DIFFERENTIAL LEARNING AND RANDOM WALK MODEL

Seungjin Choi

Department of Computer Science and Engineering, POSTECH, Korea
seungjin@postech.ac.kr

ABSTRACT

This paper presents a learning algorithm for differential decorrelation, the goal of which is to find a linear transform that minimizes the concurrent change of associated output nodes. First the algorithm is derived from the minimization of the objective function which measures the differential correlation. Then we show that the differential decorrelation learning algorithm can also be derived in the framework of maximum likelihood estimation of a linear generative model with assuming a random walk model for latent variables. Algorithm derivation and local stability analysis are given with a simple numerical example.

1. INTRODUCTION

The Hebbian rule has been widely used in the domain of unsupervised learning where no target value is available. It is a correlation learning that is based on the hypothesis of Hebb [7] which states that the concurrent activation of neurons increases the strength of connection between them. The Hebbian rule was shown to be an output variance maximizer. In contrast to the Hebbian rule, the anti-Hebbian rule updates the synaptic weights in such a way that cross-correlations between associated nodes are minimized. Hence, it is an output variance minimizer and decorrelates associated output variables.

As an alternative to the Hebbian rule, the differential Hebbian rule was studied in [8]. The motivation of the differential Hebbian rule is that concurrent change, rather than just concurrent activation, more accurately captures the *concomitant variation* that is central to inductively inferred functional relationships [8]. Under the assumption of Martingale processes, the differential Hebbian rule was shown to be a covariance learning rather than a correlation learning. The differential anti-Hebbian rule is a direct modification of the anti-Hebbian and updates the synaptic weights in a linear feedback network in such a way that the concurrent change of neurons is minimized. In this sense one can argue that the differential Hebbian rule is an output differential variance minimizer. It was first proposed in [4] and its generalization with adopting a nonlinear function was applied to the problem of independent component analysis [4].

A natural gradient algorithm for differential decorrelation was recently developed in a fully connected linear feedback network [5]. It was derived from the minimization of the objective function which measures the differential correlation. In this paper we first consider a linear feedforward network and derive a natural gradient differential decorrelation algorithm using the same objective function in [5]. Then we provide more general framework for differential learning. We show that the differential decorrelation learning algorithm can also be derived in the framework of maximum likelihood estimation of a linear generative model with assuming a random walk model for latent variables. This gives you more theoretical insight to the method of differential learning. Algorithm derivation and local stability analysis are presented.

2. DIFFERENTIAL LEARNING

2.1. Differential Anti-Hebbian Rule

Let us consider a linear feedback network (without self-feedback connections) whose ith output node $y_i(t)$ is described by

$$y_i(t) = x_i(t) + \sum_{j \neq i} w_{ij} y_j(t). \tag{1}$$

The concurrent change of two output neurons $y_i(t)$ and $y_j(t)$ is measured by the differential correlation defined by $E\{y_i'(t) y_j'(t)\}$ where

$$y_i'(t) = \frac{dy_i(t)}{dt}, \tag{2}$$

or its discrete-time counterpart is $y_i'(t) = y_i(t) - y_i(t-1)$ which is its first-order approximation. The differential variance is also defined by $E\{y_i'^2(t)\}$ which is the variance of differentiated variable (y_i is assumed to be a zero-mean random variable).

The differential anti-Hebbian rule is a direct modification of the anti-Hebbian rule. As a differential variance minimizer, the differential anti-Hebbian rule [4] has the updating equation that has the form

$$w_{ij}(t+1) = w_{ij}(t) - \eta_t y_i'(t) y_j'(t), \quad \text{for } i \neq j, \tag{3}$$

where $\eta_t > 0$ is the learning rate.

2.2. Adaptive Differential Decorrelation: A Natural Gradient Algorithm

Let us consider a linear feedforward network whose output vector $\boldsymbol{y}(t) \in \mathbb{R}^n$ is described by

$$\boldsymbol{y}(t) \quad = \quad \boldsymbol{W}\boldsymbol{x}(t), \tag{4}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the input vector to the network and $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is the synaptic weight matrix.

