International Journal of Neural Systems, Vol. 7, No. 3 (July, 1996) 223–236

LEARNING MULTIPLE CAUSES BY COMPETITION ENHANCED LEAST MEAN SQUARE ERROR RECONSTRUCTION

BAI-LING ZHANG*, LEI XU† and MINYUE FU*
*Department of Electrical and Computer Engineering,
The University of Newcastle, Callaghan, NSW 2308, Australia
†Department of Computer Science,
The Chinese University of Hong Kong, Shatin, N.T. Hong Kong

Received 11 December 1995
Revised 28 June 1996
Accepted 15 July 1996

In this paper we studied a self-organization principle that input should be best reconstructed from a factorial distributed hidden representation, which has been addressed in the literature recently. An auto-encoder network is trained by the Least Mean Square Error Reconstruction (LMSER) while the redundance in the representation is reduced by a proposed anti-Hebbian scheme, in which a penalty term called Receptive Field Overlapping Index (RFOI) is combined into the objective function for enhancing competition among nodes in the network. Our learning scheme provides a way for balancing the cooperation and competition necessary for the self-organization process thus realizes the multiple causes model, which accounts for an observed data by combining assertions from the discovered causes or features in the data. Our experiment results demonstrate again the powerful information processing capability inherent to the popular weighted sum followed by sigmoid squashing. Comparing with previous probability theory based multiple causes models, our scheme is much easier to implement and quite reliable.

1. Introduction

Many of the unsupervised learning paradigms can be seen as focusing on one of the two themes: Principle Component Analysis (PCA) and Competitive Learning (CL).[1] Since the pioneering work of Oja,[3] much advances have been made along the direction of neural learning PCA, including various models and extensions.[4–8] From the viewpoint that human perceptual system can be considered as a statistical inference engine whose function is to infer the probable causes of sensing inputs,[1] PCA learning is interesting mainly because it provides a distributed, factorial representation and such a representation can be used as multiple causes to explain a given input. Distributed representation and factorial representation are both important as the former can encode similarity while the later can maximally transfer information. However, PCA has an inherent weak point of only providing linear mapping which is trivial in many occasions. On the other hand, CL usually forms a highly nonlinear mapping from the input vector to the code by performing clustering or vector quantization. In a CL learning paradigm,[2] each cluster is represented by a processing unit that competes with others in a winner-take-all or winner-take-quota manner for each input pattern. By dividing the input space into disjoint regions, CL construct a purely local representation in which a single unit is activated in response to an input, which means an input vector has only single cause to be corresponded. Thus a question arises, how to inherit the

strong points of PCA and CL while overcome their weaknesses?

At present, many researches can be considered as a merge of PCA and CL, with an objective of forming distributed, factorial nonlinear representation. Here the nonlinearity demonstrates its diversification and different nonlinear activations (or algorithms) can bring quite different results. Sometime they are treated as nonlinear extensions of some PCA learnings or shortly termed as nonlinear PCA.[8–11] Basically, nonlinear PCA learnings have a common root in such a fact that nonlinear neurons have selectivities.[6,7] In other words, while linear neurons learn to a statistical mixture of all of the input patterns, nonlinear neurons learn input patterns discriminatively, thus partitioning the input space. This previously discovered property has been recently strictly studied from the viewpoint of statistical mechanics.[12]

Many nonlinear Hebbian or PCA-type learnings for feature extraction are closely related to the redundancy reduction principle formulated by Barlow,[13] who stressed the importance of extracting statistical relevant and independent features from sensory information in the process of cognition. The term redundancy means the statistical dependence between the components involved, and the learning refer to factorial learning because it tries to find factorial representation (code) for input. Recently many researches have been conducted toward this direction. For example, an architecture was established to perform volume-conserving transformation (i.e. determinant of the Jacobian matrix is equal to 1) for redundancy reduction, which has a property that information can be losslessly transmitted.[14,15] Starting from information theoretic concepts, a reversible cellular automata architecture was studied for performing nonlinear decorrelation.[18–20] A general predictability minimization principle is also for factorial learning, with corresponding architecture mainly composed of two type modules (predictor module and representational module).[26] It is worthy to mention that a closely related theme called "blind separation of sources" or Independent Components Analysis (ICA) has frequently been addressed in the last few years, mainly in the signal processing literature. The relationship between these two research lines has been discussed in detail in a recent paper.[21]

While factorial learning aiming at finding distributed, independent representation, a lately proposed Multiple Causes Model (MCM) further addressed the importance of causal relationship between input and such a representation or hidden causes.[22–26] From an explanative viewpoint, MCM aims at discovering a set of independent causes or generators such that each input can be completely described by the cooperative action of a few of these possible generators. In this aspect, a multiple cause model distinguish itself as compared with some single cause models such as CL and the well-known mixture of experts,[27] in which one generator or expert is only responsible for a single example. Recently, several papers have discussed the multiple causes model in the literature.

A simple scheme was previously proposed for extracting multiple independent features from the viewpoint of forming sparse representations by anti-Hebbian learning.[22] In a sparse code, the input patterns can be represented combinatorially by a relatively small number of the available units. However, there is no explanation mechanism and the success of this model strongly depends on a prior constraints on the activity patterns at the encoding layer. Specifically, a sparseness assumption was incorporated by taking the form of generating probability for each input component as a constraint for few hidden units to become active at one time. It is obvious that such an assumption is inappropriate when the generating probability for input component is not available.

Later, a form of autoencoder network was considered with the hidden units signalling features and the hidden-output weights describing the way in which features generate predictions of the inputs.[23] The conventional sigmoid at the output layer was replaced by a noisy-or activation function, which allows multiple causes to cooperate in a probabilistically justified way to activate the reconstruction units. Noisy-or scheme has a severe problem of local minimum as reported in.[24] In order for multiple causes to interact that is more competitive than the noisy-or, started from the viewpoint of learning a set of priors and conditional priors, the description length of a set of examples drawn from the input distribution is minimized in the paper.[24] Specifically, an autoencoder networks is trained to reconstruct the input on its output units with the goal of learning the underlying distributions. This scheme is a

special case of the general stochastic learning framework Helmholtz machine,[25] learning the distribution for hidden units in the recognition model is simplified by a fixed independent prior distribution and the parameter of the generative model is simply taken as interpreting probability from hidden causes.

Multiple cause model is a typical example that involve a balance between cooperation and competition. Such a cooperation and competition has been explicitly expounded in the Helmholtz machine.[25] Though the general learning algorithm is complex in the form, its principle can be approached via some appropriate simplifications in a deterministic model. In this paper, we study multiple causes model from optimization perspective, particularly, a single layer feedforward network trained with the Least Mean Square Error Reconstruction (LMSER) learning rule,[6-7] because such an auto-encoder network essentially provides an explanation mechanism for input and its ubiquitousness in neural learning paradigms has been pointed out in.[1] In a nonlinear network, LMSER learning not only provides a straightforward way for cooperatively interpreting a given input, but also implicitly takes the advantage of selectivity provided by neuron's nonlinearity.[6-7] However, with sigmoidal type non-local nonlinearity, the cooperation of the multiple causes is generally dominated. From this consideration, we discuss some approaches for enforcing the competition. Specifically, we add some constraint terms to a best reconstruction objective function to minimize the overlap between the receptive fields of two different output nodes, which approximately make the extracted features independent. For the typical independent horizontal/vertical bars example, our learning scheme can extract multiple causes satisfactorily.

