# Contextually Guided Unsupervised Learning using Local Multivariate Binary Processors

**Jim Kay**[*]
*Department of Statistics*
*University of Glasgow*
*Glasgow G12 8QQ*
Scotland, UK

**Dario Floreano**
*MicroComputing Laboratory*
*Swiss Federal Institute of Technology*
*Ecublens - Lausanne CH - 1015*
Switzerland

**W. A. Phillips**
*Centre for Cognitive and*
*Computational Neuroscience*
*University of Stirling*
*Stirling FK9 4LA*
*Scotland, UK*

July 30, 1996

**Acknowledgements:**

We thank Sue Becker, Peter Hancock and Darragh Smyth for helpful comments on this work. The work of Dario Floreano and Bill Phillips was supported by a Network Grant from the Human Capital and Mobility Programme of the European Community.

**Running Title:** Local Multivariate Binary Processors

[*] Contact author for correspondence and reprints. Tel: 0141-330-6117. Fax: 0141-330-4814. e-mail: jim@stats.gla.ac.uk.

# Contextually Guided Unsupervised Learning using Local Multivariate Binary Processors

## Abstract

We consider the role of contextual guidance in learning and processing within multi-stream neural networks. Earlier work (Kay & Phillips, 1994, 1996; Phillips et al., 1995) showed how the goals of feature discovery and associative learning could be fused within a single objective, and made precise using information theory, in such a way that local binary processors could extract a single feature that is coherent across streams. In this paper we consider multi-unit local processors with multivariate binary outputs that enable a greater number of coherent features to be extracted. Using the Ising model, we define a class of information-theoretic objective functions and also local approximations, and derive the learning rules in both cases. These rules have similarities to, and differences from, the celebrated BCM rule. Local and global versions of Infomax appear as by-products of the general approach, as well as multivariate versions of Coherent Infomax. Focussing on the more biologically plausible local rules, we describe some computational experiments designed to investigate specific properties of the processors. The main conclusions are:

1. The local methodology introduced in the paper has the required functionality.
2. Different units within the multi-unit processors learned to respond to different aspects of their receptive fields.
3. The units within each processor generally produced a distributed code in which the outputs were correlated, and which was robust to damage; in the special case where the number of units available was only just sufficient to transmit the relevant information, a form of competitive learning was produced.
4. The contextual connections enabled the information correlated across streams to be extracted, and, by improving feature detection with weak or noisy inputs, they played a useful role in short-term processing and in improving generalization.
5. The methodology allows the statistical associations between distributed self-organizing population codes to be learned.

## Keywords

Multivariate Binary Processors, Unsupervised Learning, Information Theory , Contextual Guidance, Learning Coherence, Infomax, Coherent Infomax, Population Codes

# 1 Introduction

Natural environments provide many diverse sets of data containing information about many interrelated variables or features. Neural networks can be designed so that features are first discovered by some form of unsupervised learning and then associations between them are learned by some form of supervised learning. It is also possible to combine these two goals into a single objective, however, such that the features that are discovered are those that are associated, and to do this concurrently with discovering the associations between them. Furthermore, this latter approach can include the use of learned associations to enhance the detection of weak or noisy features and to group together coherent sets of features.

The diverse sets of data may arise from different sensory modalities, from separate regions of the receptor input space within a modality, or from different streams of processing that extract different cues from the same general region of input space within a modality. Consider the perception of an ambiguous speech sound or letter, for example. Knowledge of the identity of surrounding phonemes or letters, and of the transitional probabilities between phonemes or letters within words, could be used to help disambiguate the ambiguous input, and there is a great deal of evidence that this occurs in both speech perception and reading (e.g. Massaro & Cohen, 1991). For an example at earlier stages of processing, consider the attempt to detect a faint edge indicating the position and orientation of the boundary between land and sky in a distant hazy view. The detection of an edge within any particular region will be greatly affected by the extent to which there is supportive evidence from surrounding regions, and evidence that local context of evidence of this kind plays a role in edge contrast sensitivity is provided by Polat & Sagi (1993,1994), using psychophysical studies of human subjects, and by Kapadia et al. (1995) using parallel psychophysical studies with humans and neurophysiological studies with monkeys.

This paper is therefore concerned with multi-stream neural networks that discover statistical structure both within and between the diverse data sets upon which they operate. These nets can discover and use associative relations between the features extracted within the different streams because processors within each stream receive inputs from other streams, and use them as a local context that guides both learning and processing. The input that processors receive from this local context therefore has a different role from that which they receive from their receptive field, and this has quite distinct effects on the processors' activity. The role of the local processors within these nets is to select and recode that information within their receptive field inputs that is relevant to their role in the system as a whole. The contextual input from other streams helps to specify what that role is by providing a local context within which each processor operates. The contextual input should be able to select and emphasize any 'relevant' information that is present within the receptive field input but without hallucinating evidence that is not there. That is, the local context should make predictions that enhance the selection and interpretation of data in the streams of processing to which they project, but without becoming self-fulfilling prophecies. Within this framework, information is 'relevant' if it is statistically related to the context within which it occurs. The overall form of processing produced by this approach is that of multiple streams of processing which may converge and diverge in various ways through multiple levels of processing, and which, at any level, involve many specialized processors each of which transmits information about specific local environmental variables. These processors coordinate their activities through local contextual interactions so as to produce patterns of activity across the array of processors as a whole that are as mutually 'coherent' as possible. Here 'coherent' means that their activities are mutually predictive as specified by the associative knowledge that is embodied within the strengths of the contextual connections.

From the viewpoint of multivariate statistical data processing, the learning and processing capabilities of such networks can be seen as an extension of techniques for latent structure analysis and canonical correlation. They therefore combine the selective recoding role of techniques such as principal component analysis with the predictive role of multiple regression within a single objective. In particular, canonical correlation analysis may be considered as a neural network in which there are two streams of processing with the outputs of each stream extracting those linear functions of their input data that are mutually predictive across streams (Hotelling, 1936; Becker, 1992; Kay, 1992). This technique may also deal with more than two diverse data sets (Gifi, 1990)

Computational studies by several groups of workers have shown that neural networks can use coherence across multiple streams of processing as a basis for the discovery of important features within streams (e.g. Becker, 1992; Becker & Hinton,1992; Schmidhuber & Prelinger, 1993; De Sa, 1994; Stone, 1995; Becker, 1996). Furthermore, it has also been shown that this can be used with coherence across streams to enhance the short-term processing dynamics (Kay & Phillips, 1994, 1996; Phillips et al., 1995; Der & Smyth, 1996). These latter studies were restricted to the case where each local processor produces just a single probabilistic binary output, however. If the approach is to have either technological or biological relevance it is important to explore how it could be extended to processors that produce multiple distinct outputs (Becker, 1996). Receptive field inputs within streams will usually contain more than one bit of relevant information, and it is then necessary to use processors with more than one binary unit in a way that ensures that different units within a single processor respond to different aspects of the input while still maximizing coherence across streams. The primary goal of this paper is therefore to show how this is possible within the same general information-theoretic framework that was used in the earlier studies (Kay & Phillips, 1994, 1996; Phillips et al., 1995), and to see what relevance that might have for the biological plausibility of the general approach.

We begin by considering how the goal of learning within this approach can be specified formally. The goal of feature discovery can be expressed in information-theoretic terms, and it can be regarded as recoding to reduce redundancy; see, for example, Barlow (1961, 1989), Linsker (1988) Atich & Redlich (1993), Redlich (1993) and Taylor & Plumbley (1993). The basic idea is that the flood of data to be processed can be reduced to more manageable amounts by using the statistical structure within the data to recode the information that it contains into a more efficient form. Thus patterns that recur in the raw data can be translated into codes that contain many less elements than the patterns themselves. This goal provides a valuable perspective from which to view sensory processing; it is clear and simple, and can be specified formally at the level of the local processor. An important limitation of this goal is that it is ultimately sub-ordinate to the goal of associative learning, however; there would be no point recoding information about some variable if that variable bore no relation to anything else known to the system. Proponents of the goal of feature discovery therefore see it as preparatory to associative learning and hence feature discovery and associative learning are seen as separate and successive goals. Here we study ways that feature discovery and associative learning can be combined into a single algorithm that is used throughout the various stages of processing. Information theory is used to specify the goal of this algorithm in a way that is close to that used for feature discovery and which includes that as a special case. Following Linsker (1988) we call that latter goal Infomax, and by analogy, we call the goal on which we shall focus Coherent Infomax, because it seeks to maximize the transmission of just that receptive field information that is coherently related to the context within which it occurs, whereas Infomax maximizes the transmission of whatever components are most informative within individual streams considered separately.

It is next necessary to consider how it is possible for context to affect the activity of the units to which they project without interfering with the information that those units transmit about their receptive field input. This can be done by using context to to modulate the gain of the transfer function that maps receptive field inputs into the output signal; units will then need less evidence to produce a given level of probability of an output decision if that decision is in agreement with the contextual prediction, and will require more if there is disagreement. An activation function that provides this capability was derived (Kay, 1994; Kay & Phillips, 1994) by analogy with physiological mechanisms for gain control, and it has since been used in several other studies of contextual integration (Kay &Phillips, 1994, 1996; Phillips et al., 1995; Der & Smyth, 1996). This function is also used in the studies reported here and is specified formally in Section 3.

Studies of the contextual guidance of learning and processing are worthwhile because this may be a useful computational strategy, but they are also worthwhile because there is evidence from neuroanatomy and neurophysiology that some such strategy may be embodied in the cerebral cortex. This evidence will be reviewed in detail elsewhere (Phillips & Singer, 1996), but in brief:

- voltage-dependent synaptic receptor channels are common throughout cortex and would provide a mechanism for gain control; see, for example, Fox et al. (1990);

- long-range horizontal collaterals are also common (Gilbert, 1992) and could provide contextual input to the gain-control channels;

- synchronisation of the activity of cells with non-overlapping receptive fields (Singer, 1993) may reflect the the formation of cooperative population codes resulting from contextual guidance;

- activity-dependent plasticity of the synapses mediating the response to receptive field stimulation is well established (Singer, 1990);

- there is now also evidence that the long-range horizontal connections undergo activity-dependent changes in synaptic strength (Hirsch & Gilbert, 1993; Löwel & Singer, 1992); finally, as predicted by our hypotheses, there is evidence that the long-range horizontal collaterals in V1 have a voltage-dependent – and thus gain-controlling – synaptic physiology (Hirsch & Gilbert, 1991).

In this paper the methodology proposed by Kay & Phillips (1994,1996) is extended to the case where each local processor has multivariate binary outputs. The paper proceeds as follows. In Section 2 we define basic terms, establish our notation and describe the necessary aspects of the general form of Gibbs distribution on which our multivariate binary approach is based. In Section 3, we define a general class of objective functions and derive learning rules based on the probabilistic modelling outlined in section 2; this provides a formal approach, in particular, to multivariate versions of the methodology which are based on the Infomax and Coherent Infomax objective functions. The objective function considered here may be termed *global* as it seeks to maximize the throughput of information at the output units based on information measures calculated using a multivariate probability model. This multivariate approach leads to comprehensible, but fairly complicated, learning rules that are local at the level of the processor considered as a whole, but which are not local with respect to the individual units comprising the output layer. In order to obtain learning rules that are local at the level of the individual output units – and thus biologically more plausible, we consider in Section 4 the possibility of local objective functions which lead to local learning rules. We show one way in which this is possible and derive the learning rules in this case. In Section 5 we describe some applications of the methodology in computational experiments using multi-steam networks of multivariate processors of increasing complexity with respect to the training stimuli, the connectivity and the number of local processors. Finally in Section 6 we present our conclusions and describe how this approach may be developed in future work.

# 2 Probabilistic Modelling

We consider as a basic building-block the idea of a local processor. In this paper a local processor is a local circuit composed of several output units, variously interconnected, which share a common receptive field input and can also receive contextual input from other processors in the network. This allows each local processor to represent several features simultaneously. Such processors may be linked together to form complex architectures, as desired. For example, we may consider a stream to be a vertical collection of elements through which the information flows from inputs to outputs; thus a stream may be considered as a column of interconnected local processors. Such streams may be connected together in a matrix structure via lateral connections at various levels through which contextual information can flow.

A local processor has $p$ outputs which we represent by the collection of random variables, $\mathbf{X} = \{X_1, X_2, ..., X_p\}$, where each $X_i$ is a binary random variable. Each processor receives information from two distinct sources, namely, (a) its receptive field inputs and (b) its contextual field inputs. We also represent these inputs as random variables, with $\mathbf{R} = \{R_1, R_2, ..., R_m\}$ denoting the receptive field (RF) inputs and $\mathbf{C} = \{C_1, C_2, ..., C_n\}$ the contextual field (CF) inputs, respectively. If we view a complex network composed of such local processors as a multi-layer, multi-stream system, we would normally consider the RF of each local processor to consist of information flowing from processing performed at lower layers while the CF connections are from the same layer of other streams or consist of back-projections from higher layers in the system. As defined, $\mathbf{R}, \mathbf{C}$ and $\mathbf{X}$ are random vectors and we adopt the usual convention of denoting random variables by upper-case letters and their realised values by the correponding lower-case letters. To indicate formally the possibility of incomplete connectivity, we define connection neighbourhoods for each output unit $X_i$ within a local processor. Let $\partial i(r), \partial i(c)$ and $\partial i(x)$ denote, respectively, the set of indices of the RF input units, the CF input units and the output units that are connected to the ith output unit $X_i$. The corresponding random variables are denoted, respectively, by $\mathbf{R}_{\partial i}, \mathbf{C}_{\partial i}$ and $\mathbf{X}_{\partial i}$. We term these links the RF connections, the CF connections and the WP (within-processor) connections, respectively. The set of all components of $\mathbf{X}$, excluding the ith component, is denoted by $\mathbf{X}_{-i}$. The weights on the connections into the ith output unit are given by $w_{ij}$, $v_{ij}$ and $u_{ij}$ for the jth RF input, the jth CF input and the jth output unit, respectively, and we assume that the weights connecting each pair of output units to each other are symmetric ($u_{ij} = u_{ji}$) and also that these units are not self-connected ($u_{ii} = 0$). We now define the integrated fields in relation to the ith output.

$$S_i(r) = \sum_{j \in \partial i(r)} w_{ij} R_j - w_{i0} \tag{1}$$

is the random variable representing the integrated receptive field input to the ith output.

$$S_i(c) = \sum_{j \in \partial i(c)} v_{ij} C_j - v_{i0} \tag{2}$$

is the random variable representing the integrated contextual field input to the ith output.

$$S_i(x) = \sum_{j \in \partial i(x)} u_{ij} X_j \tag{3}$$

is the random variable representing the within-processor integrated field input to the ith output.