In order to derive an adaptive decorrelation algorithm which minimizes the differential correlation between output nodes, we consider the following objective function:

$$\begin{aligned} \mathcal{J}_1(\boldsymbol{W}) \quad = \quad & \frac{1}{2} \left\{ \sum_{i=1}^{n} \log E\{y_i'^2(t)\} \right. \\ & \left. - \log \det \left(E\left\{ \boldsymbol{y}'(t) \boldsymbol{y}'^T(t) \right\} \right) \right\}. \end{aligned} \tag{5}$$

The objective function (5) is a non-negative function which takes minima if and only if $E\{y_i'(t)y_j'(t)\} = 0$, for $i, j = 1, \ldots, n$, $i \neq j$. The objective function (5) a direct consequence of the Hadamard's inequality. In fact the objective function (5) is a slight modification of the one that was used in [9]. Output values are simply replaced by their differentiated values.

In order to find a minimum for the objective function (5), we employ the natural gradient learning method that was shown to be useful in [1, 6].

We calculate the total differential $d\mathcal{J}_1(\boldsymbol{W})$ due to the change $d\boldsymbol{W}$

$$\begin{aligned} d\mathcal{J}_1(\boldsymbol{W}) &= \mathcal{J}_1(\boldsymbol{W} + d\boldsymbol{W}) - \mathcal{J}_1(\boldsymbol{W}) \\ &= \frac{1}{2} d \left\{ \sum_{i=1}^{n} \log E\{y_i'^2(t)\} \right\} \\ &\quad - \frac{1}{2} d \left\{ \log \det \left(E\{\boldsymbol{y}'(t)\boldsymbol{y}'^T(t)\} \right) \right\}, \\ &= \sum_{i=1}^{n} \frac{E\{y_i'(t) dy_i'(t)\}}{E\{y_i'^2(t)\}} - \operatorname{tr}\{(\boldsymbol{W}^{-1} d\boldsymbol{W}\} \\ &\quad - \frac{1}{2} d \left\{ \log \det \boldsymbol{C}_{x'x'}(t) \right\}, \end{aligned} \tag{6}$$

where $\boldsymbol{C}_{x'x'}(t)$ is the differential correlation matrix of $\boldsymbol{x}(t)$ defined by

$$\boldsymbol{C}_{x'x'}(t) = E\left\{ \boldsymbol{x}'(t)\boldsymbol{x}'^T(t) \right\}. \tag{7}$$

Define a modified differential matrix $d\boldsymbol{V}$ as

$$d\boldsymbol{V} = d\boldsymbol{W}\boldsymbol{W}^{-1}. \tag{8}$$

We also define a differential variance matrix $\boldsymbol{\Lambda}(t)$ which is a diagonal matrix whose ith diagonal element $\lambda_i(t)$ is estimated by

$$\lambda_i(t) = (1 - \delta)\lambda_i(t-1) + \delta y_i'^2(t), \tag{9}$$

for some small δ (say, $\delta = 0.01$).

With these defined matrices, the total differential $d\mathcal{J}_1(\boldsymbol{W})$ can be written as

$$\begin{aligned} d\mathcal{J}_1(\boldsymbol{W}) &= E\{\boldsymbol{y}'^T(t)\boldsymbol{\Lambda}^{-1}(t) d\boldsymbol{V} \boldsymbol{y}'(t)\} + \operatorname{tr}\{d\boldsymbol{V}\} \\ &\quad + d\left\{ \log \det \boldsymbol{C}_x(t) \right\}. \end{aligned} \tag{10}$$

Hence, the gradient of the objective function (5) with respect to the modified differential matrix $d\boldsymbol{V}$ is given by

$$\frac{d\mathcal{J}_1(\boldsymbol{W})}{d\boldsymbol{V}} = E\left\{ \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^T(t) \right\} - \boldsymbol{I} \tag{11}$$