2. Learning Multiple Cause Model (MCM) by Competition Enhanced LMSER

2.1. *MCM emerging from cooperation and competition*

A multiple cause model concerns two criteria. The first is the independence criterion, i.e. the occurrence of each cause or generator ought to be independent of all other causes or generators; and the hidden representations ought to be independent for interpreting a given input. Here we distinguish causes or features and hidden representations. In a nonlinear feedforward network, the former is represented by feedforward connection weights and the later is represented by nonlinear activations. The second is best reconstruction criterion which means that input could be best reconstructed in some sense from a few of the causes or generators. This criterion is also termed as an invertibility criterion in the predictability minimization principle.[26]

From the above criteria we can understand that a multiple cause model readily realizes factorial learning or redundancy reduction but not vice-versa. Most of the proposed factorial learning schemes based on some information theoretical criteria, e.g. maximal information transmission, which was initiated by Linsker's Informax principle.[28] While these models implement redundancy reduction or factorial coding, they do not provide interpretations for inputs via combinatorially using the causes.

Reconstruction commonly refers to self-association which is embodied in a number of neural network models, for example, various auto-associative memory models,[29] the ART and BAM architectures,[30] and has been studied in detail recently for linear case.[31-32] As an optimization issue, there are many specific objective functions toward best reconstruction, for example, minimal squared error, minimal cross entropy, etc. Among these objectives, minimal squared error is simple and often used, for example, in the famous error back-propagation algorithm.

Consider a nonlinear feedforward network architecture, as shown in Fig. 1(a) which has L input units, L output units, and M hidden units for representation. $L \times M$ matrix $\mathbf{W}$, $M \times L$ matrix $\hat{\mathbf{W}}$ denote the connection weights from input to hidden units and from hidden units to output, respectively. The column vector $\mathbf{w}(m) = [w_1(m), \ldots, w_L(m)]^T$ represents the weights associated with the m-th hidden unit, which has a postsynaptic potential $h_m = \sum_{j=1}^{L} w_j(m)x_j = \mathbf{x}^T\mathbf{w}(m)$. Nonlinear transfer function f acts on the postsynaptic potential, yielding nonlinear activations $\mathbf{y} = [y_1, \ldots, y_M]^T = f(\mathbf{W}^T\mathbf{x})$. We take f as a sigmoidal type with value in interval $[0, 1]$, e.g. $f(t) = \frac{1}{1+e^{-\beta t}}$, mainly from the following considerations. First, sigmoidal units with activation value in $[0, 1]$ can be interpreted as the posterior probability of the presence of some features given an input, which also can be considered as an assurance

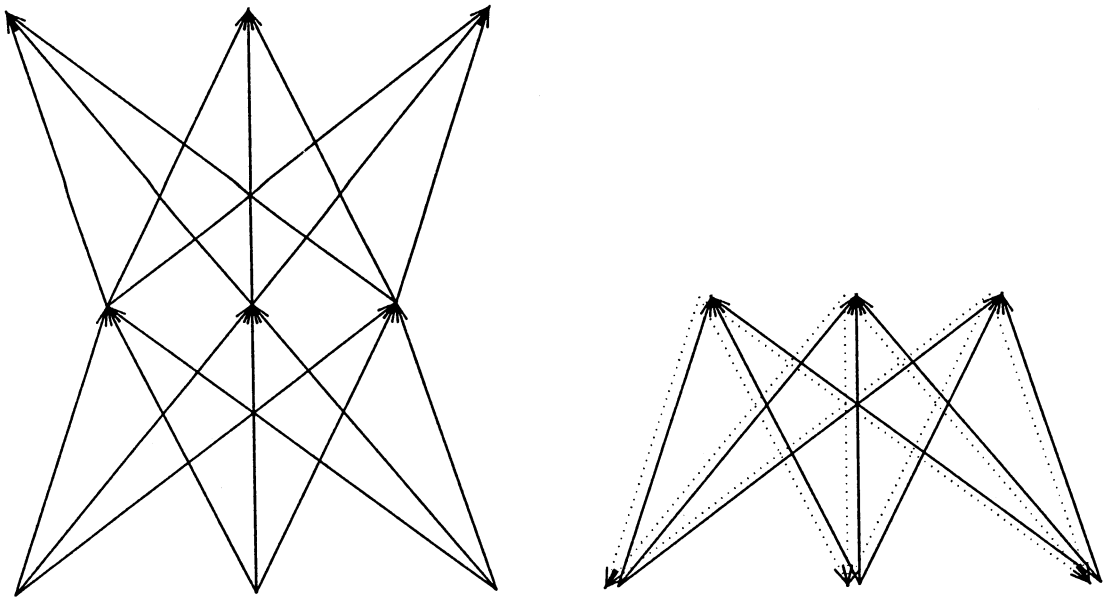


Fig. 1. Auto-encoder architecture. (a) A two-layer feedforward network, (b) An equivalent forward-backward single layer network.

measure for the corresponding cause in explaining a given input. Second, monopolar nodes more closely relates to the characteristics of biological neurons as producing a non-negative output firing rate.

The standard architecture in Fig. 1(a) is equivalent to a single-layer network in Fig. 1(b) with both bottom-up connections $\mathbf{W}$ and top-down connections $\hat{\mathbf{W}}$. In the following we mainly consider this architecture. To avoid confusion, we change the terminologies hidden units (representation) and output units in Fig. 1(a) to output units (representation) and reconstruction units in Fig. 1(b). Denote $\tilde{\mathbf{x}} = \hat{\mathbf{W}}^T\mathbf{y} = \hat{\mathbf{W}}^T f(\mathbf{W}^T\mathbf{x})$, representing a reconstruction vector of the input data $\mathbf{x}$ from the representation $\mathbf{y}$, then learning is based on the following optimization criterion:

$$J(\mathbf{W}) = \int p(\mathbf{x})\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 d\mathbf{x} = \int p(\mathbf{x})\|\mathbf{x} - \hat{\mathbf{W}}^T f(\mathbf{W}^T\mathbf{x})\|^2 d\mathbf{x} \quad (1)$$

where $p(\mathbf{x})$ is input distribution and we also use symbol p in other places for different probabilistic distributions.

Simply let $\hat{\mathbf{W}} = \mathbf{W}^T$, that is, the forward connection weights as extracted features or causes are used in the backward connection weights as reconstruction coefficients for interpreting an input. Equation (1) becomes the one layer special case of Least Mean Square Error Reconstruction (LMSER) learning principle studied by one of the present authors in the papers,[6-7] and using stochastic approximation with gradient descent, a learning algorithm for the minimization of $J(\mathbf{W})$ can be readily derived[6-7]:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k\mathbf{e}_k^T\mathbf{W}_k\mathbf{y}_k' + \mathbf{e}_k\mathbf{y}_k^T] \quad (2)$$

where k denoting a time scale and we will drop it in discussion for brevity without causing confusion, $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$ is the reconstruction error, $\mathbf{y}'$ is derivatives of $\mathbf{y}$. μ_k is a learning rate. Comparing with the Helmholtz machine, if we consider the bottom-up connections as the parameters of a recognition model for identifying causes, and the top-down connections as the parameters of a generative model for predicting or reconstructing input, then there exist an obvious difference between these two learning paradigms, as the recognition model and generative model in Helmholtz machine are treated separately with different parameters.