We take a conditional approach to the modelling of the outputs $\mathbf{X}$ given the RF and CF inputs $\mathbf{R}$ and $\mathbf{C}$ and assume that $\mathbf{X}$ follows a multivariate binary probability model (Besag, 1974), given the realised values of the RF and CF inputs, with probability mass function

$$\Pr(\mathbf{X} = \mathbf{x} | \mathbf{R} = \mathbf{r}, \mathbf{C} = \mathbf{c}) = \frac{1}{Z(\mathbf{a}, \mathbf{u})} \exp\left( \sum_{i=1}^{p} a_i x_i + \frac{1}{2} \sum_{i=1}^{p} \sum_{j \in \partial i(x)} u_{ij} x_i x_j \right), \tag{4}$$

where $Z(\mathbf{a}, \mathbf{u})$ is the normalisation constant (i.e. not a function of $\mathbf{x}$) required to ensure that the probabilities sum to unity. We assume that the output binary levels are 0 and

1; in applications in which bipolar units are used it is then necessary to adjust output probabilities using the mapping $p \mapsto 2p-1$. The terms $\{a_i\}$ and $\{u_{ij}\}$ are parameters and in general may be functions of $\mathbf{r}$ and $\mathbf{c}$. In this article we shall shortly define the $\{a_i\}$ as a function of the RF and CF inputs but will take the $\{u_{ij}\}$ to be independent of these inputs, although there are other possibilities.

Equation (4) defines a regression model in two distinct senses. Firstly, via the terms $\{a_i\}$ which may be taken to be general nonlinear functions of the RF and CF inputs, it is a nonlinear regression of the outputs as a function of all of the inputs. Secondly, when written in conditional form in terms of the distribution of the ith output given its neighbours, as in equation (5) below, it expresses, for given inputs, an auto-regression for each output unit in terms of the other output units in its neighbourhood. It is an example of what has become known as a generalized linear model and the marginal probability model for the outputs is a generalized linear mixed model, the linearity referring to the parameters $\{a_i\}$ and $\{u_{ij}\}$ (McCullagh & Nelder, 1989; Clayton, 1996).

This model, in unconditional form, was used by Ising (1925) in statistical mechanics and developed as a model for spatial data by Besag (1974); however the general methodology works also with more general graph structures (Geman & Geman, 1984). The Ising model has been employed in a different way in neural networks by Hopfield (1982) and many others within the area of recurrent networks. The formulation described in the above model has the advantage of interfacing a feed-forward network between layers with a recurrent network structure within a layer in a single coherent probabilistic framework. Not only that, but it is possible to connect the multiple output local processors themselves in a multi-layered and multi-stream network structure in a probabilistically coherent manner.

It is natural to consider the local conditional distributions for each output unit given its RF, CF and WP inputs. Under the assumed restrictions on the WP connection weights, described above, the Hammersley-Clifford theorem (Besag, 1974) ensures that working locally with the conditional distributions is equivalent to assuming a coherent global model for all the output units. If the WP units are fully connected, then it is unnecessary to invoke this thereom as the local distributions then automatically have the form given in equation (5). The local conditional distribution for the ith output, given the values of its RF, CF and WP inputs, is Bernoulli with success probability

$$\theta_i \equiv \Pr(X_i = 1 | \mathbf{R}_{\partial i} = \mathbf{r}_{\partial i}, \mathbf{C}_{\partial i} = \mathbf{c}_{\partial i}, \mathbf{X}_{\partial i} = \mathbf{x}_{\partial i}) = 1/\left(1 + \exp(-A_i)\right). \qquad (5)$$

Here $A_i = a_i + s_i(x)$, where $a_i$ may be taken to be any differentiable function of the integrated RF and CF fields and $s_i(x)$ is defined in equation (3). It then remains to specify the activation function for each output unit.

The activation function at the ith output unit is now a function of three integrated fields – how should they be combined ? It will be assumed that the general function decomposes into the following sum, although there are other possibilities.

$$A_i = A(s_i(r), s_i(c)) + s_i(x) \equiv a_i + s_i(x). \qquad (6)$$

The learning rules will be derived for the general case, however. This form in which the output integrated field is bound additively to a general function of the integrated

RF and CF fields makes the specification of the activation function consistent with the multivariate probability model for the outputs defined above in equation (4). In this paper, the function A is chosen so that the CF input can modulate the response of the ith output to the RF input and in the next section we define the particular form used in this paper.

## 3 Global Objective Function and Learning Rules

We now consider a global objective function based on the joint distribution of all outputs, RF inputs and CF inputs. In the case of multivariate outputs, we consider the general version of the objective function introduced by Kay & Phillips (1994) which is

$$F = I(\mathbf{X};\mathbf{R};\mathbf{C}) + \phi_1 I(\mathbf{X};\mathbf{R}|\mathbf{C}) + \phi_2 I(\mathbf{X};\mathbf{C}|\mathbf{R}) + \phi_3 H(\mathbf{X}|\mathbf{R},\mathbf{C}), \tag{7}$$

where, for example, $H(\mathbf{X}|\mathbf{R},\mathbf{C})$ denotes the conditional entropy in the distribution of $\mathbf{X}$ given $\mathbf{R}$ and $\mathbf{C}$, and $I(\mathbf{X};\mathbf{R}|\mathbf{C})$ is the conditional mutual information shared between $\mathbf{X}$ and $\mathbf{R}$ given $\mathbf{C}$. The three-way mutual information is defined by

$$I(\mathbf{X};\mathbf{R};\mathbf{C}) = I(\mathbf{X};\mathbf{R}) - I(\mathbf{X};\mathbf{R}|\mathbf{C}) \tag{8}$$

For the purposes of modelling it is more convenient to employ the simple rules of information theory (Hamming, 1980) and express this as

$$F = H(\mathbf{X}) - \psi_1 H(\mathbf{X}|\mathbf{R}) - \psi_2 H(\mathbf{X}|\mathbf{C}) - \psi_3 H(\mathbf{X}|\mathbf{R},\mathbf{C}), \tag{9}$$

where $\psi_1 = 1 - \phi_2, \psi_2 = 1 - \phi_1$ and $\psi_3 = \phi_1 + \phi_2 - \phi_3 - 1$. For further details, see Kay (1994) and Kay & Phillips (1994). We now present expressions for the entropic terms of equation (9) and also develop the learning rules based on the derivatives of F with respect to the weights. In the sequel, for simplicity, we shall use, in an abuse of notation, $\langle\cdots\rangle_{\mathbf{x}}$ to denote the expectation computed with respect to the probability distribution of $\mathbf{X}$. So, for example,

$$\langle p(\mathbf{x}|\mathbf{r},\mathbf{c})\rangle_{\mathbf{r}|\mathbf{c}} = \int p(\mathbf{x}|\mathbf{r},\mathbf{c})p(\mathbf{r}|\mathbf{c})d\mathbf{r} \tag{10}$$

denotes the average of the conditional probability density function, $p(\mathbf{x}|\mathbf{r},\mathbf{c})$, given by equation (4), taken with respect to the conditional distribution of $\mathbf{R}$ given that $\mathbf{C} = \mathbf{c}$; hence equation (10) gives $p(\mathbf{x}|\mathbf{c})$.

Firstly, a simple calculation based on equation (4) shows that the conditional entropy H($\mathbf{X}|\mathbf{R}$,$\mathbf{C}$) may be written as

$$H(\mathbf{X}|\mathbf{R},\mathbf{C}) = \left\langle \log Z(\mathbf{a},\mathbf{u}) - \sum_{i=1}^{p} a_i e_i - \frac{1}{2}\sum_{i=1}^{p}\sum_{j\in\partial i(x)} u_{ij} f_{ij} \right\rangle_{\mathbf{r},\mathbf{c}} \tag{11}$$

where

$$e_i = E(X_i|\mathbf{r},\mathbf{c}), \tag{12}$$

and

$$f_{ij} = E(X_i X_j|\mathbf{r}, \mathbf{c}) \tag{13}$$

denote, respectively, the conditional means of $X_i$ and $X_i X_j$ given that $\mathbf{R} = \mathbf{r}$ and $\mathbf{C} = \mathbf{c}$ $(j \in \partial i(x); i = 1, \ldots, p)$.

The other entropy terms are given by:

$$\begin{aligned} H(\mathbf{X}) &= -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} &\quad (14)\\ H(\mathbf{X}|\mathbf{R}) &= -\left\langle \int p(\mathbf{x}|\mathbf{r}) \log p(\mathbf{x}|\mathbf{r}) d\mathbf{x} \right\rangle_{\mathbf{r}} &\quad (15)\\ H(\mathbf{X}|\mathbf{C}) &= -\left\langle \int p(\mathbf{x}|\mathbf{c}) \log p(\mathbf{x}|\mathbf{c}) d\mathbf{x} \right\rangle_{\mathbf{c}} . &\quad (16) \end{aligned}$$

The required marginal and conditional distributions are given by the following equations.

$$p(\mathbf{x}) = \langle p(\mathbf{x}|\mathbf{r}, \mathbf{c}) \rangle_{\mathbf{r},\mathbf{c}} \tag{17}$$

$$p(\mathbf{x}|\mathbf{r}) = \langle p(\mathbf{x}|\mathbf{r}, \mathbf{c}) \rangle_{\mathbf{c}|\mathbf{r}} \tag{18}$$

$$p(\mathbf{x}|\mathbf{c}) = \langle p(\mathbf{x}|\mathbf{r}, \mathbf{c}) \rangle_{\mathbf{r}|\mathbf{c}} \tag{19}$$

Now we require the derivatives of the entropic terms in equations (11) and (14) – (16) with respect to the weights. Let $\alpha$ be a generic weight, that is, one of the $\{w_{ij}\}$ , $\{v_{ij}\}$ or $\{u_{ij}\}$. After some algebra we find that these derivatives are as follows.

$$\frac{\partial H(\mathbf{X}|\mathbf{R}, \mathbf{C})}{\partial \alpha} = \left\langle -\sum_{i=1}^{p} a_i \frac{\partial e_i}{\partial \alpha} - \frac{1}{2} \sum_{i=1}^{p} \sum_{j \in \partial i(x)} u_{ij} \frac{\partial f_{ij}}{\partial \alpha} \right\rangle_{\mathbf{r},\mathbf{c}} \tag{20}$$

where

$$\frac{\partial e_i}{\partial \alpha} = \sum_{k=1}^{p} \frac{\partial a_k}{\partial \alpha} \mathrm{cov}(X_i, X_k|\mathbf{r}, \mathbf{c}) \tag{21}$$

and

$$\frac{\partial f_{ij}}{\partial \alpha} = \sum_{k=1}^{p} \frac{\partial a_k}{\partial \alpha} \mathrm{cov}(X_i X_j, X_k|\mathbf{r}, \mathbf{c}). \tag{22}$$

Here cov(Y,Z|**r,c**) denotes the conditional covariance between the random variables Y and Z given that $\mathbf{R} = \mathbf{r}$ and $\mathbf{C} = \mathbf{c}$. Now

$$\frac{\partial H(\mathbf{X})}{\partial \alpha} = -\left\langle \sum_{k=1}^{p} \frac{\partial a_k}{\partial \alpha} \mathrm{cov}(X_k, \log p(\mathbf{X})|\mathbf{r}, \mathbf{c}) \right\rangle_{\mathbf{r},\mathbf{c}} \tag{23}$$

$$\frac{\partial H(\mathbf{X}|\mathbf{R})}{\partial \alpha} = -\left\langle \sum_{k=1}^{p} \frac{\partial a_k}{\partial \alpha} \mathrm{cov}(X_k, \log p(\mathbf{X}|\mathbf{r})|\mathbf{r},\mathbf{c}) \right\rangle_{\mathbf{r},\mathbf{c}} \tag{24}$$

$$\frac{\partial H(\mathbf{X}|\mathbf{C})}{\partial \alpha} = -\left\langle \sum_{i=1}^{p} \frac{\partial a_k}{\partial \alpha} \mathrm{cov}(X_k, \log p(\mathbf{X}|\mathbf{c})|\mathbf{r},\mathbf{c}) \right\rangle_{\mathbf{r},\mathbf{c}} \tag{25}$$

Collecting together terms from equations (9), (20) and (23) –(25) gives the derivatives of F as

$$\frac{\partial F}{\partial \alpha} = \left\langle \sum_{i=1}^{p} (\psi_3 a_i \frac{\partial e_i}{\partial \alpha} - \bar{O}_i) + \frac{1}{2} \sum_{i=1}^{p} \sum_{j \in \partial i(x)} u_{ij} \frac{\partial f_{ij}}{\partial \alpha} \right\rangle_{\mathbf{r},\mathbf{c}} \tag{26}$$

where

$$\begin{aligned} \bar{O}_i &= \frac{\partial a_i}{\partial \alpha}(\mathrm{cov}(X_i, \log p(\mathbf{X})|\mathbf{r},\mathbf{c}) - \psi_1 \mathrm{cov}(X_i, \log p(\mathbf{X}|\mathbf{r})|\mathbf{r},\mathbf{c}) \\ &\quad -\psi_2 \mathrm{cov}(X_i, \log p(\mathbf{X}|\mathbf{c})|\mathbf{r},\mathbf{c})). \end{aligned} \tag{27}$$

It remains to specify the partial derivatives of $a_i$ with respect to the weights. Simple calculation gives that

$$\frac{\partial a_i}{\partial w_{ij}} = \frac{\partial A_i}{\partial s_i(r)} r_j \tag{28}$$

$$\frac{\partial a_i}{\partial v_{ij}} = \frac{\partial A_i}{\partial s_i(c)} c_j \tag{29}$$

$$\frac{\partial a_i}{\partial u_{ij}} = \frac{\partial A_i}{\partial s_i(x)} x_j \tag{30}$$

In the learning rules given above, we employ for each output unit the following activation function, as specified in equation (6).

$$A_i = a_i + s_i(x), \tag{31}$$

where

$$a_i = \frac{1}{2} s_i(r)(1 + \exp(2 s_i(r) s_i(c)) \tag{32}$$

is a member of the class of activation functions introduced by Kay & Phillips (1994). The partial derivatives are given by

$$\frac{\partial A_i}{\partial s_i(r)} = \frac{1}{2} + (\frac{1}{2} + s_i(r) s_i(c)) \exp(2 s_i(r) s_i(c)) \tag{33}$$

$$\frac{\partial A_i}{\partial s_i(c)} = s_i(r)^2 \exp(2 s_i(r) s_i(c)) \tag{34}$$

$$\frac{\partial A_i}{\partial s_i(x)} = 1. \tag{35}$$

The use of gradient ascent learning, with learning-rate parameter $\eta$, leads to weight changes given by $\Delta\alpha \propto \eta\frac{\partial F}{\partial\alpha}$.

The methodology developed here for the general class of objective functions defined in equation (7) contains some important special cases in this multivariate setting. Taking $\phi_1 = 1$ and $\phi_2 = \phi_3 = 0$, and cutting the contextual connections, gives the Infomax objective function $I(\mathbf{X};\mathbf{R})$. Taking $\phi_1 = \phi_2 = \phi_3 = 0$ gives the Coherent Infomax objective function $I(\mathbf{X};\mathbf{R};\mathbf{C})$, while setting $\phi_1 = 1-\epsilon, \phi_2 = \epsilon$ and $\phi_3 = 0$ gives an information-theoretic analogue of the multivariate version of the objective function used by Schmidhuber & Prelinger (1993).