The stochastic gradient descent method leads to the updating rule for $\boldsymbol{V}$ that has the form

$$\boldsymbol{V}(t+1) = \boldsymbol{V}(t) + \eta_t \left\{ \boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^T(t) \right\}, \tag{12}$$

where $\eta_t > 0$ is the learning rate. It follows from the definition (8) that the learning algorithm for $\boldsymbol{W}$ is given by

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t \left\{ \boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^T(t) \right\} \boldsymbol{W}(t), \tag{13}$$

which is a natural gradient algorithm for adaptive differential decorrelation. The algorithm (13) is a differential version of the equivariant nonstationary source separation algorithm in [6].

We can also consider a fully connected feedback network where the input-output relation is given by

$$\boldsymbol{y}(t) = (\boldsymbol{I} - \boldsymbol{W})^{-1} \boldsymbol{x}(t). \tag{14}$$

In a similar manner, the natural gradient algorithm which finds a minimum solution to the objective function (5), can be derived [5]. It has the form

$$\begin{aligned} \Delta\boldsymbol{W}(t) &= \boldsymbol{W}(t+1) - \boldsymbol{W}(t) \\ &= \eta_t \{\boldsymbol{I} - \boldsymbol{W}(t)\} \left\{ \boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^T(t) \right\} \end{aligned} \tag{15}$$

Remark

The learning algorithm (13) can also be written as

$$\Delta\boldsymbol{W}(t) = \eta_t \boldsymbol{\Lambda}^{-1}(t) \left\{ \boldsymbol{\Lambda}(t) - \boldsymbol{y}'(t)\boldsymbol{y}'^T(t) \right\} \boldsymbol{W}(t). \tag{16}$$

Thus this differential decorrelation algorithm has properties inherited from the nonholonomic ICA algorithms [2].

3. RANDOM WALK MODEL

Let's consider a linear generative model where the observation vector $\boldsymbol{x}(t) \in \mathbb{R}^n$ is modelled as a linear transform of latent variables, i.e.,

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t), \tag{17}$$

where $\boldsymbol{s}(t)$ is an n-dimensional vector, each element of which is latent variable. Latent variables are assumed to be spatially independent. In source separation or independent component analysis (ICA), latent variables are called sources.

Now we introduce a random walk model for latent variables which is a simple Markov chain, i.e.,

$$s_i(t) = s_i(t-1) + \epsilon_i(t), \tag{18}$$

where the innovation $\epsilon_i(t)$ is assumed to have Gaussian distribution with zero mean and variance σ_i^2, i.e., $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$.. In addition, innovations $\{\epsilon_i\}$ are assumed to be mutually independent.

Let us consider the latent variables $s_i(t)$ over N-point time block. Define the vector $\boldsymbol{s}_i$ as

$$\boldsymbol{s}_i = [s_i(0), \ldots, s_i(N-1)]^T. \tag{19}$$

Then the joint probability density function of $\boldsymbol{s}_i$ can be written as

$$\begin{aligned} p_i(\boldsymbol{s}_i) &= p_i\left(s_i(0), \ldots, s_i(N-1)\right) \\ &= \prod_{t=0}^{N-1} p_i(s_i(t)|s_i(t-1)), \end{aligned} \tag{20}$$

where $s_i(t) = 0$ for $t < 0$.

The conditional probability density of $s_i(t)$ given its past samples can be written as

$$p_i(s_i(t)|s_i(t-1)) = q_i(\epsilon_i(t)), \tag{21}$$

where

$$q_i(\epsilon_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{ -\frac{\epsilon_i^2}{2\sigma_i^2} \right\}. \tag{22}$$

Combining (20) and (21) leads to

$$\begin{aligned} p_i(\boldsymbol{s}_i) &= \prod_{t=0}^{N-1} q_i(\epsilon_i(t)) \\ &= \prod_{t=0}^{N-1} q_i\left(s_i'(t)\right), \end{aligned} \tag{23}$$

where $s_i'(t) = s_i(t) - s_i(t-1)$ which is the first-order approximation of differentiation.