If the network outputs are constrained to be linear, the features that are extracted by the LMSER rule Eq. (2) span the M-dimensional principal subspace, which is same as the linear autoencoder that

has been proven to be equivalent to subspace methods. In nonlinear cases, though it still produces principal subspace, the nonlinear transformation can bring out such advantages as highly compressed code and robustness, thus providing an efficient way for signal frequencies estimation.[33-34]

An extension of the LMSER principle called min-distorted reflection theory has been proposed.[8] As to the single layer network structure, the input layer and output layer can be considered as two boundaries which continually reflects the signals bidirectionally. Specifically, consider $\hat{\mathbf{x}}^1 = \mathbf{W}f(\mathbf{W}^T\mathbf{x})$ as a reflection of $\mathbf{x}$ bounced back by the output boundary, and $\hat{\mathbf{x}}^{i+1} = \mathbf{W}f(\mathbf{W}^T\hat{\mathbf{x}}^i)$ as a reflection of $\hat{\mathbf{x}}^i$, $i = 0, 1, \ldots, K$, K denotes the number of reflection times. $\hat{\mathbf{x}}^0 = \mathbf{x}$. We hope that the distortions of reflections should be as small as possible, then the minimization objective is:

$$J(\mathbf{W}) = \sum_{i=0}^{K} E\{\|\hat{\mathbf{x}}^{i+1} - \hat{\mathbf{x}}^i\|^2\}. \tag{3}$$

Similar to the derivation of Eq. (2), we can get the following learning algorithm

$$\Delta\mathbf{W} = \mu \sum_{i=0}^{K} [\hat{\mathbf{x}}^i \mathbf{e}^{i\mathrm{T}} \mathbf{W} \mathbf{y}^{i\prime} + \mathbf{e}^i \mathbf{y}^{i\mathrm{T}}] \tag{4}$$

where $\mathbf{e}^i = \hat{\mathbf{x}}^{i+1} - \hat{\mathbf{x}}^i$ is the reconstruction error at i reflection, $\mathbf{y}^i$ is the corresponding output vector. Generally speaking, the qualitative properties of learning rules Eqs. (2) and (4) are similar, though strict theoretical analysis still seems necessary.

The sigmoid nonlinearity results in competition among the neurons for firing with a given input and makes that neurons have selectivities,[6-7] which is an important property to many learning tasks, for example, classification, clustering, generalization, etc. However, such a competition will be still quite weak for a best reconstruction learning process as the main goal of a best reconstruction learning algorithm as Eq. (2) is to cooperatively use several causes per input. In order to get a satisfactory multiple causes model, the provided cooperation must be balanced by an enforced competition which aims at finding independent causes or generators as well as assigning different responsibilities allocation among the representation units. In other words, multiple cause model emerges from an interaction of cooperation and competition.

From the above discussion, learning multiple cause model can be cast into a constrained optimization issue, with main cost reconstruction error guiding the cooperation and a penalty cost enhancing the competition, i.e. the aim of multiple causes model is

$$\text{minimize} \quad E(\mathbf{W}) = J(\mathbf{W}) + \lambda G(\mathbf{W}) \tag{5}$$

where J is a best reconstruction criterion and G is a competition criterion which is the focus of the next section. λ is a trade-off parameter for a compromise between the two optimization criteria.

2.2. *Approaches toward independence criterion*

There are two tasks for the independence criterion. First, the extracted features or causes should be independent, which means the postsynaptic potentials $h_i (i = 1, \ldots, M)$ should be factorial. Second, the output representations must assume independent responsibilities for interpreting a given input. Here we would like to point out that in many situations these two tasks can be considered as approximately equivalent in a nonlinear network. In the absence of input noise, the mutual information $I(\mathbf{y}, \mathbf{x})$ between random variables $\mathbf{y}$ and $\mathbf{x}$ is equal to the mutual information $I(\mathbf{y}, \mathbf{h})$ between $\mathbf{y}$ and the postsynaptic potentials $\mathbf{h}$ as follows[35]

$$I(\mathbf{y}, \mathbf{x}) = I(\mathbf{y}, \mathbf{h}). \tag{6}$$

If the postsynaptic potentials are factorial, i.e.

$$p(\mathbf{h}) = \prod_{\mathrm{i}=1}^{\mathrm{M}} p(\mathrm{h_i}) \tag{7}$$

and individual nonlinear transfer functions could be adapted according to

$$f_i'(h_i) = p(h_i), \quad i = 1, \ldots, M \tag{8}$$

then mutual information $I(\mathbf{y}, \mathbf{h})$ between $\mathbf{y}$ and $\mathbf{h}$

$$I(\mathbf{y}, \mathbf{h}) = \int p(\mathbf{h}) p(\mathbf{y}|\mathbf{h}) \ln \frac{p(\mathbf{y}|\mathbf{h})}{p(\mathbf{y})} d\mathbf{h} d\mathbf{y} \tag{9}$$

will be maximized,[36] which also means that the entropy of output distribution

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \ln p(\mathbf{y}) d\mathbf{y} \tag{10}$$

will be maximized.[35] Denote the mutual information of the variable $\mathbf{y}$ as $MI(\mathbf{y})$, which is a relative entropy between $p(\mathbf{y})$ and $\prod_i^M p(y_i)$

$$0 \leq MI(\mathbf{y}) = -H(\mathbf{y}) + \sum_i H(y_i) \qquad (11)$$

i.e.,

$$H(\mathbf{y}) = \sum_i H(y_i) - MI(\mathbf{y}) \qquad (12)$$

we can see that maximizing $H(\mathbf{y})$ will minimize the mutual information $MI(\mathbf{y})$, thus making the outputs independent.[a]

From Eq. (11), factorial learning, or less strictly, redundancy reduction, can be formulated as to make $MI(\mathbf{y})$ as small as possible. An obvious way is minimizing $\sum_i H(y_i)$, which can be termed as bits entropy or pixel entropy.[13] However, the probability distribution $p(y_i)$ involved in the pixel entropy is generally difficult to analytically calculate except in some special cases. Deco and Parra[15] offered a method for solving this problem by reducing it to minimize the upper bound of pixel entropies which is entropy of a sum of Gaussian distributions, i.e.

$$\begin{aligned} G &= -\sum_{i=1}^{M} \int p(y_i) \log p(y_i) dy_i \\ &\leq -\sum_{i=1}^{M} \int p(y_i) \log q(y_i) dy_i \\ &= \frac{M}{2} \log(2\pi) + \frac{1}{2} \sum_{i=1}^{M} \int p(y_i)(y_i - \overline{y_i})^2 dy_i \end{aligned} \qquad (13)$$

$\overline{y_i}$ is the mean of y_i, $q(y_i)$ is the Gaussian distribution. In a simplified form, we can minimize

$$G = \sum_{i=1}^{M} y_i^2 . \qquad (14)$$

Another way to decorrelate non-Gaussian distribution is to expand the distribution in higher orders of correlation matrix and then impose the independence condition.[14] Such a cumulant-based method is obviously complex. It needs more memory requirements for computing the relevant statistical quantities. As being observed by many researchers, nonlinear function enables a network to compute with non-gaussian statistics, and find higher-order forms of redundancy inherent in the inputs.[21]

Pixel entropy minimization can be generally realized by a competition or anti-Hebbian mechanism among the output units. This means that any two activations have some kind of inverse relationships, which will make activations decorrelated or independent. In a simple case, decorrelating sigmoidal activations can take higher-order moments of the output distributions into computation and approach independent postsynaptic potentials. Taking Taylor expansion of sigmoid $f(h) = \frac{1}{1+e^{-h}} = \sum_k b_k h^{2k+1}$, where b_j are coefficients, then the decorrelation of hidden activations, i.e.

$$\sum_{i,j} \overline{y_i y_j} = 0 \qquad (15)$$

actually means that

$$\sum_{i \neq j} \sum_{k,l} b_{ijkl} \overline{h_i^{2k+1} h_j^{2l+1}} = 0 . \qquad (16)$$

As being argued in,[21] Eq. (16) can be thought of as an approximation of independence test.