The learning rules, particularly the terms $\{\bar{O}_i\}$, are quite complicated; they are global at the level of the processor in that the weights connected to all the units are considered simultaneously. The computation of some of the average terms in equation (27) is particularly cumbersome as they involve entire conditional distributions of $\mathbf{X}$. It is possible, however, to develop approximations of the zeroth-order, but we don't present the details here. Despite this, these rules are computable when the number of units in the processor is limited, but approximations will be required if the number of output units is large; see Section 6. Primarily out of a wish to develop learning rules that are more biologically plausible, we now turn our attention to local approximations of the global objective function considered above, and we shall see that these lead to local learning rules at the level of the units within the processor.

# 4 Local Objective Functions with Local Learning Rules

In a processor with multiple output units, it is natural to consider processing in a local manner with each output unit using the information available to it from its RF, CF and WP neighbourhood connections. This suggests that we might focus on the joint distribution of each output and its RF and CF inputs, conditionally on its WP inputs. It then seems natural to consider the conditional three-way mutual information that is shared mutually by the ith output and its RF and CF inputs but not shared with its WP output units, defined by

$$I(X_i;\mathbf{R}_{\partial i};\mathbf{C}_{\partial i}|\mathbf{X}_{\partial i}) = I(X_i;\mathbf{R}_{\partial i}|\mathbf{X}_{\partial i}) - I(X_i;\mathbf{R}_{\partial i}|\mathbf{C}_{\partial i},\mathbf{X}_{\partial i}) \tag{36}$$

It is possible, in general, to decompose the global three-way mutual information as follows.

$$I(\mathbf{X};\mathbf{R};\mathbf{C}) = I(X_i;\mathbf{R};\mathbf{C}|\mathbf{X}_{-i}) + I(\mathbf{X}_{-i};\mathbf{R};\mathbf{C}) \tag{37}$$

which, given the connection structure, may be expressed as

$$I(\mathbf{X};\mathbf{R};\mathbf{C}) = I(X_i;\mathbf{R}_{\partial i};\mathbf{C}_{\partial i}|\mathbf{X}_{\partial i}) + I(\mathbf{X}_{-i};\mathbf{R};\mathbf{C}). \tag{38}$$

This decomposition may be repeated recursively and is of particular relevance when the output units represent some directional structure such as a time series or a predecessor/successor hierarchy; then the well-known general factorization of joint probability into a product of a marginal and conditional distributions allows the general three-way mutual information to be written as a sum of local conditional three-way mutual information terms. Such simplicity is not possible here, although the first-step decomposition, given in equation (38), shows that the conditional three-way information is a part of the global three-way information in a well-defined sense. Note that, under these local objective functions defined in equation (36), each output unit within the multiple-output local processor is attempting to transmit the information shared with its local RF and CF that is not being transmitted by the other units within the multiple-output processor. Hence one would expect these within-processor units to transmit slightly different aspects of the available information; but note that the outputs of these within-processor units can be correlated and hence this is not enforcing a winner-take-all scenario. There are other possible ways of defining local objective functions, but they will be discussed elsewhere.

The same conditioning idea may be applied to the other components of information within the objective function F and this leads to the specification of a local objective function for the ith output unit defined as follows

$$\begin{aligned} F_i &= I(X_i; \mathbf{R}_{\partial i}; \mathbf{C}_{\partial i}|\mathbf{X}_{\partial i}) + \phi_1 I(X_i; \mathbf{R}_{\partial i}|\mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) \\ &\quad + \phi_2 I(X_i; \mathbf{C}_{\partial i}|\mathbf{R}_{\partial i}, \mathbf{X}_{\partial i}) + \phi_3 H(X_i|\mathbf{R}_{\partial i}, \mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}), \end{aligned} \tag{39}$$

and, given our conditional approach to the modelling, we express $\{F_i\}$ in the more useful form

$$\begin{aligned} F_i &= H(X_i|\mathbf{X}_{\partial i}) - \psi_1 H(X_i|\mathbf{R}_{\partial i}, \mathbf{X}_{\partial i}) \\ &\quad - \psi_2 H(X_i|\mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) - \psi_3 H(X_i|\mathbf{R}_{\partial i}, \mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}). \end{aligned} \tag{40}$$

This means that we envisage each output unit within a local processor working to maximise $F_i$ and, because of the the fact that mutually distinct sets of weights connect into each of the outputs, this is equivalent to maximising the sum of the $F_i s$. We view this sum as a locally-based approximation to the global objective function F defined in equation (7). In the extreme case where the outputs are conditionally independent this sum is equivalent to F. Obviously, the approximation will be better the smaller are the sizes of the output neighbourhood sets relative to the number of outputs.

In our experiments we shall consider two particular forms for the $\{F_i\}$. If we take $\phi_1 = \phi_2 = \phi_3 = 0$, then equation (40) gives the conditional three-way mutual information shared amongst the output of the ith unit and its RF and CF inputs given the outputs of the neighbours of the ith output unit. Maximization of this objective function means that each unit in each processor is being adapted to maximize the coherent information it shares with its RF and CF inputs conditional on the information flowing from its neighbouring outputs within the processor to which it belongs. This objective function generalizes the coherent infomax goal proposed by Kay & Phillips (1994,1996) and we term it the **local conditional coherent infomax** criterion. Another possibility considered in our experiments is the choice $\phi_1 = 1$, and $\phi_2 = \phi_3 = 0$. In this case equation (40) gives the conditional mutual information shared between the output of the ith unit and its RF inputs given the information flowing from its neighbouring outputs within the processor. This generalizes the infomax objective function proposed by Linsker (1988) and we term it the **local conditional infomax** criterion.

We now provide formulae for the local entropic terms and the components of local information for the ith output unit. The conditional distribution of $X_i$ given its RF, CF and WP inputs is a Bernoulli distribution with success probability given by equation (5). It is easy to show that the corresponding entropy term is

$$H(X_i|\mathbf{R}_{\partial i}, \mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) = \langle \theta_i \log \theta_i + (1-\theta_i)\log(1-\theta_i) \rangle_{\mathbf{r}_{\partial i}, \mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} . \tag{41}$$

It is also easy to show that the conditional distribution of $X_i$ given its RF and WP inputs is also Bernoulli with success probability given by

$$E^{(i)}_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}} = \langle \theta_i \rangle_{\mathbf{c}_{\partial i}|\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}} \tag{42}$$

Hence a similar argument gives the following entropy term.

$$H(X_i|\mathbf{R}_{\partial i}, \mathbf{X}_{\partial i}) = \left\langle E^{(i)}_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}} \log E^{(i)}_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}} + (1 - E^{(i)}_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}}) \log(1 - E^{(i)}_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}}) \right\rangle_{\mathbf{r}_{\partial i}, \mathbf{x}_{\partial i}} \tag{43}$$

Similarly the conditional distribution of $X_i$ given its CF and WP inputs is Bernoulli with probability defined by

$$E^{(i)}_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} = \langle \theta_i \rangle_{\mathbf{r}_{\partial i}|\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} , \tag{44}$$

and the corresponding entropy is

$$H(X_i|\mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) = \left\langle E^{(i)}_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} \log E^{(i)}_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} + (1 - E^{(i)}_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}}) \log(1 - E^{(i)}_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}}) \right\rangle_{\mathbf{c}_{\partial i}, \mathbf{x}_{\partial i}} . \tag{45}$$

Also the conditional distribution of $X_i$ given its WP inputs is Bernoulli with probability

$$E^{(i)}_{\mathbf{x}_{\partial i}} = \langle \theta_i \rangle_{\mathbf{r}_{\partial i}, \mathbf{c}_{\partial i}|\mathbf{x}_{\partial i}} , \tag{46}$$

and entropy term

$$H(X_i|\mathbf{X}_{\partial i}) = \left\langle E^{(i)}_{\mathbf{x}_{\partial i}} \log E^{(i)}_{\mathbf{x}_{\partial i}} + (1 - E^{(i)}_{\mathbf{x}_{\partial i}}) \log(1 - E^{(i)}_{\mathbf{x}_{\partial i}}) \right\rangle_{\mathbf{x}_{\partial i}} . \tag{47}$$

It follows that the components of local information at the ith output unit, given in terms of equations (41), (43), (45) and (47), are as follows.

$$\begin{aligned} I(X_i; \mathbf{R}_{\partial i}; \mathbf{C}_{\partial i}|\mathbf{X}_{\partial i}) &= (47) - (45) - (43) + (41) && (48)\\ I(X_i; \mathbf{R}_{\partial i}|\mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) &= (45) - (41) && (49)\\ I(X_i; \mathbf{C}_{\partial i}|\mathbf{R}_{\partial i}, \mathbf{X}_{\partial i}) &= (43) - (41) && (50)\\ H(X_i|\mathbf{R}_{\partial i}, \mathbf{C}_{\partial i}, \mathbf{X}_{\partial i}) &= (41) && (51) \end{aligned}$$

We now present the learning rules. For all the weights they have the same general structure as those introduced by Kay and Phillips(1994). We now give the gradient ascent learning rules in relation to the ith output unit.

- **Receptive Field Connection Weights**

$$\frac{\partial F_i}{w_{is}} = \left\langle (\psi_3 A_i - \bar{O}_i)\theta_i(1-\theta_i)\frac{\partial A_i}{\partial s_i(r)} R_s \right\rangle_{\mathbf{r}_{\partial i},\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}}, \tag{52}$$

  for each RF input $s$ which connects into the ith output unit. For RF inputs $s$ that do not connect into the ith output, $\frac{\partial F_i}{\partial w_{is}} = 0$.

- **Contextual Field Connection Weights**

$$\frac{\partial F_i}{v_{is}} = \left\langle (\psi_3 A_i - \bar{O}_i)\theta_i(1-\theta_i)\frac{\partial A_i}{\partial s_i(c)} C_s \right\rangle_{\mathbf{r}_{\partial i},\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}}, \tag{53}$$

  for each CF input $s$ which connects into the ith output. For CF inputs $s$ which do not connect to the ith output, $\frac{\partial F_i}{\partial v_{is}} = 0$.

- **Within Processor Connection Weights**

$$\frac{\partial F_i}{\partial u_{is}} = \left\langle (\psi_3 A_i - \bar{O}_i)\theta_i(1-\theta_i)\frac{\partial A_i}{\partial s_i(x)} X_s \right\rangle_{\mathbf{r}_{\partial i},\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}}, \tag{54}$$

  for each output unit $s$ which connects into the ith output. For output units $s$ which do not connect to the ith output, $\frac{\partial F_i}{\partial u_{is}} = 0$.

The partial derivatives of $A_i$ are defined above in equations (33) – (35). Note that these learning rules are **local** and this results from the decision to separately maximise the local objective functions $\{F_i\}$ (or equivalently to maximise the sum of the $\{F_i\}$). They involve another level of averaging taken over the neighbouring output units. The dynamic average for the ith output unit is

$$\bar{O}_i = \log\frac{E^{(i)}_{\mathbf{x}_{\partial i}}}{(1-E^{(i)}_{\mathbf{x}_{\partial i}})} - \psi_1 \log\frac{E^{(i)}_{\mathbf{r}_{\partial i},\mathbf{x}_{\partial i}}}{(1-E^{(i)}_{\mathbf{r}_{\partial i},\mathbf{x}_{\partial i}})} - \psi_2 \log\frac{E^{(i)}_{\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}}}{(1-E^{(i)}_{\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}})}. \tag{55}$$

The dynamic averages are more complicated than in the single-unit output case and their calculation involves storing the average probability at the ith output unit for each pattern of the other outputs that connect into the ith output unit, for the combination of each of the neighbouring output and RF input patterns and for the combination of each of the neighbouring output and CF input patterns. The computational implications resulting from the need to store these conditional averages means that it is feasible to apply these learning rules only to processors of limited size. However, as noted above, connecting together many such local processors would enable a complex architecture to be constructed in which the computation required in each local processor is manageable. This issue is discussed further in section 6 and solutions are proposed.

# 5 Computational Experiments

To test the learning rules derived above for multi-unit processors with contextual input we have simulated many networks with multiple streams of processing linked by contextual connections. Here we first report three groups of experiments that use just two streams of processing, and which test specific aspects of the proposed approach. We then report simulations of networks with 25 streams of processing and up to 5,228 connections.

The experiments described in section Section 5.1 were designed to see whether the algorithm does indeed ensure that different units within multi-unit processors learn to respond to different aspects of the input. The experiments in section Section 5.2 study the way in which the information transmitted by these multi-unit processors is coded at the level of the individual units within processors. Section 5.3 demonstrates that these learning rules do indeed enable multi-unit processors to develop RF selectivities that are sensitive to just those input variables that are coherently related to the context in which they occur. Finally, in section 5.4, we study the effects of context on learning and processing in networks with many streams of processing and a pattern of connectivity within and between layers that is broadly analogous to that in cerebral cortex.

All of these experiments used the learning rules derived in Section 4. In all of the experiments the following procedures were adopted. The initial weight values were randomly chosen from the uniform distribution in the range [-0.0001, 0.0001]. The learning rate was set to 0.01 and the momentum parameter set to 0.99. The RF, CF and WP weights were repeatedly updated according to equations (52)–(54) after the presentation of all the patterns in the training set (an epoch). The WP weights were made symmetric by averaging their values after every weight–update.

Two modes of learning were employed in the experiments, namely, the Coherent Infomax mode and the Infomax mode. In Coherent Infomax mode the parameters in the objective function defined by equation (40) were set so that the local conditional coherent infomax criterion was maximised for each unit within each processor. On the other hand, in Infomax mode, the parameters were set so that the local conditional infomax criterion was maximized for each unit within each processor. These parameter values are defined for both modes in Section 4.

## 5.1 Two streams, each with two features and two units

The primary requirement of the extension of the approach from single-unit to multi-unit processors is to ensure that different units within each processor learn to respond to different aspects of the input. This experiment was designed to see whether the methodology developed in Section 4 meets this requirement. We simulated a network with two streams, each composed of a set of four receptors and a two-unit processor (Figure 1). Each unit received connections from all the receptors in its own stream (Receptive Field connections), from all the units in the other processor (Contextual Field connections), and from the other unit within the same processor (Within-Processor connection). The WP connection weights were constrained to be symmetrical. The output of each unit in the network was synchronously updated by passing the integrated inputs through the activation function three times, previous explorations having shown that this is sufficient stabilize the output values.