Take the statistical independence of latent variables and (23) into account, then we can write the joint density $p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$ as

$$\begin{aligned} p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n) &= \prod_{i=1}^{n} p_i(\boldsymbol{s}_i) \\ &= \prod_{t=0}^{N-1} \prod_{i=1}^{n} q_i\left(s_i'(t)\right). \end{aligned} \tag{24}$$

The factorial model given in (24) will be used as a optimization criterion to derive the proposed algorithm.

4. MAXIMUM LIKELIHOOD ESTIMATION

4.1. Algorithm Derivation

Here we show that the differential decorrelation algorithm (13) can be derived using the factorial model (24) in the framework of maximum likelihood estimation.

Denote a set of observation data by $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i = \{x_i(0), \ldots, x_i(N-1)\}$. Then the normalized log-likelihood is given by

$$\begin{aligned} &\frac{1}{N} \log p(\mathcal{X}|\boldsymbol{A}) \\ &= -\log|\det \boldsymbol{A}| + \frac{1}{N} \log p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n) \\ &= -\log|\det \boldsymbol{A}| + \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \log q_i(s_i'(t)). \end{aligned} \tag{25}$$

Let us denote the inverse of $\boldsymbol{A}$ by $\boldsymbol{W} = \boldsymbol{A}^{-1}$. The estimate of latent variables is denoted by $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$. With these defined variables, the objective function (that is the negative normalized log-likelihood) is given by

$$\begin{aligned} \mathcal{J}_2 &= -\log|\det \boldsymbol{W}| - \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \log q_i(y_i'(t)) \\ &= -\log|\det \boldsymbol{W}| + \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \left[\frac{[y_i'(t)]^2}{2\sigma_i^2} + \frac{1}{2} \log 2\pi\sigma_i^2 \right], \end{aligned}$$

where s_i is replaced by its estimate y_i. Neglecting the terms irrelevant to $\boldsymbol{W}$ leads to

$$\mathcal{J}_3 = -\log|\det \boldsymbol{W}| + \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \frac{[y_i'(t)]^2}{2\sigma_i^2}. \tag{26}$$

If we assume that we have sufficient number of samples (i.e., N is sufficiently large), then the ensemble average can be approximated by the sample average. In addition, for on-line learning, we replace the ensemble average by its instantaneous value. Under this, the objective function (26) is simplified as

$$\mathcal{J}_3 = -\log|\det \boldsymbol{W}| + \sum_{i=1}^{n} \frac{[y_i'(t)]^2}{2\sigma_i^2}. \tag{27}$$

We employ a natural gradient learning method to derive an updating rule to find a minimum of the objective function (27). The derivation is carried out in a similar manner that was used in Section 2.2.

We calculate the total differential $d\mathcal{J}_3(\boldsymbol{W})$ due to the change $d\boldsymbol{W}$

$$\begin{aligned} d\mathcal{J}_3(\boldsymbol{W}) &= d\left\{ \sum_{i=1}^{n} \frac{[y_i'(t)]^2}{2\sigma_i^2} \right\} - d\left\{\log|\det \boldsymbol{W}|\right\} \\ &= \sum_{i=1}^{n} \frac{y_i'(t) dy_i'(t)}{\sigma_i^2} - \mathrm{tr}\left\{d\boldsymbol{W}\boldsymbol{W}^{-1}\right\} \\ &= \boldsymbol{y}'^T(t) \boldsymbol{\Lambda}^{-1} d\boldsymbol{V} \boldsymbol{y}'(t) - \mathrm{tr}\left\{d\boldsymbol{V}\right\}, \end{aligned} \tag{28}$$

where $\boldsymbol{\Lambda}$ is the differential variance matrix that is diagonal and its ith diagonal element is estimated by (9). The nonholonomic basis $d\boldsymbol{V}$ is defined in (8).