A straightforward way of introducing competition is adding lateral inhibitory connections among the output units. However, in many cases it is neither desirable nor feasible to introduce explicit inhibitory links between competing nodes.[37] Besides, it is generally preferable to minimize the number of connections. Recently, some approaches for producing "competitive" or inhibitory effects in neural network models have been proposed,[37] e.g. the competitive activation mechanism, with which the nodes in a network compete for any externally applied activation. Unfortunately, the competitive activation mechanism is complex in the form though it is a universal principle that can be applied to any network structure.

An alternative way for controlling the spread of activation and implementing inhibitory interactions can be reached by minimizing some appropriate index which serve as the penalty cost in Eq. (5). For the convenience of discussion, such an index can

[a] Strictly speaking, condition Eq. (8) is necessary in the above discussion of information transmission in the neural network, which means the input *pdf* match the slope of sigmoid nonlinearity. Many real-world analog signals, including speech signals, are super-gaussian (with longer tails and being more sharply peaked than gaussians),[21] can be considered approximately satisfying this condition.

be termed as Receptive Fields Overlapping Index (RFOI) because joint activation of two units usually means the overlapping of the corresponding receptive fields.

Two nodes in a neural network are said to be competitors if the gain of one occurs at the expense of the other, i.e. if their functional relationships are inhibitory in nature.[38] For example, the ith node with

$$\frac{dy_i}{dt} \propto \sigma_i(\mathbf{y}) \tag{17}$$

can be said to involve competitive interactions with node j if $\frac{\partial \sigma_i}{\partial y_j} < 0$ when $i \neq j$. From this consideration, we propose the following conditions for a suitable receptive fields overlapping index.

First, the overall index G can be expressed as the summation of pairwise index g_{ij} over all possible combination of ith node and jth node ($i \neq j$), i.e.

$$G = \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}} g_{ij} \,. \tag{18}$$

Second, pairwise index g_{ij} should be a monotonically increasing function of y_i (or y_j), thus indicating their correlation in some sense.

Third, with fixed pairwise index g_{ij}, y_i can be expressed as a monotonically decreasing function of y_j and vice visa, which means that output activations inhibit each other.

Decreasing the index defined above can produce similar effect as from anti-Hebbian learning. We emphasize here that pairwise index g_{ij} is a statistical quantity over input distribution. Therefore, we can easily derive on-line algorithm with stochastic approximation.

RFOI can be designated by many function forms that meet the above requirements. In the following we discuss some of possible indexes.

1. RFOI I

$$g_{ij} = \int (y_i y_j)^k p(\mathbf{x}) d\mathbf{x} \tag{19}$$

where k is a positive integral, with $k = 1$ being the most simple one. Generally we take $k = 2$. Then using stochastic approximation, RFOI I can be replaced by

$$g_{ij} = (y_i y_j)^2 \,. \tag{20}$$

It follows from Eq. (5) that the change in weight Δw_{ij} can be made in proportional to the corresponding derivative of E by

$$\Delta \mathbf{w}(k) = \mu_k \left[\mathbf{x}\mathbf{e}^{\mathrm{T}} \mathbf{w}(k) y'_k + y_k \mathbf{e} - \lambda y'_k \mathbf{x} \sum_{l \neq k} y_l^2 \right]$$
$$k = 1, \ldots, M \,. \tag{21}$$

Equation (20) reflects a simple inverse proportional relationship between two activations via function $c/t (c > 0)$, as shown in Fig. 2(a). Decreasing g_{ij} amounts to increasing the mutual inhibition.

It is worthy to mention that the simple constraint scheme in Eq. (20) was first applied in the $G_{\max}$ learning for encouraging different output units to discover mutually exclusive features.[38] $G_{\max}$ is a powerful objective and can potentially capture arbitrarily high-order structure in the input distributions though its learning algorithm is quite complicated. Such a mechanism was also introduced into the so called Competitive Hebbian Learning (CHL).[39] The CHL is simple and effective in some feature detecting issues, but it has several shortcomings. First, the learning algorithm was proposed without solid theoretical foundations, especially the derivative of the output squashing function was dropped from the maximisation of cost function without sound reason. Second, an empirically chosen limit on the weights is a key factor for the success of the algorithm. Based on an information theoretical criterion, i.e., maximizing output variance, Deco and Obradovic[17] introduced a similar constraints in an RBF learning paradigm with output activations being normalized Gaussian function. The constraint penalizes the correlations between the outputs of Gaussian units, thus performing clustering in the input space.

RFOI I has a natural generalization with forms:

$$g_{ij} = -\int [y_i(1 - y_j)]^k p(\mathbf{x}) d\mathbf{x} \tag{22}$$

or

$$g_{ij} = -\int [(1 - y_i) y_j]^k p(\mathbf{x}) d\mathbf{x} \tag{23}$$

with y_i being regarded as a logic variable, i.e. $1 - y_i$ means the 'NO' version of y_i.

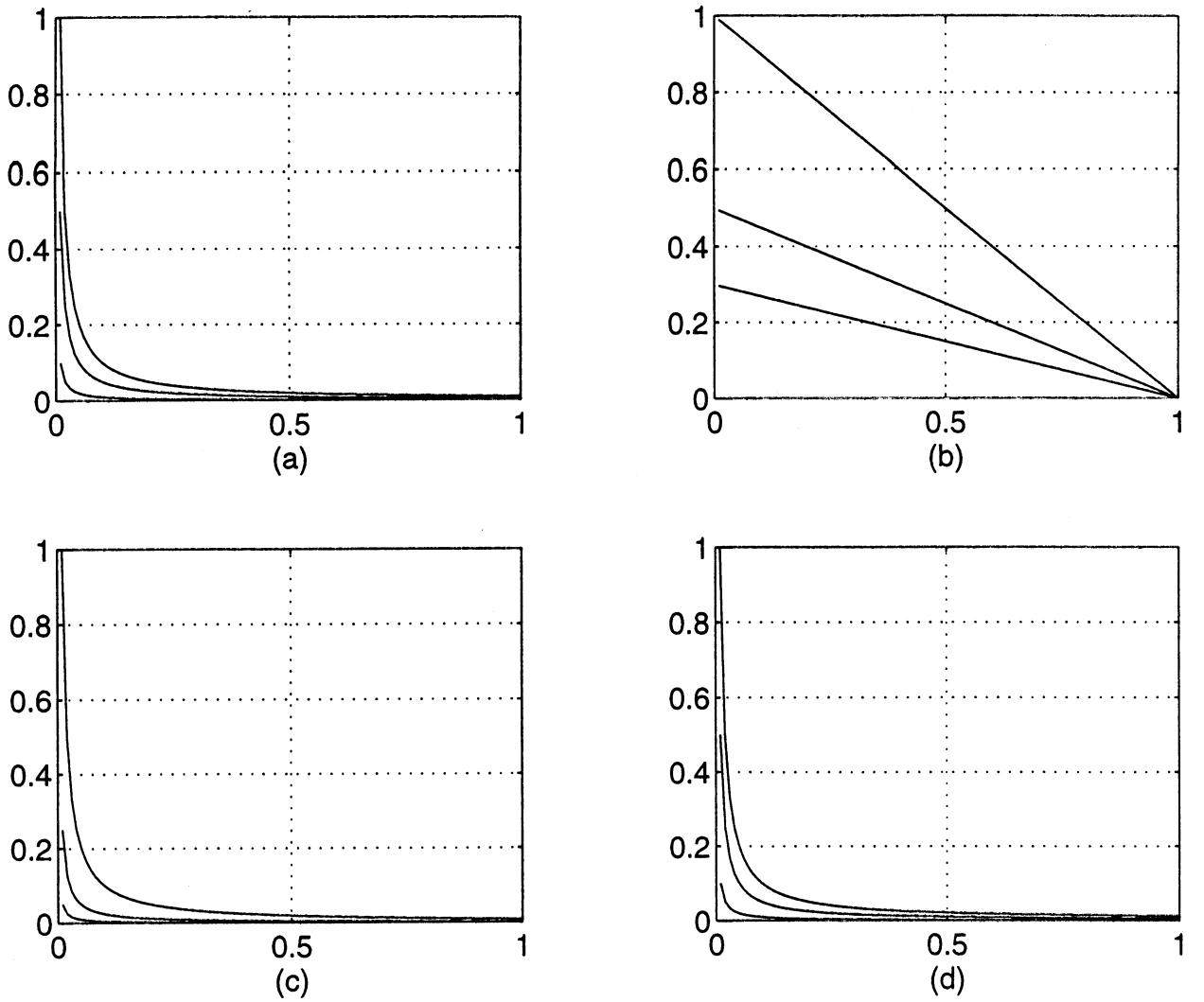


Fig. 2. Several functions that could serve as the receptive field overlapping index. (a) $c/t(c > 0)$, (b) $a - t$, $0 \leq a \leq 1$, (c) $\frac{c}{e^t - e^{-t}}$, $(c > 0)$, (d) $\frac{e^t + e^{-t}}{e^t - e^{-t}}$.

In a more general situation, the $(y_i y_j)^k$ term in RFOI I can be replaced by $g_{ij} = g(y_i y_j)$, with g being a differentiable increasing function.

2. RFOI II

$$g_{ij} = \int (y_i + y_j)^k p(\mathbf{x}) d\mathbf{x} \qquad (24)$$

with k is positive integral. This index is based on a simple linear function $a - t$, $0 \leq a \leq 1$ as shown in Fig. 2(b), i.e. any output is a linear decreasing function of other outputs. Similarly, it has following variants.

$$g_{ij} = -\int (y_i + 1 - y_j)^k p(\mathbf{x}) d\mathbf{x} \qquad (25)$$

or

$$g_{ij} = -\int (1 - y_i + y_j)^k p(\mathbf{x}) d\mathbf{x}. \qquad (26)$$

3. RFOI III

$$g_{ij} = \int y_i (e^{y_j} - e^{-y_j}) p(\mathbf{x}) d\mathbf{x} \qquad (27)$$

or

$$g_{ij} = \int y_j (e^{y_i} - e^{-y_i}) p(\mathbf{x}) d\mathbf{x}. \qquad (28)$$

This index comes from the function $\frac{c}{e^t - e^{-t}}$, $(c > 0)$, as shown in Fig. 2(c). For simplicity, we do not introduce power index k though it is also applicable here.

4. RFOI IV

$$g_{ij} = \int y_i(e^{2y_j} - 1)p(\mathbf{x})d\mathbf{x} \qquad (29)$$

or

$$g_{ij} = \int y_j(e^{2y_i} - 1)p(\mathbf{x})d\mathbf{x}. \qquad (30)$$

RFOI IV can be considered as generalization of RFOI III or established on the function $\frac{e^t+e^{-t}}{e^t-e^{-t}}$. See Fig. 2(d). In mathematics, there are many other functions that have similar characteristics as shown in Fig. 2, for example, $\frac{1}{2}\ln\frac{\sqrt{1+t^2}+1}{\sqrt{1+t^2}-1}(t \neq 0)$. While other function forms could be used, the above four type RFOI indexes are simple and their corresponding learning algorithm are quite easy to implement. To save the space, we omit the explicit learning algorithms.

Remark 1

The parameter λ in the above learning paradigm is a fitting parameter for the combination of two costs. In our experience, the range of λ can be chosen in a relatively large range without dramatically changing the performance. We leave the study on how to choose λ in future studies.

Remark 2

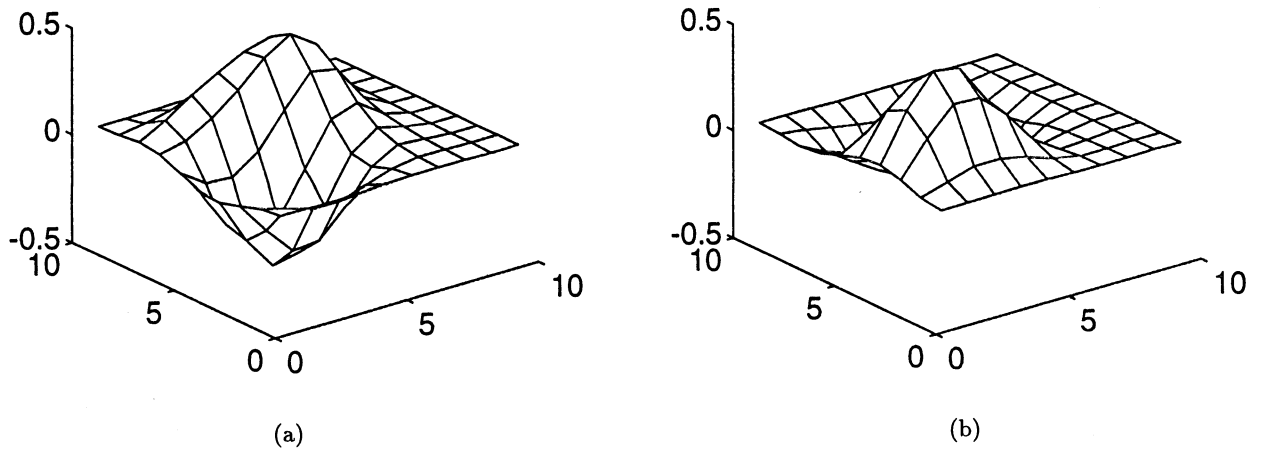
Gradient descent method, while simple to implement, suffers from some disadvantages. How to appropriately choose the learning parameter μ_k in order for the algorithm to converge and for the weight to be stable (remain bounded) is generally problem dependent. In linear output case, the μ_k should satisfy the condition $0 \leq \mu_k \leq 2\|\mathbf{x}_k\|^{-2}$ for the convergence.[33] In nonlinear situation, there is no such theoretical guideline. If μ_k is chosen too small, the convergence process may be very slow. On the other hand, instability occurs if μ_k is unsuitably chosen too large. An efficient way for solving such a problem is applying other optimization technique such as conjugate gradient.

3. Simulations

As a first example, we demonstrate that our learning scheme can partition the input space into minimally overlapping regions. We consider the problem of learning to respond to randomly placed Gaussian-shaped spots. The data generation scheme and training was similar to that used in the Competitive Hebbian Learning.[39] In the experiments, each input vector was a random located Gaussian spot, with its center at arbitrary position except that there must be two input units away from the nearest edge in the input array. In the simulations we are mainly interested in the cases of more than one output nodes in the consideration that a multi-nodes network should learn to share the input space and develop distinct regions of strong response.

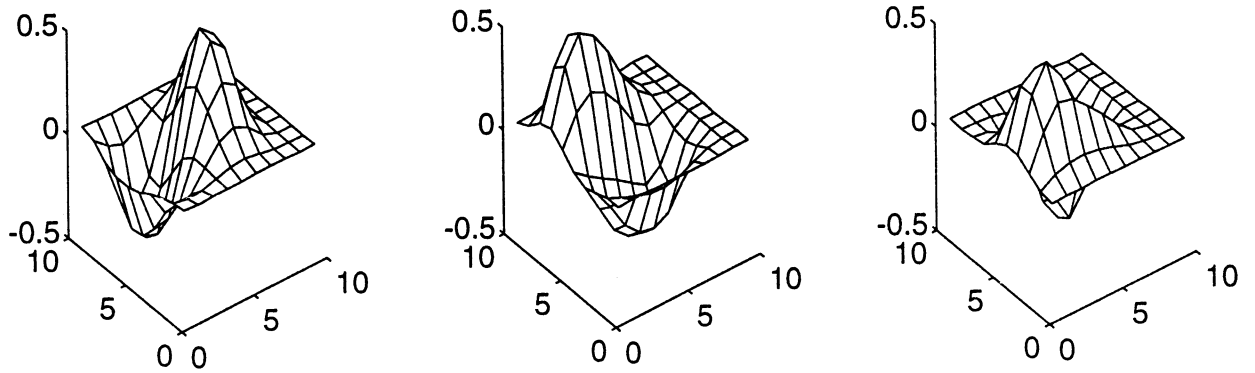
In the experiment, 100 input units with 10×10 square array were tested. The average brightness of 1000 Gaussian spots was calculated beforehand and then subtracted from each random Gaussian spot during training. Initial weights were set to small random values. The training was made with $\lambda = 1$. Typically, we test 5000-10000 training samples. Figures 3–5 illustrated typical results of two-nodes, three-nodes and four-nodes cases, respectively, with learning algorithm Eq. (21). We can find that different nodes have developed strong responses in nearly minimally overlapped different regions of the input space. The localized masks have their descriptive scopes that are narrowed to only certain regions of the full data space. The receptive fields of distinct units share their responsibility in accounting for each observed data.

The second example is a benchmark example of extracting a number of independent horizontal and vertical bars on an input pixel grid.[22–24] Figure 6 shows a test data set generated by the independent actions of 16 underlying components appearing as horizontal and vertical bars. In this example, hidden causes corresponding to the horizontal and vertical bars interact such that data pixels occurring at the intersection of bars remain black. An autoencoder network with a single hidden layer has been tested to capture the structure in these patterns using the sigmoid and noisy-or activation functions at the output layer and employing a cross-entropy error to evaluate the reconstruction.[23] This cross entropy error measure is similar to the use of a minimum description length strategy. Dayan and Zemel[24] used such an error measure and reported that the sigmoid scheme fails to capture the separate generators. The noisy-or does much better, but 73% of the time it gets stuck at a local minimum in which one or more



(a) (b)

Fig. 3. Learned responses of a pair of nodes trained on randomly placed Gaussian spots from learning algorithm Eq. (21). The nonlinear activation function is $f(t) = \frac{1}{1+e^{t}}$.



(a) (b) (c)

Fig. 4. Learned responses of three output nodes trained on randomly placed Gaussian spots from learning algorithm Eq. (21). The nonlinear activation function is same as in Fig. 3.

bars do not have individual generators. A more competitive rule was proposed in Ref. 24 with considerably improved performance, but it can still get stuck at a local minimum in 31% of time.

Our proposed learning scheme can be directly applied to binary data in such a problem. In a training data, each of the 16 possible lines are drawn with a fixed probability, for example, $\frac{1}{8}$, independently from all the others. Pixels that are part of a drawn line have the value 0, all others are 1. The network has 16 representation units. An extra node is introduced to account for the average brightness. The sigmoidal nonlinearity is taken as $f(x) = \frac{1}{1+e^{-x}}$. First, we tested with a generating probability $\frac{1}{8}$ via the learning algorithm Eq. (21). Figure 7 shows the learned weights (equalized image), which clearly reveal the generative model they embody. Next, we changed the generating probability to $\frac{2}{8}$ and $\frac{3}{8}$, respectively, with the similar results.

Another example of multiple causes structure is adopted from Saund,[23] with nine 121-dimensional test data samples shown in Fig. 8, which reflect two independent processes, one of which controls the positions of the black and white squares on the left-hand side, the other controlling the right. Similar to the experiment procedure in the above example, a