The input patterns were composed of "horizontal and vertical edges" whose sign and orientation were correlated across streams. The structure of the input can be easily understood if we visualize the input pattern to each stream arranged on a 2×2 square matrix whose entries $r_{ij}$ can take the bipolar values -1 or +1. A horizontal edge $E_H$ is defined by the difference between the sums of the two row components, whereas a vertical edge $E_V$ is defined by the difference between the sums of the two column components; the signs of the edges are given by the signs of the differences

$$E_H = \left( \sum_j r_{1j} - \sum_j r_{2j} \right), E_V = \left( \sum_i r_{i1} - \sum_i r_{i2} \right) \tag{56}$$

The four input patterns in which there is neither a vertical nor a horizontal edge were never presented. Each of the remaining twelve patterns were presented to each stream with equal probability, but with the orientation and sign of the edges being perfectly correlated across streams. The probability of each individual receptor taking the value unity during training was therefore 0.5. The learning was performed in Coherent Infomax mode.

### 5.1.1 Results

Each processor learned to detect the sign of vertical and horizontal edges by allocating one unit to each edge type, as shown by the final structure of the RF weights after training (Figure 2). The CF connections between units responding to the same type of edge were strengthened, and the CF connections between units responding to different types of edge remained around 0.0 (hence, they are not plotted in Figure 2). The WP weights displayed temporary activity during the strengthening of the appropriate contextual weights; the change in sign coincided with the sudden halt and reversal of a very short increment in the contextual weights connecting units that started to signal different edge types (not shown here). Once the pattern of contextual strength was established, the WP weights stopped their activity. These results were very stable and were replicated in several simulations starting from different initial random weights, the main variations from run to run being in which specific unit signalled each edge type and in the allocation of positive and negative outputs to the sign of each edge. Faster convergence could be achieved by increasing the learning rate which was not a crucial parameter in these simulations.

### 5.1.2 Discussion

The above results show that different units within these multi-unit processors learn to respond to different features of the information that is correlated across streams. WP weights are strengthened in order to decorrelate the activity of the units within the same processor. The shift of sign here is due to the attempt to decorrelate the "residual" activity of a unit with -say- a vertical receptive field when a horizontal edge is presented. In this case the unit will initially display a low activity whose sign might be correlated with the activity in the other unit. When the RF integrated input is small, the WP integrated input can actually change the sign of the unit output and so the WP weights are effective in decorrelating the two units. As a result of this change in sign, the RF weights will be modified in such a way that the output is still correlated, but this time in the other direction. Hence, the WP weights are strengthened in the opposite direction (here we have a zero-crossing), and these dynamics continue until the units are perfectly decorrelated.

The information transmitted by the WP connections plays a major role in the development of different selectivities for different units because, in experiments where the WP connections were removed, all the units within a processor learned to detect the edge of a single edge type (which type depending upon the initial weight settings), and in addition all the CF connections were strengthened. This was the same result obtained, as expected, when each processor had a single unit because a single bipolar unit is not sufficient to transmit the sign of two edge types. The WP connections *in this experiment*, in which the number of features to be transmitted is equal to the number of units in a processor, therefore provide a form of competitive learning that results directly from the goal of maximizing the transmission of local conditional mutual information as described in Section 4. Prior to these simulations we had no reason to suppose that the WP connections would converge to such low values that they play little role in the short-term dynamics after learning is complete. The broader significance of this finding, if any, therefore arises as an issue for further research.

The choice of the architecture employed for this experiment deserves some comment. In principle, there are no constraints on the connectivity pattern that one might choose to implement. However, it should be noted that if the processors of two different streams shared a consistent portion of the input receptors, they might simply learn to signal the information in the shared area. Since one of the reasons for employing contextual connections is discovery of information which is not locally available, we usually keep the RF inputs to different processors separate, but had successful results when several single-unit processors had partially overlapped RF inputs distributed along a continous surface. In this case each processor shared some of the receptors of other processors which, in turn, did not share those same receptors. Similarly, CF connections could be set in any arbitrary fashion. As a matter of fact, contextual connections could have been omittted in this experiment because input stimuli were perfectly correlated across streams. However, one might want to start by connecting each unit to all the units in all the other processors and let the algorithm choose which connections to strengthen during training. The automatic configuration of the CF connections is an interesting property because one does not require prior knowledge on the structure of the input patterns in order to choose a suitable architecture and thus does not run into the danger of constraining the system in unforeseen ways. As a general rule, WP connections serve the purpose of forcing apart the response of individual units; hence, they serve quite a different purpose from CF connections.

## 5.2 Two streams, each with several units and several features all fully correlated across streams

This experiment studied the way in which information is distributed across the different output units when each procesor is composed of several units and there are several input features perfectly correlated across streams. The architecture employed here was similar to that described for the previous experiment, but here each stream had nine receptors and each processor had from three to six units all fully inter-connected by symmetric WP weights. The CF connectivity was all-to-all between all output units of the two streams. (Figure 3).

The input patterns for each stream were lines presented at four different orientations: horizontal, vertical, and two opposite diagonals. The lines were presented at the centre of a 3×3 square matrix. Each of the four lines could take a negative or a positive value: a negative line was composed of -1's against a background of +1's, and a positive one was just the opposite (Figure 3, bottom). Each of the eight patterns was presented with equal probability to each stream, and the orientation and sign of each line were perfectly correlated across streams. Several simulations with networks of three, four, five, and six units per processor were run on this training set.

When the processors have three output units, one would expect that the input features would be found by simply maximising information transmission within streams by using the multivariate Infomax approach (derived in Section 3 and 4 and here implemented using local conditional infomax objective functions); but would the appropriate CF connections be learned in such a case, or would they remain at zero bacause they are not necessary to discover the relevant features? Furthermore, what use, if any, would be made of the additional output units when more than the required three are employed?

### 5.2.1 Results

We first report the results of the experiments with the three-unit processors and then the experiments involving processors with more than three units. The local conditional coherent infomax criterion was quickly maximized by all three units in both streams of the network (Figure 3, top). Learning took place at approximately the same time for all six units and the results were not sensitive to the value of the learning-rate parameter. As a result of this learning, each processor developed a distributed representation providing a unique code for each of the input patterns (Figure 4, centre). The sign of the line was signalled by reversing the sign of all the output units. The difference between the specific codes developed by the processors in the two streams depended on the initial random weight settings. The strengthening of the connections followed a similar time course (not plotted here) to that described for the previous experiment. In contrast to the previous experiment, however, the final structure of the RF weights of each unit did not mirror the appearances of the original input lines (Figure 4, bottom). With eight equi-probable input patterns and only three binary output units each unit was bound to take part in all the codes, so instead of allocating a particular unit to a particular whole input pattern that could occur in either of two signs, each unit was used as part of a distributed code by signalling a specific 'micro-structure' that occurred in each of the input patterns, these micro-structures being indicated by the RF weight structures.

The CFs learned the appropriate cross-stream predictions, even though they were not necessary to discover the relevant within-stream features. Since each unit took part in the representation of all the input patterns all the CF connections were strengthened. In order to make sure that the CF weights were properly developed so as to predict the information transmitted by the RF weights, we checked that the integrated CF input to each unit matched in sign the integrated RF input for each pattern. This was always true, except for a few cases where a weak RF integrated input of opposite sign was compensated by a relatively strong WP integrated input to that unit The final WP weights tended to approach zero, except for the few situations where they compensated the slight mismatch between RF and CF integrated inputs.

Similar results were obtained in experiments with more than three units per processor. However, since three units were sufficient for transmitting the entire information contained in eight patterns, one might wonder what use the algorithm made of the additional units. One way of assessing whether redundant units are used effectively consists in measuring the loss of information transmitted after systematic pruning of the trained units. After training a four-unit processor network to convergence, we systematically pruned out single units, tested the network on all the input patterns, and measured the loss of information. It turned out that pruning out any one unit reduced the information transmitted. This meant that all units were used in the distributed code developed for discriminating the eight patterns. We repeated the same procedure with a five-unit processor network. This time we found two units that, if pruned out (not together), did not affect the amount of information transmitted. Although these units were “redundant”, they were in fact taking part in the distributed representation and appropriately reversed their sign when the whole code for a particular orientation was reversed to signal a change in sign of the input line. Finally, the same procedure was applied to a six-unit processor network. Here we found that we could prune out *any* one unit at a time without losing information. There was no information loss also when some combinations of two units were eliminated. However, the distributed code developed by the network was not a trivial duplication of the code developed in the three-unit case described above. When the number of units per processor was larger than the minimum number necessary to transmit the coherent input information, not all units could be fully decorrelated and, therefore, the WP weights remained strong.

All these results were replicated in simulations starting from different initial random weights. As before, the main variations were in the nature of the particular ‘micro-features’ that were used to transmit the relevant information. These varied from stream to stream and from run to run. Despite these variations at the level of the detailed selectivities of individual units, however, the system always converged to a solution in which the relevant information was transmitted.

### 5.2.2 Discussion

When several units and several input features were allowed a distributed representation was obtained. Each unit specialized in detecting specific micro-features of the input patterns. Exactly what micro-features were detected in order to generate a unique code for each pattern depended not only on the properties of the input patterns, but also on the number of units per processor and on the initial random weight settings. Most of the time it was difficult to find a "label" that would describe these micro-features in common language. Other times it was easier. For example, the RF weights developed by units in the second stream of the three-unit experiment described above were easy to describe (Figure 4, bottom, last row). Unit *u2* detected the sign of the line without regard to the orientation. Unit *u3* signalled whether the line had a diagonal or a straight orientation. Finally, unit *u1* discriminated between types of diagonals and types of straight lines. On the other hand, it is less obvious how to give a simple description of the micro-features detected by units of the first stream (Figure 4, bottom, first row) in terms of our intuitive descriptions of these patterns.

The system made good use of the units exceeding the minimum number required to transmit the information in the input. The distributed code was spread across the entire population of units and, eventually, resulted in a very robust solution. In the experiments with six-unit processors, the network developed a non-trivial distributed code whose individual components all participated in the representation of the patterns, but at the same time could be eliminated without affecting the information transmitted by the processor as a whole. This property is of value because it means that good use will be made of whatever transmission capacity is available, and so it is not necessary to know prior to learning how many units will be needed. When four or more units were used per processor it would have been possible to transmit the relevant information by allocating one unit to each line orientation, as in the very simple conditions of the preceding experiment. We have never found the algorithm to develop this form of local coding in the slightly more complex conditions of either this or the following experiments.

## 5.3 Two streams, each with several units and several features only some of which are correlated across streams

A central goal of processors trained in Coherent Infomax mode is to discover just the information that is coherently related to activity in the streams from which they receive contextual input. This experiment was therefore designed to confirm that the algorithm derived for multi-unit processors in Section 4 does indeed provide this capability. Performance of a network trained in Coherent Infomax mode was compared with an identical network trained in Infomax mode. The same training parameters were used in both modes as described above and defined in Section 4.

The input patterns described in the previous section were presented with equal probability to each stream, but with the sign and orientation of only the horizontal and vertical lines being correlated across streams (Figure 5, top). Half of the information within streams was therefore correlated across streams, and half was not. The network architecture was the same as that described for the previous experiment with three units per processor, which is sufficient to transmit all the information within each stream.

### 5.3.1 Results

When the network was trained in Coherent Infomax mode, it developed a unique code for each of the correlated patterns (horizontal and vertical bars), but did not discriminate between the remaining patterns (i.e. the diagonal bars). The final value of the objective function measuring the amount of the local conditional three-way mutual information transmitted by each unit was approximately half the maximum because only half of the patterns were correlated across streams. This value was reached within about 300 epochs of training (Figure 5, centre). When the network was trained in Infomax mode, it developed a unique code for each of the four orientations within 200 epochs (Figure 5, bottom), and this code was appropriately reversed with the sign of the line. This distributed code thus had the same properties of that described in the previous experiment.

The final structure of the RF weights reflected the two different representations developed by the network when trained in the two modes. When the network was trained in Coherent Infomax mode all the RF weights corresponding to the corners of the imaginary square matrix took the same value (Figure 5, centre right). Hence, these units could discriminate only between horizontal and vertical lines. Instead, when the network was trained in Infomax mode, the RF weights displayed a structure similar to that already described for the experiments in the previous section (Figure 5, bottom right), and discriminated between all four line orientations.

### 5.3.2 Discussion

These results show that when trained in Coherent Infomax mode multi-unit processors discover just those variables that are correlated across streams. When trained in Infomax mode, however, they discover whatever features are most informative within streams, up to the limit of their transmission capacity. The operations performed by the latter are therefore wholly determined by the information in the input, whereas the former can use contextual knowledge to select just that information that is related to the context within they operate. Recognition based upon the features discovered by the Coherent Infomax approach can therefore generalize across the irrelevant input variables, because it will then be based upon descriptors that are not sensitive to those variables (Phillips & Singer, 1996).

As in all previous experiments, the fine structure of the RF weights after training depended on the number of units and on the initial random settings. However, since the input patterns employed here were simple and low-dimensional, the number of possible variations was not large. Hence, it was possible to select a processor of a network trained in Infomax mode whose RF weight structure (Figure 5, bottom right) was similar to that developed by a network trained in Coherent Infomax mode on a fully correlated data set (Figure 4, bottom, last row)

## 5.4 Many streams, each with several units and several features

This experiment studies the effects of contextual integration on the dynamics of learning and processing in a network with many streams of processing and an architecture that is broadly analogous to that of cerebral cortex. The networks used were composed of 25 streams arranged as a 5×5 matrix (Figure 6, top). Each stream had a single processor composed of a number of units fully interlinked by symmetric WP connections. Each unit in the network received RF input from all the receptors in its own stream. Each stream had nine receptors which could be visualized as a 3×3 matrix. Nearest-neighbour processors were fully interlinked via CF connections.

The input patterns were long lines with four different orientations (horizontal, vertical, and 45° diagonals) and two opposite signs shifted across the whole input surface (the 5×5 stream surface was considered as a patch of a larger input surface) Some input patterns are shown in the bottom half of Figure 6. Each line was composed of an alignment of identical small lines (as in the experiments described in the previous two sections) and these were shown at all possible positions of the input array. Altogether, there were 56 input lines presented across the 15×15 receptor array, half positive and half negative: five horizontal, five vertical, nine diagonal in one direction and nine diagonal in the other direction. Hence, the RF input patterns to each processor were identical to those employed in the experiments presented in Section 5.2: eight lines with different orientation and sign each presented with equal probability.

We ran different versions of the experiment, varying the number of units per processor (three and four ), the strength of the training patterns (four strengths were used: {+1, -1}, {+.75, -.75}, {+.5, -.5}, and {+.25, -.25}), and the values applied to the "background", i.e. to the receptors in the streams where no line was presented (all 0's, all +1's, all -1's, or equally probable +1's and -1's). After learning, the networks were tested on a variety of input data characterized by a weak signal and different types of noise to test the effects of learned contextual input on generalization to patterns corrupted by noise.

### 5.4.1 Results

All the processors in the network developed a distributed representation of the local eight patterns with the same properties already described in Section 5.2.

