, p.d. Q	2.1.3
$H(X,Y)$, $H(X\|Y)$	Joint, conditional entropy	2.1.3
$I(X;Y)$	Mutual information between X and Y	2.1.4
$D_I(p,q)$	I-divergence from p to q	2.1.5
μ_p, σ_p	Mean and variance of p	2.1.5
G	'G-max' cross-entropy measure	2.3.1
Σ_X	Covariance matrix of X	2.3.2
ΔI, $\Delta I_\Omega(X,Y)$	Information loss	2.3.3, 2.4.1
Ω	Feature r.v. (original patterns)	2.3.3
X, Y	Network input, output r.v.s	2.4.1
$\mathbf{x}$, $\mathbf{y}$	Network input, output vectors	2.4.1
$f_W(\cdot)$	Network function	2.4.2
N, M	Number of inputs, outputs	2.4.3
$D_{\mathrm{MSE}}(\Omega,Y)$	Mean Squared Error distortion	2.4.3, 3.1.1
$\mathrm{tr}(M)$	Trace of matrix M	2.4.3
$N(\mu,\sigma^2)$	Normal p.d.	2.4.3
$D_{\mathrm{CE-SLF}}$	Component-wise cross-entropy distortion	2.4.3
$D_{\mathrm{CE-MMI}}$	True cross-entropy distortion	2.4.3
J, J_W	Cost function	2.4.4
λ, γ	Lagrange multipliers	2.4.4
$\delta(\cdot)$	Dirac delta function	2.4.4
W^T	Forward weight matrix	3.1
P	Orthogonal projection	3.1.2
W^+	Moore-Penrose Pseudo-inverse	3.1.2
Φ_X	Additive noise on X (r.v.)	3.2
W_n	Value of W at step n	3.3.1
η_n	Update factor	3.3.1
$\mathrm{diag}(M)$	Diagonalisation of matrix M	3.3.1
$\mathrm{UT}(M)$, $\mathrm{UT}_+(M)$	Upper, strictly upper triangularisation	3.3.1
$W_n\Theta_n$	Forgetting factor	3.3.2
$\mathrm{vec}(M)$	Vectorisation of matrix M	3.3.2
$\|\cdot\|_F$	Matrix Frobenius norm	3.3.4

f	Frequency	4.1
$S_r(f)$, $S_c(f)$	Receptor, channel signal power density	4.1
$N_r(f)$, $N_c(f)$	Receptor, channel noise power density	4.1
B	Bandwidth limit	4.1
$C(f)$	Information capacity density	4.1
$G(f)$	Filter power spectral gain	4.1
C_T, P_T	Total channel capacity, power cost	4.1
R_r, R_c	Receptor, channel signal-to-noise ratio	4.2
R	Redundancy	4.3
C	Capacity	4.3
$\mathbf{\Psi}$	Output r.v. after additional noise	5.1
V	Inhibitory weight matrix	5.2
I_N	$N \times N$ identity matrix	5.2
S_W	Power cost for weight matrix W	5.2
k	Number of epochs	5.4.1
$\mathbf{z}$, Z	Interneurons response vector, r.v.	5.5
U^T	Forward excitation weight matrix	5.5.2
$\mathbf{t}$	Target vector	6.2.2

Chapter 1

Introduction

Perception is easy. It *must* be easy—we do it all the time, and hardly notice: it is effortless and unconscious.

Unfortunately, building machines which perceive is more tricky. The problems of speech recognition and image recognition by machine have been the object of much research effort over many years, and both are still difficult problems. One avenue of research in recent years is the field of Connectionist Models, or Artificial Neural Networks (sometimes simply *Neural Networks*). By taking ideas from the way biological neural systems are thought to operate, it is hoped that artificial systems may be built which are able to take advantage of some of the properties of biological systems. The operation of these artificial sensory systems may also help to explain the behaviour of real biological systems. However, while it is possible to analyse biological systems at a very low level of up to a few neurons, and at a high level, in terms of the function of whole areas of the brain, our understanding of the intermediate levels in rather limited. In order to proceed, we must either wait for our understanding of biological perceptual systems to progress to these intermediate levels of organisation, or we must theorise—make a 'guess' at some general principle which may be involved.

The principle we employ in this thesis is an information-theoretic one. The idea is simple: since a sensory system processes *information* about the outside world, to be used by the organism or machine to aid its ability to operate in the world, we should be able to quantify the amount of information so processed in terms of Shannon's 'Information Theory' [Shannon, 1948], well known to communications engineers. Given constraints in our system such as limited numbers of neurons, limited energy, and so on, the sensory system should attempt to retain as much *information* as possible: a principle of constrained *minimisation of information loss.*

The idea behind such a principle is not new: one of the earliest proponents was Attneave [1954], who originally suggested that information theory was important in perception a few years after Shannon's original work. He suggested, for example, that information in an image was concentrated around the edges and corners, since these are areas of greatest *uncertainty* in the image. Observing these edges and corners thus gains us more information that other areas of an image.

More recently, Linsker [1988b] suggested an 'Infomax' principle, that a perceptual system should organise itself to transmit maximum information. This is virtually identical to the minimisation of information loss across the system, although certain possibly important differences arise when considering the details. In particular, if a perceptual system can only measure, say, second order statistics of the data that it observes (i.e. covariance), the 'Infomax' principle will find an upper bound for the transmitted information, while the minimum information loss approach will find an upper bound for the *loss* of information—a *damage limitation* approach.

Despite these details, the implications of both principles are similar, and give a *data-driven* approach to the problem of developing artificial sensory systems. Although there may be constraints and costs associated with the operation of a sensory system, the prime concern should be the transmission of information: changing its representation to a more convenient form for later processing, while minimising the information lost in the process. Using this approach, we hope that we may gain a better understanding of the problem of the design of sensory systems, both biological and artificial.

Organisation

In this thesis, we explore the principle of minimisation of information loss, and its consequences; initially for supervised learning systems (systems with a *teacher*), but primarily for unsupervised learning systems. Although mainly concentrating on analysis of linear unsupervised systems, albeit with non-linear learning algorithms, towards the end we explore some of the problems of non-linear systems, especially with regard to the popular 'Winner-Take-All' approach.

In chapter 2 we review some concepts from information theory, and introduce the idea of information loss across a network as a measure of network performance. We consider the implications of this for supervised learning systems, and compare it with Linsker's 'Infomax' criterion.

In chapter 3 we examine Hebbian neural network algorithms which perform a principal components analysis (PCA) of their input data. We find that many of these algorithms can be combined in a common framework, and they all perform a direct minimisation of information loss over time, if the only noise in the system is on the inputs to the network.

In chapter 4 we consider systems with noise on both the inputs and the outputs. If we are dealing with signals such as speech or image signal whose statistics are time- or space-invariant, we can minimise the information loss in the frequency domain. We derive an analytic expression for the optimal filter which minimises information loss for a given power cost, and show how the form of this filter changes with relative input and output noise levels.

Chapter 5 considers anti-Hebbian learning algorithms, for minimising information loss for a given power cost when noise on the output of the network is dominant. We need no space-invariant assumptions in this case. We show that some anti-Hebbian learning algorithms minimise power cost while attempting to keep the information loss identical. For other architectures, we derive anti-Hebbian learning algorithms which directly decrease the constrained cost function over time. One of these leads to an architecture with inhibitory interneurons, and a simple anti-Hebbian learning algorithm with a weight decay function.

Finally, we look at the requirements for, and problems associated with non-linear systems, and how we might generalise away from the ubiquitous 'Winner-Take-All' approach.

Original Content

Although the use of information theory in general is well-known in the neural network and psychology fields, the concept of 'information loss' used in this thesis is new. It is closely related to the 'Infomax' concept of Linsker [1988b], although it has certain advantages over the latter.

In this thesis, this information loss measure is used to give new insight into low-level machine perception, specifically the importance of noise in all perceptual processes. Also, this measure is used to produce novel convergence proofs for Hebbian and anti-Hebbian algorithms. In particular, it leads to the development in this thesis of the Skew Symemtric Network and algorithm, with its implications for cortical sensory processing.

Chapter 2

Information Theory and Perceptual Processing

2.1 A Brief Introduction to Information Theory

The results in this section will be presented without proof. For more details, the reader is referred to any standard text on information theory (see e.g. [Kullback, 1959]).

2.1.1 General Concepts

The concept of an *amount of information* associated with an event was introduced by Shannon in his "Mathematical theory of communication" [Shannon, 1948], and has since been an invaluable tool in the development of communication theory, and many other fields. The general idea works as follows. If p is the probability of a particular event, then the amount of information associated with that event is $\log(1/p)$. In this way a rare event, with p small, conveys a large amount of information, while a common event, with p large, conveys a relatively small amount of information.

If there are a finite number of events i with associated probabilities p_i, $1 \leq i \leq N$, the mean amount of information about this system is known as its *entropy*

$$H = \sum_{i=1}^{N} p_i \log(1/p_i) \tag{2.1}$$

and represents the *uncertainty* in the system. For any N, the entropy H is maximised when all the events are equally probable, i.e.

$$p_i = 1/N \qquad 1 \leq i \leq N$$

in which case $H = \log(N)$. The entropy is minimised when only one event i^* is possible, i.e.

$$p_i = \begin{cases} 1 & \text{if } i = i^* \\ 0 & \text{otherwise} \end{cases}$$

in which case $H = 0$, since $p_i \log(1/p_i) = 0$ if $p_i = 1$, and $p_i \log(1/p_i) \to 0$ as $p_i \to 0$.

This entropy H is therefore a bounded non-negative measure of the information in a system with a finite number of possible events. For example, a throw of an unbiased six-sided die has entropy $H = \log(6) \approx 1.79$. When dealing with digital communication systems, this logarithm is often taken to base 2, and the result is expressed in *bits* (BInary digiTS); thus our die roll has entropy $H \approx 2.58$bits. This is convenient in communication systems, since the entropy of system with two equiprobable events is one bit. For theoretical analysis, however, it is more convenient to deal with natural logarithms, so we shall use these here.

For another view of entropy, consider the probability of a particular sequence of M independent events in this system. By the law of large numbers, as M gets large, the possible sequences are dominated by those which have approximately Mp_i occurrences of event i. The probability of each of these dominating sequences is

$$P_M = \prod_{i=1}^{N} p_i^{Mp_i}$$

or there are approximately $\mathcal{N}_M = 1/P_M$ equally probable choices for the sequence of M independent events. Taking the log of this quantity, we find

$$\begin{aligned} \log \mathcal{N}_M &= \sum_{i=1}^{N} Mp_i \log p_i \\ &= M \sum_{i=1}^{N} p_i \log p_i \\ &= MH \end{aligned}$$

so the entropy of a system can be viewed as the log of the number of ways of generating a long sequence of independent events, divided by the length of the sequence. In language modelling, the number of ways $\mathcal{N}_M/M$ as $M \to \infty$ is known as *perplexity* (see e.g. [Jelinek, 1985]).

It is possible to generalise this concept of entropy to a continuum of possible events, but only with care. If we simply let the number of possible events N increase, we find that the H can increase without bound: the more possible choices, the greater the uncertainty. In the limit as $N \to \infty$, in general $H \to \infty$ also, reflecting the notion that a continuum of events contains an infinite amount of information. If we could measure each individual event (for example a single real number) we would gain an infinite amount of information. However, any observation in a real system will leave some remaining uncertainty, due to measurement inaccuracies, for example, so we only gain a finite amount of information.

To clarify this concept, let us first introduce some notation about random variables.

2.1.2 Random Variables

We shall write a random variable (r.v.) [Papoulis, 1984] in capitals as e.g. X, which takes values in some set $\mathcal{X}$. It is a function from the set S of experimental outcomes. If continuous, the probability density is $p_X(x) = \frac{d}{dx}\Pr(X \leq x)$; if discrete, the probability distribution is $P_X(x) = \Pr(X = x)$. Where clear from context, the subscript will be omitted. We can generalise this notion for random variables which take vector values $\mathbf{x} = (x_1, \ldots, x_n)$, and write $p(\mathbf{x})$ for their probability density. If X and Y are both random variables, the *joint probability density* $p(x, y)$ is the probability density associated with the pair of random variables X, Y treated like a single vector-valued random variable.

For a function $f(X)$ of the random variable X, the *expected value* or *mean* of f over X,

$$E(f(X)) = \int f(x)p(x)\,dx$$

where $f(X)$ is the random variable which takes the value $f(x)$ whenever X takes the value x. In particular the mean of X is

$$E(X) = \int xp(x)\,dx$$

where this is meaningful: we can't talk about the expected value of X if X takes the values "apple", "banana" or "carrot", for example. In particular, the set $\mathcal{X}$ must be closed under addition and under multiplication by real numbers.

If $p(x, y)$ is a joint probability density function, we can write $p(x|y) = p(x, y)/p(y)$ for the marginal probability density of X at x given that the event $Y = y$ is known to have occurred. Also, $p(x)$ is related to $p(x, y)$ by

$$p(x) = \int_y p(x, y)\, dy.$$

2.1.3 Entropy

From the discussion at the beginning of this section, if X is a discrete random variable, the entropy $H(X)$ of X is given by

$$\begin{aligned} H(X) &= \sum_x P_X(x) \log 1/P_X(x) & (2.2) \\ &= E\left(\log\left(1/P_X(X)\right)\right) & (2.3) \end{aligned}$$

which as we noted in the discussion earlier is always positive and bounded by the number of elements in the range of X. If Q is a probability distribution such that $Q(x) \geq 0$ for all x, and $\sum_x Q(x) = 1$, the we can also write

$$H(Q) = \sum_x Q(x) \log 1/Q(x). \quad (2.4)$$

Note that one of these is the entropy of the random variable X, while the other is the entropy of a probability distribution Q: it should be clear from context which of these we are using at any one time.

In an analogous way, we shall write the entropy of a continuous random variable X as

$$H(X) = \int_X p(x) \log 1/p(x)\, dx. \quad (2.5)$$

Although notationally similar to the discrete case, this is not guaranteed to be non-negative or bounded. It is also scaling dependent. For example, if $Y = mX$ for some scalar m,

$$H(Y) = H(X) + \log m.$$

For a more general linear transform of vector-valued random variables, $Y = MX + \mathbf{c}$ where M is a square nonsingular matrix, we have

$$H(Y) = H(X) + \log \det(M)$$

so in particular $H(Y) = H(X)$ if $\det(M) = 1$. In the case of a non-linear transform $Y = f(X)$, where X and Y have the same dimensionality, we get

$$H(Y) \leq H(X) + E\left(\log \det\left(J(X)\right)\right)$$

where $J(x)$ is the jacobian of the transform [Papoulis, 1984].

Properties of Entropy

The *joint entropy* of a pair of (discrete or continuous) random variables never exceeds the sum of the individual entropies, i.e.

$$H(X, Y) \leq H(X) + H(Y) \quad (2.6)$$

with equality when X and Y are independent. We can also write the *conditional entropy* of X given Y:

$$\begin{aligned} H(X|Y) &= E(\log 1/P(X|Y)) & (2.7) \\ &= H(X, Y) - H(Y) & (2.8) \end{aligned}$$

and from (2.6) we note that $H(X|Y) \leq H(X)$. In other words, the uncertainty remaining about X after we know Y is less than the uncertainty in X alone.

2.1.4 Mutual Information

The *mutual information* between X and Y is

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) & (2.9)\\ &= H(X) + H(Y) - H(X,Y) & (2.10)\\ &= H(Y) - H(Y|X) & (2.11)\\ &= I(Y;X) & (2.12)\end{aligned}$$

which we know from (2.6) is non-negative for any X and Y, and zero if X and Y are independent. Note that $I(X;Y)$ can be written

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

Unlike entropy, mutual information $I(X;Y)$ is independent of any non-singular linear transformation of X or Y. In fact, for any function $f(X)$,

$$I(f(X);Y) \leq I(X;Y) \tag{2.13}$$

holds, with equality if $f(X)$ has an inverse.

For a discrete random variable X, the *self-information* in X, i.e. the information available about X if we observe it, is given by

$$\begin{aligned} I(X;X) &= H(X) - H(X|X) \\ = H(X) & & (2.14)\end{aligned}$$

and is thus simply the entropy of X. Note that this concept of self-information is not meaningful for continuous random variables, since there is theoretically an infinite amount of information available if we could make our observations accurate enough.

Example: Mutual Information with Additive Noise

Let the continuous random variable Y be given by

$$Y = X + \Phi$$

where X is a continuous random variable representing a 'signal' about which we wish to gain information, and Φ is a continuous random variable representing additive 'noise' which is independent of X. For the information in Y about X, we have

$$I(Y;X) = H(X) + H(Y) - H(X,Y)$$

but $H(X,Y) = H(X,\Phi)$ since the transform

$$(X,Y) \rightarrow (X,\Phi) = (X,(Y-X))$$

has determinant 1, and

$$\begin{aligned} H(X,\Phi) &= H(X) + H(\Phi) - I(X,\Phi) & (2.15)\\ &\leq H(X) + H(\Phi) & (2.16)\end{aligned}$$

with equality when X and Φ is independent, i.e. the noise is independent of the signal (which is what we are assuming). Thus our information is given by

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - (H(X) + H(\Phi)) \\ &= H(Y) - H(\Phi) & (2.17)\end{aligned}$$

provided X and Φ are independent. If this independence does not hold, we get the inequality

$$I(X;Y) \geq H(Y) - H(\Phi) \tag{2.18}$$

i.e. the entropy difference (2.17) is a lower bound for the information in Y about X.

2.1.5 I-Divergence

Another information-theoretic concept which arises in information theory is the idea of the *cross-entropy* or *I-divergence* (also known as *Kullback-Leiber* (KL) distortion) between two probability distributions p and q, which is a measure of the accuracy with which a 'suggested' probability distribution q matches the 'true' probability distribution q. This measure is defined as

$$D_I(p,q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx \tag{2.19}$$

provided p is absolutely continuous with respect to q (i.e. $q(x) = 0 \Rightarrow p(x) = 0$). This is always non-negative, since

$$\int p(x) \log p(x)\, dx \geq \int p(x) \log q(x)\, dx$$

for any probability density functions p and q. Note that this is not symmetrical, i.e. in general

$$D_I(p,q) \neq D_I(q,p)$$

so while this can be used as a distortion measure, it is not a *metric*.

We note that the we can write the mutual information $I(X;Y)$ in terms of I-Divergence, since we have

$$\begin{aligned} I(X;Y) &= E\left(\log \frac{p(X,Y)}{p(X)p(Y)}\right) \\ &= D_I(p,q) \end{aligned}$$

where $p(X,Y)$ is the true probability density of X and Y, and $q(X,Y) = p(X)p(Y)$ is the probability density for X and Y under the assumption that X and Y are independent.

For a discrete random variable X, we can get some idea of the meaning of cross-entropy, if we consider how much it would cost in bits, to transmit the values taken by X down a communication channel. If we *thought* X was distributed according to some probability distribution $Q(X)$, the best we could do (for minimum transmission cost) would be use a code word of length $\log_2 1/Q(x_i)$ bits for each possible value x_i that X could take. Thus the mean code word length we would need would be $E(\log_2 1/Q(x_i))$ bits per observation. The minimum this could be is if $Q(X)$ was in fact the true distribution $P(X)$, in which case the mean code word length would be $E(\log_2 1/P(x_i))$. The difference between these two values is simply the I-divergence between P and Q:

$$D_I(P,Q) = E\left(\log \frac{P(X)}{Q(X)}\right).$$

Thus the I-divergence represents the mean number of bits which would be wasted in transmission if we assumed that X was distributed according to $Q(X)$ instead of its true distribution $P(X)$.

Example: I-Divergence between discrete distributions

Let $P(x)$ and $Q(x)$ be discrete probability functions, such that $P(x)$ and $Q(x)$ are non-zero for $x \in \{1, \ldots, N\}$. In particular, let $Q(x) = 1/N$ and for some $0 \leq M \leq N$ let

$$P(x) = \begin{cases} 1 & \text{if } x = M \\ 0 & \text{otherwise} \end{cases}$$

as shown in Fig. 2.1 for $N = 6$ and $M = 4$ (an unbiased die roll, for example). The I-Divergence between P and Q is then

$$D_I(P,Q) = \log(N) \tag{2.20}$$

or $D_I(P,Q) = \log(6) \approx 1.79$ if taking natural logarithms (about 2.58 bits).

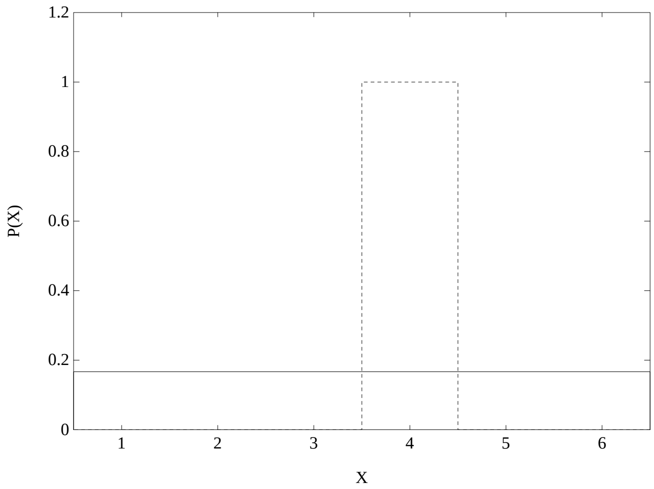


Fig. 2.1: I-Divergence between Discrete Distributions

Example: I-Divergence between Gaussians

Let p and q be normal distributions with respective means μ_p and μ_q and variances σ_p^2 and σ_q^2, i.e.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}\frac{(x-\mu_p)^2}{\sigma_p^2}\right)$$

and similarly for q. The I-Divergence between p and q is

$$\begin{aligned} D_I(p,q) &= \frac{1}{2}\log\frac{\sigma_q^2}{\sigma_p^2} + \frac{1}{2}\frac{(\mu_p-\mu_q)^2}{\sigma_q^2} + \frac{1}{2}\left(\frac{\sigma_p^2}{\sigma_q^2} - 1\right) && (2.21) \\ &= H(q) - H(p) + \frac{1}{2}\frac{(\mu_p-\mu_q)^2}{\sigma_q^2} + \frac{1}{2}\left(\frac{\sigma_p^2}{\sigma_q^2} - 1\right). && (2.22) \end{aligned}$$

In particular, if $\mu_q = 0$, $\sigma_q^2 = 36$, $\mu_p = 4$ and $\sigma_p^2 = 1$ (Fig. 2.2), $D_I \approx 1.53$ (2.20 bits).

2.2 Information Theory in Psychology

Ever since Shannon's "Mathematical Theory of Communication" [Shannon, 1948] first appeared, information theory has been of interest to psychologists and physiologists, to try to provide an explanation for the process of perception. Attneave [1954] proposed that visual perception is the construction of an economical description of a scene from a very redundant initial representation. He suggested that most of the information in a scene is concentrated around lines and corners, i.e. the regions where the content of the scene changes.

Thus it is possible that perception process is performing some sort of *data reduction* on received sensory input, to enable it to be processed and stored more easily. The maximum rate at which a person can transmit information has been measured to be around 30 or 40 bits per second [Attneave, 1959], while Kelly [1962] measured the information capacity of a single retinal channel to be 10^9 bits per second. He suggested that the visual system "takes advantage of the input statistics to perform various coding operations", to reduce some of this high data rate to something more manageable. Barlow [1961, 1987] has suggested that lateral inhibition (in the retina, for example)

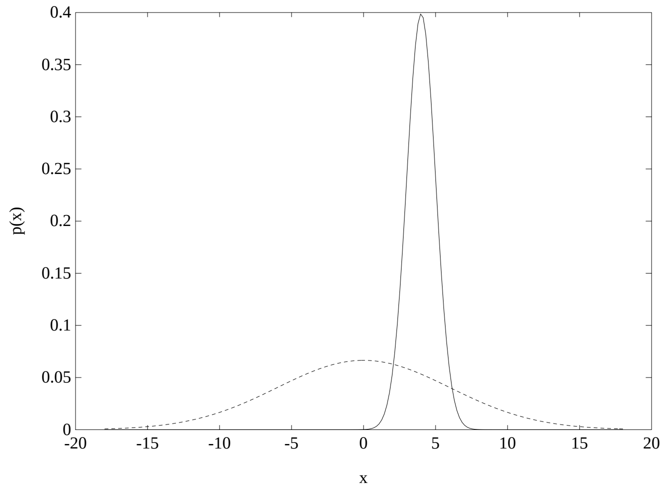


Fig. 2.2: I-Divergence between Gaussian Distributions

can help, by reducing the redundancy of an image, so the same amount of information can be represented more efficiently.

This use of information theory has had its detractors. Green and Courts [1966] for example, argued that information theory could not be used in the consideration of perception, since there is no objective 'alphabet of symbols' and also no objective transition probabilities. Despite these arguments, the feeling that information theory is a useful tool for examining perception has continued [Strizenec, 1975]. Leeuwenberg [1978] and Mellink and Buffart [1987] have continued with this idea with their 'Structural Information Theory', which has primarily been used in connection with the perception of simple visual patterns.

2.3 Unsupervised Neural Networks

One obvious application of information theory is to measure the information capacity of associative memory networks, such as the Hopfield network [Abu-Mostafa and Jaques, 1985]. Of more interest to us, however, is the use of an information measure to suggest learning algorithms for *unsupervised* neural networks [Becker, 1991].

2.3.1 Cross-Entropy Maximisation: "G-Max"

An early technique in this area, *G-Maximization*, was suggested by Pearlmutter and Hinton [1986]. Their model has N binary inputs x_i with corresponding weights w_i, $i = 1, \ldots, N$ and a single binary output y. The state of this output unit is determined stochastically in a similar way to those in the Boltzmann machine [Ackley *et al.*, 1985]. The probability that the output state is "1" is

$$P_y(1) = \sigma\left(\sum_{i=1}^{N} w_i x_i\right) \tag{2.23}$$

where the $\sigma(\cdot)$ is the logistic sigmoid function

$$\sigma(x) = \frac{1}{1+\exp(-x)}. \tag{2.24}$$

They suggested that to discover regularities in the input patterns,

> "...the unit should respond to patterns in its inputs that occur more often than would be expected if the activities of the individual input lines were assumed to be independent."

To this end, they chose as their target function the cross-entropy distortion

$$G = \sum_{y=0}^{1} P(y) \log \frac{P(y)}{Q(y)}$$

between the true output probability distribution $P(y)$, and the 'independent' probability distribution $Q(y)$ of the output, if the inputs are assumed to be independent. This can be differentiated with respect to each of the weights w_i, and a hill-climbing technique used to find the maximum.

The algorithm is run in two phases: first with real data to accumulate statistics about $P(y)$; and then with simulated data with inputs generated independently to accumulate statistics about $Q(y)$. For this second phase, the inputs have the same likelihood of being on or off individually as they did in the first phase, but now independent of the state of the other input lines. When this was tested on a 10×10 "retina" exposed to randomly positioned and oriented edges, the output unit typically developed an centre-surround antagonistic response: the unit developed a positive (or negative) reponse to receptors in the center of the retina, with the opposite reponse to the surrounding annulus.

This model was difficult to extend to more than one unit, since some other scheme has to be used to prevent them all learning the same features of the input. Another possible disadvantage was the requirement to generate synthetic input data during the second pass. It seemed that a different approach would be required for networks with many output units.

2.3.2 Maximising of Mutual Information between Output Units: "I-Max"

More recently, Becker and Hinton [1989] suggested that the information *between* output units could be used as the objective function for an unsupervised learning technique (Fig. 2.3). In a visual system, this scheme would attempt to extract higher-order features of the visual scene which are coherent over space (or time). For example, if two networks each produce a single output from two separate but neighbouring *patches* of a retina, the objective of their algorithm is to maximise the mutual information $I(Y_1; Y_2)$ between these two outputs. A steepest-ascent procedure can be used to find the maximum of this mutual information function, both for binary- and real-valued units.

One application of this principle is the extraction of depth from random-dot stereograms [Julesz, 1971]. Nearby patches in an image usually view objects of a similar depth, so if the mutual information between neighbouring patches is to be maximised, the outputs from both output units y_1 and y_2 should correspond to the information which is common between the patches, rather than that which is different. In other words the outputs should both learn to extract the common depth information rather than any other property of the random dot patterns.

For binary-valued units, with each unit similar to that used by the G-max scheme described above, the mutual information $I(Y_1; Y_2)$ between the two output units is

$$I(Y_1; Y_2) = H(Y_1) + H(Y_2) - H(Y_1, Y_2) \tag{2.25}$$

so if the probability distributions $P(y_1)$, $P(y_2)$ and $P(y_1, y_2)$ are measured, this mutual information can be calculated directly. Of course, it is sufficient to measure $P(y_1, y_2)$ only, since

$$P(y_1) = \sum_{y_2=0}^{1} P(y_1, y_2)$$

and similarly for $P(y_2)$. The derivative of (2.25) can be taken with respect to the weights in the network for each different input pattern, so enabling the steepest-ascent procedure to be used.