A interesting point here is that the differential $d\mathcal{J}_3(\boldsymbol{W})$ is identical to the differential $d\mathcal{J}_1(\boldsymbol{W})$ even though objective functions $\mathcal{J}_1(\boldsymbol{W})$ and $\mathcal{J}_3(\boldsymbol{W})$ are slightly different. Thus a natural gradient learning algorithm which finds a minimum solution to the objective function (27) has the form

$$\Delta \boldsymbol{W}(t) = \eta_t \left\{ \boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t) \boldsymbol{y}'(t) \boldsymbol{y}'^T(t) \right\} \boldsymbol{W}(t). \tag{29}$$

4.2. Differential ICA Algorithm

In general, the objective function (27) has the form

$$\mathcal{J}_4 = -\log|\det \boldsymbol{W}| - \sum_{i=1}^{n} \log q_i(y_i'(t)), \tag{30}$$

where $q_i(\cdot)$ is the probability density function for $\epsilon_i(t)$. The differential decorrelation algorithm (29) is derived from assuming $q_i(\cdot)$ being Gaussian. Allowing $q_i(\cdot)$ to have a general distribution, leads to the differential version of ICA algorithm that has the form

$$\Delta \boldsymbol{W}(t) = \eta_t \left\{ \boldsymbol{I} - \varphi(\boldsymbol{y}'(t)) \boldsymbol{y}'^T(t) \right\} \boldsymbol{W}(t), \tag{31}$$

where

$$\varphi(\boldsymbol{y}'(t)) = \left[\varphi_1(y_1'(t)), \ldots, \varphi_n(y_n'(t))\right] \tag{32}$$

and

$$\varphi_i(y_i') = -\frac{d \log q_i(y_i')}{dy_i'}. \tag{33}$$

Remarks

- The algorithm (31) was derived in an ad hoc manner in [4]. Here we show that the algorithm (31) can be derived in the framework of maximum likelihood estimation with a random walk model.
- The algorithm (31) can be viewed as a special case of temporal ICA algorithm [3] where the spatiotemporal generative model was employed.

5. LOCAL STABILITY ANALYSIS

In this section, we show that the stationary points of (29) are locally stable. To this end we calculate the Hessian $d^2\mathcal{J}_3$ in terms of the modified differential matrix $d\boldsymbol{V}$ and show that it is positive.

For shorthand notation, we omit the time index t in the following analysis. The Hessian $d^2\mathcal{J}_3$ is computed as

$$\begin{aligned} d^2\mathcal{J}_3 &= E\left\{\boldsymbol{y}'^T d\boldsymbol{V}^T \boldsymbol{\Lambda}^{-1} d\boldsymbol{V}\boldsymbol{y}' + \boldsymbol{y}'^T \boldsymbol{\Lambda}^{-1} d\boldsymbol{V} d\boldsymbol{V}\boldsymbol{y}'\right\} \\ &= E\left\{\boldsymbol{y}'^T d\boldsymbol{V}^T \boldsymbol{\Lambda}^{-1} d\boldsymbol{y}'\right\} + E\left\{\boldsymbol{y}'^T \boldsymbol{\Lambda}^{-1} d\boldsymbol{V} d\boldsymbol{y}'\right\} \\ &= \sum_{i,j} \frac{\lambda_i}{\lambda_j}(dv_{ji})^2 + \sum_{i,j} dv_{ij} dv_{ji}, \end{aligned} \tag{34}$$

where the statistical expectation is taken at the solution which satisfies the condition $E\{y_i' y_j'\} = 0$ for $i \neq j$.