The learning dynamics were little affected by the number of units per processor, the strength of the input, or the values applied to the background streams, so we describe in detail just the results obtained with the three-unit processors. The local conditional mutual information terms reached their maximum value of about 0.1 after 500 epochs (Figure 8); this maximum value was given by the fact that for each processor ($a$) only 8 out of 56 patterns featured a line, and ($b$) when a line was presented, only a subset of the other processors linked by CF connections signalled the same input feature (those aligned within the input line orientation), because the others belonged to the background. After learning, the effect of the CF connections was assessed by testing the network with four different forms of a horizontal line presented ot the central row of the 5×5 matrix of streams (Figure 7), and for each of them we measured the output probability of the processor in the middle of that line, that is, the processor at the centre of the whole array. The input to the remaining twenty streams was set to zero. The horizontal lines presented to the five streams in the central row were designed to test the effect of no context, supportive context, or opposing context on the response to a non-saturating RF input. When the RF input signal was strong (magnitude 1.0) the output of all the units in the central processor was strong, both with and without input from the CF connections. When the RF signal was weak (magnitude 0.125) and there was no input from the CF connections (Figure 7a), the output of the units in the central processor was weakened too (Figure 7c); however, when the signals from the CF connections were allowed, a single iteration was sufficient to restore the maximum output strength (Figure 7b). On the contrary, when the neighbouring processors were presented with RF evidence for a line with equally weak strength but opposite sign, the output of all the units in the central processor was further dampened in a single iteration (Figure 7d).

The network was then tested with noisy inputs. Two types of noise were used. One consisted in adding a random number from a uniform distribution in the range [-0.25, +0.25] to already weak RF inputs (signal strength 0.25); the other consisted in randomly flipping the sign of 20% of the RF weak inputs (signal strength 0.25). A horizontal line was presented on the central row of streams and corrupted according to either noise condition. When the CF connections were cut the RF weights alone were sufficient to provide the correct pattern of output (the sign of the units was correct for all the processors when compared to the output or normal strong RF input plotted in the top of Figure 7), but the output strength was weak in most cases (Figure 7, left column). When the CF connections were intact, the contextual input boosted the unit outputs in the correct direction in a single iteration (Figure 7, right column).

### 5.4.2 Discussion

Although these were much larger nets than in the previous experiments the number of epochs required for learning was not greatly increased, being approximately 700 epochs (Figure 8), as compared with the 200 epochs required for the two-stream network with only 8 patterns fully correlated across streams.

The CF connections learned to correctly predict the RF features signalled by the units to which they projected. The post-training tests with weak RF inputs showed that contextual signals from neighbouring processors which were presented with the same RF inputs boosted the unit outputs in the right direction, whereas contextual signals from the same processors presented with opposite RF inputs further dampened the unit outputs (Figure 7). The role of the learned CF connection weights was also shown by tests with noisy weak inputs (Figure 9). In this case the network displayed double generalization. The RF weights generalized to the noisy input by getting the unit signs correct and the CF weights generalized to the varying output strengths by boosting all the unit activations in the right direction. Although the relevant RF variables in the patterns used during training could have been discovered using the Infomax approach, the CF connections learned using the Coherent Infomax approach provides the network with the additional ability of exploiting contextual information when the primary RF evidence is uncertain. There is a great deal of evidence from cognitive psychology and psychophysics that local context plays a major role in disambiguating ambiguous inputs in human information processing (Phillips & Singer, 1996), and this is a computational strategy that is obviously likely to be of value in technological applications. Learned CFs therefore merit further study as a mechanism for implementing this strategy. The choice of nearest-neighbour CF connectivity in the above architectures reflected our prior knowledge of the input patterns to be presented to the net. Since the input lines were always presented on collinear streams (horizontal, vertical, and diagonals), there ought to be no correlation between the outputs of non-collinear processors; consequently, as shown in subsection 5.1, the corresponding hypothetical CF connection weights ought not to be strengthened. In order to test this hypothesis, we added a set of "spurious" CF connections to three random processors (Figure 10, top); we then trained the network presenting each input line on a background of equally probable +1's and -1's and recorded the sum of the (absolute value of) weight magnitudes for a set of CF connections from a nearest-neighbour processor and for the spurious CF connections. The comparisons confirmed our hypothesis: CF connections from nearest-neighbour processors were strengthened (Figure 10, left column), whereas the spurious CF connections remained close to zero (Figure 10, right column). When the same experiment was repeated with a background of all 1's (or all -1's), the spurious CF connections were initially slightly strengthened, but this did not affect the network dynamics because the spurious CF signal was the same for all patterns and for all units. Detailed prior knowledge of the CF connectivity appropriate for the inputs to be received is therefore not necessary to this approach.

# 6 Conclusions and Further Work

We have shown how the use of associations between streams to guide feature discovery within streams, and to enhance short-term processing through the use of contextual predictions, can be extended to deal with the case in which the processors have multivariate binary outputs. The general approach was presented in Section 3 including the definition of global objective functions and the derivation of the concomitant learning rules. This provides algorithms for a general class of information-theoretic objective functions, including Infomax and Coherent Infomax. The learning rules are quite complicated, but applicable as long as the number of outputs in the processor is not too large. The usefulness of these algorithms remains to be tested, however. The learning rules obtained in this global approach are not local at the level of the individual units within a processor and so, in Section 4, we turned our attention to local approximations to the global objective function. We have defined local objective functions at the level of the individual units and derived the corresponding learning rules. These rules have been obtained for a general class of objective functions and, in particular, provide as a by-product a local version of Infomax for multi-unit binary processors. The approach outlined in section 4 in really quite general and it is readily seen that the multivariate binary processors can be combined in many ways in multi-stream, multi-layer networks, even though only two-layer nets have been used in our experiments. These experiments demonstrate the feasibility of the methodology.

Some useful conclusions about the sensitivities of individual units emerged during the course of the experiments and may be summarised as follows:

- Different units within processors become sensitive to different aspects of the input, and this seems to be due, at least in part, to the influence of the within-processor connections;

- Different units within a processor do not normally discover factorial codes, that is, sensitivity to features that are uncorrelated. They can do so in the special case where the number of units available is only just sufficient to transmit the relevant information. Otherwise, all available units are used in a form of coarse-coding with different units detecting units that are correlated. Their outputs will therefore usually be correlated and thus contain some redundancy. The removal of redundancy is therefore not an emergent property of this algorithm, but robustness to damage is!;

- In very simple cases the features look like the patterns presented, but usually they do not, and are instead micro-features that are sometimes easily interpreted in relation to the input but usually not. This suggests that when trying to interpret the sensitivities of individual units in either biological or technological networks we should not assume that those sensitivities will be well matched to our intuitions concerning the featural composition of the stimuli presented;

- There is considerable variability in single-unit sensitivities across streams and across runs of the algorithm from different start states and different exact sequences of input – even given the same overall population. Local processors learn to transmit the relevant information, however, so performance is more constant at the population encoding level than at the local encoding level. This means that what matters for the algorithm is the total information to be transmitted – not the particular way in which this is divided up for encoding at the local unit level. Applied to neurobiology, this predicts greater variability of coding at the level of single units than at the level of the population codes. These results therefore encourage the growing movement within neurobiology towards the study of population codes.

An central aspect of our approach is the use of context to guide both learning and processing. The main conclusions concerning the role of the CFs can be summarized as follows:

- The information that is provided by the CFs enables local multi-unit processors to discover that region in the information space of their RF input that is correlated across streams;

- They are learned appropriately even when they are not necessary to discover the relevant features within streams;

- They enhance the short-term processing by providing contextual predictions that improve feature detection when the RF inputs are weak or noisy. They could therefore play a useful role in improving generalization;

- The correct patterns of CF connection strengths are learned even though the associative relations are between complex self-organising population codes. This shows that it is computationally feasible to learn the statistical associations between distributed self-organizing population codes, so we cannot assume that the discovery of associations between input variables implies that those variables must have been given either a single-unit code or an arbitrary explicit name.

The above results provide encouragement for the study of multi-unit processors which use context to guide learning and processing, and they raise various issues that merit further research:

- Simple input data were used to elucidate some of the essential properties of the algorithm and the to test the feasibility of the generalization to multi-unit processors. Further work is now needed to study the application of the approach derived in section three and four to real-world technological problems;

- The implementation described here assumes binary variables, but the general approaches outlined in sections 3 and 4 have been developed also for a very general class of Gibbs distributions (Kay, 1996a) and it is intended to implement them using different probability models;

- In the experiments described in this paper the learning rules were applied in batch mode. It is possible to implement them using on-line learning and this will be used in future work. On-line learning has been applied in experiments with single-unit processors (Kay, 1995; Kay and Phillips, 1996) and given the similarity of the learning rules derived in Section 4 to those in the single-unit case, it is straightforward to develop on-line versions;

- The computational complexity of the approach deserves some comment and further study. It is required to store conditional averages, given in equations (42), (44) and (46), in order to compute the dynamic averages in the learning rules. This is more complicated in the multivariate case because time averages require to be kept at each unit for each distinct RF, CF and WP pattern. Clearly this limits the scalability of the capacity of each multivariate processor. Of course, within a multi-stream, multi-stage network one can envisage that even a limited representation capacity for each processor would still enable the network as a whole to perform useful tasks. Hence the scalability issue could be circumvented to some extent by the adoption of a different, more segmented, architecture. In real world applications, however, it has been shown (Kay, 1994; Kay and Phillips, 1996) that the dependence of these averages on the RF inputs can vanish and then it is only the dependence on the CF and WP patterns that remains. In addition, possible solutions to the problem have been devised (Kay, 1994), and generalized to the multivariate case (Kay, 1996b), but not yet tested. The conditional averages depend only on the integrated RF, CF and WP fields. When the dimensions of these fields are large it is expected, due to a version of the central limit theorem, that they may be viewed as continous random variables following a Gaussian distribution. Using this approach, approximations to these conditional expectations have been calculated; these involve storing only nine parameters, irrespective of the dimensionality of the RF and CF input-spaces; hence the storage cost per processor is nine times the number of output units. Furthermore, these nine parameters may be updated on-line using recursive formulae. An alternative approach, based on simple non-parametric estimation of the conditional averages as functions the integrated fields, may also be computed on-line and involves a storage cost that depends only on the resolution employed in the nonparametric estimators and the number of units in the processor, but not on the dimensionality of the RF and CF input-spaces; hence, the storage cost is linear in the number of output units in the processor;

- The form of the general class of learning rules derived formally from the information-theoretic objectives in Section 4 (see also Kay & Phillips, 1994, 1996) closely resembles the structure of the BCM learning rule which has already been shown to be both computationally powerful and to possess functionality similar to that found in biological forms of learning. See Intrator & Cooper (1995) for a review of the BCM rule, and Kirkwood et al. (1996) for further detailed evidence on its relevance to synaptic plasticity in mammalian visual cortex. Both rules have a threshold that depends on a dynamically-computed average of prior activity, the main difference between them being that the Coherent Infomax approach takes contextual inputs into account when computing this average whereas the BCM rule does not. In both rules the synaptic strengths from active units are adapted depending on whether or not the post-synaptic activity exceeds this threshold. Given the computational power of these learning rules, and their close approximation to biological forms of learning, the similarities and differences between them merit much further study.

## References

Atick, J. J. & Redlich, A. N. (1993) Convergent algorithm for sensory receptive field development. *Neural Computation*, **5**, 45–60.
Barlow, H. B. (1961) Possible principles underlying the transformations of sensory messages. In Rosenblith, W. A. (Ed.), *Sensory communication* (pp. 217-234). Boston: MIT Press.
Barlow, H. B. (1989) Unsupervised learning. *Neural Computation*, **1**, 295-311.
Becker, S. (1992) *An information-theoretic unsupervised learning algorithm for neural networks.* PhD Thesis. University of Toronto.
Becker, S. (1996) Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, **7**, 7-31.
Becker, S. & Hinton, G. E. (1992) Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, **355**, 161-163.
Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society: Series B*, **36**, 192-236.
Clayton, D. G. (1996) Generalized linear mixed models. In W. R. Gilks, S. Richardson & D. J. Spiegelhalter (Eds.), *Markov chain monte carlo in practice* (pp. 275-301). London. Chapman & Hall.
De Sa, V. (1994) *Unsupervised classification learning from cross-modal environmnetal structure.* PhD Thesis. University of Rochester.
Der, R. & Smyth, D. (1996) Local online learning of coherent information. *Neural Networks.* (accepted conditional on minor revision).
Fox, K., Sato, H. & Daw, N. (1990) The effect of varying stimulus intensity on NMDA-receptor activity in cat visual cortex. *Journal of Neurophysiology*, **64**, 1413-1428.
Geman, S. & Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721-741.
Gifi, A. (1990) *Nonlinear multivariate analysis.* New York: Wiley.
Gilbert, C. D. (1992) Horizontal integration and cortical dynamics. *Neuron*, **9**, 1-13.
Hamming, R. W. (1980) *Coding and information theory.* Englewood Cliffs: Prentice-Hall.
Hirsch, J. A. & Gilbert, C. D. (1991) Synaptic physiology of horizontal connections in cat visual cortex. *Journal of Neuroscience*, **11**, 1800-1809.
Hirsch, J. A. & Gilbert, C. D. (1993) Long-term changes in synaptic strength along specific intrinsic pathways in the cat visual cortex. *Journal of Physiology*, **461**, 247-262.
Hopfield, J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of*

*Sciences*, **79**, 2553-2558.

Hotelling, H. (1936) Relation between two sets of variates. *Biometrika*, **28**,321-377.

Intrator, N. & Cooper, L. N. (1995) BCM theory of visual cortical plasticity. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 153-156). Boston: MIT Press.

Ising, E. (1925) Beitrag zur theorie des ferromagnetismus. *Zeitschrift Physik*, **31**, 253.

Kapadia, M. K., Ito, M, Gilbert, C. D. & Westheimer, G. (1995) Improvements in visual sensitivity by changes in local context; Parallel studies observers and V1 of alert monkeys. *Neuron*, **15**, 843-856.

Kay, J. (1992) Feature discovery under contextual supervision using mutual information. *Proc. Int. Joint Conf. on Neural Networks (Baltimore MD)*, **Book 4**, 79-84.

Kay, J. (1994) Information-theoretic neural networks for unsupervised learning: Mathematical and statistical considerations. *Technical Report* . Scottish Agricultural Statistics Service.

Kay, J. (1995) Information-theoretic neural networks for unsupervised learning. Talk given at the ICMS Workshop on Statistics and Neural Networks (Edinburgh, Scotland).

Kay, J. (1996a) Contextual guidance of unsupervised learning: Formulation for general multivariate processors. In preparation.

Kay, J. (1996b) Contextual guidance of unsupervised learning: Some approximations for large networks. In preparation.

Kay, J. & Phillips, W. A. (1994) Activation functions, computational goals and learning rules for local processors with contextual guidance. *Technical Report*, **CCCN-15**. Centre for Cognitive and Computational Science, University of Stirling.

Kay, J. & Phillips, W. A. (1996) Activation functions, computational goals and learning rules for local processors with contextual guidance. Submitted to *Neural Computation.*

Kirkwood, A., Rioult, M. G. & Bear, M. F. (1996) Experience-dependent modifications of synaptic plasticity in visual cortex. *Nature*, **381**, 526-528.