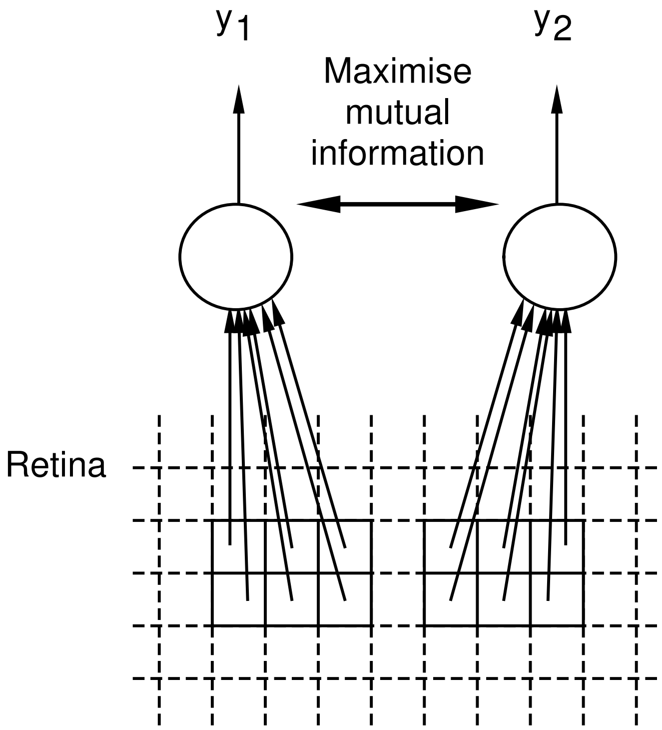


Fig. 2.3: Maximising Mutual Information between Output Units

For real-valued outputs it would be impossible to measure the entire probability distribution $P(Y_1; Y_2)$, so instead it is assumed that the two outputs have a Gaussian probability distribution, and that one of the outputs is a noisy version of the other, with independent additive Gaussian noise. In this case, the information $I(Y_1; Y_2)$ between the two outputs can be calculated from the variances of one of the outputs (the 'signal') and the variance of the difference between the outputs (the 'noise') as

$$I(Y_1; Y_2) = \frac{1}{2} \log \frac{\sigma^2_{Y_1}}{\sigma^2_{Y_1 - Y_2}} \tag{2.26}$$

where $\sigma^2_{Y_1}$ is the variance of the output of the first unit, and $\sigma^2_{Y_1-Y_2}$ is the variance of the difference between the two outputs. There are also alternative symmetrical objective functions which can be used instead [Becker and Hinton, 1989, Becker, 1991].

Now, if we accumulate the mean and variance of both Y_1 and $Y_1 - Y_2$, it is possible to find the derivative of (2.26) for each input pattern, with respect to each weight value. Thus the weights in the network can be updated in a steepest-ascent procedure to maximise $I(Y_1; Y_2)$, or at least the approximation to $I(Y_1; Y_2)$ given by (2.26).

Becker and Hinton found that unsupervised networks using this principle could learn to extract depth information from random-dot stereograms with either binary- or continuous-valued shifts, as appropriate for the type of outputs used, although in some cases it helped to force the units to share weight values, i.e. enforcing the idea that the units should calculate the same function of the input. They generalised their scheme to allow networks with hidden layers, and also to allow multiple output units, with each unit maximising the mutual information between itself and a value predicted from its neighbouring units. This latter scheme allowed the system to discover an interpolation for curved surfaces.

More recently, Zemel and Hinton [1991] have generalised this procedure to allow for more than one output per module. In their network, they use 4 outputs per unit to attempt identify four degrees of freedom in 2-dimensional objects: horizontal and vertical position, orientation, and size. The objects used were simple dipole objects with a left half and a symmetric right half. The mutual information measure is now

$$I(Y_1; Y_2) = \frac{1}{2} \log \frac{\det(\Sigma_{Y_1+Y_2})}{\det(\Sigma_{Y_1-Y_2})} \tag{2.27}$$

where $\Sigma_{Y_1+Y_2}$ is the covariance matrix of the sum of Y_1 and Y_2 (now vectors, of course), and $\Sigma_{Y_1-Y_2}$ is the covariance matrix of their difference. By measuring the degree of mismatch between the two representations, the model can tell roughly how much one half of an object is perturbed away from the position and orientation which is consistent with the other half of the object.

2.3.3 Minimising Information Loss

Both of these procedures outlined above use information theory measures in their formulation, but neither is concerned directly with the actual information transmitted through to the output units about the input. Rather they attempt to discover regularities in the input data to a single output unit (G-Max) or between different output units (I-Max).

An alternative to these procedures is to try to minimise the *information loss* across a network [Plumbley, 1987, Plumbley and Fallside, 1988a]

$$\Delta I = I(X; \Omega) - I(Y; \Omega) \tag{2.28}$$

where $I(X; \Omega)$ is the information in the input of the network about some feature variable Ω, and $I(Y; \Omega)$ is the remaining information at the output of the network about this feature variable. For most purposes, this is equivalent to Linsker's "Infomax" principle [Linsker, 1988b, Linsker, 1988a] that the information transmitted through the network $I(Y; \Omega)$ should be maximised; however, we prefer the information loss formulation, (for reasons which should become apparent when we consider non-Gaussian distributions). Linsker has successfully applied his Infomax principle to linear filters [Linsker, 1989a] and the generation of Kohonen-like ordered maps [Linsker, 1989c].

It is also possible to maximise $I(Y;\Omega)$ more directly, in a supervised learning scheme, where we have access to the feature variable Ω as a *teacher*. We maximise $I(Y;\Omega)$, hence minimising ΔI, by making Y follow Ω as closely as possible.

In the following, we shall investigate the consequences of attempting to minimise information loss ΔI, initially for supervised neural networks, and then in unsupervised networks. We shall see that some familiar learning algorithms can be shown to decrease information loss as they proceed, and it is possible to modify old algorithms, and design new ones, in order to fulfil this requirement to minimise the information loss across the network.

2.4 Information Loss across a Network

2.4.1 Properties of Information Loss

The simple information loss measure which we are considering has several useful properties which help with our consideration of unsupervised (and supervised) networks. Let us write

$$\Delta I_\Omega(X,Y) = I(X;\Omega) - I(Y;\Omega) \tag{2.29}$$

for the information lost about Ω across the transform $X \rightarrow Y$.

Non-negative across a transform

Firstly, for any transform $Y = f(X)$, from (2.13) we know that $I(Y;\Omega) \leq I(X;\Omega)$, so the expression (2.29) is non-negative, with equality when $f(\cdot)$ is reversible.

Non-negative with additive noise

Consider a system with additive noise $Y = X + \Phi$, where Φ is independent of Ω and X, we can write

$$\begin{aligned}\Delta I_\Omega(X,Y) &= I(X;\Omega) - I(Y;\Omega) \\ &= H(X) - H(X,\Omega) - (H(Y) - H(Y,\Omega)).\end{aligned} \tag{2.30}$$

Since Φ is independent of X and Ω, $H(X,\Phi) = H(X) + H(\Phi)$, and $H(X,\Phi,\Omega) = H(X,\Omega) + H(\Phi)$. Thus, remembering that the transform $(X,\Phi) \rightarrow (X,Y)$ has determinant 1, we can rewrite (2.30) as

$$\begin{aligned}\Delta I_\Omega(X,Y) &= H(X,\Phi) - H(X,\Phi,\Omega) - H(Y) + H(Y,\Omega) \\ &= H(X,Y) - H(X,Y,\Omega) - H(Y) + H(Y,\Omega) \\ &= I(X;(Y,\Omega)) - I(X;Y) \\ &\geq 0\end{aligned} \tag{2.31}$$

again from (2.13), since Y is a transform from the pair (Y,Ω).

This result suggests that unsupervised learning to minimise this information loss, given that only X and Y are available (and not Ω), may be possible under certain circumstances. If we could keep the term $I(X;(Y,\Omega))$ fixed, or at least place an upper bound on it, by maximising $I(X;Y)$ we could minimise or place a lower bound on $\Delta I_\Omega(X,Y)$. We shall return to this idea later, when we consider principal component analysing networks.

Additive in series

Over a chain of $n-1$ networks transforming X_1 to $X_2, \cdots, X_{n-1}$ to X_n, we can verify that

$$\Delta I_\Omega(X_1,X_N) = \sum_{i=1}^{n-1} \Delta I_\Omega(X_i,X_{i+1}) \tag{2.32}$$

so the information lost across any transform in this series can never be regained. Each transform should therefore attempt to minimise its own information loss, to try to minimise the information lost over the whole chain.

Of course, in trying to minimise the information loss across itself, it is possible that an early network in the chain may transform the data into a form which would make it more difficult for a later network to minimise its own information loss. For example, suppose the first network in a chain has two outputs, and learns that it can minimise its information loss if it sends all its data down only one of its outputs. Also suppose that the following network always ignores the output chosen by the preceding network: its algorithm doesn't allow it to 'listen' to that particular input. In this case, the whole system would do better if the first network chose a suboptimal solution, allowing the second network to extract at least some information from the input which it does listen to. However, provided we make certain symmetry assumptions about the classes of functions that the networks are allowed to take (e.g. the networks should be capable of using all available inputs), this problem should not arise.

2.4.2 Minimisation of Information Loss by Supervised Learning

A direct method to minimise the information loss

$$\Delta I_\Omega(X,Y) = I(X;\Omega) - I(Y;\Omega)$$

is to attempt to maximise $I(Y;\Omega)$ if the Ω is available at the output of the network as a teacher. The information in the input to the network, $I(X;\Omega)$ is independent of the network weights, but the output Y is determined according to the network function $f_W(\cdot)$ for a weight vector (or matrix) W, such that

$$Y = f_W(X)$$

so the information $I(Y;\Omega)$ at the output of the network is dependent on f_W, and hence W. Now, we can write

$$I(Y;\Omega) = H(\Omega) - H(\Omega|Y)$$

where $H(\Omega)$ is fixed, if we minimise the uncertainty in Ω given Y, $H(\Omega|Y)$, we will in turn minimise the information loss $\Delta I_\Omega(X,Y)$. For supervised learning, we try to minimise this term $H(\Omega|Y)$ by making Y follow Ω as 'closely' as possible.

Bayesian classification: Two classes

Suppose that the output from our transform is a single binary random variable Y, which is attempting to match a binary target Ω (i.e. both Y and Ω take values in {0,1}). Define the Bayesian error

$$\Phi = \begin{cases} 0 & \text{if } Y = \Omega \\ 1 & \text{otherwise} \end{cases}$$

from which we can verify that

$$H(\Omega, Y) = H(\Phi, Y)$$

so

$$\begin{aligned} H(\Omega|Y) &= H(\Omega,Y) - H(Y) \\ &= H(\Phi,Y) - H(Y) \\ &= H(\Phi|Y) \\ &\leq H(\Phi) \end{aligned} \tag{2.33}$$

with equality when Φ is independent of the estimator Y. Now, if $p_i = \Pr(\Phi = i)$ for $i \in \{0, 1\}$, we can write

$$H(\Phi) = -\sum_{i=0}^{1} p_i \log p_i$$

which monotonically increases with the Bayesian probability of error, p_1, if $p_1 \leq 1/2$. Thus if we minimise the probability of error, we will minimise the upper bound on the information loss about Ω across any network with output Y. By minimising the probability of error, we adopt a minimax strategy to the minimisation of information loss across the transform.

Bayesian classification: n classes

The analysis for n classes is more involved. The distribution of 'wrong' classes becomes important, with the most effective minimisation when the errors are as evenly distributed between the classes as possible, or at least, the errors are consistently distributed amongst the other classes.

To see this, consider a set of n possible outcomes $1, \ldots, n$, with an error function $\Phi = (\Omega - Y) \bmod n$. Now,

$$H(\Phi) = p_0 \log(1/p_0) + (1 - p_0) \log(1/(1 - p_0)) + (1 - p_0) H(\Phi|\Phi \neq 0)$$

where $H(\Phi|\Phi \neq 0)$ is the conditional entropy of Φ given that we know that $\Phi \neq 0$, i.e. we know that *some* error was made. In other words, this is the entropy of errors within the 'wrong' classes. When $H(\Phi|\Phi \neq 0)$ is maximal, i.e. when the probability of making an error into any of the other classes is equal, minimising the Bayesian error probability will bound the information loss, provided the probability of error $\Pr(\Phi \neq 0) \leq (n-1)/n$. However, if the distribution of errors is not uniform, the information loss will not be tightly bound in this way. In the extreme case of many classes, where there is an implicit ordering among the classes, Bayesian classification would no longer appropriate.

This has implications for speech processing, for example, where it is common to attempt to minimise the probability of a phoneme error, where the number of possible phonemes could be as high as 60 or more [Robinson and Fallside, 1990, Lippmann, 1989]. For this task, it is highly likely that the error distribution is *not* uniform, and not consistent for each true phoneme, since it is easier to confuse 'close' vowels than some consonants, for example.

If the minimum Bayesian error probability is the final output required from the system, this is no problem, but in this case we might like to use this phoneme recogniser as a building block as part of a word or sentence recogniser. Minimising the Bayesian error probability at the phoneme level will not guarantee that we minimise the information loss across the phoneme recogniser stage, and we may not get the performance we would like from a complete system as a word recogniser.

Essentially, the 'pick the biggest' transformation loses a lot of information, and if we would like the whole system to perform better, we should keep more for the next stage. One way of doing this is to keep the probabilities of some $N > 1$ most probable outputs: this will be optimal if the probabilities of making an error to any of the other $n - N$ outputs are equal, and will thus be a better fit than if we simply kept the identity of the top one. This idea is related to the 'N-BEST' technique used in speech recognition [Young, 1984, Schwartz and Austin, 1991].

2.4.3 Minimising Information Loss by Minimising Distortion

In addition to the Bayesian error minimisation techniques outlined above, we can also bound our information loss target by minimisation some distortion measure between the output and target values, in the case of continuous-valued outputs. This type of distortion minimisation is used when training Multi-Layer Perceptrons with the Error Back-Propagation algorithm, for instance [Rumelhart *et al.*, 1986].

General Mean Squared Error

If we write the error $\Phi = \Omega - Y$, where Y is the output of the network, and Ω is the target, then we have

$$H(\Omega|Y) \leq H(\Phi)$$

with equality if Φ is independent of Y. If we could minimise $H(\Phi)$ directly, we could then bound the information loss as before, by bounding $H(\Omega|Y)$. Unfortunately, in contrast to the discrete case, it is no longer possible to minimise $H(\Phi)$ directly. However, we can *bound* the value of $H(\Phi)$ by minimising the mean squared error, as follows.

If the covariance matrix of Φ is given by $\Sigma_\Phi = E(\Phi\Phi^T)$, then the entropy of Φ is bounded by

$$H(\Phi) \leq \frac{1}{2} \log \left((2\pi e)^N \det \Sigma_\Phi \right)$$

where N is the dimensionality of the output vector (and hence Φ), with equality if Φ has a multivariate normal distribution [Shannon, 1948]. Also, since we can bound the value of $\det \Sigma_\Phi$ by bounding the trace, $\mathrm{tr}\Sigma_\Phi$, we can bound $H(\Phi)$ by bounding the mean squared error

$$\begin{aligned} D_{\mathrm{MSE}}(\Omega, Y) &= E(|\Phi|^2) \\ &= \mathrm{tr}(\Sigma_\Phi) \end{aligned}$$

providing Φ has zero mean.

The best fit we will get on all of these bounds occurs when the error Φ is zero mean spherical Gaussian, and independent of the estimator Y. If these conditions are not satisfied, the bounds so produced will not be tight, and we may not minimise the information loss even if we do minimise the mean squared error. We can summarise the bounds we use in the I-divergence diagram, or *I-Diagram* in Fig. 2.4. The vertices give probability distributions, and the edges give I-divergences between the probability distributions at the two edges (with the 'true' distribution at the arrow head). In this diagram, the notation

$$\prod_i P(X_i)$$

represents the probability distribution over the vector $X = (X_1, \ldots, X_N)$, with

$$P(X) = P(X_1)P(X_2)\cdots P(X_N).$$

Since the probability distributions indicated are all maximum entropy (or minimum cross-entropy) distributions, the I-divergences on the arrows on this diagram are additive, even though I-divergences are not additive in general [Csiszár, 1975, Shore and Johnson, 1981].

We can see from this diagram that there is a large amount of room for 'slop': i.e. minimising the mean squared error will not guarantee to maximise $I(\Omega; Y)$, at the bottom of the diagram, unless the error $\Phi = \Omega - Y$ is independent spherical Gaussian.

Matching probabilities

If the target values for Ω in the previous section represent a probability vector, the situation is simpler than in the general case above. In particular, it is often the case that one of the target values is 1 (on), representing the class to be associated with the network input, while the other outputs are off (zero). In fact, the mean squared error will be minimised when the outputs form the a-posteriori Bayesian probability estimates of the output class given the input, $Y_i = P(\Omega_i|X)$, *if this is possible* [Hampshire and Pearlmutter, 1990]. The same is true for the component-wise cross-entropy measure introduced by Solla, Levin and Fleisher (CE-SLF) [Solla *et al.*, 1988]:

$$D_{\mathrm{CE-SLF}}(\Omega, Y) = \sum_{i=1}^{N} \Omega_i \log \frac{\Omega_i}{Y_i} + (1 - \Omega_i) \log \frac{1 - \Omega_i}{1 - Y_i}.$$

If the network is not sufficiently 'flexible' to form the exact a-posteriori probabilities, it will attempt to form a minimum mean-squared-error or minimum cross-entropy approximation to this value, depending on the distortion measure in use. Fig. 2.5 shows an I-diagram for the CE-SLF distortion measure, from which we can see that this is simpler than the general case for minimum mean squared error. In this case, this measure attempts to minimise the sum of the information loss across the transform from X to each output Y_i. This does not guarantee that a 'pick the best' from the output of a network trained using CE-SLF will have a Bayesian error rate less than one trained with the normal MSE measure. In fact, the opposite may be true, since Bayesian probability of

$$
\begin{tikzcd}
N(0, N\bar{\sigma}) \arrow[r, "(D_{\mathrm{MSE}}(\Omega, Y))"] \arrow[d, equal] & -\infty & \\
N(0, \bar{\sigma})^N \arrow[d] & & \\
P\left(\left(1/N \sum_i (\Omega_i - Y_i)^2\right)^{1/2}\right)^N \arrow[d] & & \\
\prod_i P(\Omega_i - Y_i) \arrow[r, "\sum_i I(\Omega_i - Y_i; Y_i)"] \arrow[d] & \prod_i P(\Omega_i | Y_i) \arrow[d] & \\
P(\Omega - Y) \arrow[r, "I(\Omega - Y; Y)"] & P(\Omega | Y) = P(\Omega - Y | Y) \arrow[d, equal] & \\
P(\Omega) \arrow[r, "\boxed{I(\Omega; Y)}"] \arrow[d, "H(\Omega)"'] & P(\Omega | Y) \arrow[r, "H(\Omega | Y)"] & P(\Omega | Y, \Omega) \arrow[d, equal] \\
0 & & 0
\end{tikzcd}
$$

Fig. 2.4: I-diagram for General Mean Squared Error. In this form of I-diagram, the vertices are each labelled with a probability density (PD). The arrows between edges, where labelled, give a particular interpretation for the I-divergence between the PDs at each end of the arrows: for example, the I-divergence between $P(\Omega - Y)$ and $P(\Omega - Y|Y)$ is $I(\Omega - Y; Y)$, the mutual information between $\Omega - Y$ and Y. If the label on an arrow appears in parenthesis (as does $D_{\mathrm{MSE}}(\Omega, Y)$ on this diagram), this is not the true I-divergence itself, but a number which determines the I-divergence. A special vertex marked "0" denotes a discrete probability distribution which is fully determined (i.e. $P(x_i) = 1$ for some i). Similarly, "$-\infty$" is used to denote a probability density which is a delta function. This diagram shows that minimising the mean squared error distortion, at the top of the diagram, only loosely maximises the mutual information $I(\Omega; Y)$, at the bottom of the diagram. In particular, the form of the PD $P(\Omega - Y)$ will determine how loose the connection between $D_{\mathrm{MSE}}(\Omega, Y)$ and $I(\Omega; Y)$ will be: i.e. how much 'slop' exists in the system.

$$
\begin{tikzcd}
\prod_i((1-Y_i),Y_i) \arrow[r, "D_{\mathrm{CE-SLF}}(\Omega,Y)-C"] \arrow[d] & \prod_i((1-P(\Omega_i|X)),P(\Omega_i|X)) \arrow[r, "\prod H(\Omega_i|X)"] \arrow[d, equal] & \prod_i((1-\Omega_i),\Omega_i) \arrow[d, equal] \\
\prod_i P(\Omega_i|Y_i) \arrow[r, "\boxed{\sum_i \Delta I_\Omega(X,Y_i)}"] & \prod_i P(\Omega_i|X) \arrow[r, "\prod H(\Omega_i|X)"] \arrow[d, "\sum_i D_I(\Omega,\Omega_i|X)"'] & \prod P(\Omega_i|X,\Omega_i) \arrow[d, equal] \\
P(\Omega) \arrow[r, "I(\Omega;X)"] \arrow[d, "H(\Omega)"'] & P(\Omega|X) \arrow[r, "H(\Omega|X)"] & P(\Omega|X,\Omega) \arrow[d, equal] \\
0 & & 0
\end{tikzcd}
$$

Fig. 2.5: I-Diagram for Solla, Levin and Fleisher Cross-Entropy.

error is the mean-squared-error between two binary vectors, each with one component at one and the others at zero. From our information-loss viewpoint, however, the CE-SLF measure represents an improvement over MSE.

A further improvement can be gained if we realise that the CE-SLF measure is treating outputs as individual probability estimators, with the 'on' probability being Y_i, and the 'off' probability being $1 - Y_i$. In this case, we need only ensure that $0 \leq Y_i \leq 1$ for this to be a valid probability distribution. If, however, we were to normalise the outputs so that $\sum_i Y_i = 1$, then we could compare the probability vector Y to the actual probability vector Ω directly, using the distortion measure

$$D_{\mathrm{CE-MMI}}(\Omega, Y) = \sum_{i=1}^{N} \Omega_i \log \frac{\Omega_i}{Y_i} \tag{2.34}$$

as shown in Fig. 2.6. Bridle [1990] has taken this approach, using his 'softmax' function

$$
\begin{tikzcd}
 & (Y_1,\ldots,Y_N) \arrow[r, "D_{\mathrm{CE-MMI}}(\Omega,Y)"] \arrow[d] & (\Omega_1,\ldots,\Omega_N) \arrow[d, equal] \\
P(\Omega) \arrow[r, "\boxed{I(\Omega;Y)}"] \arrow[d, "H(\Omega)"'] & P(\Omega|Y) \arrow[r, "H(\Omega|Y)"] & P(\Omega|Y,\Omega) \arrow[d, equal] \\
0 & & 0
\end{tikzcd}
$$

Fig. 2.6: I-diagram for CE-MMI distortion measure

$$Y_i = \frac{\exp(O_i)}{\sum_j \exp(O_j)} \tag{2.35}$$

where O_i are the activations of the final layer units. This ensures the outputs are positive and sum to one, and 'probability scoring' which is the cross-entropy measure in (2.34), and is related to MMI (Maximum Mutual Information) training in Hidden Markov Models (HMMs) [Bahl *et al.*, 1986].

This softmax function combines well with the cross-entropy measure, since if Y is given by (2.35), and the error

$$S = E\left(D_{\mathrm{CE-MMI}}(\Omega, Y)\right),$$

then [Bridle, 1990]

$$\frac{\partial S}{\partial O_i} = \Omega_i - Y_i \tag{2.36}$$

which is a generalisation of a similar result for the combination of the logistic sigmoid function with the CE-SLF distortion measure.

The Perceptron Learning Procedure from Cross-Entropy

This result of (2.36) above is independent of any scaling applied to O_i before the softmax function, i.e. if we use

$$Y_i = \frac{\exp(kO_i)}{\sum_j \exp(kO_j)} \tag{2.37}$$

for any constant k. For some small ϵ, provided $|O_i - O_j| \geq \epsilon$ for all i and j, and for all input presentations to the network, as $k \to \infty$ we get

$$Y_i \to \begin{cases} 1 & \text{if } O_i > O_j \text{ for all } j \neq i \\ 0 & \text{otherwise} \end{cases}$$

but the error derivative remains proportional to the original, i.e.

$$\frac{1}{k}\left(\frac{\partial S}{\partial O_i}\right) = \Omega_i - Y_i$$

so if the outputs Ω_i are either zero or one, the derivative (scaled by $1/k$) will be 0 if the network output is correct, and 1 if it is incorrect. For the Gaussian radial basis functions (RBFs) considered by Bridle [1990], this leads to a rule which moves the centres of the RBFs until they are at the centroids of the *incorrectly* classified data.

It is easy to see, however that the same reasoning will lead to the Perceptron Learning Rule, i.e. for the single-output system

$$O = W^T X$$

with

$$Y = \begin{cases} 1 & \text{if } O > 0 \\ 0 & \text{otherwise} \end{cases},$$

the learning rule

$$\Delta W = \begin{cases} 0 & \text{if } Y = \Omega \\ (\Omega - Y)X & \text{otherwise} \end{cases}$$

is equivalent to a steepest decent procedure using the componentwise cross entropy measure $D_{\mathrm{CE-SLF}}(\Omega, Y)$ and the sigmoid output function

$$Y = \frac{1}{1 + \exp\left(-kW^T X\right)}$$

as $k \to \infty$.

2.4.4 Information Loss and Infomax

The minimisation of information loss criterion [Plumbley, 1987, Plumbley and Fallside, 1988a] for a network from X to Y,

$$\Delta I_{\Omega}(X, Y) = I(\Omega; X) - I(\Omega; Y)$$

and Linsker's 'Infomax' criterion [Linsker, 1988b, Linsker, 1988a], i.e. maximisation of

$$I(\Omega; Y)$$

are apparently identical, since $I(\Omega; X)$ is independent of any transform we choose from X to Y. The advantage of minimising loss of information rather than simply maximising final information arises when we only measure certain parameters of the probability distributions of X and Y: commonly we only measure the covariance matrices of these distributions.

Consider the single system in Fig. 2.7. For this system, we have $X = \Omega + \Phi_X$, and $Y = X + \Phi_Y$,

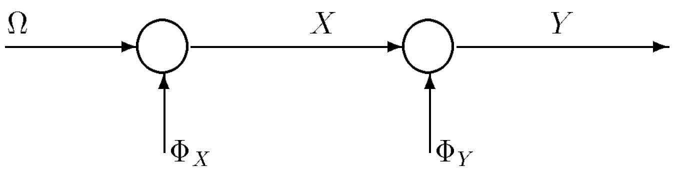


Fig. 2.7: Information loss with additive noise

i.e.

$$Y = \Omega + (\Phi_X + \Phi_Y)$$

where Φ_X and Φ_Y are independent additive noise terms.

Infomax for non-Gaussian signals

For the Infomax criterion, we will be measuring

$$I(\Omega; Y) = H(Y) - H(\Phi_X + \Phi_Y)$$

with a view to maximising this quantity. Since Φ_X and Φ_Y are fixed, this will be equivalent to maximising $H(Y)$.

Suppose that for simplicity we assume Y to be zero mean, and we only measure the variance of Y, $\sigma_Y^2 = E(Y^2)$. This will place an *upper bound* of

$$H_{\max}(Y) = \frac{1}{2} \log \left(2\pi e \sigma_Y^2\right)$$

on the entropy $H(Y)$, thus placing an upper bound on the transmitted information $I(\Omega; Y)$. This upper bound will be achieved for the Gaussian density $N(0, \sigma_Y^2)$, but any other distribution will have less entropy, and hence less transmitted information. Hence if we only measure the variance, we can only maximise the transmitted information if the output Y is Gaussian: for any other probability distribution for Y, all we know is that we have maximised the *upper bound* on $I(\Omega; Y)$.

Information loss for non-Gaussian signals

If instead we consider the information *loss*

$$\begin{aligned} \Delta I_\Omega(X, Y) &= I(\Omega; X) - I(\Omega; Y) &\quad (2.38) \\ &= H(X) - H(Y) - H(\Phi_X) + H(\Phi_X + \Phi_Y) &\quad (2.39) \end{aligned}$$

we can improve upon the situation outlined above. Essentially, if the noise term Φ_Y is Gaussian, for Y to be non-Gaussian, X must be non-Gaussian also. So if $I(\Omega; Y)$ is less than the maximum value for a particular variance measurement, so $I(\Omega; X)$ will also be less than its maximum.