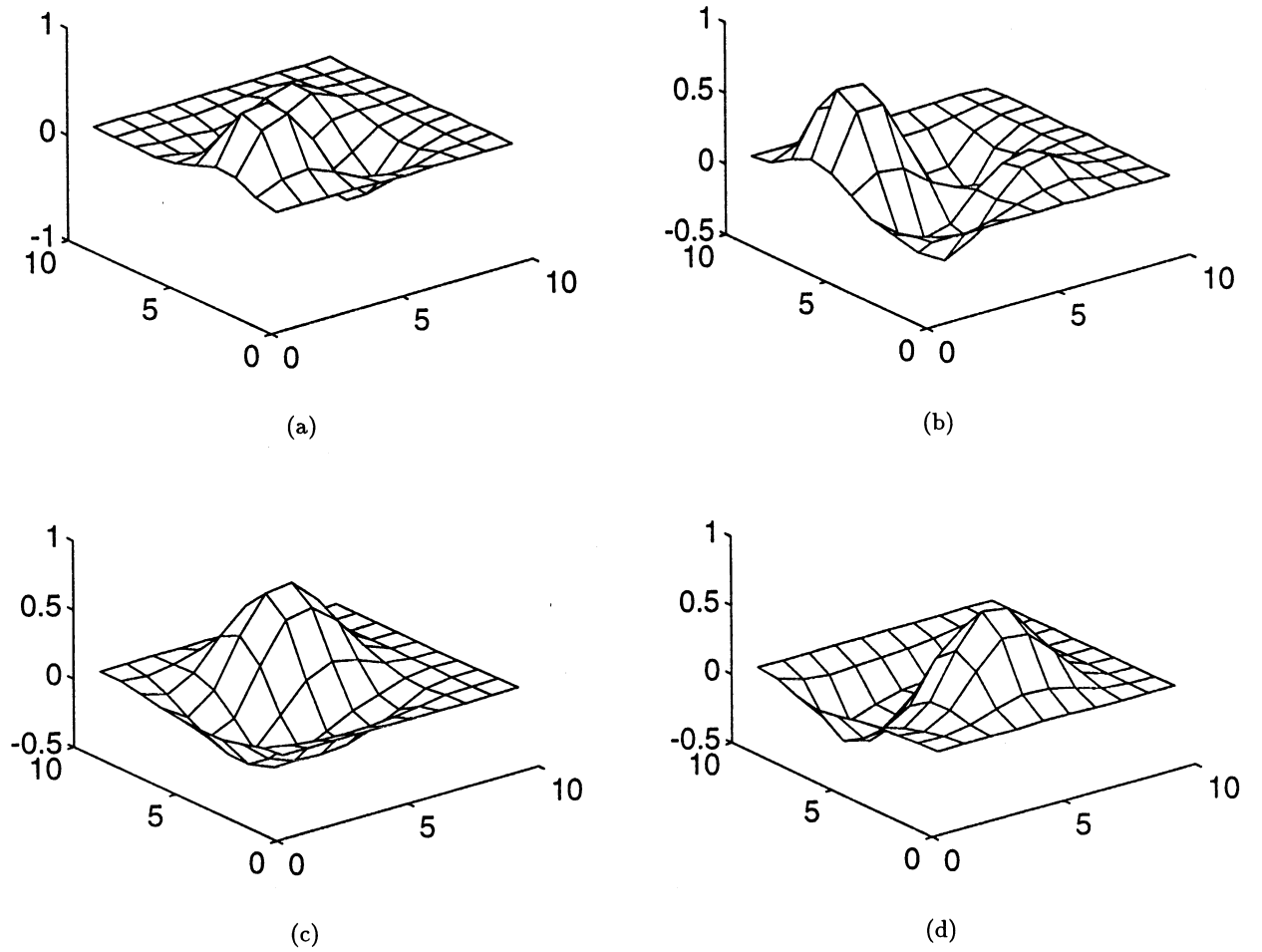


Fig. 5. Learned responses of four output nodes trained on randomly placed Gaussian spots from learning rule Eq. (21). The nonlinear activation function is same as in Fig. 3.

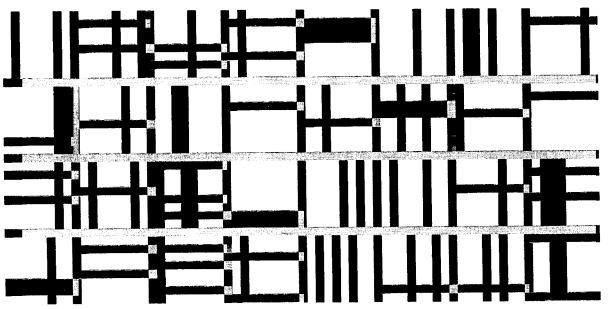


Fig. 6. Samples of horizontal and vertical bars in a 10×10 grid. Each bar is generated with probability $\frac{1}{8}$.

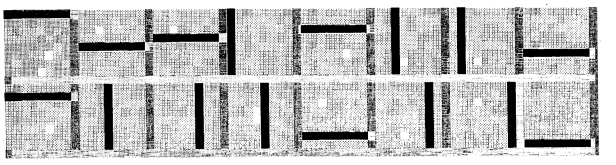


Fig. 7. Multiple causes representation for 2000 randomly generated horizontal and vertical bars discovered by the learning rule Eq. (21).

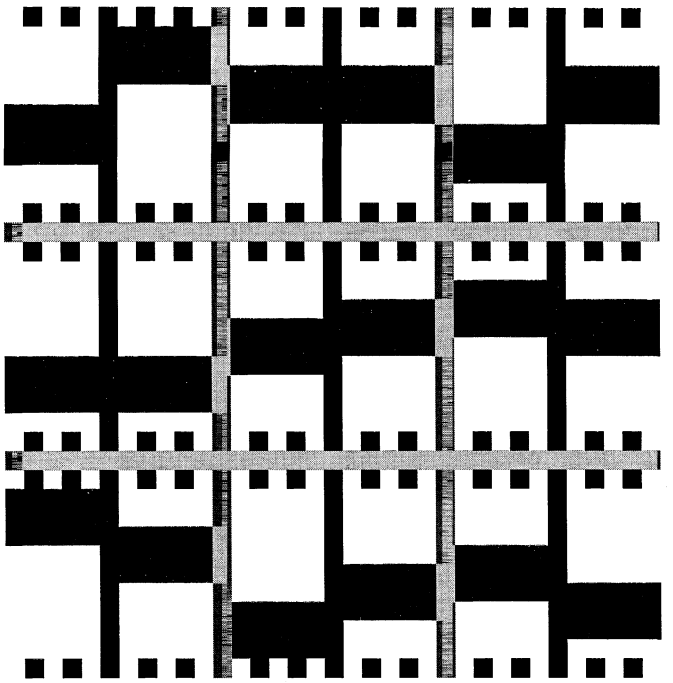


Fig. 8. Typical 121-dimensional test data samples designed by Saund, which exhibit multiple cause structure. Independent processes control the position of the block rectangle on the left- and right-hand sides.

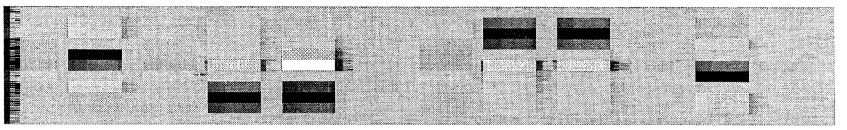


Fig. 9. The corresponding multiple causes representation for 2000 randomly generated data as shown in Fig. 8.

network with six representation units (plus another extra node for the average brightness) and sigmoidal nonlinearity $f(x) = \frac{1}{1+e^{-x}}$ is trained with algorithm Eq. (21). Figure 9 demonstrated an experiment result, showing the multiple cause representation for these data.

4. Discussions and Conclusions

Factorially representing the environment is an important object of unsupervised learning. Factorial codes has following advantages[26]:

1. Optimal input segmentation.
2. Speeding up supervised learning.
3. Occam's razor.
4. Novelty detection.

In addition, any factorial representation realizes maximal information transmission (under certain mild conditions). Learning multiple causes model is even more challenging in the sense that it need to not only find factorial representations, but also model the interaction of the representations with input for generating a set of patterns, which involve a delicate balance between cooperation and competition. Furthermore, learning multiple causes model also implicitly emphasizes the importance of distributed representation which expresses information by the ensemble behavior of a collection of microfeatures.[2]

The principle of learning multiple cause model is explicitly embodied in the self-supervised learning framework Helmholtz machine.[25] In Helmholtz machine, the competition and cooperation are

realized by two coupled modules, recognition module and generative module, with the former guiding the self-organization degree of formed representation and the later guiding the quality of reconstruction from the representation. Note that the independence criterion in recognition module is implicitly embodied in an assumption that the hidden representation is factorial. As been argued in Ref. 1, any method that communicates each hidden activity separately and independently will tend to result in factorial codes because any mutual information between hidden units will cause redundancy in the communicated message, so the pressure to keep the message short will squeeze out the redundancy. Stressing the causal relationship between the data and representation (code) from statistical inference, the Helmholtz machine is particularly important in such areas as source coding.

Our learning scheme can be considered as following the principle of Helmholtz machine[25] by a much simplified deterministic learning in an auto-encoder network. If we consider the bottom-up connections as the parameters of a recognition module and top-down connections as the parameters of a generative module, one important assumption in our learning scheme is that these two modules should be reciprocal, thus making the learning much easier. However, we would like to emphasize that the Helmholtz machine is much general, because it formulates learning as statistical inference which may underlie the mechanism of human perception. Though the typical task of multiple cause model of extracting horizontal/vertical bars can be easily tackled by our learning scheme, other tasks such as detecting the directions of shift patterns which was exemplified by Helmholtz machine is difficult to realize by our learning algorithms, possibly because the balance between cooperation and competition in this case is more difficult to control than the simpler horizontal/vertical bars example. Though the Helmholtz machine performs better for some learning tasks, our proposed scheme for learning multiple causes model is still interesting as it is based on simple constrained optimization principle.

References

1. G. E. Hinton and R. S. Zemel 1994, "Autoencoders, minimum description length and Helmholtz free energy," in *Neural Info. Processing Syst.* **6**, eds. J. D. Cowan *et al.* (Morgan Kaufmann).
2. D. E. Rumelhart and D. Zipser 1985, "Feature discovery by competitive learning," *Cognitive Sci.* **3**, 75–112.
3. E. Oja 1982, "A simplified neuron model as a principle component analyzer," *J. Math. Biol.* **15**, 267–273.
4. E. Oja 1989, "Neural networks, principal components, and subspaces," *Int. J. Neural Systems* **1**, 61–68.
5. E. Oja, H. Ogawa and J. Wangviwattana 1991, "Learning in nonlinear constrained Hebbian networks," in *Artificial Neural Networks,* eds. T. Kohonen *et al.*, 385–390.
6. L. Xu 1991, "Least MSE reconstruction for self-organization: (I) Multi-layer neural nets and (II) further theoretical and experimental studies on one layer nets," in *Proc. Int. Joint Conf. on Neural Networks-1991-Singapore*, 2363–2373.
7. L. Xu 1993, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks* 6, 627–648.
8. L. Xu 1994, "Theories for unsupervised learning: PCA and its nonlinear extensions," in *Proc. of 1994 IEEE Int. Conf. on Neural Networks* **II**, 1252–1257.
9. L. Xu 1994, "Beyond PCA learnings: From linear to nonlinear and from global representation to local representation," in *1994 Proc. Int. Conf. on Neural Information Processing*, 943–949.
10. A. Sudjianto and M. H. Hassoun 1995, "Statistical basis of nonlinear Hebbian learning and application to clustering," *Neural Networks* **8**, preprint.
11. C. Fyfe and R. Baddeley 1995, "Non-linear data structure extraction using simple Hebbian network," *Biological Cybernetics* **72**, 533–541.
12. J. L. Shapiro and A. Prugel-Bennett 1994, "Non-linear statistical analysis and self-organizing Hebbian networks," *Adv. Neural Info. Processing Syst.* **6**, eds. J. D. Cowan *et al.* (Morgan Kaufmann), pp. 404–414.
13. H. Barlow 1989, "Unsupervised learning," *Neural Comp.* **1**, 295–311.
14. G. Deco and W. Brauer 1995, "Redundancy reduction with information-preserving nonlinear maps," *Neural Networks* **4**, 525–535.
15. G. Deco and L. Parra 1995, "Non-linear feature extraction by redundancy reduction in an unsupervised neural network," preprint.
16. G. Deco and B. Schürmann 1995, "Learning time series evolution by unsupervised extraction of correlations," *Phys. Rev. E* **51**, 1780–1790.
17. G. Deco and D. Obradovic 1994, "Decorrelated Hebbian learning for clustering and function approximization," *Neural Comp.* 6, 401–405.
18. J. Atick and A. Redlich 1992, "What does the

retina know about natural scenes," *Neural Comp.* **4**, 196–210,
19. A. Redlich 1993, "Redundancy reduction as a strategy for unsupervised learning," *Neural Comp.* **5**, 289–304.
20. A. Redlich 1993, "Supervised factorial learning," *Neural Comp.* **5**, 750–766.
21. A. J. Bell and T. J. Sejnowski 1995, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Comp.* **7**, 1129–1159.
22. P. Földiak 1990, "Forming sparse representations by local anti-Hebbian learning," *Biol. Cybern.* 64, 165–170.
23. E. Saund 1995, "A multiple causes mixture model for unsupervised learning," *Neural Comp.* **7**, 51–71.
24. P. Dayan and R. S. Zemel 1995, " Competition and multiple cause model," *Neural Comp.* **7**, 565–579.
25. P. Dayan, G. E. Hinton, R. M. Neal and R. S. Zemel 1995, "The Helmholtz machine," *Neural Comp.* **7**, 889–904.
26. J. Schmidhuber 1992, "Learning factorial codes by predictability minimization," *Neural Comp.* **4**, 863–879.
27. R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton 1991, "Adaptive mixtures of local experts," *Neural Comp.* **3**, 79–87.
28. R. Linsker 1988, "Self-organization in perceptron network," *IEEE Computer* **21**, 105–117.
29. H. Bourlard and Y. Kamp 1988, "Auto-association by multilayer perceptron and singular value decomposition," *Biol. Cybern.* **59**, 291–294.
30. S. Haykin 1995, *Neural Networks: A Comprehensive Foundation* (Macmillan College Publishing Company, New York).
31. P. Baldi and K. Hornik 1989, "Neural networks for principal component analysis: Learning from examples without local minima," *Neural Networks* **2**, 53–58.
32. F. Palmieri 1994, "Hebbian learning and self-association in nonlinear neural network," in *Proc. IEEE Internat Conf. on Neural Networks*, 1258–1261.
33. J. Karhunen and J. J. Joutsensalo 1994, "Representation and separation of signals using nonlinear PCA type Learning," *Neural Networks* **7**, 113–127.
34. J. Karhunen and J. J. Joutsensalo 1995, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks* **8**, 549–562.
35. J.-P Nadal and N. Parga 1994, "Nonlinear neurons in the low-noise limit: A factorial code maximizes information transfer," *Network: Comp. in Neural Systems* **5**, 565–581.
36. J. J. Atick 1992, "Could information theory provide an ecological theory of sensory processing?" *Network: Comp. in Neural Syst.* **3**, 213–251.
37. J. A. Reggia, Y. Peng and P. Bourret 1991, "Recent applications of competitive mechanisms," in *Neural Networks: Advances and Applications,* ed. E. Gelenbe (Elsevier Science Publishers, North-Holland), pp. 33–62.
38. B. A. Pearlmutter and G. E. Hinton 1986, "G-maximization: An unsupervised learning procedure for discovering regularities," in *Networks for Computing: Am. Inst. Physics Conf. Proc.* **151**, ed. J. S. Denker, 333–338.
39. R. H. White 1992, "Competitive Hebbian learning: Algorithm and demonstrations," *Neural Networks* **5**, 261–275.