Linsker, R. (1988) Self-organization in a perceptual network. *Computer* **21**, 105-117.

Löwel, S. & Singer, W. (1992) Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, **255**, 209-212

McCullagh, P. & Nelder, J. A. (1989) *Generalized linear models.* London: Chapman & Hall.

Massaro, D. W. & Cohen, M. M. (1991) Integration versus interactive activation: The joint influence of stimulus and context in perception. *Cognitive Psychology*, **23**, 558-614.

Phillips, W. A., Kay, J. & Smyth, D. (1995) The discovery of structure by multi-stream networks of local processors with contextual guidance. *Network: Computation in Neural Systems*, **6**, 225-246.

Phillips, W. A. & Singer, W. (1996) In search of common foundations for cortical computation. *Behavioural and Brain Sciences*, In Press.

Polat, U. & Sagi, D. (1993) Lateral interactions between spatial channels: Suppression and facilitation revealed by lateral masking experiments. *Vision Research*, **33**, 993-999.

Polat, U. & Sagi, D. (1994) The architecture of perceptual spatial interactions. *Vision Research*, **34**, 73-78.

Redlich, A. N. (1993) Redundancy reduction as a strategy for unsupervised learning. *Neural Computation*, **5**, 289-304.

Schmidhuber, J. & Prelinger, D. (1993) Discovering predictable classifications. *Neural Computation*, **5**, 625-635.

Singer, W. (1990) Search for coherence: A basic principle of cortical self-organization. *Concepts in Neuroscience*, **1**, 1-26.

Singer, W. (1993) Synchronization of cortical activity and its putative role in information processing and learning. *Annual Review of Physiology*, **55**, 349-374.

Stone, J. V. (1995) Learning perceptually salient visual parameters through spatio-temporal smoothness constraints. *Neural Computation.* In Press.

Taylor, J. G. & Plumbley, M. D. (1993) Information theory and neural networks. In J. G. Taylor (Ed.) *Mathematical approaches to neural networks* (pp. 307-340). North Holland: Elsevier.

## Nomenclature

$X_i$ random variable representing the output at the ith unit

$R_j$ random variable representing the jth RF input

$C_j$ random variable representing the jth CF input

$\mathbf{X}$ random vector representing the ouputs of a processor

$\mathbf{R}$ random vector representing the RF inputs

$\mathbf{C}$ random vector representing the CF inputs

$\mathbf{x}$ a vector of realised values of the random vector $\mathbf{X}$

$\mathbf{r}$ a vector of realised values of the random vector $\mathbf{R}$

$\mathbf{c}$ a vector of realised values of the random vector $\mathbf{C}$

$\mathbf{X}_{\partial i}$ random vector representing the outputs that are connected to the ith output unit

$\mathbf{R}_{\partial i}$ random vector representing the RF inputs that are connected to the ith output unit

$\mathbf{C}_{\partial i}$ random vector representing the CF inputs that are connected to the ith output unit

$\mathbf{x}_{\partial i}$ a vector of realised values of the random vector $\mathbf{X}_{\partial i}$

$\mathbf{r}_{\partial i}$ a vector of realised values of the random vector $\mathbf{R}_{\partial i}$

$\mathbf{c}_{\partial i}$ a vector of realised values of the random vector $\mathbf{C}_{\partial i}$

$\mathbf{X}_{-i}$ all the components of the random vector $\mathbf{X}$ excluding the ith one

$\partial i(x)$ the set of indices of the output units that are connected to the ith output unit

$\partial i(r)$ the set of indices of the RF inputs that are connected to the ith output unit

$\partial i(c)$ the set of indices of the CF inputs that are connected to the ith output unit

$w_{ij}$ the weight on the connection between the jth RF input and the ith output unit

$w_{i0}$ the RF bias for the RF inputs connected to the ith output unit

$v_{ij}$ the weight on the connection between the jth CF input and the ith output unit

$v_{i0}$ the CF bias for the CF units connected to the ith output unit

$u_{ij}$ the weight on the connection between the ith and jth output units

$S_i(x)$ random variable representing the integrated WP field input to the ith output unit

$S_i(r)$ random variable representing the integrated RF field input to the ith output unit

$S_i(c)$ random variable representing the integrated CF field input to the ith output unit

$s_i(x)$ a realised value of $S_i(x)$

$s_i(r)$ a realised value of $S_i(r)$

$s_i(c)$ a realised value of $S_i(c)$

$\Pr(\mathbf{X} = \mathbf{x} | \mathbf{R} = \mathbf{r}, \mathbf{C} = \mathbf{c})$ the probability mass function of the Ising model

$Z(\mathbf{a}, \mathbf{u})$ the normalisation factor in the Ising model

$a_i$ a parameter in the Ising model associated with the ith output; it is a differentiable function of the RF and CF inputs and the weights on their connections into the ith output unit

$\theta_i$ the output probability at the ith output unit; it is a function of all inputs and weights connecting into the ith output

$A_i$ the activation function at the ith output unit

$A(s_i(r), s_i(c))$ the part of the activation function at the ith output that specifies how the integrated RF and CF fields are to be combined

$H(\mathbf{X})$ the Shannon entropy associated with the distribution of the output units

$H(\mathbf{X}|\mathbf{R})$ the Shannon entropy associated with the conditional distribution of the output units given the RF inputs

$H(\mathbf{X}|\mathbf{C})$ the Shannon entropy associated with the conditional distribution of the outputs given the CF inputs

$H(\mathbf{X}|\mathbf{R}, \mathbf{C})$ the Shannon entropy associated with the conditional distribution of the outputs given both the RF and the CF inputs

$p(\mathbf{x}|\mathbf{r}, \mathbf{c})$ the probability that the outputs are $\mathbf{x}$ given that the RF inputs are $\mathbf{r}$ and the CF inputs are $\mathbf{c}$

$p(\mathbf{x}|\mathbf{r})$ the probibility that the outputs are $\mathbf{x}$ given that the RF inputs are $\mathbf{r}$

$p(\mathbf{x}|\mathbf{c})$ the probability that the outputs are $\mathbf{x}$ given that the CF inputs are $\mathbf{c}$

$p(\mathbf{x})$ the marginal probability that the outputs are $\mathbf{x}$

$I(\mathbf{X}; \mathbf{R}; \mathbf{C})$ the three-way mutual information shared amongst the random vectors $\mathbf{X}, \mathbf{R}$ and $\mathbf{C}$ (Coherent Infomax Objective Function)

$I(\mathbf{X}; \mathbf{R}|\mathbf{C})$ the conditional mutual information shared between the random vectors $\mathbf{X}$ and $\mathbf{R}$ given $\mathbf{C}$

$I(\mathbf{X};\mathbf{C}|\mathbf{R})$ the conditional mutual information shared between the random vectors **X** and **C** given **R**

$I(\mathbf{X};\mathbf{R})$ the mutual information shared between the random vectors **X** and **R** (Infomax Objective Function)

$\langle \ldots \rangle_{\mathbf{X}}$ the operation of taking the theoretical mean of ... with respect to the distribution of **X**

$E(\ldots|\mathbf{r},\mathbf{c})$ the theoretical mean of ... taken with respect to the conditional distribution of **X** given **R** and **C**

$E^{(i)}_{\mathbf{x}_{\partial i}}$ the theoretical mean of the output probability at the ith output unit, taken with respect to the conditional distribution of neighbouring RF and CF inputs given the neighbouring WP outputs

$E^{(i)}_{\mathbf{c}_{\partial i},\mathbf{x}_{\partial i}}$ the theoretical mean of the output probability at the ith output unit, taken with respect to the conditional distribution of neighbouring RF inputs given both the neighbouring CF inputs and WP outputs

$E^{(i)}_{\mathbf{r}_{\partial i},\mathbf{x}_{\partial i}}$ the theoretical mean of the output probability at the ith output unit, taken with respect to the conditional distribution of neighbouring CF inputs given both the neighbouring RF inputs and WP outputs

# Figure Legends

### Figure 1

Architecture of the two-stream network of multi-unit processors used for edge detection. A stream is defined as a vertical structure composed of an input field and one (or more) layers of processors. Each processor is composed of two (or more) units which receive input from the same receptors. Each unit has Receptive Field (RF) connections $\{w_{ij}\}$ from all the receptors within its own stream, Contextual Field (CF) connections $\{w_{ij}\}$ from all units of processors in other streams, and Within-Processor connections $\{u_{ij}\}$ from the units within the same processor.

### Figure 2

Development of weight strengths for a network discovering the sign of horizontal and vertical edges correlated across streams. The final RF weights for each unit are arranged as a square matrix of intensity values to facilitate readability: white is strongly excitatory and black is strongly inhibitory. The asymmetric layout of excitation and inhibition indicates to which orientation each unit responds. RF weights for units belonging to the same processor are grouped vertically. The plots between units of the same processor display the strength of WP weights during training (one plot for each processor because the weights are symmetric). The plots between units of different processors display the strength of CF connections which are increased during learning (the arrows indicate the direction of signal flow). The CF connections between units of different processors that respond to different edge types remain around zero and are not displayed.

### Figure 3

**Top**: Network architecture; the number of units per processor was varied in different experiments (3, 4, 5, and 6).

**Bottom**: Pairs of input patterns presented to the network during training. Each input pattern to a stream is arranged as a square matrix to visualize it as an oriented line with a negative or positive sign; negative lines are formed by -1's (gray squares) on a +1's (white squares) background, and positive lines are formed of +1's on a -1's background. The sign and orientation of each line is correlated across streams.

**Figure 4**

Discovery of lines whose sign and orientation are correlated across streams; the data refer to a network of three-unit processors.

**Top**: Average local conditional coherent infomax of all six units in the network (error bars indicate standard deviations).

**Centre**: Distributed representation of each pattern after learning; the output of each unit is indicated by its sign. The orientation of the input lines are graphically displayed on the right (only in the negative version) and their sign is spelled beside the unit outputs.

**Bottom**: Weights of the RF connections for each unit in the network (white is strongly positive and black is strongly negative) arranged as a square matrix.

**Figure 5**

Comparison between the behaviours of the model when trained in Coherent Infomax mode and in Infomax mode on partially correlated data. The networks have three units per processor; only the data of one processor for each training mode is plotted.

**Top**: Input patterns (only line orientation displayed) are presented with equal probability to each stream, but only horizontal and vertical lines are correlated in orientation and sign across streams.

**Centre**: Network trained in Coherent Infomax mode develops specific distributed codes for correlated input patterns and a single code for all remaining patterns. During training, the local conditional coherent infomax criterion reaches only half the maximum because only half of the input patterns are correlated across streams. The RF weights after training have equal value on the four corners showing that the units do not discriminate between diagonal lines.

**Bottom**: Network trained in Infomax mode develops different distributed representations for all the input patterns, as indicated by the full maximization of the local conditional two-way mutual information. The micro-features discovered by the RF weights are sufficient to discriminate all the eight patterns; those plotted here are qualitatively similar to those plotted in Figure 4.

**Figure 6**

Several streams arranged as a lattice.

**Top**: Architecture and magnification of a single stream with 4 units per processor. The thick lines between processors show the CF connectivity between nearest-neighbour processors. Each unit receives RF input from all the receptors in its own stream, CF connections from all the units in the nearest-neighbour processors, and WP connections from all the units within its own processor.

**Bottom**: Some examples of the 56 input lines used during learning; each line is composed of an alignment of small lines with identical sign and orientation.

This makes the RF input to each individual processor identical to that used in the previous experiments.

**Figure 7**

Assessing the role of the CF connections to the central processor after learning. Each graph plots the bipolar transformation of the output probability of the three units when a long horizontal line is presented.

**a**: Input signal strength is 1.0 (the same output values can be obtained with and without CF connections).

**b**: Input signal strength is 0.125 for the receptors of all the streams in the central row carrying evidence for a positive horizontal line; output value is stable after a single iteration.

**c**: Weak horizontal line presented only to the central stream; input to other streams is set to zero (the same effect can be obtained by allowing input to the streams and cutting the CF connections).

**d**: Central processor is presented with weak horizontal line and other streams are presented with weak horizontal line of opposite sign.

**Figure 8**

Learning in a network of three-unit processors.

**Left**: Maximization of the local conditional three-way mutual information for each unit in the network.

**Right**: The same measures separately plotted for the units of three types of processors: a processor on the corner of the lattice (linked by CF connections to 3 processors), a processor on the side (linked by CF connections to 5 processors), and a processor in the centre (linked by CF connections to 8 processors).

**Figure 9**

Testing the network with noisy inputs.

**Top**: A horizontal line is presented to all the processors in the central row (Top right). The pattern of response of all the units when the RF input is strong (magnitude 1.0) and clear is plotted on the top left of the figure.

**Centre**: The RF input is weak (magnitude 0.25) and is corrupted by adding uniform noise in the range [-0.25, +0.25]. The output of all the units are plotted without the contribution from the CF connections (left) and with the contribution of the CF connections after a single iteration (right).

**Bottom**: The RF input is weak (magnitude 0.25) and is corrupted by flipping the sign of 20% random RF inputs. As above, the output of all the units are plotted without the contribution from the CF connections (left) and with the contribution of the CF connections after a single iteration (right).

**Figure 10**

Addition of "spurious" CF connections between non-collinear processors in a network of three-unit processors.

**Top**: Sets of spurious CF connections are indicated by thick arrows.

**Left column**: Each graph plots the cumulative strength of CF connections from a nearest-neighbour processor for one of the three processors during learning.

**Right column**: Plots of cumulative strength of spurious connections for the same processor during learning. The cumulative strength is the sum of the absolute values of all nine incoming CF connections for each processor.