Let us look at the way that $\Delta I_\Omega(X, Y)$ varies with $p_X(X)$ such that the constraints

$$\int p_X(x)\, dx = 1 \tag{2.40}$$

and

$$E(X^2) = \sigma_X^2 \tag{2.41}$$

hold. We can use the calculus of variations to find the maximum information loss for a given σ_Y, and hence σ_X. Provided the output noise Φ_Y is Gaussian with zero mean, we find

$$\begin{aligned} \frac{d}{dp_X(x)}\Delta I_\Omega(X,Y) &= -\log p_X(x) + \int p_{\Phi_Y} \log p_Y(x+\phi) + 1 && (2.42) \\ &= -\log p_X(x) - H(\Phi_Y) - D_I(p_{\Phi_Y}, p_{Y-x}) + 1 && (2.43) \end{aligned}$$

where $p_{Y-x}(\phi) = p_Y(\phi + x)$, and $D_I(\cdot,\cdot)$ is the I-divergence measure described in section 2.1.5. Note that the form of the input noise Φ_X does not appear in this expression: it need only be additive. To maximise $\Delta I_\Omega(X,Y)$ under the constraints (2.40) and (2.41), we should maximise

$$J = \Delta I_\Omega(X,Y) - \lambda \int p_X(x)\,dx - \gamma \int x^2 p_X(x)\,dx$$

over $p_X(x)$, for lagrange multipliers λ and γ. This leads to the condition

$$\begin{aligned} \frac{d}{dp_X(x)}J &= -\log p_X(x) - H(\Phi_Y) - D_I(p_{\Phi_Y}, p_{Y-x}) + 1 - \lambda - \gamma x^2 \\ &= 0 \end{aligned}$$

which we can verify is satisfied for Gaussian output noise

$$p_{\Phi_Y}(\phi_Y) = \frac{1}{\sqrt{2\pi\sigma^2_{\Phi_y}}} \exp\left(-\frac{1}{2}\frac{\phi_y^2}{\sigma^2_{\Phi_Y}}\right)$$

when

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma_x^2}\right). \quad (2.44)$$

If the second derivative of J is negative definite in the direction of our constraints, i.e.

$$\begin{aligned} F &= \int\int (\Delta p_X(x))\,(\Delta p_X(x'))\,\frac{d^2}{dp_X(x)dp_X(x')}J\,dx\,dx' && (2.45) \\ &\leq 0 && (2.46) \end{aligned}$$

for any $\Delta p_X(X) = p_X^+(X) - p_X^-(X)$ such that (2.40) and (2.41) hold for both p_X^+ and p_X^-, then the solution of (2.44) will be a maximum for the information loss. For this second derivative, we find

$$\begin{aligned} \frac{d^2}{dp_X(x)dp_X(x')}J &= -\frac{1}{p_X(x)}\delta(x-x') + \int p_{\Phi_Y}(\phi)\frac{1}{p_Y(\phi+x)}p_{\Phi_Y}(\phi+x-x')d\phi \\ &= -\frac{\delta(x-x')}{p_X(x)} + \int \frac{p_{\Phi_Y}(y-x)p_{\Phi_Y}(y-x')}{p_Y(y)}dy && (2.47) \end{aligned}$$

which is symmetrical in x and x'. Thus we get

$$\begin{aligned} F &= \int\int\int (\Delta p_X(x))\,(\Delta p_X(x'))\left(-\frac{\delta(x-x')}{p_X(x)} + \frac{p_{\Phi_Y}(y-x)p_{\Phi_Y}(y-x')}{p_Y(y)}\right)dx\,dx'\,dy \\ &= \int \frac{(\Delta p_Y(y))^2}{p_Y(y)}\,dy - \int \frac{(\Delta p_X(x))^2}{p_X(x)}\,dx && (2.48) \end{aligned}$$

where

$$p_Y(y) = \int p_{\Phi_Y}(\phi)p_X(y-\phi)\,d\phi$$

and

$$\Delta p_Y(y) = \int p_{\Phi_Y}(\phi)\Delta p_X(y-\phi)\,d\phi.$$

So for the Gaussian to be a maximum, F must be strictly negative, so we must have

$$\int \frac{(\Delta p_X(x))^2}{p_X(x)}\,dx > \int \frac{(\Delta p_Y(y))^2}{p_Y(y)}\,dy. \tag{2.49}$$

We can verify that this is possible in the case where $p_X(x)$ and $p_Y(y)$ are approximately constant whenever $\Delta p_X(x)$ and $\Delta p_Y(y)$ are significant, and the noise Φ is small. For Gaussian $\Delta p_X(x)$, we find that

$$\int (\Delta p_X(x))^2 \, dx \propto \frac{1}{\sigma_{\Delta X}}$$

and, since the variance of $\Delta p_Y(y)$ would be greater than $\Delta p_X(x)$, we will have

$$\int (\Delta p_X(x))^2 \, dx > \int (\Delta p_Y(y))^2 \, dy.$$

This is not a proof that $F < 0$, but it does lead us to conjecture that F in (2.48) is negative.

If this is the case, the Gaussian probability density for X (and hence Y, since Φ_Y is Gaussian), would be not only the maximum transmitted information condition, but also the maximum information *loss* condition. Thus by minimising the information loss for a Gaussian distribution, we would minimise the upper bound on the information loss across corrupting Gaussian noise: a 'damage limitation' approach.

2.4.5 Applying Information Loss

In practice, the application of information loss minimisation is essentially identical to Linsker's 'Infomax' [Linsker, 1989c, Linsker, 1989b], except we bear in mind that if we always deal with Gaussian distributions, we are upper-bounding the loss of information rather than upper-bounding the transmitted information. This upper-bounding also applies when certain components of the input are ignored completely, as we shall see when we consider principal component analysis networks.

Most of the following sections on information loss minimisation will concentrate on linear unsupervised learning systems, for analytical tractability. In all these sections, the assumptions that we make about the input noise are always important. Conceptually, these assumptions make up for the fact that we are not allowed access to Ω, which would supply our targets in a supervised learning system.

Chapter 3

Principal Component Analysis

Principal Component Analysis is a popular statistical tool for removing redundancy from data in a linear fashion, and deserves special mention. It has various names in different signal and data processing fields, including Factor Analysis, (discrete) Karhunen-Loève Transform (KLT), and the Hotelling Transform in image processing [Watanabe, 1985, Gerbrands, 1981, Gonzalez and Wintz, 1987]. Various unsupervised algorithms have already been suggested for a linear neural network to find the Principal Components or Principal Subspace of their input data [Oja, 1982, Oja, 1983, Oja and Karhunen, 1985, Williams, 1985, Bourlard and Kamp, 1987, Földiák, 1989, Sanger, 1989a], so this represents an important aspect of our consideration of unsupervised learning in general.

3.1 Transforming for Principal Components

Consider the N-input, M-output linear network

$$y_j = \sum_{i=1}^{N} w_{ij} x_i \tag{3.1}$$

where x_i is the ith input unit, y_j is the jth output unit, and w_{ij} is the weight connecting the ith input to the jth output. Alternatively, this can be written in matrix notation as

$$\mathbf{y} = W^T \mathbf{x} \tag{3.2}$$

where $\mathbf{x}$ is the N-dimensional input vector and $\mathbf{y}$ is the M-dimensional output vector, and W is an $N \times M$ weight matrix (Fig. 3.1). The inputs $\mathbf{x}$ are instances of some random variable X with mean $\boldsymbol{\mu}_x = E(\mathbf{x})$ and covariance matrix $\Sigma_x = E((\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{x} - \boldsymbol{\mu}_x)^T)$. We shall assume for simplicity that $\boldsymbol{\mu}_x = 0$ (we can always force this to be the case by centralizing the input vector to produce a modified input $\tilde{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}_x$ before applying the weight matrix).

Let ζ_i and λ_i, $i = 1, 2, \ldots, N$, be the eigenvectors and corresponding eigenvalues of the covariance matrix Σ_x, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. If $M = N$, and the columns of W (rows of W^T) are the eigenvectors ζ_i in order, then this network is said to perform a Karhunen-Loève Transform (KLT) on the input vectors $\mathbf{x}$ [Gerbrands, 1981].

3.1.1 Properties of the KLT

Since the covariance matrix Σ_x of the zero-mean input vectors is real and symmetric, its eigenvectors ζ_i form an orthonormal set. Thus $WW^T = W^T W = I_N$, and the output covariance matrix

$$\begin{aligned} \Sigma_y &= W^T \Sigma_x W && (3.3) \\ &= W^T W \Lambda && (3.4) \\ &= \Lambda && (3.5) \end{aligned}$$

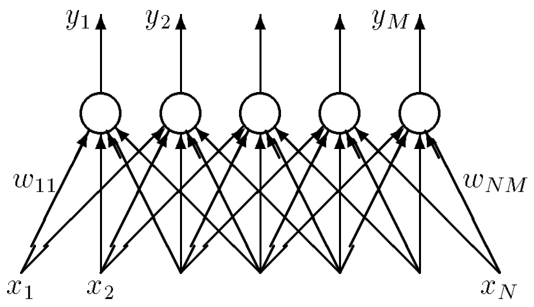


Fig. 3.1: A Single Layer Linear Network.

where

$$\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix} \tag{3.6}$$

Thus the output has the properties of

Decorrelation The activations of the output units are uncorrelated; and

Decreasing Variance The variances of the output units are simply the eigenvalues λ_i of the input covariance matrix, and monotonically decrease with output unit number.

The decorrelation property of the outputs is important in Factor Analysis [Watanabe, 1985], since the object here is to attempt to identify independent factors which combine in a linear fashion to give rise to the observed data. In addition, since $WW^T = I_N$, the input $\mathbf{x}$ can be reconstructed from the output $\mathbf{y}$ using

$$\mathbf{x} = W\mathbf{y} \tag{3.7}$$

If there are fewer output units available than input units ($M < N$) we can simply drop the higher numbered outputs $M < j \leq N$ to give us

$$\mathbf{y} = W_M^T \mathbf{x} \tag{3.8}$$

where W_M is the $N \times M$ matrix whose M columns are the first M eigenvectors of Σ_x. The outputs will still be uncorrelated, with $W_M^T W_M = I_M$, but now W_M^T is not invertible, so we can now only estimate the input from the output

$$\hat{\mathbf{x}} = W_M \mathbf{y} \tag{3.9}$$

with squared error

$$\begin{aligned} S_{\text{MSE}} &= \operatorname{tr}\left(E\left((\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T\right)\right) && (3.10) \\ &= \operatorname{tr}\left(\Sigma_x(I_N - W_M W_M^T)\right) && (3.11) \\ &= \sum_{i=M+1}^{N} \lambda_i. && (3.12) \end{aligned}$$

Since the eigenvalues were chosen to be monotonically decreasing, this represents a least-squared-error (LSE) reconstruction for $\mathbf{x}$ from $\mathbf{y}$: the **LSE reconstruction** property.

3.1.2 The Projection Induced by W_M

If we are only interested in the LSE reconstruction property, and not the decorellation property, we only need to find the principal subspace rather than all the principal components. To see this, we consider the *projection* induced by the W_M, which describes the principal subspace.

If we let $P_M = W_M W_M^T$, we can verify that $P_M^2 = P_M^T = P_M$. Thus P_M is an *orthogonal projection* [Golub and van Loan, 1983] onto the range of W_M^T (In the particular case of $N = M$, $P_M = I_N$).

Now we can rewrite the reconstructed signal $\hat{\mathbf{x}}$ as

$$\begin{aligned}\hat{\mathbf{x}} &= W_M W_M^T \mathbf{x} &(3.13)\\ &= P_M \mathbf{x} &(3.14)\end{aligned}$$

with reconstruction mean squared error as

$$S = \operatorname{tr}((I - P_M)\Sigma_x) \tag{3.15}$$

Thus P_M is the projection on to the principal eigenspace of $\mathbf{x}$.

In fact, it can be shown that for any full rank $N \times M$ matrix W whose columns span the same subspace as any M eigenvectors of Σ_x, the LSE reconstruction is given by

$$\hat{\mathbf{x}} = P\mathbf{x} \tag{3.16}$$

where $P = (W^+)^T W^T$, with $W^+ = (W^T W)^{-1} W^T$, the Moore-Penrose psuedo-inverse of W [Strang, 1976] (where $(W^T W)^{-1}$ exists). This gives the same reconstruction squared error as above. Thus if we are only interested in the minimum squared error, we can use any weight vector which induces the same projection P_M onto the M-dimensional principal eigenspace, i.e. any matrix W whose columns span the same subspace as the M principal eigenvectors of $\mathbf{x}$.

Of course, using any general matrix W which spans the principal eigenspace is not guaranteed to give us uncorrelated outputs. If the object of the exercise is simply to produce an output $\mathbf{y}$ which can be used to reconstruct the input with least squared error, any correlation at the outputs is of no consequence. We shall re-examine this point later.

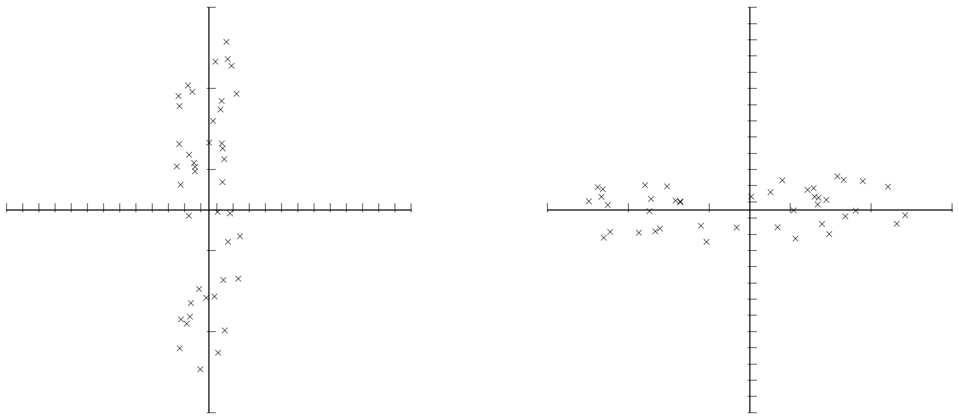
3.1.3 Problems With PCA

The application of Principal Components Analysis to real data suffers from the *scaling problem*, i.e. the principal components produced depend on the way the data is scaled. As a simple illustration, consider the data shown in Fig. 3.2. The data points have 2 parameters, and we wish to reduce this to a single parameter using PCA. Now, if we scale the axes as shown in Fig. 3.2(a), the apparent single principal component is the one in the direction of the axis x_2. However, if they are scaled as shown in Fig. 3.2(b) the apparent principal component is the one in the direction of the axis x_1. This type of problem is also evident in other forms of data classification tasks, such as cluster analysis [Hand, 1981, p157]. Of course it would not normally be useful to attempt PCA on this data, since the values on the two axes are independent in this case, but it does serve to illustrate an extreme case of the problem.

When the axes are similar types of data, measured over the same values, the obvious solution to this is simply to use $1:1$ relative scaling between the axes. But what if the measurements are from completely different quantities, such as height vs. IQ? To avoid this problem, it is customary to normalise the axes so that each has unit variance [Watanabe, 1985, p212] before attempting Principal Components Analysis. However, the justification for this is unclear.

Even if the data is normalised so that each component has unit variance, another related problem arises if we have multiple observations of a single data variable. In our hypothetical example, suppose we have only one opportunity to measure height, but four to measure IQ. We have two obvious approaches:

1. Treat the four measurements of IQ as separate, and perform a principal component analysis on the resulting 5-dimensional data. Or;



(a) (b)

Fig. 3.2: The Scaling Problem in PCA

2. Take the mean of the four IQ measurements, and find the principal component of the remaining 2-dimensional data (using the mean as one of the components).

If we do not normalise so that each component has unit variance, we are left with our original scaling problem of how to scale the data appropriately. However, if we do normalise the variance of the data, the two approaches outlined above are not equivalent, and will give an inconsistent value for the principal component. To try to resolve these problems, we shall now consider principal component analysis from an information theory viewpoint.

3.2 Information Lost by Principal Components Analysis

As for supervised learning networks, we can construct an Information Loss framework [Plumbley, 1987, Plumbley and Fallside, 1988b] for the unsupervised linear system (3.2)

$$\mathbf{y} = W^T \mathbf{x} \tag{3.17}$$

with $M < N$. Now, we wish to minimise the information loss about the pattern instances ω which are instances of some random variable Ω. These patterns are transformed by the 'world', together with some corrupting noise, into the observed input vector instances $\mathbf{x}$ to our network. I.e.

$$X = U(\Omega, \Phi_U) \tag{3.18}$$

Unlike the supervised case, we can make no observations of the patterns Ω, so cannot minimise the information loss directly. We can make no further progress unless we can make some assumption about the relationship between Ω and X.

The assumption we make is that the input X can be considered to be corrupted by spherical additive normally distributed (Gaussian) noise Φ_X, so that an input instance is generated according to

$$\mathbf{x} = \omega_{\mathbf{x}} + \phi_{\mathbf{x}} \tag{3.19}$$

where $\omega_{\mathbf{x}}$ represents the pattern information in $\mathbf{x}$ before corruption by the noise (Fig. 3.3). Thus we have independent equal variance noise on each input. The output vectors $\mathbf{y}$ are assumed to contain no additional noise, so

$$\begin{aligned} \mathbf{y} &= W^T \mathbf{x} && (3.20) \\ &= \omega_{\mathbf{y}} + \phi_{\mathbf{y}} && (3.21) \end{aligned}$$

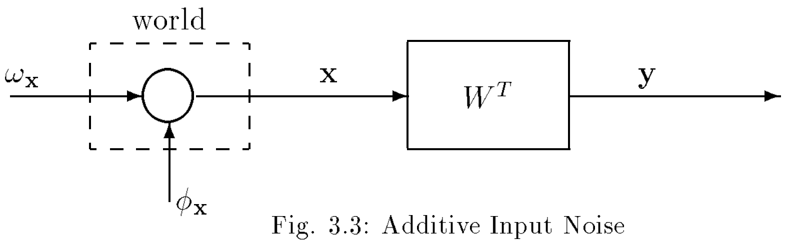


Fig. 3.3: Additive Input Noise

where $\omega_{\mathbf{y}} = W^T \omega_{\mathbf{x}}$ and $\phi_y = W^T \phi_{\mathbf{x}}$.

The information loss across the network,

$$\Delta I_\Omega(X, Y) = I(X; \Omega) - I(Y; \Omega) \tag{3.22}$$

where $I(X;\Omega)$ and $I(Y;\Omega)$ are the information in the input to the network and the output from the network respectively, about the patterns Ω. If we could measure the probability distribution of X, $P_X(\mathbf{x})$ directly, we could ignore $I(X;\Omega)$, which is not a function of W. By using a W such that $I(Y;\Omega)$ is maximised, we minimise our information loss $\Delta I_\Omega(X,Y)$.

As an example, let us restrict ourselves for the moment to unitary matrices W^T, i.e. matrices such that $W^T W = I_M$. Since the output $\mathbf{y}$ is corrupted by additive noise only, we know that

$$I(Y; \Omega) = H(Y) - H(\Phi_Y). \tag{3.23}$$

But since $W^T W$ is unitary and the input noise $\phi_{\mathbf{x}}$ is spherical Gaussian, the output noise $\phi_{\mathbf{y}} = W^T \phi_{\mathbf{x}}$ is also spherical Gaussian with covariance independent of W, so $H(\Phi_Y)$ is independent of W also[Plumbley and Fallside, 1988a]. It remains for us to find a unitary W so that $H(Y)$ is maximised.

To maximise the true $H(Y)$ would require us to accurately measure the true input probability distribution $P_X(\mathbf{x})$, and deduce which W would give rise to a maximal $H(Y)$ given this $P_X(\mathbf{x})$. Although this measurement may be possible for certain discrete distributions, for a general distribution this would be impractical, since measuring $P_X(\mathbf{x})$ accurately would be out of the question. In addition, once $P_X(\mathbf{x})$ had been measured, finding a W to maximise $H(Y)$ would not be a trivial task in itself.

One possibility is to make some *assumption* about the input probability distribution. For example, we could assume that X is normally distributed with zero mean so that $P_X(\mathbf{x}) = N(0, \Sigma_x)$. Linsker [1988a] points out that in this case the information $I(Y;\Omega)$ is maximised for a one-output system ($M = 1$) when the weight vector W is in the direction of the principal component of Σ_x. In fact, for an M-output network, the information $I(Y;\Omega)$ is maximised for normally distributed X when W spans the M-dimensional principal subspace of Σ_x. So for a zero-mean Gaussian input signal, the information loss across the network will be minimised when the outputs are the M principal components of the input, or some output which covers the same subspace. With this assumption that the input signal is normally distributed with zero mean, it is sufficient to measure its covariance matrix Σ_x to completely determine the probability distribution $P_X(X)$.

Unfortunately, this is apparently a 'best-case' assumption, which does not necessarily hold for any other input probability distribution. For any distribution $P_Y(\mathbf{y})$ with covariance matrix $\Sigma_y = W^T \Sigma_x W$, except for the Gaussian distribution with this covariance, the entropy $H(Y)$ will always be less than the entropy of the Gaussian distribution with this covariance [Shannon, 1948]. Therefore the information $I(Y;\Omega)$ in (3.23) will always be less than the information for the equivalent Gaussian signal (call this $I^*(Y;\Omega)$). We have found an *upper bound* on the information which could be transmitted through the network, so we have maximised the information capacity of the system, but there is no guarantee that a non-Gaussian signal will achieve this capacity. In game-theoretic terms, Nature could conspire against us, so that although the direction we chose was the principal component, it would not be the direction with maximum information. Formulated in this way, the principal component analysis approach does not appear as useful as it might be.

3.2.1 An Upper Bound on Information Loss

Fortunately, we can take another approach to the problem. If the signal at the output is non-Gaussian, then the signal at the input must also be non-Gaussian. So, for a given input covariance Σ_x, if the output information $I(Y;\Omega)$ is less than the capacity $I^*(Y;\Omega)$, the the input information $I(X;\Omega)$ will also be less than the input information $I^*(X;\Omega)$ for the equivalent Gaussian input signal. By considering the information loss $\Delta I_\Omega(X,Y)$ in (3.22) rather than the transmitted information $I(Y;\Omega)$ alone, it is possible to *bound* the information loss for any input probability distribution, given that only the input covariance matrix Σ_x is measured.

In other words, we wish to place the lowest bound on the information capacity lost from the input to the output, instead of placing the highest bound on the information capacity at the output itself. Noting that since Y is a function of X, we have $I(X;\Omega) = I(X,Y;\Omega)$, our information loss

$$\Delta I_\Omega(X,Y) = I(X,Y;\Omega) - I(Y;\Omega) \tag{3.24}$$

Let us express the input X as the sum of two terms

$$X = X_y + X_{\bar{y}} \tag{3.25}$$

where $X_y = W^T(W^TW)^{-1}Y$ (assuming W full rank) so that $X_y = PX$ with P an orthogonal projection. Observing that

$$H(X,Y) - H(X,Y,\Omega) = H(X_{\bar{y}},Y) - H(X_{\bar{y}},Y,\Omega) \tag{3.26}$$

since the determinants of both transforms are equal, after a little manipulation we get

$$\begin{aligned}\Delta I_\Omega(X,Y) &= I(X_{\bar{y}};Y,\Omega) - I(X_{\bar{y}};Y) && (3.27)\\ &\le I(X_{\bar{y}};Y,\Omega) && (3.28)\\ &\le I(X_{\bar{y}};\Omega) && (3.29)\end{aligned}$$

since $I(A;B) \le I(A, f(B))$ for any $f(\cdot)$. The first inequality is an equality if $X_{\bar{y}}$ is independent from Y. As before, we can write

$$X_{\bar{y}} = \Omega_{X_{\bar{y}}} + \Phi_{X_{\bar{y}}} \tag{3.30}$$

with the obvious interpretation, leading to

$$I(X_{\bar{y}};\Omega) = H(X_{\bar{y}}) - H(\Phi_{X_{\bar{y}}}). \tag{3.31}$$

Now $\Phi_{X_{\bar{y}}} = (I-P)\Phi_X$, so provided W remains full rank, the rank of the orthogonal projection $(I-P)$ remains constant, so $H(\Phi_{X_{\bar{y}}})$ remains constant. It remains for us to minimise $H(X_{\bar{y}})$.

For a Gaussian distribution, this is minimised if $X_{\bar{y}}$ consists of the smallest $N-M$ components of X, i.e. the components remaining after the M principal components have been removed. This, then is the principal component analysis solution, and coincides with the minimum mean squared reconstruction error $\mathrm{tr}(X_{\bar{y}})$. Any other probability distribution for $X_{\bar{y}}$ will have an entropy not more than this value, so the information loss $\Delta I_\Omega(X,Y)$ will be upper-bounded by the information loss for the equivalent Gaussian distribution.

Thus the solution for W^T which spans the same space as the M principal components of X is not only the solution for maximum information *capacity* $I(Y;\Omega)$, but also the solution for the lowest bound on the information *loss*, $\Delta I_\Omega(X,Y)$. So although we may not be able to guarantee that the transmitted information is maximised, we can be sure that the information lost by the network has been bounded as tightly as possible, given the parameters which we have measured.

3.2.2 Implications for the Scaling Problem

We have shown that the technique of Principal Components Analysis, or the use of any transformation matrix spanning the same subspace as the principal components of a distribution, will place an upper bound on the information lost by that transformation, provided there is additive spherical Gaussian error on the input vectors.

In fact, it is possible to relax this condition slightly [Plumbley and Fallside, 1988a], provided that the entropy of $\Phi_{X_{\bar{y}}}$, the component of the input noise Φ_X which is not a function of the output Y, has a *lower* bound. Of course, this would mean that the information loss upper bound found by the Principal Component Analysis procedure would not be as tight as if $H(\Phi_{X_{\bar{y}}})$ was constant.

We can now view the scaling problem that arises in principal component analysis as the problem of determining an appropriate scaling for the input components to get as close to spherical Gaussian noise as possible, to fit as tight a bound as possible to the information loss. If the inputs are all similar, it is reasonable to assume that the noise on all the components is equal, so 1:1 scaling is appropriate. However, if the input components are not measurements of similar quantities, we should attempt to estimate the noise, or uncertainty, in each of the components with respect to the patterns that we are interested in.

To see why this is so, imagine that one of the components of the input was virtually noise-free, while the rest had a significant amount of noise. The information conveyed by the low-noise component could be much greater than the information conveyed by the other components, even though its variance may be smaller than that of the others.

The technique of normalising each input component to unit variance is therefore, in effect, making the assumption that the variance of the noise on each input component is in proportion to the variance of the signal on that input. This is where the inconsistency arises with multiple measurements, such as the four measurements of I.Q. in our example of section 3.1.3. If the mean of the I.Q. measurements are taken before applying principal components analysis, any independent Gaussian error on the individual measurements will give rise to a Gaussian error on the mean of half the variance. Thus the mean value should be scaled to twice the scaling which would have been used for an individual measurement.

This gives a consistent way to deal with both of these cases. The proportion of each component in the final principal component is the same in both cases, provided, or course, that our assumptions about independence of measurements are correct. It suggests that the customary normalisation is often successful when the noise in each component of the data points is approximately in proportion with their variance; this may even be a reasonable assumption to make when no other information is available. However, it is now clear what assumption is being made when this normalisation is used, and in some cases it may be apparent that an alternative scaling is appropriate.

3.3 Neural Network Algorithms for PCA

Our particular interest is in unsupervised learning, so we would like a stochastic approximation algorithm, preferably using only local operations, to find the principal components or principal subspace. Several such algorithms already exist, with various proofs of convergence [Krasulina, 1970, Oja, 1982, Oja and Karhunen, 1985, Williams, 1985, Sanger, 1989a] as well as similar algorithms designed to find the *minimum* eigenvalue of the input covariance matrix, for Pisarenko's 'harmonic retrieval' method [Reddy *et al.*, 1982].

We shall see that these algorithms are members of a general class of stabilised Hebbian learning algorithms. By examining the behaviour of the subspace spanned by the transformation matrix W, or rather the behaviour of the orthogonal projection P onto that subspace, we can separate the common Hebbian part of the algorithms from the stabilisation part, and show that they produce an identical algorithm for adapting the projection P. First, let us review the algorithms in this class.

3.3.1 The Algorithms

As before, we have a linear system with $\mathbf{y}_n = W_n^T \mathbf{x}_n$ at the nth time step. The basis of many unsupervised learning algorithms is the *Hebbian* algorithm, of the form

$$W_{n+1} = W_n + \eta_n \mathbf{x}_n \mathbf{y}_n^T \tag{3.32}$$

where η_n is a small update factor, after D. O. Hebb's postulation that a cell which excites another cell should have its connections strength increased if the cells tend to fire together [Hebb, 1949]. Unfortunately, if this rule is used without modification, the weights will tend to increase without limit [von der Malsburg, 1973]. One possibility is to renormalise the weights to each output cell, to keep the sum of the weights constant [von der Malsburg, 1973], another to use a hard bound on the weight values [Linsker, 1986], in effect limiting some p-norm of the weight vector.

Krasulina's principal component finder

Krasulina [1970] suggested a stochastic approximation algorithm for computation of the single principal vector ($M = 1$)

$$\begin{aligned} W_{n+1} &= W_n + \eta_n \left(\mathbf{x}_n y_n - W_n \frac{y_n^2}{W_n^T W_n} \right) &(3.33)\\ &= W_n + \eta_n \left(\mathbf{x}_n \mathbf{x}_n^T W_n - W_n \frac{W_n^T \mathbf{x}_n \mathbf{x}_n^T W_n}{W_n^T W_n} \right) &(3.34)\end{aligned}$$

where W_n converges to a vector in the direction of the principal eigenvector of Σ_x as $n \to \infty$, with certain conditions on Σ_x, η_n and W_n. However, as pointed out by Oja and Karhunen [1985], the weight vector can grow very large with this algorithm.

The Oja single principal component neuron

Starting from the Hebbian approach, Oja [1982] observed that the application of the Hebbian algorithm (3.32) in a single output ($M = 1$) system, followed by a normalisation stage after each time step to ensure the 2-norm of W remains at unity, i.e.

$$\begin{aligned} \tilde{W}_{n+1} &= W_n + \eta_n \mathbf{x}_n y_n &(3.35)\\ W_{n+1} &= \tilde{W}_{n+1} / |\tilde{W}_{n+1}|_2 &(3.36)\end{aligned}$$

can be expanded as

$$W_{n+1} = W_n + \eta_n \left(\mathbf{x}_n y_n - W_n y_n^2 \right) + O(\eta_n^2) \tag{3.37}$$

which can be written

$$W_{n+1} = W_n + \eta_n \hat{\mathbf{x}}_n y_n + O(\eta_n^2) \tag{3.38}$$

where $\hat{\mathbf{x}}_n = \mathbf{x}_n - W_n y$. The weight vector W in this algorithm converges to a unit vector which is the dominant eigenvector of Σ_x (or the negative of this vector).

Oja and Karhunen algorithm for M components

Oja's algorithm above has been generalised by Oja and Karhunen [1985] to find the M largest eigenvalues in order with an algorithm based on the Gram-Schmidt orthogonalization (GSO) of the weight matrix W_n at each stage. A simplified version of their algorithm is

$$W_{n+1} = W_n + \eta_n \left(\mathbf{x}_n \mathbf{y}_n^T - W_n \mathrm{diag}(\mathbf{y}_n \mathbf{y}_n^T) - 2 W_n \mathrm{UT}_+(\mathbf{y}_n \mathbf{y}_n^T) \right) \tag{3.39}$$

where $\mathrm{UT}_+(\cdot)$ denotes a 'strictly upper triangular' matrix operator which sets all the entries on or below the diagonal to zero, and $\mathrm{diag}(\cdot)$ is the operator which sets all entries off the diagonal to zero.

Sanger's 'Generalized Hebbian Algorithm'

The algorithm above is similar to Sanger's 'Generalized Hebbian Algorithm' [Sanger, 1989a], which is also based on a GSO idea:

$$W_{n+1} = W_n + \eta_n \left(\mathbf{x}_n \mathbf{y}_n^T - W_n \mathrm{UT}(\mathbf{y}_n^T \mathbf{y}^n) \right) \tag{3.40}$$

where this time $\mathrm{UT}(\cdot)$ is an upper triangular operator which sets all entries below the diagonal *only* to zero (i.e. the diagonal is retained).

Williams' 'Symmetric Error Correction' Algorithm

Finally, Oja and Karhunen [1985] mention the algorithm

$$W_{n+1} = W_n + \eta_n \left(\mathbf{x}_n \mathbf{y}_n^T - W_n \mathbf{y}_n \mathbf{y}_n^T \right) \tag{3.41}$$

which can be written as

$$W_{n+1} = W_n + \eta_n \hat{\mathbf{x}}_n \mathbf{y}_n^T \tag{3.42}$$

where $\hat{\mathbf{x}}_n = \mathbf{x}_n - W_n \mathbf{y}_n^T$ (c.f. algorithm (3.38)). This is the symmetric version of Williams' 'Symmetric Error Correction' algorithm [Williams, 1985], and Oja uses this as a learning equation for his 'Subspace Network' [Oja, 1989].

3.3.2 A Common Framework

As may already be apparent, all these algorithms for our system $\mathbf{y}_n = W^T \mathbf{x}_n$ can be written in the form

$$W_{n+1} = W_n + \eta_n \left(\mathbf{x}_n \mathbf{x}_n^T W_n + W_n \Theta_n \right) + O(\eta_n^2) \tag{3.43}$$

where Θ_n is some $M \times M$ matrix which may be a function of W_n and/or $\mathbf{x}_n$. The term $W_n \Theta_n$ is sometimes called a *forgetting* or *weight decay* term [Kohonen, 1984], and embodies the intuitive idea that connections in a neural network should decay if they are not used.

We shall show that any algorithm in this class leads to identical conditions for convergence of the orthogonal projection P_n onto the range of W_n, independent of the particular choice of Θ_n. We shall show that this orthogonal projection converges to the projection $\tilde{P}_M$ onto the M-dimensional principal subspace of Σ_x.

Let us assume that the weight matrix W_n has full rank for all n. Then the orthogonal projection onto the range of W_n is

$$P_n = W_n (W_n^T W_n)^{-1} W_n^T. \tag{3.44}$$

since $P_n^2 = P$, $P_n^T = P_n$, and the ranges of P_n and W_n are equal [Golub and van Loan, 1983]. First, we show that algorithm (3.43) leads to an algorithm for P_n which is independent of Θ_n, apart from terms proportional to η_n^2 or smaller.

Theorem 3.3.1 *If $W_n^T W_n$ is invertible and both $\|W_n^T W_n\|$ and $\|(W_n^T W_n)^{-1}\|$ are bounded for all n, the algorithm (3.43) for W_{n+1} leads to the following algorithm for P_{n+1}*

$$P_{n+1} = P_n + \eta_n \Delta P_n + O(\eta_N^2) \tag{3.45}$$

where $\Delta P_n = P_n A_n (I - P_n) + (I - P_n) A_n P_n$ and $A_n = \mathbf{x}_n \mathbf{x}_n^T$.

Proof. Write (3.43) as

$$W_{n+1} = W_n + \eta_n \Delta W_n + O(\eta_N^2) \tag{3.46}$$

where $\Delta W_n = \mathbf{x}_n \mathbf{x}_n^T W_n - W_n \Theta_n$. Now let $R_n = W_n^T W_n$, so

$$\begin{aligned} R_{n+1} &= W_{n+1}^T W_{n+1} \\ &= \left(W_n + \eta_n \Delta W_n + O(\eta_n^2) \right)^T \left(W_n + \eta_n \Delta W_n + O(\eta_n^2) \right) \\ &= R_n + \eta_n \Delta R_n + O(\eta_n^2) \end{aligned}$$

where $\Delta R_n = W_n^T \Delta W_n + (\Delta W_n^T) W_n$. Thus, provided $\eta_n \|R_n^{-1}\| << 1$ for all n,

$$\begin{aligned} R_{n+1}^{-1} R_{n+1} &= I \\ &= R_{n+1}^{-1}\left(R_n + \eta_n \Delta R_n + O(\eta_n^2)\right) \end{aligned}$$

which gives us

$$R_{n+1}^{-1} = R_n^{-1} - \eta_n R_n^{-1}(\Delta R_n) R_n^{-1} + O(\eta_n^2)$$

and so

$$\begin{aligned} P_{n+1} &= W_{n+1}(W_{n+1}^T W_{n+1})^{-1} W_{n+1}^T \\ &= (W_n + \eta_n \Delta W_n + O(\eta_n^2)) R_{n+1}^{-1} (W_n + \eta_n \Delta W_n + O(\eta_n^2))^T \\ &= P_n + \eta_n \Delta P_n + O(\eta_n^2) \end{aligned}$$

where

$$\begin{aligned} \Delta P_n &= (I - W_n R_n^{-1} W_n^T) \Delta W_n R_n^{-1} W_n^T + W_n R_n^{-1} \Delta W_n^T (I - W_n R_n^{-1} W_n^T) \\ &= \left((I - P_n) \Delta W_n R_n^{-1} W_n^T\right) + \left((I - P_n) \Delta W_n R_n^{-1} W_n^T\right)^T . \end{aligned}$$

Now $\Delta W_n = \mathbf{x}_n \mathbf{x}_n^T W_n - W_n \Theta_n$, so

$$\begin{aligned} \Delta P_n &= \left((I - P_n)(\mathbf{x}_n \mathbf{x}_n^T W_n - W_n \Theta_n) R_n^{-1} W_n^T\right) \left((I - P_n)(\mathbf{x}_n \mathbf{x}_n^T W_n - W_n \Theta_n) R_n^{-1} W_n^T\right)^T \\ &= \left((I - P_n) \mathbf{x}_n \mathbf{x}_n^T W_n R_n^{-1} W_n^T\right) \left((I - P_n) \mathbf{x}_n \mathbf{x}_n^T W_n R_n^{-1} W_n^T\right)^T \\ &= (I - P_n)(\Sigma_X)_n P_n + P_n (\Sigma_X)_n (I - P_n) \end{aligned}$$

where $(\Sigma_X)_n = \mathbf{x}_n \mathbf{x}_n^T$. Q.E.D.

This independence from $\Theta_n$is rather convenient, since we can now go on to show several properties of the whole class of algorithms without reference to the precise algorithm we are using. In particular, they can be viewed as stochastic algorithms which minimise least mean squared reconstruction error, and find the least upper bound on informatiominimisationn loss with spherical Gaussian input noise and no output noise.

Minimising Squared Error

Following Oja and Karhunen [1985], if certain reasonable conditions on the form of the sequences A_n and η_n are satisfied, P in (3.45) tends almost surely (a.s.) to the asymptotically stable solutions of the ordinary differential equation (o.d.e.)

$$dP/dt = P\Sigma_x(I - P) + (I - P)\Sigma_x P \tag{3.47}$$

where $\Sigma_x = E(\mathbf{x}_n \mathbf{x}_n^T) = E(A_n)$ is the covariance matrix of the input sequence $\mathbf{x}_n$. An algorithm which works in this way is termed a *stochastic approximation* algorithm: we can relate the algorithm for W to the o.d.e.

$$dW/dt = \Sigma_x W + W\Theta \tag{3.48}$$

in a similar way.

Let us now examine the behaviour of the differential equation (3.47), to give us an insight into what is going on with these algorithms. First, we note that P_n in (3.47) performs a (constrained) steepest descent search in mean squared error, even though W_n in, say (3.42) does not [Williams, 1985].

Theorem 3.3.2 *In the o.d.e. (3.47), P performs a (constrained) steepest descent search towards the minimum least mean squared reconstruction error*

$$S = E(|\mathbf{x}_n - \hat{\mathbf{x}}_n|^2) \tag{3.49}$$

where $\hat{\mathbf{x}_n} = P\mathbf{x}_n$.

Proof. Using various identities for the trace of a matrix, in particular $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ and $\mathrm{tr}(A) = \mathrm{tr}(A^T)$, we can re-write S as follows:

$$\begin{aligned} S &= E\left((\mathbf{x}_n - P\mathbf{x}_n)^T(\mathbf{x}_n - P\mathbf{x}_n)\right) \\ &= \mathrm{tr}\left((I-P)E(\mathbf{x}_n\mathbf{x}_n^T)(I-P)\right) \\ &= \mathrm{tr}((I-P)\Sigma_x) \end{aligned}$$

since $(I-P)^2 = (I-P)$. Now since $P = P^2$,

$$dP/dt = (dP/dt)P + P(dP/dt)$$

so

$$(I-P)dP/dt = (dP/dt)P$$

and

$$(I-P)dP/dt = (I-P)(dP/dt)P$$

giving

$$dP/dt = (I-P)(dP/dt)P + P(dP/dt)(I-P).$$

Now, the decrease in S,

$$\begin{aligned} dS/dt &= -\mathrm{tr}((dP/dt)\Sigma_x) \\ &= -\mathrm{tr}((I-P)(dP/dt)P\Sigma_x + P(dP/dt)(I-P)\Sigma_x) \\ &= -\mathrm{tr}\left((P\Sigma_x(I-P) + (I-P)\Sigma_x P)(dP/dt)\right) \end{aligned}$$

so the o.d.e.

$$dP/dt = P\Sigma_x(I-P) + (I-P)\Sigma_x P$$

represents a constrained steepest descent search in P-space for minimum S, since this would give us

$$dS/dt = -tr((dP/dt)^T(dP/dt)) \tag{3.50}$$
$$= -|\mathrm{vec}(dP/dt)|^2 \tag{3.51}$$
$$\leq 0 \tag{3.52}$$

with equality when $dP/dt = 0$, where $vec(B)$ for an $N \times M$ matrix B is the NM dimensional vector obtained by concatenating the columns of B. Q.E.D.

So we have a steepest descent in P-space, which is stationary when $P\Sigma_x(I-P) = P(dP/dt) = 0$, or $P\Sigma_x = P\Sigma_x P$. But since $P = P^T$ and $\Sigma_x = \Sigma_x^T$ we also have $P\Sigma_x = \Sigma_x P$, so P and Σ_x commute at the stationary points of the algorithm (3.47). This means that P must have the same eigenvectors as Σ_x [Strang, 1976, p185], so P must be a orthogonal projection onto the space spanned by some of the eigenvectors of Σ_x.

While it would require analysis of the stability of these solutions to find the convergence set that way, we shall instead look for a Lyapunov function for the algorithm which will also tell us the domain of attraction for the solution we find. One possibility for such a Lyapunov function might be the information capacity of the system (with spherical Gaussian noise on the inputs), since we know from Section 3.2 that if P is the principal subspace, this information capacity will be maximised.

Information Capacity

From Section 3.2, the information $I(Y;\Omega)$ for our system is

$$I = I(Y;\Omega) \tag{3.53}$$
$$= \frac{1}{2}\log\left(\frac{\det(\Sigma_y)}{\det(\Sigma_{\phi_y})}\right) \tag{3.54}$$
$$= \frac{1}{2}\left(\log\left(\det\left(W^T\Sigma_x W\right)\right) - \log\left(\det\left(\sigma_{\phi_x}^2 W^T W\right)\right)\right) \tag{3.55}$$

since $\Sigma_{\phi_x} = \sigma_{\phi_x}^2 I_N$. We shall show that this is a non-decreasing function of time if algorithm (3.43) is used.

Using the identity

$$\frac{d}{dt}\det(B) = \operatorname{tr}\left(\operatorname{adj}(B)\frac{dB}{dt}\right) \tag{3.56}$$

where $\operatorname{adj}(B)$ is the adjugate of B, we get

$$\begin{aligned} \frac{d}{dt}\log(\det(B)) &= \frac{1}{\det(B)}\operatorname{tr}\left(\operatorname{adj}(B)\frac{dB}{dt}\right) && (3.57)\\ &= \operatorname{tr}\left(B^{-1}\frac{dB}{dt}\right) && (3.58)\end{aligned}$$

provided B is invertible (i.e. $\det(B) \neq 0$). Thus the derivative of I in (3.55) is

$$\begin{aligned} \frac{dI}{dt} &= \operatorname{tr}\left((W^T\Sigma_x W)^{-1}W^T\Sigma_x\frac{dW}{dt} - (\sigma_{\phi_x}^2 W^T W)^{-1}\sigma_{\phi_x}^2 W^T\frac{dW}{dt}\right) && (3.59)\\ &= \operatorname{tr}\left((W^T\Sigma_x W)^{-1}W^T\Sigma_x(I - W(W^TW)^{-1}W^T)\frac{dW}{dt}\right) && (3.60)\\ &= \operatorname{tr}\left((W^T\Sigma_x W)^{-1}W^T\Sigma_x(I-P)\frac{dW}{dt}\right). && (3.61)\end{aligned}$$

Now, since $PW = W$ (or $(I-P)W = 0$) we get

$$\begin{aligned} \frac{dP}{dt}W + P\frac{dW}{dt} &= \frac{dW}{dt} && (3.62)\\ \frac{dP}{dt}W &= (I-P)\frac{dW}{dt} && (3.63)\end{aligned}$$

and this gives

$$\frac{dI}{dt} = \operatorname{tr}\left((W^T\Sigma_x W)^{-1}W^T\Sigma_x\frac{dP}{dt}W\right) \tag{3.64}$$

which again is dependent only on the form of the algorithm for P, and not for W. Substituting in $dP/dt = (I-P)\Sigma_x P + P\Sigma_x(I-P)$ we get

$$\begin{aligned} \frac{dI}{dt} &= \operatorname{tr}\left((W^T\Sigma_x W)^{-1}W^T\Sigma_x(I-P)\Sigma_x W\right) && (3.65)\\ &= \operatorname{tr}\left((I-P)\Sigma_x W(W^T\Sigma_x W)^{-1}W^T\Sigma_x(I-P)\right) && (3.66)\\ &\geq 0 && (3.67)\end{aligned}$$

with equality when $(I-P)\Sigma_x W = 0$ (since $(W^T\Sigma_x W)^{-1}$ is positive definite). Thus our class of 'stabilised Hebbian' algorithms (3.43), when considered in the continuous o.d.e. domain, causes the information capacity of the system to be a non-decreasing function of time, and is only stationary when $dP/dt = 0 = (I-P)\Sigma_x P$, so P must be a projection onto some of the eigenvectors of Σ_x as before.

Unfortunately, this still does not tell us directly which of the stationary points are stable, and the domain of attraction of these. It is possible to get a stronger version of this, however, which will give us the domain of attraction. We base this on the notion that P will never find the space spanned by the M principal eigenvectors if it does not have a component in the direction of each of them already. In other words, if $\det(W^T Q_M W) = 0$ where Q_M is the projection onto the M-dimensional principal eigenspace of Σ_x, our class of algorithms won't find the principal eigenspace. Here Q_M is acts like a sort of pseudo-covariance matrix, which gives unity weight to the M principal components of Σ_x, and zero weight to the remaining $N - M$ components.

To use Q_M, notice that in deriving (3.64) above, we did not use any particular properties of Σ_x (apart from being square and symmetrical). So, let us use a modified version of our target function from (3.55):

$$I_{Q_M} = \frac{1}{2}\log\left(\frac{\det\left(W^T Q_M W\right)}{\det\left(\Sigma_{\phi_y}\right)}\right) \tag{3.68}$$

with $W^T Q_M W$ replacing $\Sigma_y = W^T \Sigma_x W$. Now provided $\det(W^T Q_M W) \neq 0$ we get

$$\begin{aligned}
\frac{d}{dt} I_{Q_M} &= \operatorname{tr}\left((W^T Q_M W)^{-1} W^T Q_M \frac{dP}{dt} W\right) &(3.69)\\
&= \operatorname{tr}\left((W^T Q_M W)^{-1} W^T Q_M (I-P)\Sigma_x W\right) &(3.70)\\
&= \operatorname{tr}\left((W^T Q_M W)^{-1} W^T Q_M \Sigma_x W - (W^T Q_M W)^{-1} W^T Q_M P \Sigma_x W\right) &(3.71)\\
&= \operatorname{tr}\left((W^T Q_M W)^{-1} W^T Q_M \Sigma_x W\right) - \operatorname{tr}\left((W^T W)^{-1} W^T \Sigma_x W\right) &(3.72)\\
&= \operatorname{tr}\left((W^T Q_M W)^{-1} W^T Q_M \Sigma_x Q_M W\right) - \operatorname{tr}\left(P\Sigma_x\right) &(3.73)
\end{aligned}$$

remembering that $Q_M^2 = Q_M$ and Q_M has the same eigenvectors as Σ_x, so $Q_M \Sigma_x = Q_M(Q_M \Sigma_x) = Q_M \Sigma_x Q_M$. If we now take advantage of the fact that Q_M is the orthogonal projection onto the subspace spanned by the top M eigenvectors of Σ_x, we can write $Q_M = \zeta\zeta^T$ where the ζ is the orthonormal matrix whose columns are the top M eigenvectors of Σ_x.

With this expansion we can write $W^T Q_M W^T = W^T \zeta\zeta^T W$ where $W^T\zeta$ is an $M \times M$ nonsingular square matrix, since

$$\begin{aligned}
\det(W^T\zeta)^2 &= \det(W^T Q_M W^T) &(3.74)\\
&\neq 0 &(3.75)
\end{aligned}$$

so we get

$$\begin{aligned}
\frac{d}{dt} I_{Q_M} &= \operatorname{tr}\left((\zeta^T W)^{-1}(W^T\zeta)^{-1} W^T \zeta\zeta^T \Sigma_x \zeta\zeta^T W\right) - \operatorname{tr}\left(P\Sigma_x\right) &(3.76)\\
&= \operatorname{tr}(Q_M \Sigma_x) - \operatorname{tr}(P\Sigma_x). &(3.77)
\end{aligned}$$

From our earlier analysis, we know that $\operatorname{tr}(P\Sigma_x)$, which is the normalised output variance from our network, is maximised when P spans the M principal eigenvectors of Σ_x, i.e. when $P = Q_M$. Thus we have that, provided $\det(W^T Q_M W) \neq 0$,

$$\frac{d}{dt} I_{Q_M} \geq 0 \tag{3.78}$$

with equality at the solution $P = Q_M$.

Therefore we can use the quantity I_{Q_M} (or rather $\max_P(I_{Q_M}) - I_{Q_M}$) as our Lyapunov function, since I_{Q_M} is a strictly increasing function of t, except at the point $P = Q_M$, when it is stationary. So we can finally state the following theorem:

Theorem 3.3.3 *If the $M \times N$ weight matrix W has full rank for all t, the o.d.e.*

$$\frac{dW}{dt} = \Sigma_x W - W\Theta \tag{3.79}$$

is asymptotically stable on the set where the columns of W span the principal M eigenvectors of Σ_x. The domain of attraction of this set is the set of W such that for all of the principal eigenvectors ζ_i, $1 \leq i \leq M$ of Σ_x, $W^T\zeta_i \neq 0$.

To summarise, we have shown that the class of algorithms (3.43)

$$W_{n+1} = W_n + \eta_n\left(\mathbf{x}_n \mathbf{x}_n^T W_n + W_n \Theta_n\right) + O(\eta_n^2)$$

converges to a W which spans the M-dimensional principal eigenspace of Σ_x, provided that the matrix W_n remains full rank. This convergence is independent of Θ_n. This result is an M-output generalization of one of Kohonen's theorems about a generalized forgetting law in the single-unit case [Kohonen, 1984, Theorem 4.2].

The particular form of Θ_n is required to ensure W_n does not degenerate to a singular matrix as the algorithm progresses: this would be the case for the unstabilised Hebbian learning algorithm (3.32), for example.

3.3.3 Orthonormalising the Weights

So far, we have shown that our class of stabilised Hebbian algorithms leads to a weight matrix whose columns span the M-dimensional principal eigenspace of the input data. In the process, the algorithm (or at least its o.d.e. equivalent) descends monotonically in mean-squared reconstruction error, using the orthogonal projector $P = W(W^T W)^{-1} W^T$ for the reconstruction; and also ascends monotonically in information capacity $I(Y; \Omega)$ if the noise on the inputs is spherical and Gaussian. This is all valid only provided the weight matrix W remains full rank.

The unstabilised Hebbian algorithm (3.32)

$$W_{n+1} = W_n + \eta_n \mathbf{x}_n \mathbf{x}_n^T W_n$$

will not stabilise: there is nothing to prevent W_n from growing very large, and nothing to prevent all the columns of W_n tending towards the direction of the single principal eigenvector of Σ_x individually. For this algorithm W_n will not remain full rank.

The other algorithms depend on the particular form of Θ_n to keep W_n non-degenerate. In fact, the Symmetric Error Correction algorithm (3.42), for which we have $\Theta_n = \mathbf{y}\mathbf{y}^T$, and the Oja and Karhunen algorithm (3.39), for which $\Theta_n = \mathrm{diag}(\mathbf{y}\mathbf{y}^T) + 2\mathrm{UT}_+(\mathbf{y}\mathbf{y}^T)$ are similar in that in both these algorithms we have

$$\Theta_n + \Theta_n^T = 2\mathbf{y}\mathbf{y}^T \tag{3.80}$$

which we can use to our advantage in the following analysis.

We would like to show that W for these algorithms tends to an orthonormal matrix, i.e. $W^T W = I$ or $W^T W - I = 0$. So, let us consider the cost function

$$J = \|W^T W - I\|_F^2 \tag{3.81}$$

where $\|B\|_F = \mathrm{tr}(B^T B)^{1/2}$ is the Frobenius norm of B. From this we find that the derivative of our cost function, using the o.d.e. $dW/dt = \Sigma_x W - W\Theta$, is

$$\begin{aligned}
\frac{dJ}{dt} &= 2\,\mathrm{tr}\left(\left(W^T W - I\right) \left(W^T \frac{dW}{dt} + \frac{dW}{dt}^T W \right) \right) && (3.82)\\
&= 2\,\mathrm{tr}\left(\left(W^T W - I\right) \left(2W^T \Sigma_x W - W^T W \Theta - \Theta^T W^T W\right) \right) && (3.83)\\
&= 2\,\mathrm{tr}\left(\left(W^T W - I\right) \left(2W^T \Sigma_x W - W^T W(\Theta + \Theta^T)\right) \right) && (3.84)\\
&= 4\,\mathrm{tr}\left((W^T W - I)(I - W^T W)(W^T \Sigma_x W) \right) && (3.85)\\
&= -4\,\mathrm{tr}\left((W^T W - I)(W^T \Sigma_x W)(W^T W - I) \right). && (3.86)\\
& && (3.87)
\end{aligned}$$

Thus J in (3.81) is a non-decreasing function of t, provided W has full rank and Σ_x is positive definite, and is stationary only when $W^T W = I$ i.e. W has been orthonormalised.

It is also possible to show that $(W^T W)^{-1}$ converges to I using a similar function

$$J_- = \|(W^T W)^{-1} - I\|_F^2 \tag{3.88}$$

which is also a decreasing function of t except when $W^T W = I$. Thus the 2-norms of the deviation of $W^T W$ and $(W^T W)^{-1}$ are both bounded towards I by $J|_{t=0}$ and $J_-|_{t=0}$ forcing $W^T W$ to remain invertible, if it was initially. Thus if W is initially of full rank, W has full rank for all t.

In conjunction with the result from the previous section, we now have that the stabilised Hebbian algorithms with stabilisation term satisfying $\Theta_n + \Theta_n^T = 2\mathbf{y}\mathbf{y}^T$ converge to an orthonormal weight matrix W which spans the M-dimensional principal subspace of the data.

It is now a simple matter to show that W in Oja and Karhunen's Gram-Schmidt Orthogonalizing algorithm converges to the M principal eigenvalues in order, by observing that the algorithm can be written

$$\frac{d}{dt} W^{(i)} = \Sigma_x W^{(i)} - W^{(i)} \mathrm{diag}\left((W^{(i)})^T \Sigma_x W^{(i)} \right) - W^{(i)} 2\mathrm{UT}_+ \left((W^{(i)})^T \Sigma_x W^{(i)} \right) \tag{3.89}$$

where $W^{(i)}$ is the matrix composed of the first i columns of W, $0 \leq i \leq M$. Thus for all $0 \leq i \leq M$, $W^{(i)}$ converges to the orthonormal matrix which spans the i principal eigenvectors of Σ_x, so by induction the columns of W converge to the M principal eigenvectors of W.

3.3.4 Discussion

We have seen that the class of stabilised Hebbian learning algorithms considered in this chapter have the useful property of finding the principal subspace of the inputs fed to the single-layer linear network $\mathbf{y} = W^T\mathbf{x}$ (Fig. 3.1). This subspace produces the minimum mean square reconstruction error. In addition, provided the input noise is spherical and Normally distributed, this produces the maximum information capacity of the system, and minimum upper bound on the information loss.

In addition, we have seen that these algorithms find the extrema for these mean-squared error and information measures in a monotonic manner. It wasn't just 'lucky' that the stable weight matrices happen to maximise the information capacity, say—the algorithm (or at least its ordinary differential equation equivalent) causes this to monotonically increase with time.

Some of the algorithms considered also orthonormalise their weight matrix, so that the output units tend to represent different components of the input signal. This is sufficient to prevent the weight matrix from becoming degenerate, but is rather more powerful than necessary if the mean squared reconstruction error, or information loss with noise on the input, is all that we are interested in.

If this is so, is it useful for the outputs from principal components analysis or factor analysis to be uncorrelated? Wouldn't any output which covered the principal subspace suffice, even if the outputs were highly correlated? Perhaps something is missing.

In fact, what is missing is the processing that happens to the output $\mathbf{y}$ after it emerges from the network. In any processing system, there will be noise added to this output as well as the noise on the input, and we should take account of this when deciding on our output representation. Decorrelated outputs are likely to be less susceptible to noise than highly correlated ones [Barlow and Földiák, 1989]. We consider this idea in more detail in the following chapter.

Chapter 4

Filtering for Optimal Information Capacity

In the previous chapter, we considered the single-layer linear network

$$\mathbf{y} = W^T \mathbf{x}$$

where the N-dimensional input vector $\mathbf{x}$ is corrupted by additive spherical Gaussian noise (Fig. 3.3). For this system, any $N \times M$ matrix W whose columns span the same space as the M principal eigenvectors of Σ_x, the covariance matrix of $\mathbf{x}$, is optimal from both a minimum mean squared reconstruction sense, and a minimum information loss sense. The optimality of this result is unaffected by any correlations in the output $\mathbf{y}$, provided that the weight matrix W is not degenerate.

In any real system, however, the output $\mathbf{y}$ is also subject to the effects of corrupting noise. In a computer simulation, this is caused by the finite number of digits available to represent any real number: i.e. quantisation noise. In a biological system, noise may be caused perhaps by various random fluctuations of cell potentials, or by noise in the synaptic transmission process [Laughlin *et al.*, 1987]. The information we can transmit therefore *does* depend on the way it is represented at our output $\mathbf{y}$.

Attneave [1954] and Barlow [1961, 1987] have argued for many years that a perceptual system such as the human visual system should attempt to reduce the redundancy in a visual scene. Neurons have a limited maximum discharge rate that they can sustain, and the output from each neuron will be corrupted by some amount of noise. If the outputs of neurons are decorrelated from each other, more information can be represented than if the outputs are highly redundant, and *lateral inhibition* is suggested as a natural candidate to perform this decorrelation [Barlow, 1987, Barlow and Földiák, 1989].

This concept of efficient representation of information is particularly important in the retina, say, since all the information received by the retinal receptors has to pass through the optic nerve before it reaches the higher processing areas in the brain [Kuffler *et al.*, 1984]. The optic nerve is therefore an *information bottleneck*, like a limited capacity channel in communication theory [Shannon, 1949]. Srinivasan, Laughlin and Dubs [Srinivasan *et al.*, 1982, Laughlin, 1987] propose that neurons in in the early stages of a visual system should have a centre-surround antagonism to form a type of 'predictive coding'. The intensity at the centre of a receptive field is predicted from the intensities at the receptors which surround it, and the *difference* between this predicted value and the actual value is transmitted instead of the original. They show that the receptive field profiles predicted by a linear predictive coding (LPC) approach match well with those observed in the fly compound eye in both space and time and these predictions even match the way that the lateral inhibition increases with signal-to-noise ratio in the visual signal [Laughlin, 1987].

When we are dealing with low-level visual coding, it is convenient to treat the visual stimulus as an input whose characteristics are translation-invariant. In particular the covariance or auto-correlation between the signal at two receptors is a function of the distance between the receptors only. In the time domain, this means the correlation between the signal at a receptor at two instances in time should depend only on the relative time between the two instances, and not on the absolute time of either of the instances. We can then consider the problem in the frequency domain instead of the space or time domain, which make the analysis considerably simpler.

For example, in the time domain, linear predictive coding attempts to decorrelate successive values of the received signal. In the frequency domain, this is equivalent to trying to flatten or *whiten* the power spectrum of the resulting signal. In the analysis which follows, we shall also ignore the particular problems associated with the use of pulse-coded information transmission in biological systems, and approximate this to a linear system with additive white Gaussian noise.

4.1 Whitening Filters

It is a well known result from communication theory [Shannon, 1949] that if a signal is to be transmitted through a power-limited channel which has white Gaussian noise and bandwidth limit B, a *whitening filter* should be used to get maximum information capacity for the signal through the system (Fig. 4.1). However, this type of whitening filter is only optimal for a signal where the noise on the input signal is insignificant in comparison to the noise in the channel. This may not be the case for very low light levels in a visual system.

We shall see that it is possible to analyse the case for noise on the input signal as well as the channel, and that this gives rise to a filter similar to the whitening filter for low noise levels, but has an optimum cutoff frequency without the need for an arbitrary bandwidth limit. At high input noise levels, the filter changes character, and qualitatively exhibits the changes in filter observed in biological systems at low light levels, i.e. high input noise levels [Plumbley and Fallside, 1991]. First we shall review the case of a noiseless input signal.

Consider the noise-free receptor system shown in Fig. 4.2. $S_r(f)$ is the signal power spectrum at the receptor, $S_h(f) = S_r(f)$ is the signal power spectrum before the filter, $G_h(f)$ is the power spectral gain of the filter, $S_c(f)$ is the signal power in the channel, and $N_c(f)$ is the noise power in the channel. We assume a channel bandwidth B.

We treat a channel in the frequency domain as a continuum of elemental information channels, with a 'capacity density function' $C(f)$. Thus a small frequency range f to $f + \Delta f$ will have capacity approximately $(\Delta f)C(f)$. With a linear time- (or space-) invariant filter, the output signal component at each frequency is simply the product of the input signal component at the same frequency with the corresponding filter response component. We cannot move information transmitted at one frequency to another; we can only amplify or suppress the signal at each frequency independently. This simplifies the analysis considerably, since we are now dealing with a continuum of independent elemental channels identified with our frequency range, with a gain term (due to the filter) associated with each channel.

Following Shannon [Shannon, 1949], the capacity of a channel with signal spectrum $S(f)$, noise spectrum $N(f)$ and bandwidth B is given by

$$C_{\mathrm{T}} = \int_0^B C(f)\, df$$

where

$$C(f) = \log\left(\frac{S(f) + N(f)}{N(f)}\right).$$

In our case, $S(f) = S_c(f) = G_h(f)S_r(f)$ and $N(f) = N_c(f)$. To transmit this information requires total power

$$P_{\mathrm{T}} = \int_0^B S_c(f)\, df$$

thus to maximise C_{T} for a given power P_{T} we should maximise

$$J = \int_0^B C(f) - \lambda S_c(f)\, df$$

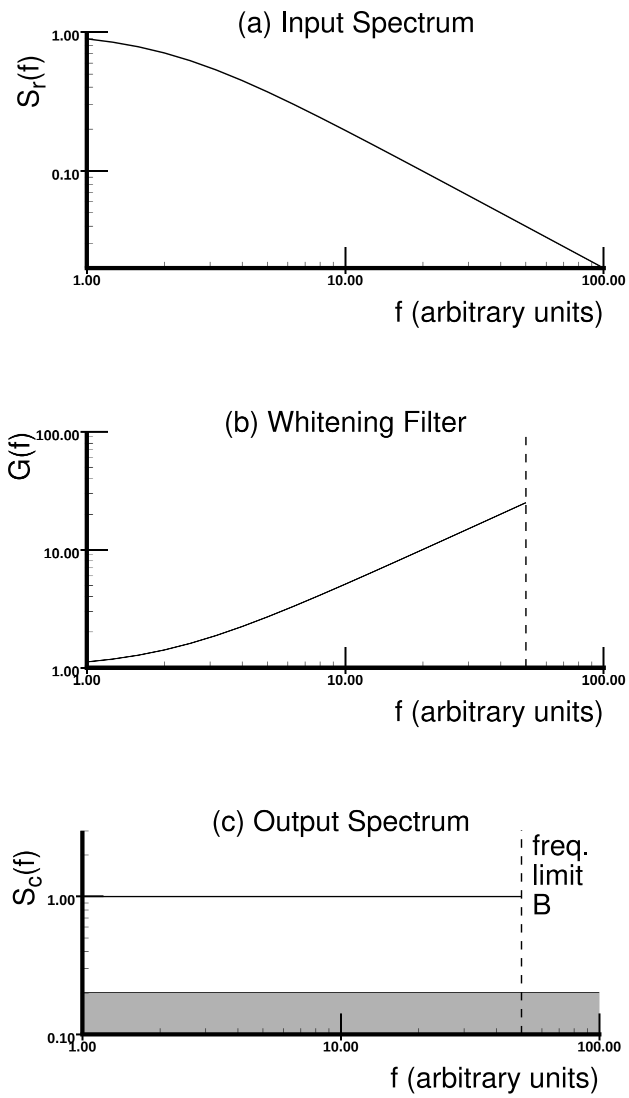


Fig. 4.1: A Whitening Filter.

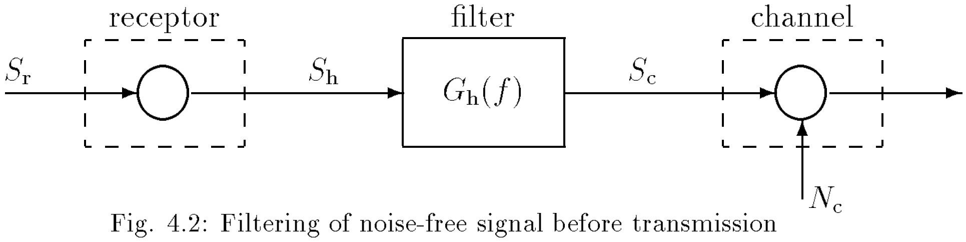


Fig. 4.2: Filtering of noise-free signal before transmission

(where $\lambda \in \mathbb{R}$ is a Lagrange multiplier) by suitable choice of the filter power gain $G_h(f) \geq 0$. This leads to the condition [Shannon, 1949]:

$$S_c(f) + N_c(f) = \gamma \tag{4.1}$$

with $\gamma = 1/\lambda$.

In the case of white noise in the channel ($N_c(f) = N_o$), $S_c = G_h(f)S_r(f)$ should be constant. With this simplification, then, the filter should be a whitening filter, so that $G_h(f) \propto S_r(f)^{-1}$ within the bandwidth limit $0 \leq f \leq B$, with $G_h(f) = 0$ for $f > B$.

There are a couple of problems with this whitening filter that we hinted at earlier. One is the 'available bandwidth', and the other is excessive boosting of noisy components in the input signal.

Let us consider the available bandwidth first. In a communication system, it is reasonable to expect that we are given a certain channel bandwidth within which to transmit our signal, and we should make use of this as best we can. However, in a biological system we might expect the problem to be posed another way round: if we need more bandwidth to transmit our signal more efficiently, it is possible that evolution would provide this additional bandwidth if it was useful. This may also be true in a communication system where the bandwidth limit is not fixed beforehand. In other words, we would also like the bandwidth limit to be fixed by the optimisation process.

Now consider the noisy components on the input signal. If the input signal to our whitening filter (Fig. 4.1(a)) had an appreciable amount of white (flat spectrum) noise on it, by the time we get to the high-frequency components of the signal which are significantly boosted by the whitening filter, we would end up spending much of our available channel power transmitting the noise on the signal instead of the signal itself. We may be able to get more efficient transmission of information by using a bit more power to transmit the low-frequency, high-quality components of the signal, and less on the high frequency, low-quality components.

In fact, by considering both the input and output noise, we shall see that we solve both of these problems. Not only are lower quality components of the input signal boosted less than higher quality components, but below a certain signal-to-noise ratio (SNR) on the input, the filter cuts off completely.

4.2 Information Transmission with Receptor Noise

Consider a simple receptor system as shown in Fig. 4.3. The receptor picks up the signal of

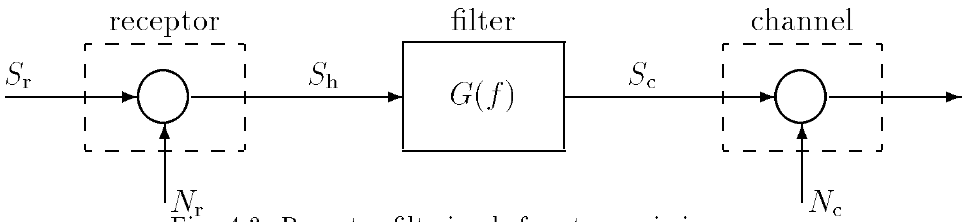


Fig. 4.3: Receptor filtering before transmission

interest, with power spectrum $S_r(f)$, together with additive Gaussian noise $N_r(f)$ (which includes both external noise and noise internal to the receptor), to give the output $S_h(f) = S_r(f) + N_r(f)$ from the receptor. This is then passed through the filter with power spectral gain $G_h(f)$ to give the input to the neuron channel with power spectrum $S_c(f) = G_h(f)S_h(f)$. Finally, Gaussian noise $N_c(f)$ is added to the signal as it is transmitted through the channel. No bandwidth limit is assumed.

Our optimisation task is still to transmit the maximum information about the signal, for a fixed total power cost. We assume that the only signal of interest arises from $S_r(f)$. Thus the 'signal' at the output of the system is

$$S(f) = G_h(f)S_r(f)$$

and the 'noise' is

$$N(f) = G_h(f)N_r(f) + N_c(f).$$

With these expressions for $S(f)$ and $N(f)$, as before, we maximise

$$J = \int_0^\infty C(f) - \lambda S_c(f)\, df \tag{4.2}$$

with

$$C(f) = \log\left(\frac{S(f) + N(f)}{N(f)}\right)$$

by suitable choice of the filter power spectral gain $G_h(f) \geq 0$. From the calculus of variations, the condition for this is

$$\frac{S_r(f)N_c(f)}{N(f)\,(S(f) + N(f))} - \lambda\,(S_r(f) + N_r(f)) = 0$$

or, dropping the '(f)'s and re-arranging,

$$(S_cN_r + N_cS_h)(S_c + N_c) - \gamma S_rN_c = 0 \tag{4.3}$$

where $S_h = S_r + N_r$, $S_c = G_hS_h$ and $\gamma = 1/\lambda$. If $N_r = 0$ (no input noise), we get

$$(S_c + N_c) = \gamma$$

as in (4.1). Otherwise, introducing signal-to-noise ratio terms $R_c = S_c/N_c$ and $R_r = S_r/N_r$, we have

$$(R_c + R_r + 1)(R_c + 1) - \frac{\gamma}{N_c}R_r = 0 \tag{4.4}$$

which leads to the solution

$$R_c = \frac{1}{2}\left(\sqrt{R_r^2 + \frac{4\gamma}{N_c}R_r} - (R_r + 2)\right) \tag{4.5}$$

from which G_h can be determined. The solution is valid (i.e. G_h non-zero) whenever

$$R_r > \frac{1}{(\gamma/N_c) - 1}. \tag{4.6}$$

Therefore, for white channel noise ($N_c(f) = N_{c_0}$), the optimal filter response is zero for all frequencies where the receptor signal-to-noise ratio R_r is less than a cut-off value $R_{r_{\min}} = 1/\left(\gamma/N_{c_0} - 1\right)$. Fig. 4.4 shows a typical solution for $G_h(f)$, for white receptor and channel noise.

4.2.1 Bounding curves

For very low input noise, we would expect the solution given above to approach Shannon's result (4.1). It is also instructive to explore the other bounding curves, to give us a qualitative idea of the the general solution (4.4).

For the bounds, we shall consider approximations to the curve at extreme values of the receptor signal-to-noise ratio R_r, and the channel signal-to-noise ratio R_c. We would expect many real sensory signals to have high receptor signal-to-noise ratio (SNR) at low temporal and spatial frequencies, so we shall start with high receptor SNR R_r.

1. $R_r \gg 1$ and $R_r \gg R_c$ (very high receptor SNR). This is the condition we might expect in a sensory system under normal operating conditions, at relatively low frequencies. In this case, the limiting noise is the channel noise N_c.

   Rearranging (4.4) we get

$$\frac{R_c^2}{R_r} + \left(1 + \frac{2}{R_r}\right) R_c + 1 + \frac{1}{R_r} = \frac{\gamma}{N_c} \tag{4.7}$$

$$S_c\left(1 + O(1/R_r) + O(R_c/R_r)\right) = \gamma - N_c(1 - O(1/R_r))$$

$$S_c = \gamma - N_c - N_c O(1/R_r) - (\gamma - N_c)(O(1/R_r) + O(R_c/R_r))$$

   giving an upper bound for R_c of

$$R_c + 1 \leq \gamma/N_c. \tag{4.8}$$

   In other words $S_c + N_c \approx \gamma$, confirming the analysis for the noise-free case.

   For white channel noise N_c, as for the noise-free case the channel signal power S_c should be constant, leading to the filter power gain

$$G_h \propto R_r^{-1} \tag{4.9}$$

   which can be seen on the first part of the curve in Fig. 4.4.

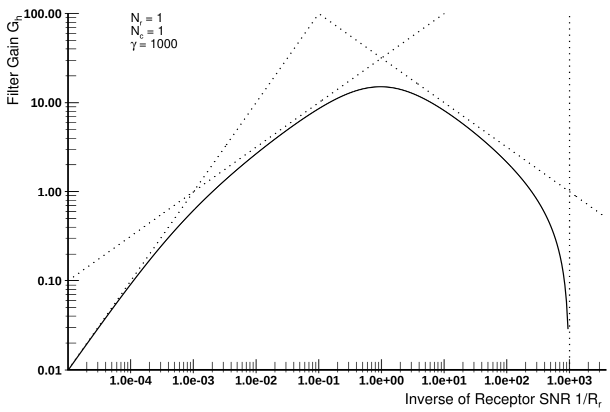


Fig. 4.4: Typical boundaries for general solution

2. $R_c \gg R_r \gg 1$ (high receptor SNR, very high channel SNR). In this case, the receptor noise N_r starts to become significant. However, the major proportion of the signal S_c in the channel is still due to the filtered receptor signal $G_h S_r$, rather than noise.

   Again rearranging (4.4) we get:

$$1 + \frac{R_r + 2}{R_c} + \frac{R_r + 1}{R_c^2} = \frac{1}{R_c^2}\left(\frac{\gamma}{N_c} R_r\right)$$

$$R_c^2\left(1 + O(R_r/R_c) + O(1/R_c)\right) = \frac{\gamma}{N_c} R_r$$

$$R_c^2 = \frac{\gamma}{N_c} R_r\left(O(R_r/R_c) + O(1/R_c)\right)$$

$$R_c^2 \leq \frac{\gamma}{N_c} R_r. \tag{4.10}$$

Substituting for R_c we get

$$(G_h N_r R_r (1 + 1/R_r))^2 \leq \gamma N_c R_r \tag{4.11}$$

$$\begin{aligned}(G_h N_r R_r)^2 &\leq \gamma N_c R_r \left(1 - O\left(1/R_r\right)\right) \\ &\leq \gamma N_c R_r\end{aligned}$$

or

$$G_h \leq N_r^{-1} \left(\gamma N_c / R_r\right)^{1/2}.$$

Thus in the case of white receptor and channel noise, we have approximately

$$G_h \propto R_r^{-1/2}. \tag{4.12}$$

This mode can be seen on the second part of Fig. 4.4. This is no longer a whitening filter: as the signal at the receptor reduces, the gain of the amplifier does increase a certain amount, but not as much as in the very low noise case.

3. $R_c \gg 1 \gg R_r$ (low receptor SNR, high channel SNR). This is similar to the previous case, but the 'signal' in the channel S_c is now mainly composed of noise at the receptor, rather than true signal S_r.

   Substituting for R_c in (4.10) we get

$$(G_h N_r (1 + O(R_r)))^2 \leq \gamma N_c R_r$$

   so

$$(G_h N_r)^2 \leq \gamma N_c R_r \tag{4.13}$$

   or

$$G_h \leq N_r^{-1} (\gamma N_c R_r)^{1/2}.$$

   Thus for white noise N_c and N_r

$$G_h \propto R_r^{1/2} \tag{4.14}$$

   The filter is no longer attempting to whiten the signal: to do so would be wasteful, since the 'signal' S_c in the channel is mainly due to the filtered receptor noise $G_h N_r$ rather than any useful signal. However, transmitting a small amount of very low quality signal such as this is still worthwhile, since the power cost is so low compared with the cost to transmit more of a higher quality signal in a frequency band which is already used significantly.

4. $R_r \ll 1$ and $R_c \ll 1$ (low receptor and channel SNR). This is really a limiting case, just before the optimal filter cuts off completely at the lower limit $R_{r_{\min}} = 1/(\gamma/N_c - 1)$.

   Again from (4.4) we get:

$$(R_c + 1)^2 \approx \frac{\gamma}{N_c} R_r$$

$$R_c \approx \frac{1}{2} \left(\frac{\gamma}{N_c} R_r - 1 \right) \tag{4.15}$$

   so, for the case of white noise,

$$G_h \propto \frac{\gamma R_r}{N_c} - 1 \tag{4.16}$$

Of course, from (4.6) we see that $R_c = 0$ for $R_r < 1/((\gamma/N_c) - 1)$ so cases 2, 3 and 4 above may not be approached unless $\gamma \gg N_c$, allowing a low threshold for R_r.

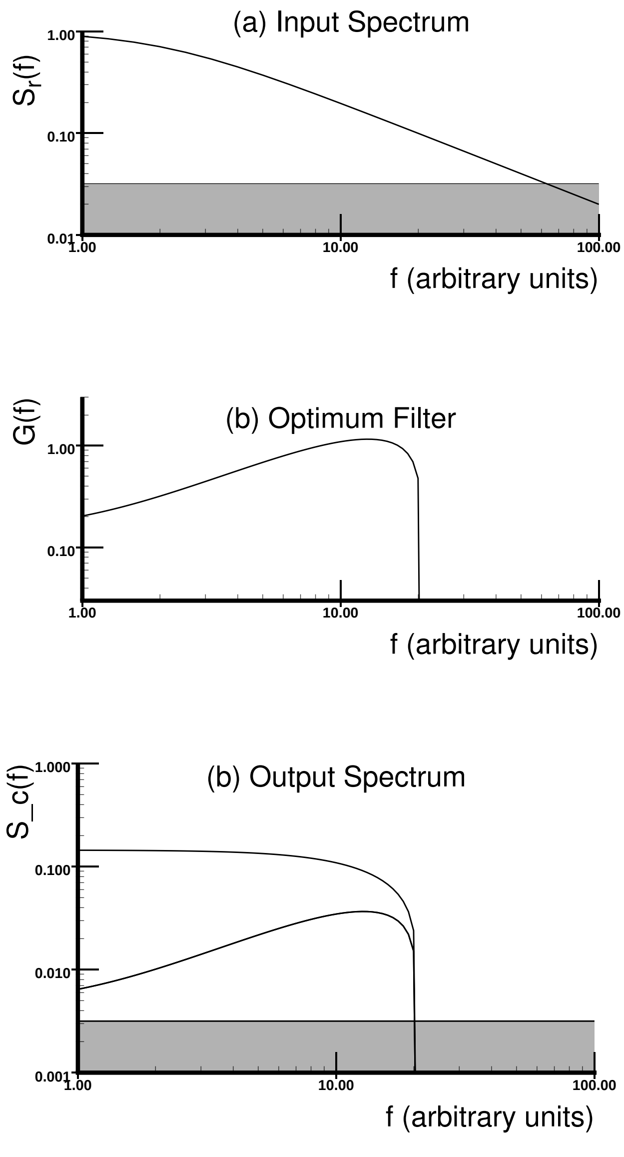


Fig. 4.5: Optimum filter with low input noise.

4.2.2 Typical Filter

Fig. 4.5 shows root power spectra of a typical optimal filter for signals with power spectral density of $S(f) = 1/(1+(f/f_c)^2)$, with $f_c = 2$. The filter response is shown in Fig. 4.5(b). In this case, the input SNR threshold is set relatively high, i.e. $R_{r_{\min}} \approx 10$ with $\gamma \approx 1.4N_c$, so there is a relatively swift transition from case 1 above to the cut-off.

In contrast, Fig. 4.6(a) compares this example with the equivalent filter for two orders of magnitude reduction in SNR, and realignment of the operating point (to $\gamma \approx 25N_c$). It can be seen that the high-frequency boost has allmost disappeared. This loss of high frequency boost at high noise levels (low signal levels) is the same qualitative effect as that observed in biological visual systems (Fig. 4.6(b)).

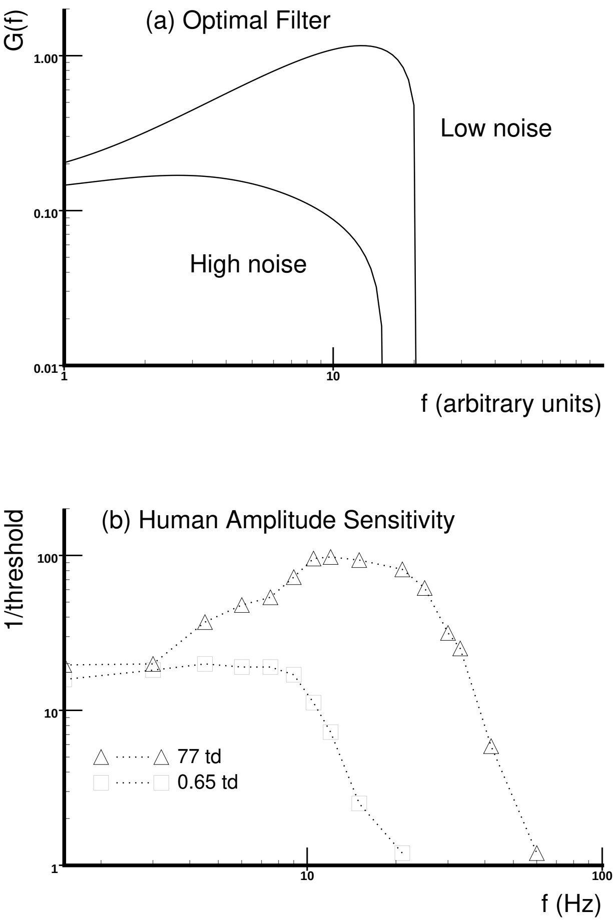


Fig. 4.6: Optimal filter change with receptor noise level. Data in (b) adapted from [Kelly, 1962]

4.3 Information Loss and Redundancy Reduction

Atick and Redlich [1989a, 1989c, 1989b] independently arrived at the same form of optimal filter, from a form of *redundancy reduction* for a given information loss. If we write the *redundancy* R of our system transforming X to Y as

$$R = 1 - \frac{I(\Omega; Y)}{C}$$

where C is the capacity of the channel for a given power cost, we minimise R for a fixed information loss about Ω

$$\begin{aligned} \Delta I_\Omega(X,Y) &= I(\Omega;X) - I(\Omega;Y) \\ &= \epsilon \end{aligned}$$

Since $I(\Omega;Y) \leq C$, the channel capacity, and $I(\Omega;Y)$ is fixed, this is equivalent to minimising C for a fixed $I(\Omega;Y)$, i.e. minimising a cost function

$$J = C - \lambda I(\Omega;Y) \tag{4.17}$$

for a lagrange multiplier λ [Atick and Redlich, 1989c].

4.3.1 Equivalence with Constrained Information Loss

Minimising J in (4.17) is equivalent to maximising the transmitted information $I(\Omega;Y)$ for a given capacity C, since this condition would require us to maximise

$$J' = I(\Omega;Y) - \frac{1}{\lambda}C \tag{4.18}$$

which has the same stationary points as (4.17), and is minimised when (4.17) is maximised.

The capacity C is determined only by the available power

$$S_{c_\mathrm{T}} = \int_0^\infty S_c(f)\,df,$$

the channel noise N_{c_T}, and the bandwidth B of the channel (which we will suppose is fixed and limited, for the moment). By the Shannon-Hartley channel capacity formula [Shannon, 1949], we have

$$C = B\log\left(1 + \frac{S_{c_\mathrm{T}}}{N_{c_\mathrm{T}}}\right) \tag{4.19}$$

so

$$\begin{aligned} \frac{\partial C}{\partial S_c(f)} &= B\frac{N_{c_\mathrm{T}}}{S_{c_\mathrm{T}} + N_{c_\mathrm{T}}}\frac{\partial}{\partial S_c(f)}\left(\frac{S_{c_\mathrm{T}}}{N_{c_\mathrm{T}}}\right) \\ &= \frac{B}{S_{c_\mathrm{T}} + N_{c_\mathrm{T}}}. \end{aligned} \tag{4.20}$$

Thus from (4.18) we get

$$\frac{\partial}{\partial S_c(f)}J' = \frac{\partial}{\partial S_c(f)}I(\Omega;Y) - \frac{1}{\lambda}\frac{B}{S_{c_\mathrm{T}} + N_{c_\mathrm{T}}} \tag{4.21}$$

$$= \frac{\partial}{\partial S_c(f)}I(\Omega;Y) - \lambda' \tag{4.22}$$

where

$$\lambda' = \frac{1}{\lambda}\frac{B}{S_{c_\mathrm{T}} + N_{c_\mathrm{T}}}$$

in other words, we are are maxmising the same quantity as in (4.2) considered before, but with a different Lagrange multiplier.

Thus these two alternative techniques, of minimisation of information loss with limited power cost on the one hand; and minimising redundancy with fixed information on the other, lead to an identical formulation for the form of the response required.

4.3.2 Results of Atick and Redlich

The results of Atick and Redlich [1989b] show a remarkable quantitative agreement with human and monkey contrast sensitivity curves. Using a small number of parameters, and an assumption that visual images are scale-invariant [Field, 1987, Field, 1989] up to a size limited by the receptive field of visual ganglion cells, they find very close fits for both temporal and spatial contrast response (see [Atick and Redlich, 1989b] for more details).

These results do strongly suggest that the mammalian visual system does indeed minimise information loss for a given power cost, or alternatively, minimises redundancy for a given information loss.

4.4 Discussion

In the preceeding analysis, we have derived a filter for maximising the information transmitted about a noisy signal through a channel which is itself noisy, with the restriction that the power available to transmit the signal is restricted.

If we compare the optimal filter in Fig. 4.5, derived for relatively low input noise, we can see that its reponse is similar to the whitening filter in Fig. 4.1. However, the frequency limit is now determined by the filter itself, rather than pre-defined by the problem. Of course, this will be determined by the values for receptor and channel noise spectra used, and the operating point γ which we choose. It would therefore be possible to optimally use a particular avalilable bandwidth by altering γ to fit, i.e. choosing γ such that $R_r((\gamma/N_c) - 1) \approx 1$ at the bandwidth limit.

The assumption that we could treat the filter response at each frequency independently helped to make the analysis for optimal information capacity possible. As always, we suffer from lack of knowledge of the precise probability distribution at each frequency component, and we must do what we can with the information which we measure. In this case, we measure only the power spectrum, i.e. the variance of each frequency component, and we must do as well as we can using that information only.

The other problem that this analysis gives us is that the filter which we derive would not be practically realisable. The cutoff for the filter is much too sharp to be constructed using any finite-length filter. In fact, we have not built any restrictions into the 'construction cost' of the filter at all. In a practical system, a filter will be constructed in the time domain using delays, or in the spacial domain using weighted sums of neigbouring receptor values. Thus there will be a cost associated with large filters in both space and time, so it would be interesting to extend this analysis to include these costs. Linsker [1989a], for example, has considered this type of cost for a system with input noise only, by multiplying the input noise variance associated with inputs by a factor proportional to the distance from the centre.

Since we would like to construct this filter using a neural network, we would also like the filter to be adaptive. Thus this analysis gives us some insight to the type of filter which we might expect, but we would like an adaptive algorithm to give it to us. In the next chapter we look at some neural network learning algorithms to try to approach this problem, primarily for the more tractible low-input-noise case.

Chapter 5

Anti-Hebbian Learning

We have seen that decorrelation of the output units of a network is important if these outputs are to be transmitted through some noisy process, and a resource limitation means that we cannot simply amplify the outputs to overcome this noise completely. When input noise becomes significant, the outputs will be less decorrelated. Intuitively, if the input noise is more significant, we cannot rely on the information between neighbouring inputs, to simply subtract one from the other, as we would have done in the case where the input noise is small. Linsker [1988a] noted a similar effect with a two-unit network.

In a neural network system, we would like to use a stochastic approximation algorithm to perform this decorrelation for us. The use of 'anti-Hebbian' or 'inhibitory learning' algorithms has already been suggested by various authors to make cells learn either to respond to different patterns [Easton and Gordon, 1984, Kohonen, 1984, Marshall, 1990a, Marshall, 1990b], or to form decorrelated outputs [Barlow, 1987, Barlow and Földiák, 1989, Földiák, 1989]. Decorrelation between units in each layer of a conventional supervised Error Back Propagation network has also been shown to give faster learning times on some problems [Orphandis, 1990].

Kühnel and Tavan [1990] showed that the anti-Hebbian learning algorithm can be derived from minimisation of mutual information between every pair of output units, which supports the intuitive notion that output units of a network should all represent different quantities, rather than all representing the same thing. The mutual information between all pairs of outputs is minimised when all the outputs are independent from each other, or in the case of zero-mean multivariate Gaussian outputs, uncorrelated from each other. However, while this does minimise the duplication of information in the outputs, some duplication may be necessary to overcome significant levels of output noise, if the variance of one of the outputs is much greater than the others, say.

In this chapter we consider forms of the anti-Hebbian learning rule, and show how they relate to minimisation of information loss, or alternatively maximization of information capacity. We first consider a linear predictive network and a network with recurrent, self-inhibiting connections.

Extending this analysis, we consider a novel skew-symmetric network which uses inhibitory interneurons to perform the required decorrelation. This network, together with a simple combination of Hebbian and anti-Hebbian learning, also attempts to minimise the information loss across the system. As well as arguably being more 'biologically plausible' than the self-inhibiting network forms, this type of network offers a potential saving in the number of connections required.

5.1 Outline

All the networks we shall be considering in this chapter have the same number N of inputs as outputs. Thus we can write

$$\mathbf{y} = W^T \mathbf{x}$$

where as before $\mathbf{x}$ is the N dimensional input vector, $\mathbf{y}$ is the output vector (also N dimensional), and the $N \times N$ matrix W^T is the effective forward transform. The precise form of W will change from one network to another. In addition, we assume additive spherical Gaussian noise ν on the output, giving a final useful output of

$$\psi = \mathbf{y} + \nu.$$

Again we wish to minimise the information loss

$$\Delta I_\Omega(X, \Psi) = I(X; \Omega) - I(\Psi; \Omega)$$

for a given power cost

$$S_W = \mathrm{tr}(\Sigma_Y)$$

(or alternatively minimise S_W for a given $\Delta I_\Omega(X, \Psi)$). As before, since $I(X; \Omega)$ is fixed for any given W, we wish to maximise the information transmitted about Ω through the system $I(\Psi; \Omega)$.

For the moment, we shall assume the effect of the input noise is negligible, so $I(\Psi; \Omega) \approx I(\Psi; X)$. Thus to maximise the information rate through the network about Ω, we must maximise the information rate $I(\Psi; X)$. If we further assume that X is zero-mean Gaussian with covariance matrix Σ_X, then we have

$$\begin{aligned} I(\Psi; X) &= H(\Psi) - H(\nu) \\ &= \log\left((\det \Sigma_\Psi)^{1/2}\right) - \log\left((\det \Sigma_\nu)^{1/2}\right) \end{aligned} \tag{5.1}$$

with the noisy output covariance matrix $\Sigma_\Psi = W^T \Sigma_X W + \Sigma_\nu$. Since the noise entropy $H(\nu)$ is independent of W, it suffices to maximise $H(\Psi)$.

5.1.1 Small Output Noise

For some networks, we may also wish to consider the case of very small output noise ν, in which case we can make the approximation

$$\begin{aligned} \Sigma_\Psi &\approx \Sigma_Y \\ &= W^T \Sigma_X W. \end{aligned}$$

so $H(\Psi) \approx H(Y)$. Note that for the spherical Gaussian noise of covariance matrix $\Sigma_X = \sigma_X^2 I_N$ to be considered to be *small* in comparison with the output $\mathbf{y}$ for this purpose, σ_ν^2 must be small in comparison to each of the eigenvalues of the output covariance matrix Σ_Y. To see this, consider $\sigma_\nu^2 I_N$ as a small perturbation to Σ_Y. Then

$$\begin{aligned} \log\det(\Sigma_Y + \sigma_\nu^2 I_N) &\approx \log\det(\Sigma_Y) + \mathrm{tr}(\Sigma_Y^{-1}\sigma_\nu^2 I_N) \\ &= \log\det(\Sigma_Y) + \sigma_\nu^2 \,\mathrm{tr}(\Sigma_Y^{-1}) \end{aligned} \tag{5.2}$$

so any small eigenvalue of Σ_Y would produce a correspondingly large eigenvalue of Σ_Y^{-1}, giving a large perturbation $\sigma_\nu^2 \,\mathrm{tr}(\Sigma_Y^{-1})$.

With this small output noise assumption, we have

$$\begin{aligned} I(\Psi; X) &= H(\Psi) - H(\nu) \\ &\approx H(Y) - H(\nu) \\ &= \log\left((\det W^T \Sigma_X W)^{1/2}\right) - \log\left((\det \Sigma_\nu)^{1/2}\right) \end{aligned} \tag{5.3}$$

but

$$\begin{aligned} \det(W^T \Sigma_X W) &= \det(W)\det(\Sigma_X)\det(W^T) \\ &= \det(W)^2 \det(\Sigma_X) \end{aligned}$$

so maximising $I(\Psi; X)$ (without constraints) reduces to maximising $\det(W)$. Alternatively, if $\det(W)$ is constant, the transmitted information is constant, and we get the most efficient transmission of information when the power cost is minimised.

5.2 Forward Linear Prediction

The first type of network we shall consider is the forward linear prediction arrangement, illustrated in Fig 5.1. Each output y_i is formed from the corresponding input x_i, less some weighted sum of

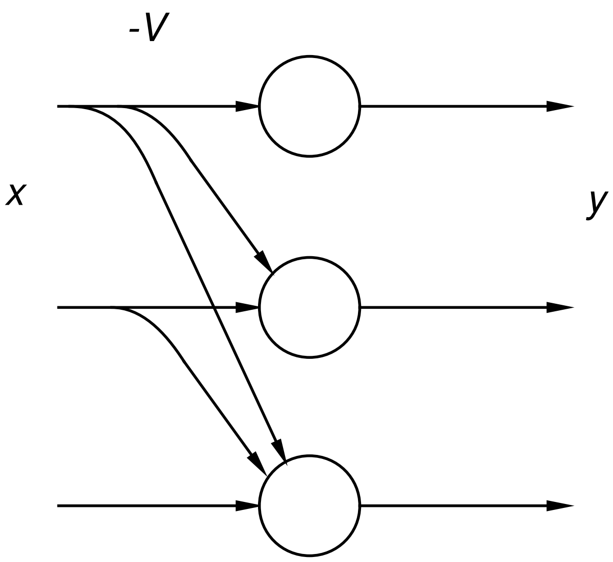


Fig. 5.1: Forward Linear Prediction Network.

lower numbered inputs, i.e.

$$y_i = x_i - \sum_{j=1}^{i-1} v_{ij} x_j$$

or, in matrix notation,

$$\begin{aligned} \mathbf{y} &= \mathbf{x} - V\mathbf{x} \\ &= (I_N - V)\mathbf{x} \end{aligned} \tag{5.4}$$

where V is a strictly lower triangular weight matrix. Thus we have our effective forward weight matrix $W^T = (I_N - V)$, where W must be upper triangular (W^T is lower triangular) with ones down the leading diagonal.

Now, since W is triangular, its eigenvalues are its unity diagonal entries, so $\det(W)$ is constant for all strictly lower triangular matrices V. If we assume that the output noise is small, as outlined above, the transmitted information $I(\Psi; X)$ is directly proportional to $\det(W)$, so is constant for all strictly lower triangular V.

So to get most efficient information transmission for small output noise, it suffices to minimise the power cost

$$S_W = \operatorname{tr}(W^T \Sigma_X W)$$

where $\operatorname{tr}(\cdot)$ is the trace of the matix. In normal linear prediction terminology, we wish to minimise the mean squared error at the output $\mathbf{y}$, so each y_i should be the residual error after the best prediction of x_i from $x_1, \ldots, x_{i-1}$ has been subtracted. This error y_i will therefore be orthogonal to each of the lower numbered inputs $x_1, \ldots, x_{i-1}$.

A simple algorithm to achieve this is the anti-Hebbian algorithm

$$(v_{ij})_{n+1} = (v_{ij})_n + \eta_n (y_i)_n (x_j)_n \tag{5.5}$$

for $j < i$, or in matrix form

$$V_{n+1} = V_n + \eta_n \mathrm{LT}_+(\mathbf{y}_n \mathbf{x}_n^T) \tag{5.6}$$

where $\mathrm{LT}_+(B)$ is the matrix B with all entries on or above the diagonal set to zero (i.e. forces it to be strictly lower triangular). This gives us an equivalent stochastic approximation o.d.e.

$$\begin{aligned} \frac{dV}{dt} &= \mathrm{LT}_+\left(E(\mathbf{y}\mathbf{x}^T)\right) \\ &= \mathrm{LT}_+\left(W^T \Sigma_X\right) \end{aligned} \tag{5.7}$$

for which

$$\frac{dW}{dt} = -\mathrm{UT}_+(\Sigma_X W). \tag{5.8}$$

The derivative of our power cost S_W in (5.2) is

$$\begin{aligned} \frac{d}{dt} S_W &= \mathrm{tr}(W^T \Sigma_X \frac{dW}{dt} + \frac{dW^T}{dt} \Sigma_X W) \\ &= 2\,\mathrm{tr}(W^T \Sigma_X \frac{dW}{dt}). \end{aligned} \tag{5.9}$$

Substituting (5.8) in this we get

$$\begin{aligned} \frac{d}{dt} S_W &= -2\,\mathrm{tr}(W^T \Sigma_X \mathrm{UT}_+(\Sigma_X W)) \\ &= -2\,\mathrm{tr}(\mathrm{LT}_+(W^T \Sigma_X) \mathrm{UT}_+(\Sigma_X W)) \\ &= -2\|dW/dt\|_F^2 \\ &\leq 0 \end{aligned}$$

with equality when $dW/dt = 0$. At the solution, we have $\mathrm{LT}_+(W^T \Sigma_X) = 0$, i.e. each output y_i is decorrelated from all the preceding *inputs* x_i.

Also, since $W^T \Sigma_X$ is upper triangular, and W is upper triangular, $\Sigma_Y = W^T \Sigma_X W$ is also upper triangular. But Σ_Y is symmetric, so it must be diagonal. So, at the solution, the outputs are decorrelated.

This algorithm, or rather its equivalent o.d.e., performs a constrained minimisation of S_W, with the constraint that V must remain strictly lower triangular. Thus the anti-Hebbian learning algorithm (5.5) will search for the least power cost solution to the problem of transmitting the information through the network, given this constraint on V.

5.3 Recurrent Linear Prediction

The next type of network we shall consider is the recurrent linear predictive network shown in Fig 5.2. This network also decorrelates the outputs if an anti-Hebbian algorithm is used. In contrast to the previous network, each output y_i is predicted from all lower-numbered *outputs* y_j for $1 \leq j < i$.

For this network we can write

$$\mathbf{y} = \mathbf{x} - V\mathbf{y}$$

or

$$\mathbf{y} = (I_N + V)^{-1}\mathbf{x}$$

where V is again strictly lower triangular. This gives us an effective forward transform matrix

$$W^T = (I_N + V)^{-1}$$

which again has unit determinant since $(I_N + V)^{-1}$ has unit determinant for all strictly lower triangular V. So as for the forward linear prediction network, if we assume small output noise, we

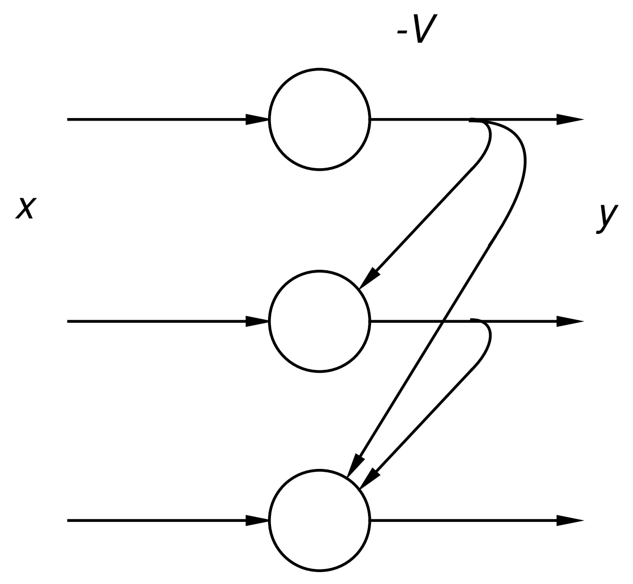


Fig. 5.2: Recurrent Linear Prediction Network

should minimise the power cost S_W, since the transmitted information $I(\Psi; X)$ is independent of V.

If we use the obvious anti-Hebbian rule for V as before we get

$$(v_{ij})_{n+1} = (v_{ij})_n + (y_i)_n (y_j)_n \tag{5.10}$$

for $j < i$. This gives our equivalent o.d.e. to be

$$\frac{dV}{dt} = \mathrm{LT}_+(\Sigma_y) \tag{5.11}$$

which gives the o.d.e. for the effective forward matrix as

$$\begin{aligned}\frac{dW^T}{dt} &= -(I_N + V)^{-1} \frac{dV}{dt} (I_N + V)^{-1} \\ &= -W^T \left(\mathrm{LT}_+(W^T \Sigma_x W) \right) W^T.\end{aligned}$$

Substituting this into (5.9) as before, we get

$$\frac{d}{dt} S_W = -2 \operatorname{tr} \left(W^T \Sigma_X W \left(\mathrm{UT}_+(W^T \Sigma_X W) \right) W \right).$$

To take this further, we first notice that W is upper triangular, as the inverse of another upper triangular matrix $(I_N + V)$. Thus the product $\mathrm{UT}_+(W^T \Sigma_X W) W$ is strictly upper triangular, and since we can easily verify that

$$\operatorname{tr}(A(\mathrm{UT}_+(B))) = \operatorname{tr}((\mathrm{LT}_+(A))B) = \operatorname{tr}((\mathrm{LT}_+(A))(\mathrm{UT}_+(B)))$$

we get

$$\begin{aligned}\frac{d}{dt} S_W &= -2 \operatorname{tr}(\mathrm{LT}_+(W^T \Sigma_X W) \mathrm{UT}_+(W^T \Sigma_X W) W) \\ &= -2 \operatorname{tr} \left(\mathrm{UT}_+(W^T \Sigma_X W) W \, \mathrm{LT}_+(W^T \Sigma_X W) \right)\end{aligned}$$

Since W has all unit eigenvalues, and is therefore positive definite, the right hand side above is strictly negative unless $\mathrm{UT}_+(W^T\Sigma_X W) = 0$, i.e. Σ_Y is diagonal.

So, for this recurrent linear predictor network, the anti-Hebbian learning algorithm (5.8) causes S_W to monotonically decrease with t until the outputs are decorrelated from each other. Since the transmitted information $I(\Psi; X)$ is independent of V for small output noise, this minimum S_W gives us the most efficient information transmission through this network.

We have seen that both of these asymmetrical forward and recurrent linear prediction networks will learn to decorrelate their output units, and thus minimise the output power cost. Provided the output noise is 'small', this will be the most efficient regime for information transmission, since the transmitted information is independent of the connection values.

However, Linsker [1988a] pointed out that simple decorrelation of the outputs is not optimal if the output noise is significant. What we would really like is the optimal trade-off between information transmission and power cost, rather than simply minimising the power cost alone.

5.4 Symmetrical Recurrent Anti-Hebb Learning

The asymmetric networks in the previous sections were designed to decorrelate their outputs, by linear prediction of each successive output from the previous outputs, while keeping the information rate constant. A more usual form of network is the symmetrical recurrent network shown in Fig. 5.3. This type of network is sometimes known as the 'Backward Inhibition' model in vision

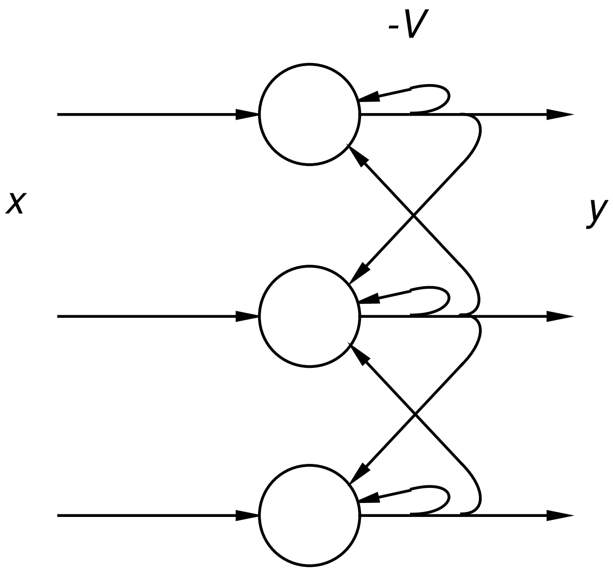


Fig. 5.3: A Network with recurrent lateral connections.

literature [Cornsweet, 1970, Hall, 1979], where the self-inhibition connections from each node back to itself are sometimes omitted. It also appears as the 'Novelty Filter' described by Kohonen [1984], and without the self-inhibitory connections as the decorrelating network described by Barlow and Földiák [1989]. Földiák [1989] also uses this decorrelating network between Oja [1982] principal component neurons.

In this network we have

$$\mathbf{y} = \mathbf{x} - V\mathbf{y} \tag{5.12}$$

or

$$\mathbf{y} = (1 + V)^{-1}\mathbf{x} \tag{5.13}$$

provided $(1+V)$ is invertible. Thus our effective forward matrix $W^T = (1+V)^{-1}$.

As ever, the output $\mathbf{y}$ is going to be sent through a limited-power noisy channel of available power S_W with spherical additive Gaussian noise ν to give a final signal $\psi = \mathbf{y} + \nu$. We shall assume that both ν and X (and therefore Y and Ψ) are zero-mean normally distributed, and that the noise ν is spherical so that $\Sigma_\nu = \sigma_\nu^2 I$. To simplify the analysis, we shall tentatively assume the initial lateral inhibition weight matrix V is symmetrical (and thus W is also symmetrical).

The difference now is that the network is fully recurrent, so the weight matrix no longer has constant determinant, as it did for the previous two cases. This time we will actually be optimising the trade-off between information capacity and power cost, rather than simply minimising the power cost for the same amount of information capacity.

For a given V (and hence W), the information rate through the network is

$$\begin{aligned} I_W &= I(\Psi; X) && (5.14)\\ &= \log\left((\det\Sigma_\Psi)^{1/2}\right) - \log\left((\det\Sigma_\nu)^{1/2}\right) && (5.15)\end{aligned}$$

and the channel transmission power

$$S_W = \operatorname{tr}(\Sigma_Y). \tag{5.16}$$

To maximise the transmitted information I_W for a given $S_W = S$, we maximise a modified function

$$J_W = I_W - \frac{1}{2}\lambda S_W$$

where λ is a Lagrange multiplier, chosen such that $S_W = S$ at the maximum of J_W.

Rather than finding the maximum of J_W directly, we wish to construct an algorithm to do it for us by modifying V. Taking a Lyapunov function approach, we wish to find an expression for dV/dt, where t is a scalar parameter ('time'), for which $dJ_W/dt \geq 0$ for V within a suitable domain of attraction, with equality when $J_W = J^\star$, the maximum value of J_W. We will then be able to deduce that V converges to the optimal solution $W^\star$ as $t \to \infty$. Now since the entropy of the output noise $H(\nu)$ is independent of V, we have

$$\begin{aligned} \frac{d}{dt} I_W &= \frac{d}{dt}\left(\log\left((\det\Sigma_\Psi)^{1/2}\right) - \log\left((\det\Sigma_\nu)^{1/2}\right)\right) \\ &= \frac{1}{2}\operatorname{tr}\left(\Sigma_\Psi^{-1}\frac{d}{dt}\Sigma_\Psi\right) \\ &= \frac{1}{2}\operatorname{tr}\left(\Sigma_\Psi^{-1}\frac{d}{dt}\Sigma_Y\right) && (5.17)\end{aligned}$$

and

$$\frac{dS_W}{dt} = \operatorname{tr}\left(\frac{d}{dt}\Sigma_Y\right) \tag{5.18}$$

so

$$\begin{aligned} \frac{d}{dt} J_W &= \frac{d}{dt}(I_W - \lambda S_W) && (5.19)\\ &= \frac{1}{2}\operatorname{tr}\left((\Sigma_\Psi^{-1} - \lambda I_N)\frac{d}{dt}\Sigma_Y\right). && (5.20)\end{aligned}$$

Now

$$\begin{aligned} \frac{d}{dt}\Sigma_Y &= \frac{d}{dt}(W^T \Sigma_X W) && (5.21)\\ &= -\left(\left(\Sigma_Y \frac{dV}{dt} W\right) + \left(W^T \frac{dV}{dt}\Sigma_Y\right)\right) && (5.22)\end{aligned}$$

since $dW/dt = -W(dV/dt)W$, so

$$\begin{aligned} \frac{d}{dt} J_W &= -\operatorname{tr}\left((\Sigma_\Psi^{-1} - \lambda I_N)\Sigma_Y \frac{dV}{dt} W\right) & (5.23) \\ &= +\operatorname{tr}\left(\Sigma_\Psi^{-1}(\lambda\Sigma_\Psi - I_N)\Sigma_Y \frac{dV}{dt} W\right). & (5.24) \end{aligned}$$

We wish to find an algorithm dV/dt for which dJ_W/dt in (5.24) is a Lyapunov function. For example, the anti-Hebbian rule of Barlow and Földiák [1989] would give us an algorithm of the form

$$(v_{ij})_{n+1} = (v_{ij})_n + \eta_k (y_i)_n (y_j)_n \qquad \text{for } i \neq j$$

where $(v_{ij})_n$ is the inhibitory weight between output units i and j at time step k. Under suitable stochastic approximation assumptions, this would give the same solution as the o.d.e.

$$\frac{dV}{dt} = \operatorname{offdiag}(\Sigma_Y)$$

where offdiag$(\cdot)$ returns only the off-diagonal elements of the matrix, with the diagonal elements set to zero. The diagonal entries of V (the connections from units back to themselves) are zero for all t, i.e. $v_{ii} = 0$ for all i. This is obviously stationary when offdiag$(\Sigma_Y) = 0$, i.e the outputs are decorrelated as before. Unfortunately, this algorithm does not guarantee to increase J_W, since it is possible to make dJ_W/dt negative (for $N = 2$ try $\Sigma_\nu \approx 0$, $\lambda = 1/1000$, with

$$V = \begin{bmatrix} 0 & -1/2 \\ -1/2 & 0 \end{bmatrix} \qquad \text{and} \qquad \Sigma_Y = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

as an example). Instead, we shall try to choose the learning algorithm to force J_W to decrease.

Let us first consider the low-noise case $\Sigma_\nu \ll \Sigma_Y$, so $\Sigma_\Psi \approx \Sigma_Y$. In this case

$$\begin{aligned} \frac{d}{dt} J_W &\approx \operatorname{tr}\left(\Sigma_Y^{-1}(\lambda\Sigma_Y - I_N)\Sigma_Y \frac{dV}{dt} W\right) \\ &= \operatorname{tr}\left((\lambda\Sigma_Y - I)\frac{dV}{dt} W\right) \\ &= \operatorname{tr}\left(\frac{dV}{dt} W(\lambda\Sigma_Y - I_N)\right) \end{aligned}$$

from which we can immediately see that choosing our algorithm to be

$$\frac{dV}{dt} = \lambda\Sigma_Y - I_N \tag{5.25}$$

gives

$$\begin{aligned} \frac{d}{dt} J_W &= \operatorname{tr}\left((\lambda\Sigma_Y - I_N) W (\lambda\Sigma_Y - I_N)\right) \\ &\geq 0 \end{aligned}$$

with equality when $\Sigma_Y = \lambda^{-1} I_N$, provided our forward transfer matrix $W = (I_N + V)^{-1}$ remains positive definite. This is certainly the case if V is small, and also possible at the solution, when $W^{-2} = (1/\lambda)\Sigma_X$, provided Σ_X is positive definite.

In fact, W does remain positive definite, as the following argument due to Dayan [1991] demonstrates. Since W is real and symmetric it has a complete set of orthonormal eigenvectors e_i, $1 \leq i \leq N$, with associated eigenvalues l_i. The matrix W is positive definite if and only if eigenvalues l_i are strictly positive for all i. The input covariance matrix Σ_X has all its eigenvalues less than some maximum value $\sigma^2_{X_{\max}}$. Then we have

$$\begin{aligned} \frac{dl_i}{dt} &= e_i^T \frac{dW}{dt} e_i \\ &= e_i^T \left(W^2 - \lambda W \Sigma_Y W\right) e_i \\ &= e_i^T W^2 e_i - \lambda e_i^T W^2 \Sigma_X W^2 e_i \\ &\geq -l_i^4 \lambda \sigma^2_{X_{\max}} \end{aligned}$$

since

$$\begin{aligned} e_i^T W^2 e_i &= l_i \\ &> 0 \end{aligned}$$

and

$$\begin{aligned} e_i^T W^2 \Sigma_X W^2 e_i &= l_i^4 e_i^T \Sigma_X e_i \\ &\leq l_i^4 \sigma_{X_{\max}}^2 \end{aligned}$$

and λ is positive. Therefore each eigenvalue l_i of W decreases to zero at a rate no faster than $t^{-1/3}$, and W remains positive definite.

Our algorithm is then

$$\frac{d}{dt} v_{ij} = \begin{cases} \lambda E(y_i y_j) - 1 & \text{if } i = j \\ \lambda E(y_i y_j) & \text{otherwise} \end{cases} \tag{5.26}$$

which leads to the following stochastic approximation algorithm:

$$v_{ij}(k+1) = v_{ij}(k) + \begin{cases} \eta_k (y_i(k)^2 - \frac{1}{\lambda}) & \text{if } i = j \\ \eta_k y_i(k) y_j(k) & \text{otherwise} \end{cases} \tag{5.27}$$

This is the Barlow and Földiák [1989] anti-Hebbian algorithm outlined above, with an additional 'self-inhibition' term which forces the variance of all the output units to be $1/\lambda$, as well as decorrelating them. In fact, we can also verify that the algorithm

$$\frac{dV}{dt} = \lambda \Sigma_\Psi - I \tag{5.28}$$

$$= \lambda \Sigma_Y - (1 - \lambda \sigma_\nu^2) I \tag{5.29}$$

(replacing Σ_Y in (5.25) with Σ_Ψ) is valid for the spherical noise case $\Sigma_\nu = \sigma_\nu^2 I$, as

$$\frac{d}{dt} J_W = \operatorname{tr}\left(\Sigma_\Psi^{-1} (\lambda \Sigma_\Psi - I) \Sigma_Y \frac{dV}{dt} W \right) \tag{5.30}$$

$$= \operatorname{tr}\left(\Sigma_\Psi^{-1} (\lambda \Sigma_\Psi - I) \Sigma_Y (\lambda \Sigma_\Psi - I) W \right) \tag{5.31}$$

$$= \operatorname{tr}\left((\lambda \Sigma_\Psi - I) \Sigma_\Psi^{-1/2} \Sigma_Y^{1/2} W \Sigma_Y^{1/2} \Sigma_\Psi^{-1/2} (\lambda \Sigma_\Psi - I) \right) \tag{5.32}$$

$$\geq 0 \tag{5.33}$$

if W is positive definite, with equality when

$$\Sigma_Y = (\lambda')^{-1} I \tag{5.34}$$

where $\lambda' = \left(\lambda^{-1} - \sigma_\nu^2\right)^{-1}$. This is the same as the low-noise version, but with a modified λ.

5.4.1 A Two Dimensional Example

To test this fully recurrent Hebbian algorithm, we took a random sample of 100 zero-mean normally distributed points in two dimensions. The major axis of the covariance matrix of Σ_X is at an angle of $\pi/6$ from the horizontal, and the covariance matrix has standard deviations of 10 units along the major axis, and 1 along the minor axis. The weight matrix V was initialized to zero, and updated after each pass through the data set according to the low-noise algorithm

$$\frac{dV}{dt} = \eta \left(\hat{\Sigma}_Y - \lambda^{-1} I \right)$$

where $\hat{\Sigma}_Y = \overline{yy^T}$, $\eta = 0.03$, and $\lambda = 1/2$.

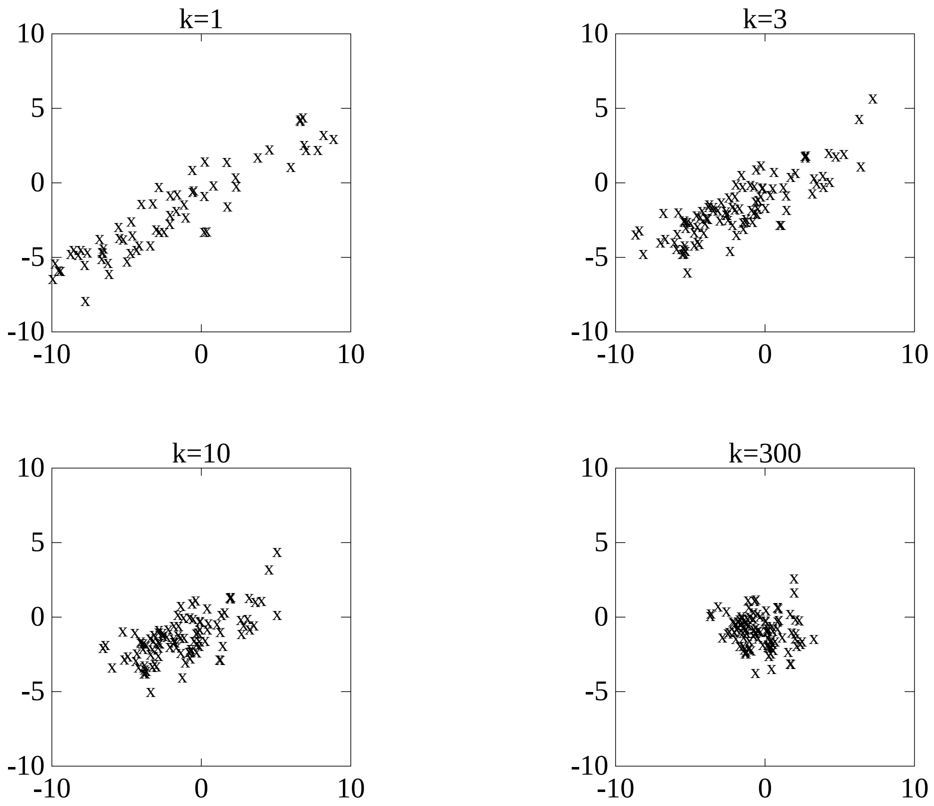


Fig. 5.4: Output distributions from the recurrent Hebbian learning network for time $k = 0, 3, 10$, and 300.

k	W	Σ_Y
1	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 88.8349 & 53.1395 \\ 53.1395 & 33.6472 \end{bmatrix}$
3	$\begin{bmatrix} 1.0363 & 0.6372 \\ 0.6372 & 0.3746 \end{bmatrix}$	$\begin{bmatrix} 15.4404 & 8.5302 \\ 8.5302 & 6.5814 \end{bmatrix}$
10	$\begin{bmatrix} 1.6947 & 1.0606 \\ 1.0606 & 0.5932 \end{bmatrix}$	$\begin{bmatrix} 8.3436 & 4.1485 \\ 4.1485 & 4.0353 \end{bmatrix}$
300	$\begin{bmatrix} 4.5515 & 2.8709 \\ 2.8709 & 1.5700 \end{bmatrix}$	$\begin{bmatrix} 2.2013 & 0.1223 \\ 0.1223 & 2.0743 \end{bmatrix}$

Table 5.1: Sequence of W and Σ_Y

Fig. 5.4 plots the distribution of output points for $k = 1, 3, 10$ and 300, where k is the number of passes through the data set (epochs). A deliberately small value of η was to prevent convergence within the first few epochs. Since the weights V were initialised to zero, this plot for "k=1" shows the distribution of input points. It can be seen from these plots that the distribution of output points is squashed (but not rotated) until equally distributed, and with a standard deviation of about 2 or 3 in all directions. In fact, the sequence of Σ_Y is shown in Table 5.1, and is converging towards

$$\lambda^{-1} I = 2I = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

as predicted. Fig. 5.5 shows the 'learning curve' for the target function $J_W = I_W - \lambda S_W$ during this test.

5.5 Skew-Symmetric Interneurons

Although the network described in the previous section does search for a maximum information capacity with a limited power constraint, there are some objections that can be levelled against it. In biological systems, it is unusual (if not impossible) to find neurons which can inhibit other neurons of the same class [Kuffler *et al.*, 1984]. In addition, the self-inhibition connections are rather cumbersome in that they must adapt differently from the inhibitory connections to other units. The number of connections is also large, being N^2 for N output units.

It may be the case that interneurons could help here. In the retina, for example, the horizontal cells seem to act as inhibitory interneurons to help produce the centre-surround antagonistic response of the later ganglion cells [Dowling, 1987]. If interneurons such as these are trying to optimise the information transmission with limited 'power', the best solution will be found when all the outputs are decorrelated, and with the same variance (if we allow our units to have both positive and negative responses). However, we may be able to get a solution with a small number $M < N$ of interneurons, if the correlation between units is caused by a relatively small number of components.

The general idea works as follows. Assume that the responses of two units are positively correlated together, and we wish to decorrelate them. If we use an Oja-like neuron [Oja, 1982] to find the principal component of their responses, its weight vector would 'line up' with the direction of maximum response of the two units. If we then subtract an appropriate amount of the response of this Oja interneuron from the responses of the other two, we may be able to decorrelate their outputs, with the use of only a *single* interneuron.

Fig. 5.6 shows the type of network we are suggesting. In vector notation, the response of the interneurons is given by

$$\mathbf{z} = V^T \mathbf{y}$$

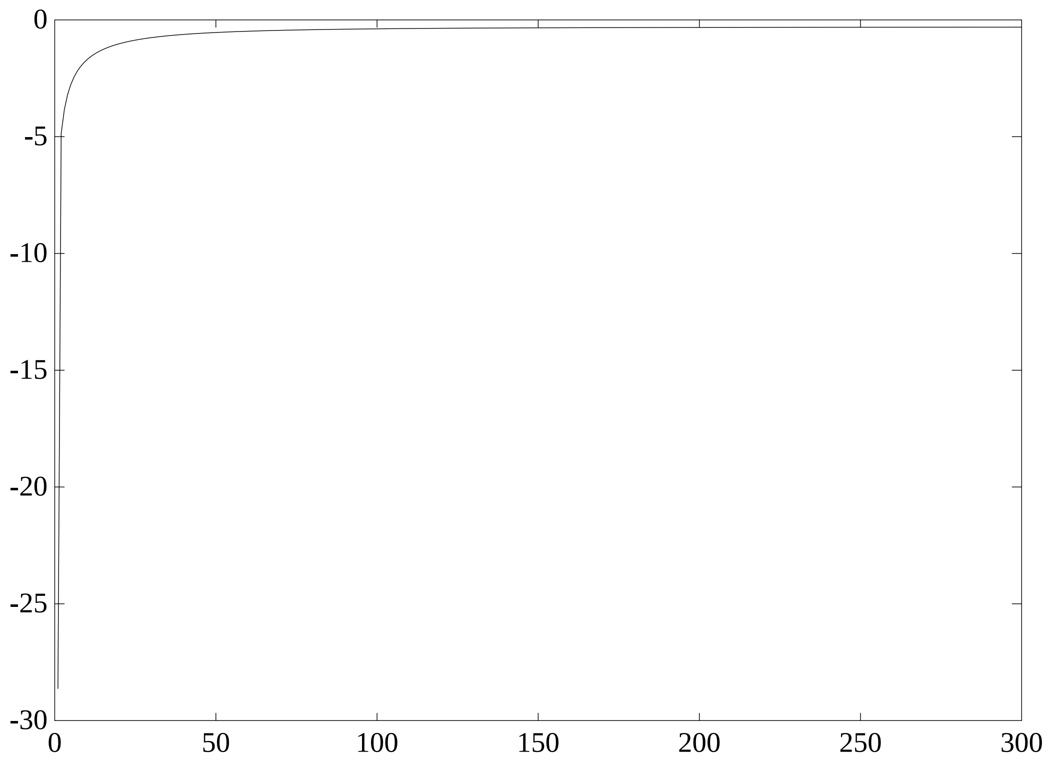


Fig. 5.5: Learning curve for the 2-D trial

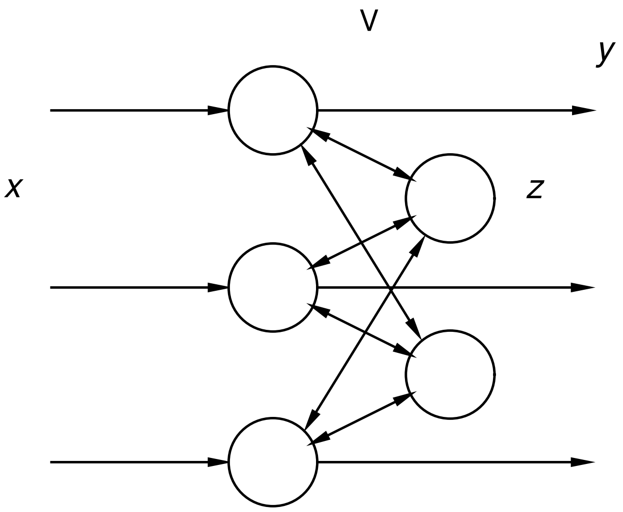


Fig. 5.6: Network with inhibitory interneurons

and the output response

$$\mathbf{y} = \mathbf{x} - V\mathbf{z}$$

thus if V is positive, the interneurons are positively excited by the $\mathbf{y}$ units, but inhibit them in return. Thus we get

$$\mathbf{y} = (I_N + VV^T)^{-1}\mathbf{x}$$

so our effective forward matrix W is given by $W = (I_N + VV^T)^{-1}$, provided the inverse exists. We now have VV^T instead of the V in the network described in the previous section, forcing W to be symmetrical and positive definite. This gives the derivative of W as

$$\frac{dW}{dt} = -W\left(\frac{dV}{dt}V^T + V\frac{dV^T}{dt}\right)W$$

which, for the low output noise case, gives the derivative of our constrained information function J_W in (5.20) to be

$$\frac{d}{dt}J_W = -\operatorname{tr}\left((I_N - \lambda\Sigma_Y)\left(\frac{dV}{dt}V^T + V\frac{dV^T}{dt}\right)W\right). \tag{5.35}$$

In order for us to have any hope of finding an algorithm for V which forces J to increase, we must introduce a term in $(\lambda\Sigma_Y - I_N)$. In the previous section, both ends of each of the lateral inhibitory connections were directly connected to $\mathbf{y}$ units, so a local algorithm of this form was possible. For the network in this section, we are now connected between $\mathbf{y}$ and $\mathbf{z}$ units, so a local algorithm could be of the form

$$\begin{aligned}\left(\frac{dV}{dt}\right)_1 &= E(\mathbf{y}\mathbf{z}^T) \\ &= \Sigma_Y V\end{aligned}$$

which would be a simple (anti-) Hebbian update rule. In fact, we can see that the addition of a weight decay term sometimes used to stabilise similar algorithms [Easton and Gordon, 1984] would enable us to produce

$$\begin{aligned}\frac{dV}{dt} &= \lambda\Sigma_Y V - V \\ &= (\lambda\Sigma_Y - I_N)V\end{aligned} \tag{5.36}$$

which is of the form we are looking for. Substituting this into (5.35) we get

$$\begin{aligned}\frac{d}{dt}J_W &= \operatorname{tr}\left((I_N - \lambda\Sigma_Y)(VV^TW)(I_N - \lambda\Sigma_Y)\right) \\ &\quad + \operatorname{tr}\left(V^T(I_N - \lambda\Sigma_Y)(W)(I_N - \lambda\Sigma_Y)V\right) \\ &\geq 0\end{aligned} \tag{5.37}$$

with equality when $(\lambda\Sigma_Y - I_N)V = 0$ i.e. $dV/dt = 0$. (Note that both W and

$$VV^TW = VV^T\left(VV^T + (VV^T)^2\right)^{-1}VV^T$$

are positive definite). So with the algorithm (5.36), J_W is a strictly increasing function of t, except when V has converged.

This algorithm leads to a particularly simple implementation, vis.

$$(v_{ij})_{n+1} = (v_{ij})_n + \eta_n((y_i)_n(z_j)_n - \lambda^{-1}(v_{ij})_n) \tag{5.38}$$

which is of Hebbian form, but with a weight decay term derived explicitly to optimise the information transmission, rather than an alternative biological justification in terms of competition for resources [Hirai, 1980, Grossberg, 1986].

5.5.1 Properties at Convergence

At convergence, we have

$$\lambda \Sigma_Y V = V$$

or

$$\begin{aligned} \Sigma_X (V V^T) &= \lambda^{-1} (I_N + V V^T) V V^T (I_N + V V^T) \quad (5.39) \\ &= (V V^T) \Sigma_X \quad (5.40) \end{aligned}$$

i.e. Σ_X and (VV^T) commute, so they share the same eigenvectors [Strang, 1976]. This doesn't necessarily mean that the columns of V are M of the eigenvectors of Σ_X, but it does mean that they span the same subspace as some $L \leq M$ eigenvectors of the input covariance matrix Σ_X. The subspace that is spanned by those eigenvectors is decorrelated to have variance λ^{-1}. The maximum of J_W is found when this subspace is the span of the top L eigenvectors of Σ_X. So, not only do we decorrelate the inputs a certain extent, but we also find the space spanned by the principal eigenvectors as a by-product of this process.

In contrast with the previous symmetric algorithm, it is not possible to *increase* the variance of any component of the input, since $\mathbf{y} = (I_N + VV^T)^{-1}\mathbf{x}$, so each component of Σ_Y must be less than or equal to the corresponding component of Σ_X. The algorithm attempts to force all the components to have variance λ^{-1}, which is not possible if that component of the input $\mathbf{x}$ originally had variance less than this value. In this case that component will be simply left alone. This could prevent small-variance input components, possibly swamped by input noise, from being amplified and wasting information capacity. This would the case even if there are 'spare' interneurons which could potentially be used to decorrelate an extra component.

5.5.2 Another View of the Skew-Symmetric Network

Since the interneurons in this skew-symmetric network search for a principal subspace of the input distribution in order to decorrelate the outputs $\mathbf{y}$, an alternative view of this network is to consider the interneurons $\mathbf{z}$ themselves at the output (Fig. 5.7) In this case, we have separate

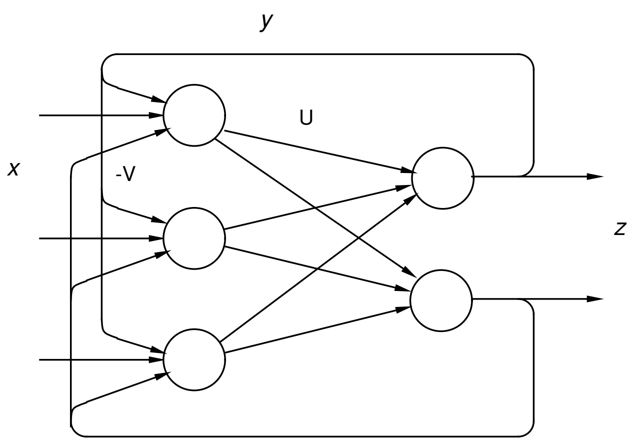


Fig. 5.7: The Skew-Symmetric Network, with the interneurons as outputs.

forward excitation connections U, and backward inhibition connections V, with

$$\mathbf{z} = U^T \mathbf{y} \qquad \text{and} \qquad \mathbf{y} = \mathbf{x} - V\mathbf{z}$$

i.e.

$$\mathbf{z} = U^T (I_N + VU^T)^{-1} \mathbf{x} \quad (5.41)$$

If initially we let $U = V$, both U and V have access to identical local information (i.e. they are both connected to $\mathbf{z}$ and $\mathbf{y}$ units), so we can use the same algorithm (5.36) for both of them. Thus they remain the same for all t, and produce the same result as above, but without explicitly forcing the forward excitation and backward inhibition to be the same all the time.

We can also verify that if $U = V$, components of Σ_Z in the direction of eigenvectors with large ($\gg 1$) eigenvalues of $VU^T = UU^T$ will have variance varying with the root of the variance of the corresponding input component. For example, for a single input and output, at the solution we have

$$E(z^2) = E\left((uy)^2\right) = u^2 E(y^2)$$

but

$$\begin{aligned} E(y^2) &= 1 \\ &= (1+u^2)^{-2} E(x^2) \\ &\approx u^{-4} E(x^2) \end{aligned}$$

so

$$E(z^2) \approx \left(E(x^2)\right)^{1/2}.$$

So although this interpretation does not cause the $\mathbf{z}$ output units to be decorrelated, they are *less* correlated (by a power of a half) than they were before.

This result is rather suggestive, since the analysis of our optimal filter in the previous chapter gave a requirement for a similar square-root-variance requirement if the input noise is significant, but not large (bound 2 on p. 43).

This arrangement also has another interesting interpretation: the V connections are reminiscent of the cortical back-projections found in the primate cortex [Kuffler *et al.*, 1984, Rolls, 1989]. This network therefore suggests that cortical back-projections could be acting as decorrelators for the neurons in the first area, allowing them to transmit their information efficiently to the second area. In doing so, the second area receives the most important information-bearing components of the first area.

While this is in no way conclusive, and doesn't even take into account non-linearities in the system, it is rather interesting since it makes the following prediction: if this model is a possible simple model of cortical processing, the effect of cortical back-projections should be *inhibitory* rather than excitatory, although not completely so. Under this interpretation cortical back-projections would act as regulators, forcing all events and correlations of events to be weighted equally.

5.5.3 Example Simulations

To illustrate the skew-symmetric network algorithm on a simple example, we show the algorithm operating on the same test data as section 5.4.1, under a couple of different conditions.

One Interneuron ($N = 2,\ M = 1$)

For this example (Fig. 5.8), we have a simple output interneuron, with the direction of the weight vector V indicated by the line on the plots. We can see that as the algorithm progresses, the weight vector lines up with the principal component of the input data, while decorrelating the data as best it can. In this example we used $\lambda^{-1} = 2$, which is the target for the final output covariance components.

The eigenvalues of the input covariance matrix were 164.1 and 1.14, while those of the final output were 2 and 1.23. Thus the covariance target had been achieved for the principal component of the input, while leaving the other component approximately unchanged. The weight found was [0.859 0.511], which compares favourably with the true principal component of the input covariance matrix of [0.862 0.5050].

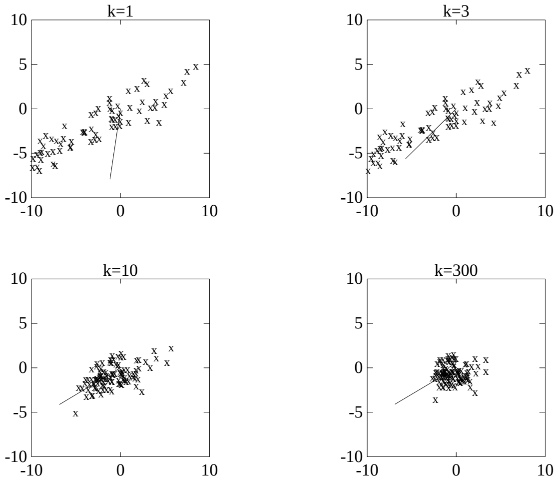


Fig. 5.8: Output Sequence for $M = 1$

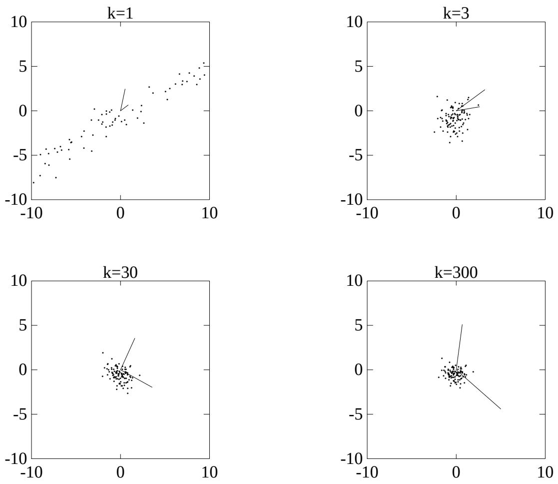


Fig. 5.9: Algorithm operating with two interneurons

Two Interneurons ($N = M = 2$)

In this example (Fig. 5.9), we have two interneurons, and a target variance of $\lambda^{-1} = 0.5$ which is less than both of the input covariance components. The lines show both the direction and size of the weight vectors associated with both outputs.

As the algorithm runs, the input sequence is decorrelated to within simulation accuracy. The weight vectors tend to move apart, but do not end up at right angles—but then they are not required to do so, provided the eigenvectors of the weight matrix tend towards the eigenvectors of the input covariance matrix. In fact, the eigenvectors of the weight matrix are at [0.858 0.513] and [-0.513 0.858], which again compare favourably with the principal eigenvectors of the input distribution at [0.862 0.5050].

The eigenvalues of the interneurons' final covariance matrix Σ_Z were 9.19 and 0.25 (a ratio of 36.5) compared with 164.1 and 1.14 for the input covariance matrix Σ_X (a ratio of 143.5). Thus the signal at the interneurons is less correlated than that at the input, although not quite by a power of a half as would be for the large-V regime, since the weight values required to normalise the smaller of the two input components is not much larger than 1.

5.6 Discussion

In this chapter, we started with the aim of finding a learning algorithm which could optimise transmission of information, but where the output units were susceptible to corrupting noise.

Initially, we considered Linear Predictive networks, which attempt to minimise the variance of each of the output units, by subtracting the optimal amount of each of the preceding inputs or outputs from it. These networks do not change the information transmitted through the network (provided the noise on the output is small), so they get the most efficient information transmission when the variance of all the outputs is minimised, thus minimising the power required to transmit that information. At the optimal solution, the outputs are decorrelated.

Moving on to fully recurrent lateral inhibition, we found that we could optimise information transmission provided we forced the outputs not only to be decorrelated, but also to have the same variance. In other words, the optimal probability distribution of output vectors is a spherical Gaussian distribution. The algorithm required is similar to the anti-Hebbian algorithm suggested by Barlow and Földiák [1989], but with an additional 'self-inhibitory' element for connections from each unit back to themselves. Provided the recurrent inhibitory connections are symmetrical, this algorithm will optimise information transmission, even in the case of significant noise on the output.

Finally, we moved on to the use of inhibitory interneurons in a novel skew-symmetric network, and found that a simple anti-Hebbian algorithm with a weight-decay term was required to optimise information transmission through this system. In this network, the interneurons find the principal subspace of the input distribution, and 'neutralise' these components by subtracting away most, but not all, of the original. The remaining signal has the same variance in all the high-variance directions 'found' by the interneurons.

This skew-symmetric network has several interesting properties, when compared with the previous ones. Firstly, if the input covariance matrix has only a few very large components, it can be effectively decorrelated using only that number of interneurons. If M is the number of interneurons, with N inputs, the input could be partially decorrelated using $2MN$ connections, instead of N^2. If the number of components of the input distribution with large variance is much less than $N/2$, this could offer a considerable saving.

In its other guise, the interneurons of this skew-symmetric network could be the outputs rather than simply interneurons. In the process of optimally decorrelating the outputs, the interneurons would span the same space as the top eigenvectors of the input covariance matrix. As we saw in chapter 3, this would be optimal if the the significant noise was on the input rather than the output, with not too much noise on the output.

We could use this to suggest a possible model for simple inter-area cortical processing, as follows. Let us assume that information is to be transmitted from $\mathbf{y}$ neurons in area A to $\mathbf{z}$

neurons in area B, over a relatively long distance. The information is therefore susceptible to corrupting noise across this connection. The $\mathbf{y}$ neurons each send a single axon to area B, which then splits until it synapses on the $\mathbf{z}$ neurons, with excitatory connection weights given by U (Fig. 5.10). The $\mathbf{z}$ neurons in area B each send an axon back to area A, which synapses onto the $\mathbf{y}$ neurons.

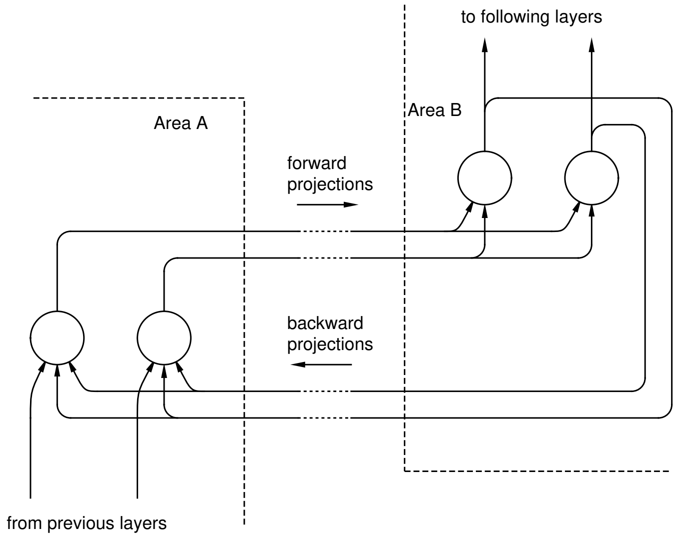


Fig. 5.10: A Simple Model of Cortical Back-Projections

With the algorithm outlined above, the outputs of the $\mathbf{y}$ neurons will be decorrelated with equal variance, and the axons from area A to area B will have optimal information transmission for the power used to transmit it. Meanwhile, the response of the $\mathbf{z}$ units in area B spans the first few principal components of the original input $\mathbf{x}$ to the $\mathbf{y}$ neurons in area A, thus retaining optimal information about the original input $\mathbf{x}$.

For other types of neural network processing, it may not be possible to provide the inhibitory back-projections from the following areas. In this case, local interneurons could perform a similar task instead (Fig. 5.11). This may be the case in the mammalian retina, for example, where the horizontal cells and amacrine cells may be providing this decorrelating function.

This suggestion is not attempting to be a detailed model of cortical function: there are many factors which have not been taken into account. Biological neural systems use pulse-modulated information transmission, and hence can only have a positive response. The response of cells is non-linear rather than linear, as we have used in our analysis so far. The 'running cost' associated with a cell may not be a simple mean squared amplitude, and the noise introduced may not be white additive Gaussian. However, even taking these objections into account, we feel that this model is sufficiently interesting to merit further work.

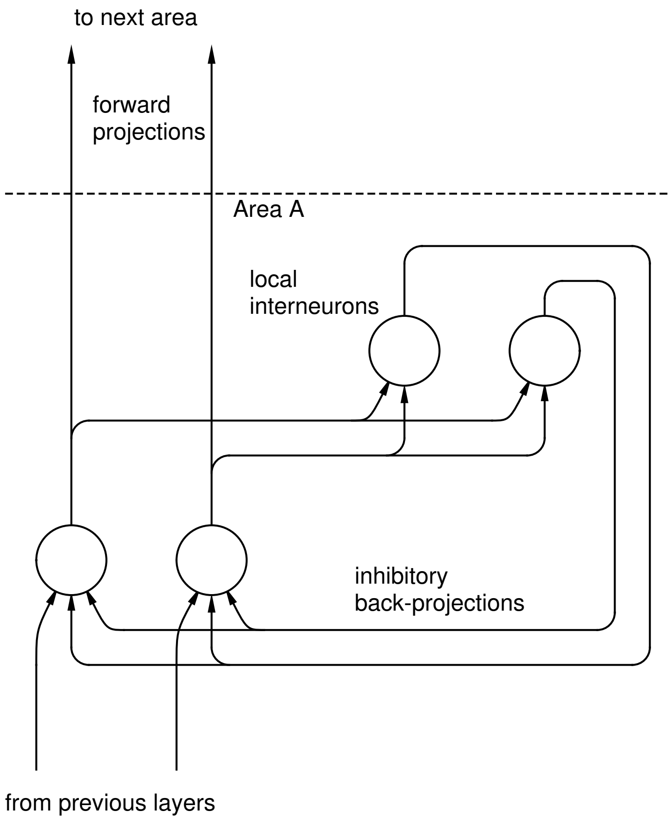


Fig. 5.11: Decorrelating with Local Interneurons

Chapter 6

Conclusions

6.1 General Conclusions

In most of this thesis, we have restricted our consideration to linear unsupervised learning systems, which primarily measure second order statistics only (i.e. variances of the data). The approach of minimising information loss, or bounding information loss, has led to several interesting results and led directly to the derivation of the popular Hebbian and anti-Hebbian learning algorithms.

Initially, we showed that the principle of minimisation of information loss has particular implications for supervised learning schemes such as Error Back Propagation. In particular, the use of the 1-of-N representation to minimise Bayesian error is only a good match when the distribution of errors is relatively constant over all possible classes. If the purpose of the system is to minimise Bayesian error, this is no problem, since the learning scheme has tried to optimise the required performance measure directly. However, if it is to be used as part of a larger system, a phoneme recogniser used as part of a word or sentence recogniser for example, learning to minimise Bayesian error on phonemes will not necessarily improve the final word or sentence recognition performance.

The relationship between minimisation of information loss across a system, and minimisation of final error, either Bayesian error or due to some distortion measure, means that minimisation of information loss across initial unsupervised pre-processing networks is a reasonable approach to take. We have considered in some detail the operation and development of unsupervised learning algorithms with this objective in mind.

A variety of Hebbian learning algorithms have been shown to be similar as far as reduction of information loss is concerned. They all attempt to directly decrease information loss over time for a system where the noise on the input to the network is dominant, and they find the space spanned by the principal components of the input data.

For significant output noise, we considered the information loss in the frequency domain taking the cost of information transmission into account. We get a non-trivial trade-off between the extremes of small input noise and small output noise. Atick and Redlich have used a very similar approach to produce very close curve fits to observed responses of biological visual systems, at various noise levels. This heavily suggests that biological systems somehow take advantage of this constrained minimisation of information loss.

Finally, for the extreme of dominant output noise, we have examined various anti-Hebbian learning algorithms, and show that some of these attempt to minimise the transmission power cost by decorrelating their outputs, while keeping the actual transmitted information constant. For a more complex network we derive a learning algorithm for directly reducing the constrained information loss function over time. This leads naturally to a Skew-Symmetric Network, which uses interneurons to perform the decorrelation function. This network offers a possible explanation for both the operation of biological interneurons in the retina and back-projections found in the cortex. We feel this would be an interesting candidate for further work.

This work represents a start along the road of the serious use of unsupervised learning networks in sensory system design. The eventual aim of this work is to attempt to take most of the 'difficult' part of difficult sensory tasks, such as speech recognition and object recognition, into the unsupervised pre-processing stages. By working harder on the way that sensory data can be represented in neural network systems, we hope to reduce the supervised part to something almost trivial in comparison. In this way, the problems of 'local minima' and so on associated with Error Back Propagation and other supervised learning schemes will be very much reduced.

Of course, this is a long way off. However, we feel that with continued effort in this direction, we may be able to gain much in the way of an increased understanding of the representation and processing of sensory information.

6.2 Further Work

For any practical sensory system, we must allow the use of non-linear processing stages before we can hope to solve the problem of designing artificial sensory systems. In much of the preceding analysis, we have often assumed that the input data to the network or sensory system was distributed according to a multivariate Gaussian probability density. In the absence of this assumption, the best we can do is to assert that an *upper bound* on the information loss is minimised instead. If the probability density associated with the data is far from Gaussian, this bound may be very loose, and may not be particularly useful.

6.2.1 Non-linear pre-processing

One simple way of introducing non-linearities is in the form of a simple fixed non-linear processing stage before a linear adaptive stage. This is the approach taken with the use of Volterra Expansions [Rayner and Lynch, 1989]. In this approach, the input vector $\mathbf{x}$ is expanded into a larger vector $\mathbf{x}'$ consisting of the polynomial expansion of $\mathbf{x}$, possibly truncated after the first few terms. For example, if we have

$$\mathbf{x} = (x_1, x_2, x_3)$$

this would give us

$$\mathbf{x}' = (1, x_1, x_2, x_3, x_1^2, x_1x_2, \ldots, x_3^2, x_1^3, x_1^2x_2, \ldots, x_3^3).$$

This has been applied to supervised learning networks, and has been shown to allow fast learning algorithms to be applied to the network, since the adaptive part is now entirely linear.

An example from a biological system is the light-sensitive rod and cone receptors in the retina. These have an approximately logarithmic response [Dowling, 1987]. Since a visual image is composed primarily of light reflected from surfaces, at any point the received intensity x is the product of the incident light and the surface reflectivity, i.e.

$$x(i,j) = l(i,j)r(i,j)$$

where the illumination l and the reflectance r are independent. A logarithmic response in the receptors would give us

$$x' = \log(x) = \log(l) + \log(r)$$

i.e. the sum of two independent terms. This is the basis of *homomorphic filtering*, as used in both image processing [Gonzalez and Wintz, 1987] and speech processing [Rabiner and Schafer, 1978]. If $\log(l)$ and $\log(r)$ have Gaussian distributions, then $\log(x)$ will also have a Gaussian distribution.

Any non-linear pre-processing stage has particular implications if used before, say, a Principal Component Analysing network of chapter 3. The assumption which we made in that chapter was that the input signal to the PCA stage contained spherical additive Gaussian noise, in particular that the noise is independent of signal level. If we use a logarithmic stage before our PCA network, we are effectively making the assumption that the noise on the input signal is *multiplicative* rather than additive, i.e.

$$x = (1+\nu)\omega$$

for some small independent noise term ν, or

$$log(x) \approx \log(\omega) + \nu.$$

Of course, this is unlikely to be valid for very small signal levels, since there may also be additive noise on the received signal which may swamp this multiplicative noise: but this may be reasonable at normal signal levels.

Unfortunately, there is no similar interpretation for the Volterra non-linearity outlined previously: it seems unlikely that we could have independent additive noise on all of x_1, x_2 and $x_1 x_2$ at the same time.

Higher-order statistics

Another way to introduce non-linearities in a neural network sensory system is within the network itself. This may enable the network to take advantage of statistics in the input data of higher than second order, i.e. other than simple means and variances.

Biological suggestive evidence that this may be important comes from the form of the receptive fields of simple cells in the visual cortex [Kuffler *et al.*, 1984]. These have been compared to 2-dimensional Gabor functions [Daugman and Kammen, 1987], the product of a sinusoid with a Gaussian, as shown in Fig. 6.1. Sanger [Sanger, 1989b] and Linsker [Linsker, 1989b] have found

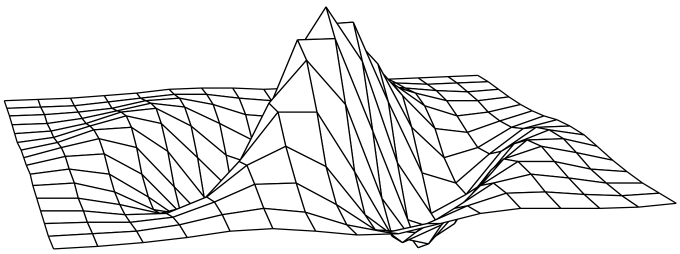


Fig. 6.1: A 2-dimensional Gabor function

that the principal components of patches of a typical image include this 'line detector' form of solution. However, the principal components also include, for example, components with four alternating (positive and negative) quadrants to their receptive fields, which apparently have no biological equivalent.

The problem here could be due to the image data having significant higher than second order statistics, leading to a very non-Gaussian distribution, for which a PCA approach would be inappropriate. One way to look at this is to consider the way that images tend to be composed of objects with edges. If only the second order statistics were important, a linear combination of a number of original images would produce images with similar second order statistics to the originals. However, we are unlikely to observe images with a half-intensity left-to-right edge overlayed on a half-intensity top-to-bottom image, except in the rare case of a translucent material overlayed on another object. One simple edge at a certain point of an image virtually precludes the appearance of another simple edge at the same point. This gives rise to the type of probability distribution illustrated in Fig. 6.2, for the response of two Gabor functions at right angles, to an image. This probability distribution is obviously non-Gaussian, since it is rare for the Gabor filters to have a significant response at the same time as each other.

These higher-order statistics may be behind the use of these Gabor-like filters in the mammalian visual system (there are suggestions that Gabor functions *per se* may not provide a completely accurate description of the response of cortical cells [Field, 1987, Stork and Wilson, 1989], although qualitatively the form of the responses is similar).

6.2.2 Winner-Take-All: The ultimate non-linearity

Arguably the simplest form of non-linearity commonly used in artificial neural networks (and other applications) is the Winner-Take-All (WTA) or 'pick the biggest' scheme [Grossberg, 1978,

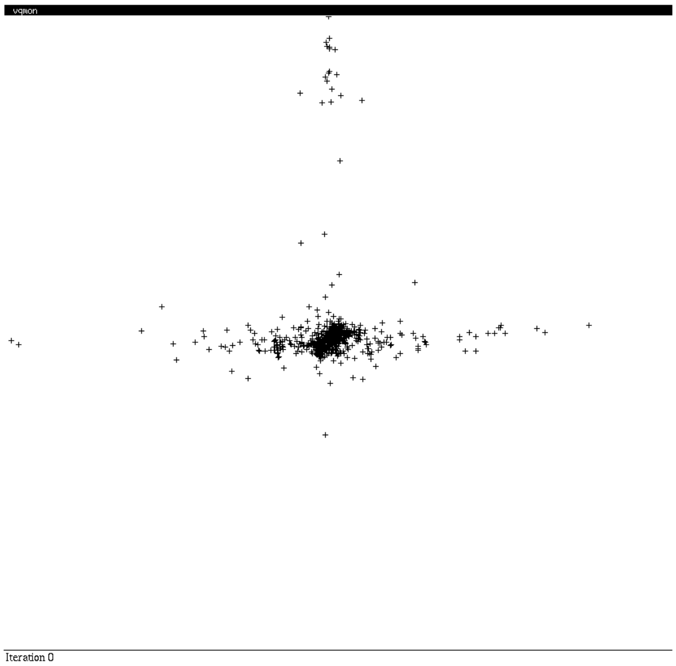


Fig. 6.2: A scatter plot for the response of two Gabor filters

Kohonen, 1984, Rumelhart and Zipser, 1985, Moore, 1988, Linde *et al.*, 1980]. Essentially, a WTA network has the same number N of outputs as inputs, and the output which corresponds to the largest input component is set to 1, while the others are set to zero. In other words, we have

$$y_i = \begin{cases} 1 & \text{if } o_i \geq o_j \qquad 1 \leq j \leq N \\ 0 & \text{otherwise} \end{cases}$$

(although this value may be undefined if there is not a single 'biggest' input).

Essentially, an element of this type allows any probability distribution over input data to be approximated in a piecewise-constant manner: the areas for which a particular input i is largest will be approximated by the single output state $y_i = 1, y_j = 0$ for $j \neq i$. Many schemes which use this technique are essentially modifications from the vector quantisation (VQ) approach [Linde *et al.*, 1980]. In a VQ system, each output y_i has an associated *target* vector $\mathbf{t}_i$, which is used in conjunction with a distortion measure $D(\mathbf{t}_i, \mathbf{x})$. The non-zero output $i^*$is chosen according to the target with minimum distortion from the output, i.e.

$$i^* = \arg\min_i D(\mathbf{t}_i, \mathbf{x}).$$

The targets are adjusted according to a simple procedure which finds a local minimum in the mean distortion ('error').

Since we can consider these outputs as additive noise which the vector quantisation process has added to our data, we can relate this in to our information loss principle in a very straight-forward way. In the case of a mean-squared-error measure $D(\mathbf{t}, \mathbf{x}) = |\mathbf{t} - \mathbf{x}|^2$, the information loss will be bounded in the same way as additive noise through the system considered in section 2.4.4. Thus minimising the mean squared error, say, will minimise a bound on the information loss, provided the input $\mathbf{x}$ has additive noise.

Non-linear dimensionality reduction

One interesting variant of this system is the Kohonen self-organising Feature Map [Kohonen, 1982, Kohonen, 1984], which maps input data in a high-dimensional space onto a position on a (usually) 2-dimensional grid, although other 3-dimensional maps have also been used [Ritter *et al.*, 1989]. The feature map approach is similar to the VQ approach, although additional constraints are built into the feature map to force target vectors which are neighbours on the feature map grid to become similar.

The use of this feature map approach is based on the idea that the patterns of interest occupy some low dimensional manifold within the high-dimensional input space. For example, although a steady-state vowel may be presented to a stage of a sensory system as a number of log FFT parameters say a 128-dimensional input space, it can be also be described by its position on a vowel quadrilateral—a two-dimensional representation [Kohonen, 1984]. The feature map, then, is attempting to perform a non-linear dimension reduction using a set of piecewise-constant approximations.

The Error Back-Propagation technique has also been suggested for non-linear dimension reduction, using a constrained version of its auto-encoder setup [Saund, 1989]. This has the advantage that its is not restricted to a piecewise-constant approximation of the input, so may perform better with a smaller number of units.

Intrinsic dimensionality

The *intrinsic dimensionality* of a set of data is the dimensionality of the manifold which it occupies within its representation space. It is sometimes known as the *fractal dimension*, since fractal curves, although drawn on a plane using lines, have a fractional value for their intrinsic dimensionality. This dimensionality is measured by observing the way that the number of spheres needed to *cover* the data (i.e. enclose all the data points) varies with the size of the spheres. Fig. 6.3 illustrates this idea. In (a) and (b), for a 2-dimensional object, the number of spheres

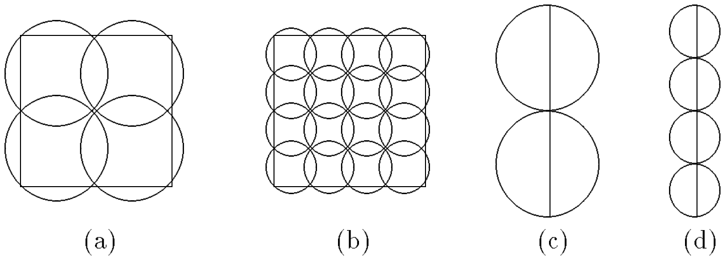


Fig. 6.3: Illustration of intrinsic dimension

goes up with second power of the inverse of radius; while in (c) and (d), for a one-dimensional object, the number of spheres goes up with the first power of the inverse of the radius. The intrinsic dimensionality can be computed using various methods, including a K-nearest-neighbour approach [Pettis *et al.*, 1979], observing the distance from the neighbours as K changes; and a vector quantisation approach, observing the mean squared error for various numbers of target vectors [Plumbley, 1989].

The feature map approach is mainly suited to extracting a manifold of only a few dimensions. If we needed a 6-dimensional feature map, with an accuracy of 1 in 10 in each dimension, we would need $10^6 = 1000000$ units. Although the vowel quadrilateral may only be two-dimensional, estimates of intrinsic dimensionality from speech data with varying speakers and utterances suggest that a value of 12 or 15 dimensions may be required to represent the variation in speech [Baydal *et al.*, 1989].

The determination of intrinsic dimensionality from a data set is not entirely straightforward, however. While it is easy to imagine our input data occupying some low-dimensional manifold of the input space, any noise on the input will perturb this data until it occupies possibly all of the input space dimensions. It may also be possible that in one region of our input space, the data occupies roughly 2 dimensions, while in another, more dimensions are occupied. In a speech example, it may be that vowels occupy two dimensions, while consonants occupy a higher-dimensional manifold in the space.

In order to proceed with our representation problem, we may have to look again at the idea of intrinsic dimensionality, to discover if something more appropriate can be done for real datasets.

6.2.3 Generalising Winner-Take-All

The Winner-Take-All approach is very popular, and has been used with success in simple cases. However, for a more complex sensory system, it is too restricted for what we need. As a form for representing perceptual information, it only allows one possibility to be represented at once: in speech recognition, this may correspond to the presence of one vowel or one consonant.

Real data is rarely this simple. Perhaps the identity of the vowel is uncertain, so a probabilistic representation may be more appropriate. Perhaps there is other information in the input data which we would like to represent: the speaker identity, speaking rate, emphasis, and so on. Having a single output unit for each possible combination of these variables would obviously be wasteful, and would require a huge number of units, as a high-dimensional feature map would do.

One approach to this problem, taken in image processing, is to divide the input image up into patches, and treat each patch independently. By using a one-of-N representation for each patch, the entire image is represented in parallel, using one group of units for each patch. A possible generalisation of this idea may be to attempt to represent the input values as the sum of several partial independent sub-representations. If the sub-representations are orthogonal to each other, we will have a similar situation to the separate processing of patches outlined above: if not, a more complex representation is possible.

Linear interpolation and Temporal Decomposition

A similar approach has been suggested for the recognition of speech in noise [Kadirkamanathan and Varga, 1991], where a noisy speech signal is represented as the sum of a speech and a noise model (in practice, they use either the speech or the noise model for each frequency component, whichever dominates). In a similar vein, Atal [Atal, 1983] suggested that co-articulation in speech could be modelled by 'Temporal Decomposition' (TD). Here some representation (log area ratios) of the speech waveform at any point in time is linearly interpolated from the vowels and consonants which occur just before and just after that point in time [Niranjan, 1990].

Rather than approximating the input data in a piecewise-constant way, TD uses a piecewise-linear interpolation between extremes. Thus if we regard the vowel and consonant extremes as target vectors $\mathbf{t}_i$, $1 \leq i \leq N$, we attempt to construct a representation of our input data using the sum of positive amounts of our target values, i.e.

$$\hat{\mathbf{x}} = \sum_i a_i \mathbf{t}_i$$

with $a_i \geq 0$, and also possibly $a_i \leq 1$ for true linear interpolation. This suggests a possible rôle for systems capable of representing positive values only, such as the pulse-frequency system used in biological systems [Kuffler *et al.*, 1984]. Rather than simply attempting to represent the presence or absence of a number of possibilities, perhaps we should instead represent the *amount* of each of these possibilities present in the input data.

Of course, if required, a system with positive-only values for a_i can be used to represent both positive and negative amounts of a given target, by using positive and negative versions of the target vectors thus:

$$\hat{\mathbf{x}} = \sum_i a_i^+ \mathbf{t}_i^+ + a_i^- \mathbf{t}_i^-$$

where $t_i^- = -t_i^+$. In particular, simulations suggest that the Skew-Symmetric Net describes in section 5.5 can develop positive and negative versions of its interneurons, if they are given a rectification non-linearity.

If the probability distribution of the input is not symmetrical about the origin, the use of separate positive and negative targets may enable a more efficient representation to be found than by forcing the positive and negative targets to be identical.

Bibliography

[Abu-Mostafa and Jaques, 1985] Yaser S. Abu-Mostafa and Jeannine-Marie St. Jaques. Information capacity of the Hopfield model. *IEEE Transactions on Information Theory*, IT-31:461–464, 1985.

[Ackley *et al.*, 1985] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Journal of Cognitive Science*, 9:147–169, 1985.

[Atal, 1983] B. S. Atal. Efficient coding of LPC parameters by temporal decomposition. In *Proceedings of ICASSP-83*, Boston, MA., 1983.

[Atick and Redlich, 1989a] Joseph J. Atick and A. Normal Redlich. Predicting ganglion and simple cell receptive field organizations from information theory. Technical Report IASSNS-HEP-89/55, NYU-NN-89/1, School of Natural Sciences, Institute for Advanced Study, Princeton; Center for Neural Science, New York University, October 1989.

[Atick and Redlich, 1989b] Joseph J. Atick and A. Normal Redlich. Quantitative tests of a theory of retinal processing: Contrast sensitivity curves. Technical Report IASSNS-HEP-90/51, NYU-NN-90/2, School of Natural Sciences, Institute for Advanced Study, Princeton; Center for Neural Science, New York University, October 1989.

[Atick and Redlich, 1989c] Joseph J. Atick and A. Normal Redlich. Towards a theory of early visual processing. Technical Report IASSNS-HEP-90/10, NYU-NN-90/2, School of Natural Sciences, Institute for Advanced Study, Princeton; Center for Neural Science, New York University, October 1989.

[Attneave, 1954] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61:183–193, 1954.

[Attneave, 1959] Fred Attneave. *Applications of Information Theory to Psychology.* University of Illinois Press, New York, Chicago, London, 1959.

[Bahl *et al.*, 1986] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of ICASSP-86*, pages 49–52, 1986.

[Barlow and Földiák, 1989] Horace B. Barlow and Peter Földiák. Adaptation and decorrelation in the cortex. In Durbin et al. [1989], chapter 4, pages 54–72.

[Barlow, 1961] H. B. Barlow. Three points about lateral inhibition. In W. Rosenblith, editor, *Sensory Communication*, pages 782–786. MIT Press, 1961.

[Barlow, 1987] H. B. Barlow. A theory about the functional role and synaptic mechanism of visual after-effects. In C. Blakemore, editor, *Vision: Coding and Efficiency.* Cambridge University Press, 1987.

[Baydal *et al.*, 1989] E. Baydal, G. Andreu, and E. Vidal. Estimating the intrinsic dimensionality of discrete utterances. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-37:755–757, 1989.

[Becker and Hinton, 1989] Suzanna Becker and Geoffrey E. Hinton. Spatial coherence as an internal teacher for a neural network. Technical Report CRG-TR-89-7, Department of Computer Science, University of Toronto, December 1989.

[Becker, 1991] Suzanna Becker. Unsupervised learning procedures for neural networks. *International Journal of Neural Systems*, 1991. (To appear).

[Bourlard and Kamp, 1987] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular-value decomposition. Manuscript M217, Philips Research Laboratory, Brussels, November 1987.

[Bridle, 1990] John S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Françoise Fogelman Soulié and Jeanny Hérault, editors, *Neurocomputing - Algorithms, Architectures and Applications*, volume F68 of *NATO ASI Series*, pages 227–236. Springer-Verlag, Berlin, 1990.

[Cornsweet, 1970] Tom N. Cornsweet. *Visual Perception.* Academic Press, New York, 1970.

[Csiszár, 1975] I. Csiszár. I-divergence geometry of probability distributions and minimisation problems. *The Annals of Probability*, 3(1):146–158, 1975.

[Daugman and Kammen, 1987] John G. Daugman and Daniel M. Kammen. Image statistics, gases, and visual neural primitives. In *Proceedings of the IEEE First International Conference on Neural Networks, San Diego, CA.*, volume IV, pages 163–175, 1987.

[Dayan, 1991] Peter Dayan. Personal Communication, 1991.

[Dowling, 1987] John E. Dowling. *The Retina: an approachable part of the brain.* Harvard University Press, Cambridge, MA., 1987.

[Durbin *et al.*, 1989] Richard Durbin, Christopher Miall, and Graeme Mitchison, editors. *The Computing Neuron.* Computation and Neural Systems. Addison-Wesley, Wokingham, England, 1989.

[Easton and Gordon, 1984] Paul Easton and Peter E. Gordon. Stabilisation of Hebbian neural nets by inhibitory learning. *Biological Cybernetics*, 51:1–9, 1984.

[Field, 1987] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4:2379–2394, 1987.

[Field, 1989] David J. Field. What the statistics of natural images tell us about visual coding. In *Proceedings of SPIE, Los Angeles, CA.*, January 1989.

[Földiák, 1989] Peter Földiák. Adaptive network for optimal linear feature extraction. In *Proceedings of the International Joint Conference on Neural Networks*, pages 401–405, Washington D.C., June 18–22 1989.

[Gerbrands, 1981] Jan J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 14:375–381, 1981.

[Golub and van Loan, 1983] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* North Oxford Academic, Oxford, England, 1983.

[Gonzalez and Wintz, 1987] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing.* Addison-Wesley, Reading, MA, second edition, 1987.

[Green and Courts, 1966] R. T. Green and M. C. Courts. Information theory and figure perception: The metaphor that failed. *Acta Psychologica*, 25:12–36, 1966.

[Grossberg, 1978] Stephen Grossberg. A theory of visual coding, memory, and development. In E. L. J. Leeuwenberg and H. F. J. M. Buffort, editors, *Formal Theories of Visual Perception*, pages 7–26. Wiley, Chichester, UK, 1978.

[Grossberg, 1986] Stephen Grossberg. The adaptive self-organization of serial order in behavior: Speech, language and motor control. In E. C. Schwab and H. C. Nusbaum, editors, *Pattern Recognition by Humans and Machines: Volume 1: Speech Perception*, pages 187–294. Academic Press, London, 1986.

[Hall, 1979] Ernest L. Hall. *Computer Image Processing and Recognition.* Academic Press, New York, 1979.

[Hampshire and Pearlmutter, 1990] John B. Hampshire and Barak A. Pearlmutter. Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function. In Touretsky, Elman, Sejnowski, and Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School.* Morgan Kaufmann, San Mateo, CA, 1990.

[Hand, 1981] D. Hand. *Discrimination and Classification.* Wiley, Chichester, UK, 1981.

[Hebb, 1949] Donald O. Hebb. *The Organization of Behavior.* Wiley, New York, 1949.

[Hirai, 1980] Yuzo Hirai. A template matching model for pattern recognition: Self-organization of templates and template matching by a disinhibitory neural network. *Biological Cybernetics*, 38:91–101, 1980.

[Jelinek, 1985] Frederick Jelinek. The development of an experimental discrete dictation recogniser. *Proceedings of the IEEE*, 73(11):1616–1624, 1985.

[Julesz, 1971] B. Julesz. *Foundations of Cyclopean Perception.* University of Chicago Press, Chicago, IL, 1971.

[Kadirkamanathan and Varga, 1991] Maha Kadirkamanathan and Andrew P. Varga. Simultaneous model re-estimation from contaminated data by "composed hidden Markov modelling". In Proceedings of ICASSP-91 [1991], pages 897–900.

[Kelly, 1962] D. H. Kelly. Information capacity of a single retinal channel. *IRE Transactions on Information Theory*, IT-8:221–226, 1962.

[Kohonen, 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[Kohonen, 1984] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, New York, second edition, 1984.

[Krasulina, 1970] T. P. Krasulina. Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices. *Automation and Remote Control*, 31(2):215–221, 1970.

[Kuffler *et al.*, 1984] 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.

[Kühnel and Tavan, 1990] Hans Kühnel and Paul Tavan. The anti-Hebb rule derived from information theory. In R. Eckmiller, G. Hartmann, and G. Hauske, editors, *Parallel Processing in Neural Systems and Computers*, pages 187–190. Elsevier Science Publishers, North-Holland, 1990.

[Kullback, 1959] S. Kullback. *Information Theory and Statistics.* Wiley, New York, 1959.

[Laughlin *et al.*, 1987] Simon B. Laughlin, J. Howard, and Barbara Blakeslee. Synaptic limitations to contrast coding in the retina of the blowfly calliphora. *Proceedings of the Royal Society of London, Series B*, 231:437–467, 1987.

[Laughlin, 1987] Simon B. Laughlin. Form and function in retinal processing. *Trends in Neuro-Sciences*, 10:478–483, 1987.

[Leeuwenberg, 1978] E. L. J. Leeuwenberg. Quantification of certain visual pattern properties: Salience, transparency, similarity. In E. L. J. Leeuwenberg and H. F. J. M. Buffort, editors, *Formal Theories of Visual Perception*, pages 277–298. Wiley, Chichester, UK, 1978.

[Linde *et al.*, 1980] Yoseph Linde, Andrés Buzo, and Robert M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, COM-28:84–95, 1980.

[Linsker, 1986] Ralph Linsker. From basic network principles to neural architecture: Emergence of spatial-opponent cells. *Proceedings of the National Academy of Science U.S.A.*, 83:7508–7512, 1986.

[Linsker, 1988a] Ralph Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, March 1988.

[Linsker, 1988b] Ralph Linsker. Towards an organisational principle for a layered perceptual network. In Dana. Z. Anderson, editor, *Neural Information Processing Systems (Denver, CO. 1987)*, pages 485–494. American Institute of Physics, New York, 1988.

[Linsker, 1989a] Ralph Linsker. An application of the principle of maximum information preservation to linear systems. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1 (Denver, CO., Nov.–Dec. 1988)*, pages 186–194. Morgan Kaufmann, San Mateo, CA, 1989.

[Linsker, 1989b] Ralph Linsker. Designing a sensory processing system: What can be learned from principal components analysis? Technical Report RC 14983, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 19598, 1989.

[Linsker, 1989c] Ralph Linsker. How to generate ordered maps by maximising the mutual information between input and output signals. Technical Report RC 14624, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 19598, 1989.

[Lippmann, 1989] Richard P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1(1):1–38, 1989.

[Marshall, 1990a] Jonathan A. Marshall. Adaptive neural methods for multiplexing oriented edges. In *Proceedings of the SPIE Symposium on Advances in Intelligent Systems*, Boston, November 1990.

[Marshall, 1990b] Jonathan A. Marshall. Representation of uncertainty in self-organizing neural networks. In *International Neural Network Conference INNC-90-PARIS*, pages 809–812, Palais des Congres, Paris, France, 1990. Kluwer Academic Publishers.

[Mellink and Buffart, 1987] H. Mellink and H. Buffart. Abstract code network as a model of perceptual memory. *Pattern Recognition*, 20:143–151, 1987.

[Moore, 1988] Barbara Moore. ART1 and pattern clustering. In David Touretzky, Geoffrey Hinton, and Terrence Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 174–185, San Mateo, CA., 1988. Morgan-Kaufmann.

[Niranjan, 1990] Mahesan Niranjan. *Modelling and Classifying Speech Patterns*. Ph. D. thesis, Cambridge University Engineering Department, May 1990.

[Oja and Karhunen, 1985] Erkki Oja and Juha 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.

[Oja, 1982] Erkki Oja. A simplified neuron model as a principal component analyser. *Journal of Mathematical Biology*, 15:267–273, 1982.

[Oja, 1983] Erkki Oja. *Subspace Methods of Pattern Recognition*. John Wiley & Sons, New York, 1983.

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

[Orphandis, 1990] Sophocles J. Orphandis. Gram-Schmidt neural nets. *Neural Computation*, 2:116–126, 1990.

[Papoulis, 1984] Athanasios Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, second edition, 1984.

[Pearlmutter and Hinton, 1986] Barak A. Pearlmutter and Geoffrey E. Hinton. G-maximization: An unsupervised learning procedure for discovering regularities. In J. S. Denker, editor, *Proceedings of Neural Networks for Computing*, pages 333–338. American Institute of Physics, 1986.

[Pettis *et al.*, 1979] Karl W. Pettis, Thomas A. Bailey, Anil K. Jain, and Richard C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:25–37, 1979.

[Plumbley and Fallside, 1988a] Mark D. Plumbley and Frank Fallside. An information-theoretic approach to unsupervised connectionist models. In David Touretzky, Geoffrey Hinton, and Terrence Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 239–245, San Mateo, CA., 1988. Morgan-Kaufmann.

[Plumbley and Fallside, 1988b] Mark D. Plumbley and Frank Fallside. An information-theoretic approach to unsupervised connectionist models. Technical Report CUED/F-INFENG/TR.7, Cambridge University Engineering Department, 1988.

[Plumbley and Fallside, 1989] Mark. D. Plumbley and Frank Fallside. Sensory adaptation: An information-theoretic viewpoint. In *Proceedings of the International Joint Conference on Neural Networks, Washington D.C.*, volume II, page 598, 1989.

[Plumbley and Fallside, 1991] Mark. D. Plumbley and Frank Fallside. The effect of receptor signal-to-noise levels on optimal filtering in a sensory system. In *ICASSP-91*, volume 4, pages 2321–2324, May 1991.

[Plumbley, 1987] Mark D. Plumbley. Speech recognition using unsupervised connectionist models. First Year Report, Cambridge University Engineering Department, 1987.

[Plumbley, 1989] Mark D. Plumbley. An information-theoretic approach to connectionist models. Unpublished manuscript, September 1989.

[Proceedings of ICASSP-91, 1991] *Proceedings of ICASSP-91*, Toronto, May 1991.

[Rabiner and Schafer, 1978] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, New Jersey, 1978.

[Rayner and Lynch, 1989] P. Rayner and M. R. Lynch. A new connectionist model based on a non-linear adaptive filter. In *Proceedings of ICASSP-89*, April 1989.

[Reddy *et al.*, 1982] V. U. Reddy, B. Eghart, and T. Kailath. Least squares type algorithm for adaptive implementation of pisarenko's harmonic retrieval method. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30:399–405, 1982.

[Ritter *et al.*, 1989] Helge Ritter, Thomas Martinez, and Klaus Schulten. Topology-conserving maps for motor control. In L. Personnaz and G. Dreyfus, editors, *Neural Networks from Models to Applications. (Proceedings of nEuro '88)*, pages 579–591. I.D.S.E.T., Paris, 1989.

[Robinson and Fallside, 1990] Tony Robinson and Frank Fallside. Phoneme recognition from the TIMIT database using recurrent error propagation networks. Technical Report CUED/F-INFENG/TR.42, Cambridge University Engineering Department, March 1990.

[Rolls, 1989] Edmund Rolls. The representation and storage of information in neuronal networks in the primate cerebral cortex and hippocampus. In Durbin et al. [1989], chapter 8, pages 125–159.

[Rumelhart and Zipser, 1985] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9:25–50, 1985.

[Rumelhart *et al.*, 1986] 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.* Bradford Books/MIT Press, Cambridge, MA, 1986.

[Sanger, 1989a] Terence D. Sanger. Optimal unsupervised learning in a single-layer feedforward neural network. *Neural Networks*, 1989. (submitted).

[Sanger, 1989b] Terence D. Sanger. An optimality principle for unsupervised learning. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1 (Denver, CO., Nov.–Dec. 1988)*, pages 11–19. Morgan Kaufmann, San Mateo, CA, 1989.

[Saund, 1989] Eric Saund. Dimensionality-reduction using connectionist networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:304–314, March 1989.

[Schwartz and Austin, 1991] Richard Schwartz and Steve Austin. A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses. In Proceedings of ICASSP-91 [1991], pages 701–704.

[Shannon, 1948] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:370–423, 623–656, 1948.

[Shannon, 1949] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37:10–21, 1949.

[Shore and Johnson, 1981] John E. Shore and Rodney W. Johnson. Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, IT-27(4):472–482, July 1981.

[Solla *et al.*, 1988] Sara A. Solla, Esther Levin, and Michael Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2, 1988.

[Srinivasan *et al.*, 1982] M. V. Srinivasan, S. B. Laughlin, and A. Dubs. Predictive coding; a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London, Series B*, 216:427–459, 1982.

[Stork and Wilson, 1989] David. G. Stork and Hugh R. Wilson. Do Gabor functions provide appropriate descriptions of visual cortical receptive fields? Submitted for publication, 1989.

[Strang, 1976] Gilbert Strang. *Linear Algebra and its Applications.* Academic Press, New York, 1976.

[Strizenec, 1975] Michal Strizenec. Information and mental processes. In Libor Kubat and Jiri Zeman, editors, *Entropy and Information in Science and Philosophy*, pages 149–153. Elsevier Scientific, Amsterdam, 1975.

[von der Malsburg, 1973] Christoph von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100, 1973.

[Watanabe, 1985] Satosi Watanabe. *Pattern Recognition: Human and Mechanical.* John Wiley & Sons, New York, 1985.

[Williams, 1985] Ronald J. Williams. Feature discovery through error-correction learning. ICS Report 8501, University of California, San Diego, 1985.

[Young, 1984] S. Young. Generating multiple solutions from connected word DP recognition algorithm. *Proceedings of the Institute of Acoustics*, pages 351–354, 1984.

[Zemel and Hinton, 1991] Richard S. Zemel and Geoffrey E. Hinton. Discovering viewpoint-invariant relationships that characterize objects. In *Advances in Neural Information Processing Systems 3*. Morgan Kaufmann, San Mateo, CA, 1991. Proceedings of NIPS*90, Denver, CO. (to appear).