For a pair (i, j), $i \neq j$, the summand in the first term in (34) can be rewritten as

$$\begin{aligned} &\frac{\lambda_i}{\lambda_j}(dv_{ji})^2 + \frac{\lambda_j}{\lambda_i}(dv_{ij})^2 + 2dv_{ij}dv_{ji} \\ &= \begin{bmatrix} dv_{ij} & dv_{ji} \end{bmatrix} \begin{bmatrix} \frac{\lambda_j}{\lambda_i} & 1 \\ 1 & \frac{\lambda_i}{\lambda_j} \end{bmatrix} \begin{bmatrix} dv_{ij} \\ dv_{ji} \end{bmatrix}, \end{aligned} \tag{35}$$

which is always non-negative. Hence $d^2\mathcal{J}_3$ is always positive. Therefore the algorithm (29) is locally stable around the solutions.

6. A NUMERICAL EXAMPLE

A simple numerical example is given to evaluate the validity of the differential decorrelation algorithm (29). Three independent colored Gaussian random variables is linearly mixed to generate the observation vector $\boldsymbol{x}(t)$ with a differential correlation matrix

$$\boldsymbol{C}_{x'x'} = \begin{bmatrix} 8.367 & 3.274 & 2.448 \\ 3.274 & 1.349 & 0.943 \\ 2.448 & 0.943 & 0.790 \end{bmatrix}. \tag{36}$$

We applied the algorithm (29) with the differential variance matrix being fixed as $\boldsymbol{\Sigma} = \boldsymbol{I}$ and with a constant learning rate, $\eta = .001$. Fig. 1 shows the evolution of $E\{y_1'(t)y_2'(t)\}$ as an example. Other differential correlations were also suppressed in a similar fashion.

7. DISCUSSION

In this paper we have presented a natural gradient learning algorithm for differential decorrelation, the goal of which is to minimize the correlation between differentiated random variables. We showed that the differential decorrelation algorithm could be derived from learning a linear generative model by the maximum likelihood estimation under a random walk model. We also discussed a differential version of the natural gradient ICA algorithm and showed that it could also be derived under the random walk model. The differential correlation algorithm (29) or the differential ICA algorithm (31) could be generalized by adopting higher-order differentiation. This generalization is currently under investigation.

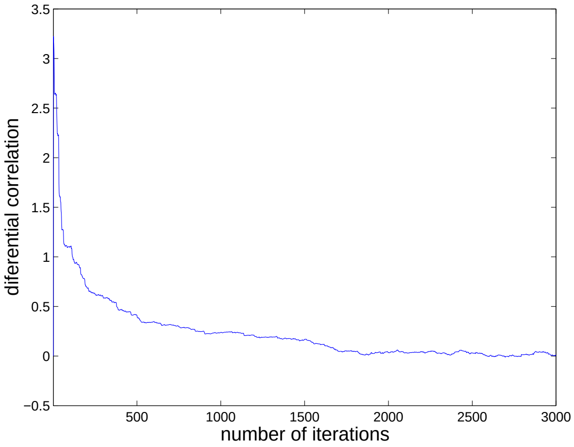

Fig. 1. Differential correlation between y_1 and y_2.

8. ACKNOWLEDGMENTS

This work was supported by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and an International Cooperative Research Project, by KOSEF-ARIEL, and by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

9. REFERENCES

[1] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[2] S. Amari, T. P. Chen, and A. Cichocki. Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Computation*, 12(6):1463–1484, 2000.

[3] H. Attias and C. E. Schreiner. Blind source separation and deconvolution: The dynamic component analysis algorithms. *Neural Computation*, 10:1373–1424, 1998.

[4] S. Choi. Differential Hebbian-type learning algorithms for decorrelation and independent component analysis. *Electronics Letters*, 34(9):900–901, 1998.

[5] S. Choi. Adaptive differential decorrelation: A natural gradient algorithm. In *Proc. ICANN*, pages 1168–1173, Madrid, Spain, Aug. 2002.

[6] S. Choi, A. Cichocki, and S. Amari. Equivariant nonstationary source separation. *Neural Networks*, 15(1):121–130, 2002.

[7] D. O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

[8] B. Kosko. Differential Hebbian learning. In *Proc. American Institute of Physics: Neural Networks for Computing*, pages 277–282, 1986.

[9] K. Matsuoka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.