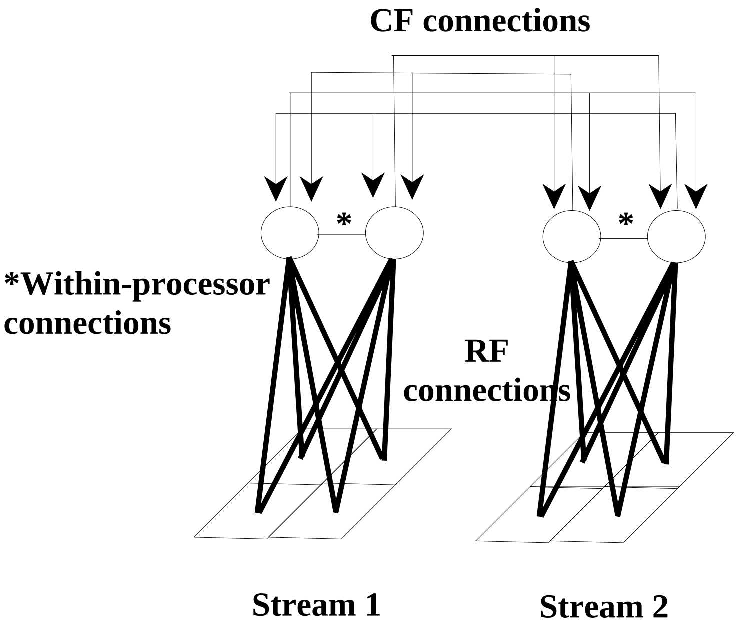

CF connections
*
*
*Within-processor connections
RF connections
Stream 1
Stream 2

Figure 1:

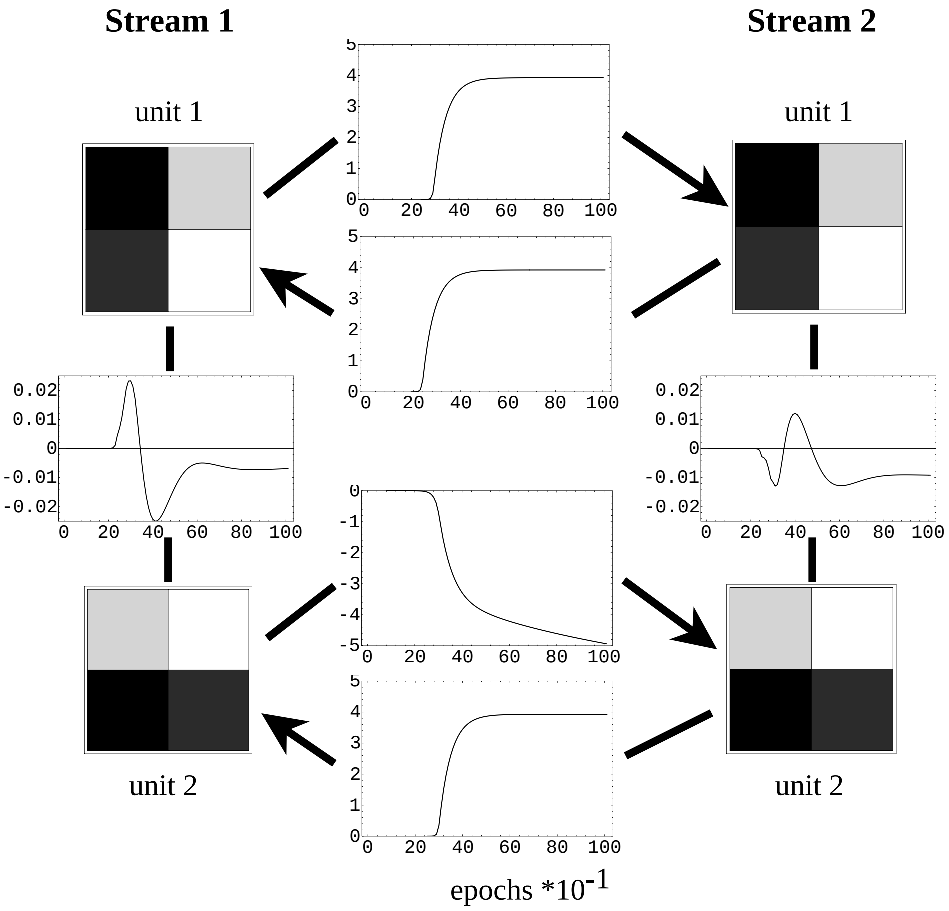

Stream 1
Stream 2
unit 1
unit 1
unit 2
unit 2
0
1
2
3
4
5
0.02
0.01
0
-0.01
-0.02
-1
-2
-3
-4
-5
0
20
40
60
80
100
epochs *10$^{-1}$

Figure 2:

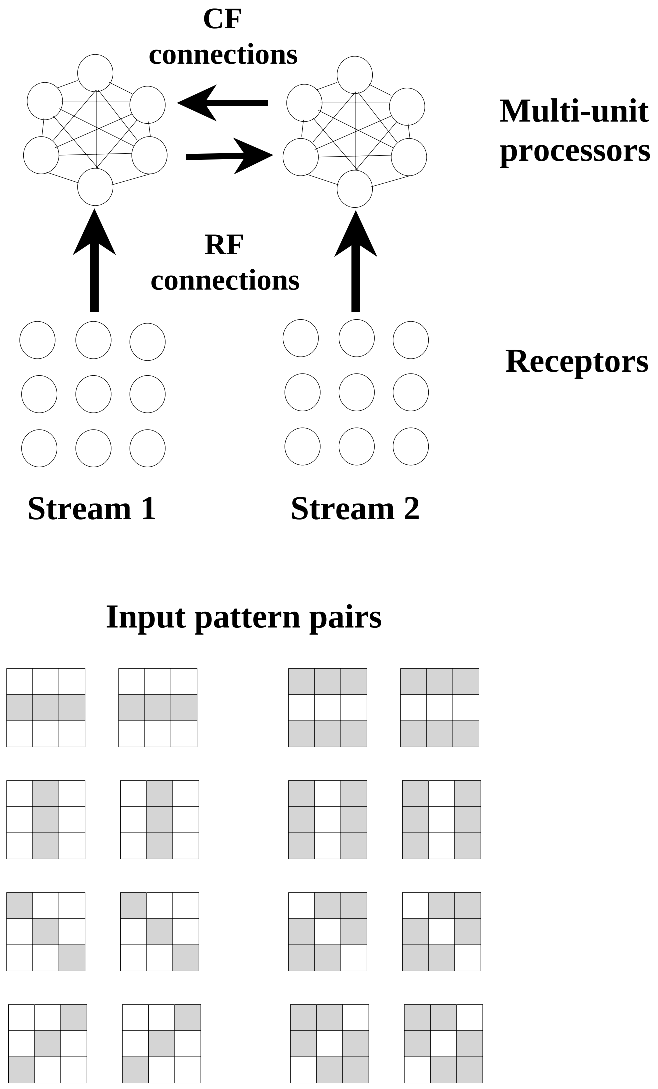

CF
connections
Multi-unit
processors
RF
connections
Receptors
Stream 1
Stream 2
Input pattern pairs

Figure 3:

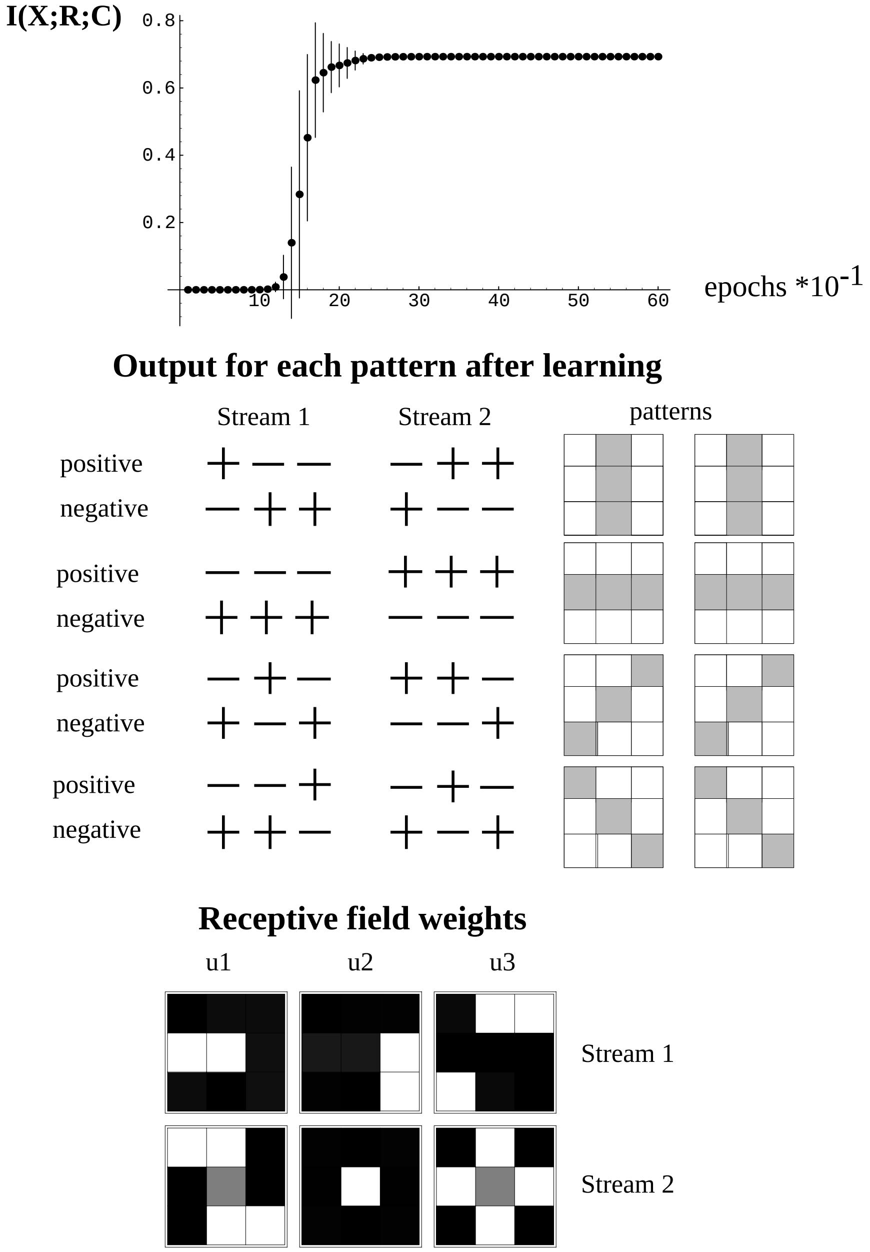

I(X;R;C)
0.8
0.6
0.4
0.2
10
20
30
40
50
60
epochs *10^-1
Output for each pattern after learning
Stream 1
Stream 2
patterns
positive + − −  − + +
negative − + +  + − −
positive − − −  + + +
negative + + +  − − −
positive − + −  + + −
negative + − +  − − +
positive − − +  − + −
negative + + −  + − +
Receptive field weights
u1
u2
u3
Stream 1
Stream 2

Figure 4:

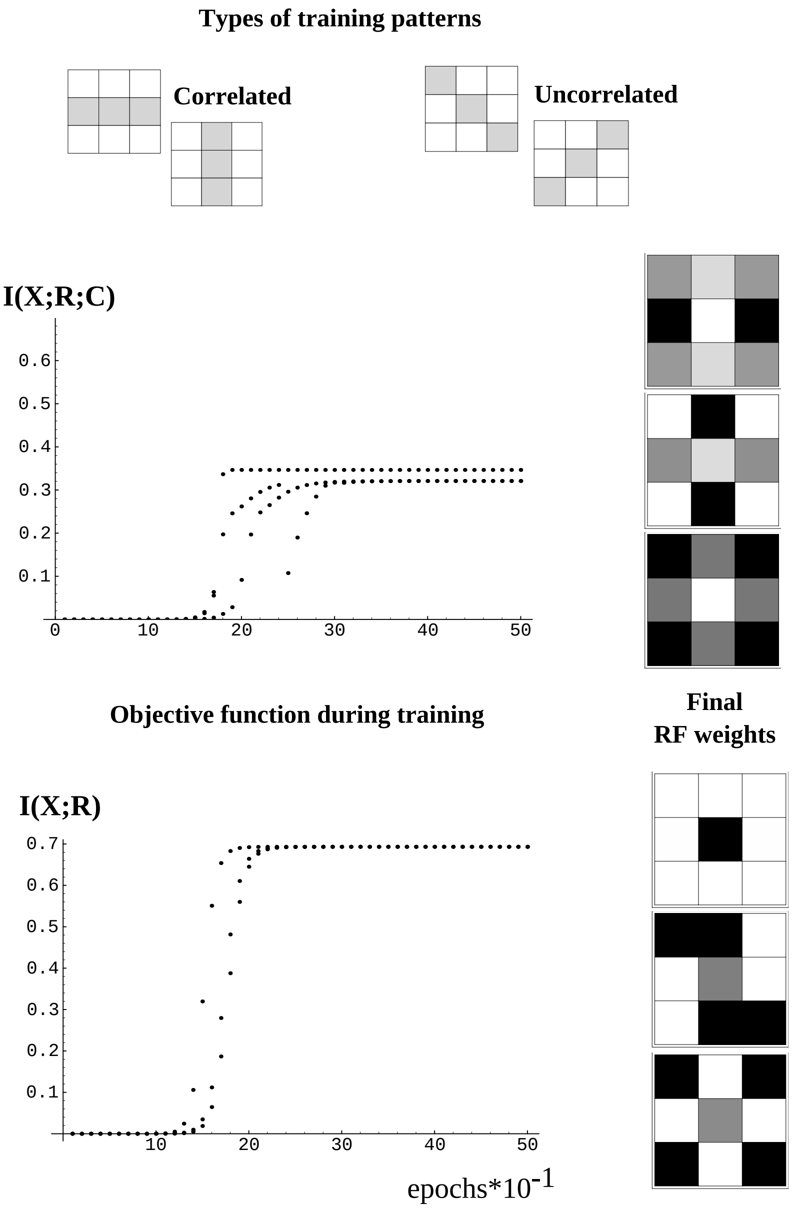

Types of training patterns
Correlated
Uncorrelated
I(X;R;C)
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
Objective function during training
Final
RF weights
I(X;R)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
50
epochs*10^-1

Figure 5:

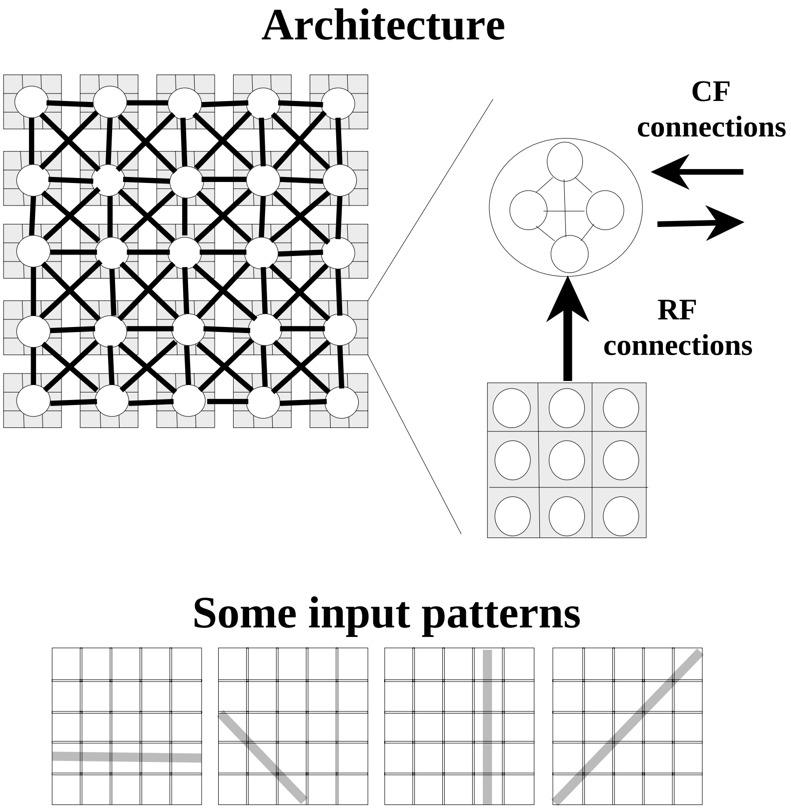

Architecture
CF
connections
RF
connections
Some input patterns

Figure 6:

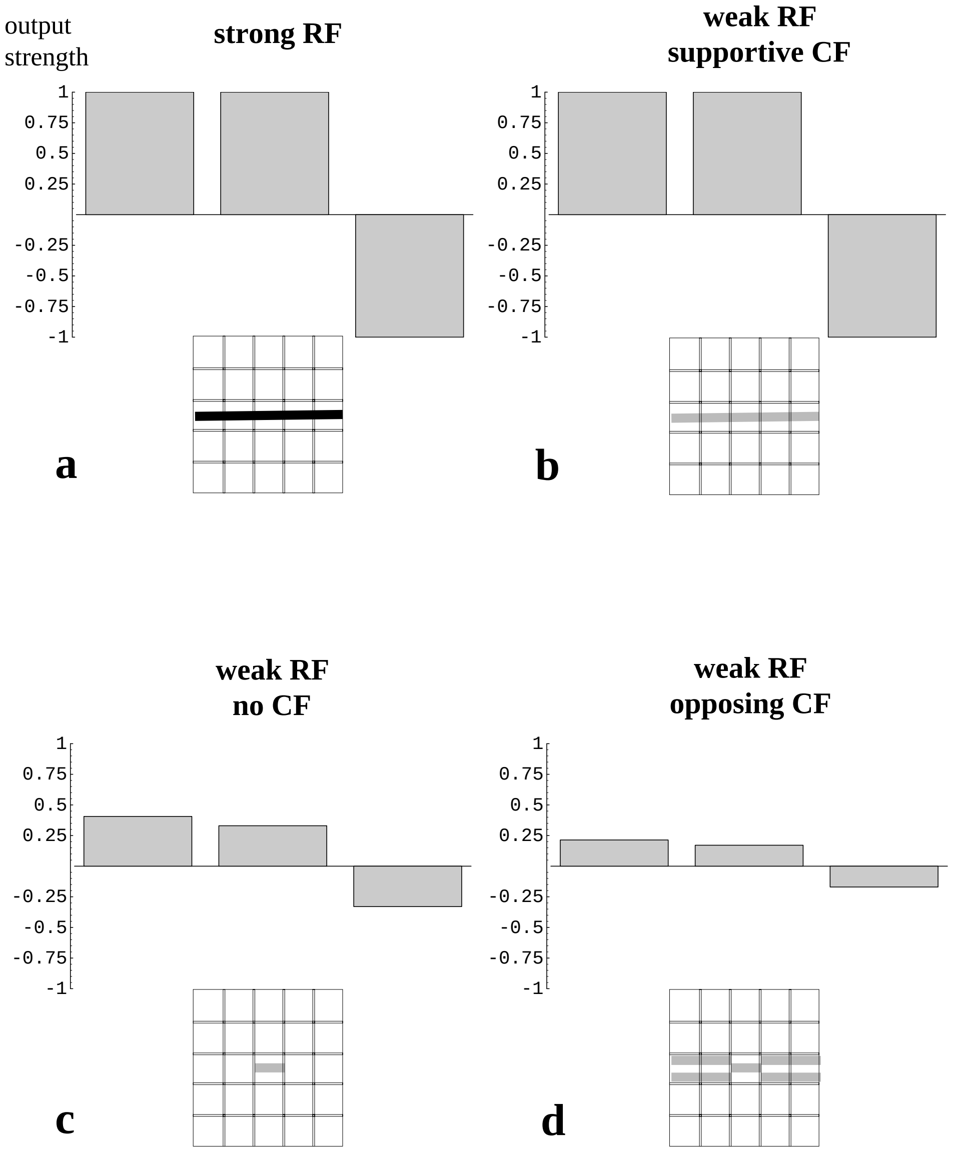

output
strength
strong RF
weak RF
supportive CF
weak RF
no CF
weak RF
opposing CF
1
0.75
0.5
0.25
-0.25
-0.5
-0.75
-1
a
b
c
d

Figure 7:

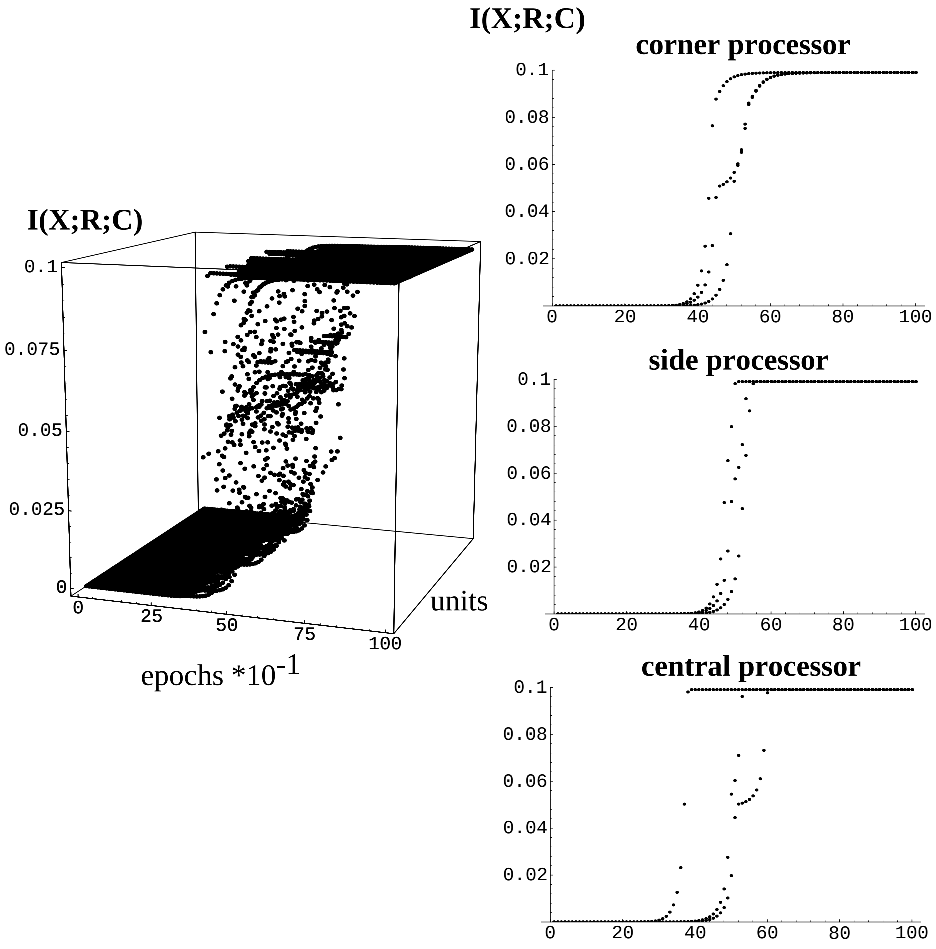
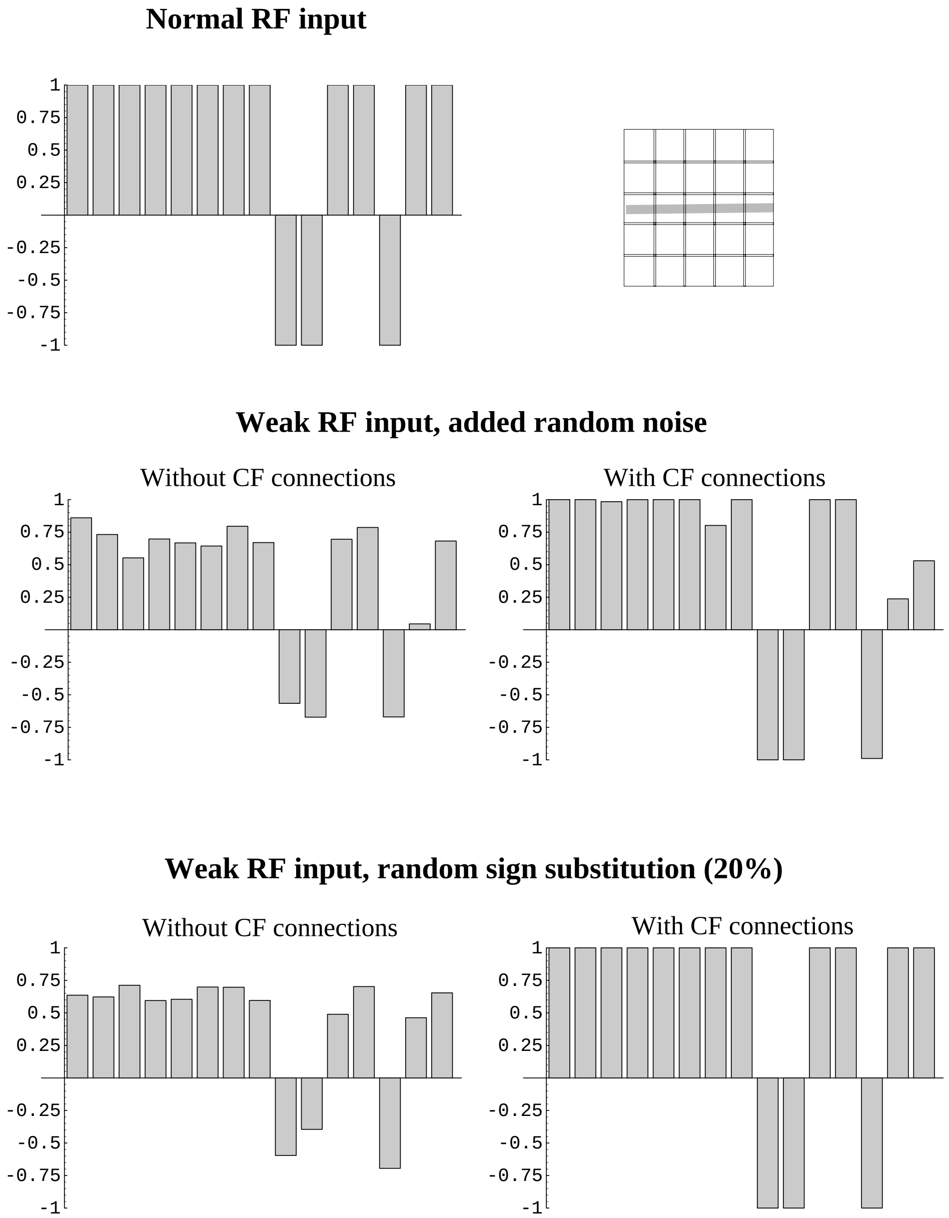

I(X;R;C)
0.1
0.075
0.05
0.025
0
0
25
50
75
100
units
epochs *10^-1
I(X;R;C)
corner processor
0.1
0.08
0.06
0.04
0.02
0
20
40
60
80
100
side processor
0.1
0.08
0.06
0.04
0.02
0
20
40
60
80
100
central processor
0.1
0.08
0.06
0.04
0.02
0
20
40
60
80
100

Figure 8:



Normal RF input

Weak RF input, added random noise

Without CF connections

With CF connections

Weak RF input, random sign substitution (20%)

Without CF connections

With CF connections

Figure 9:

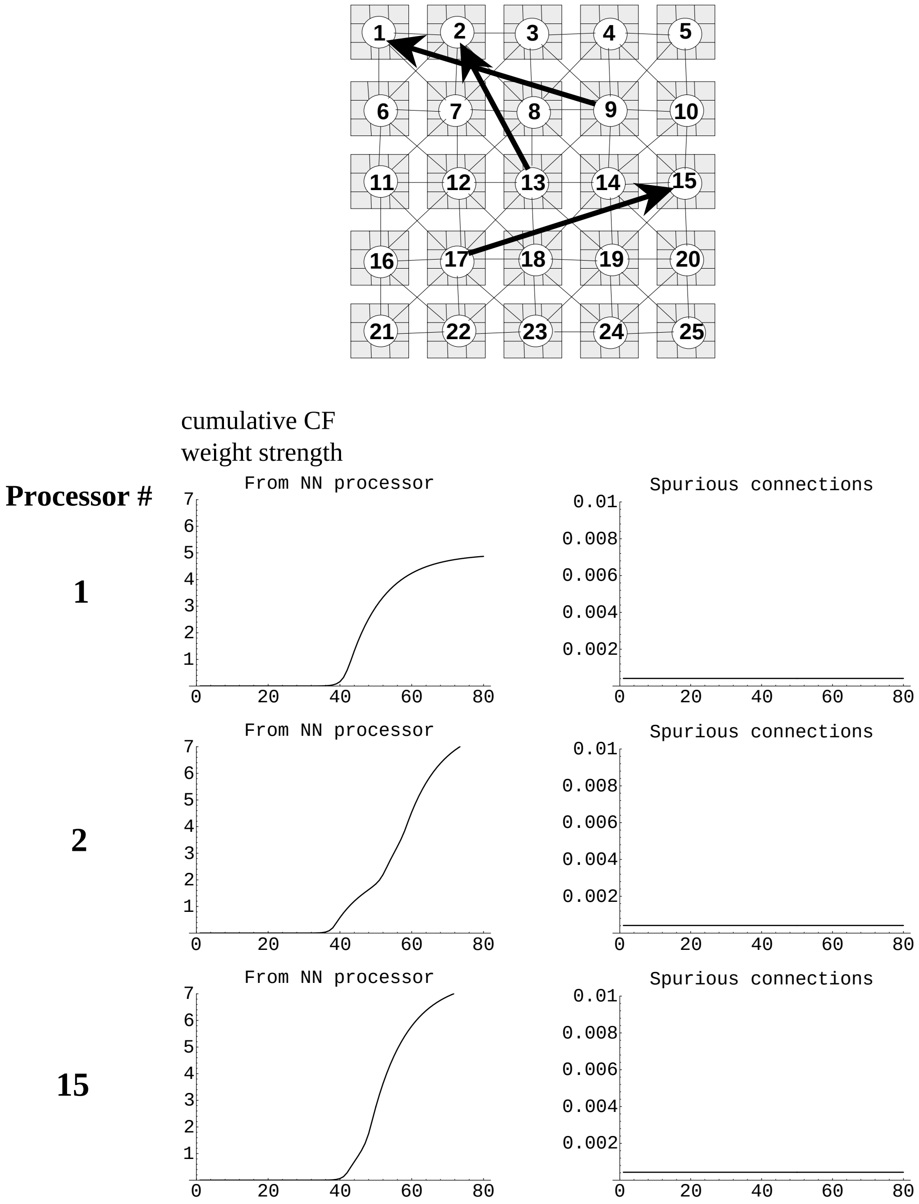

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
cumulative CF
weight strength
Processor #
1
2
15
From NN processor
Spurious connections
7
6
5
4
3
2
1
0.01
0.008
0.006
0.004
0.002
0
20
40
60
80

Figure 10: