Vol. 40 No. 10 Transactions of Information Processing Society of Japan Oct. 1999

Regular Paper

Principal Component Analysis by Entropy-Likelihood Optimization

MAHDAD N. SHIRAZI,† FERDINAND PEPER† and HIDEFUMI SAWAI†

This paper proposes a principal component analysis (PCA) criterion whose optimization yields the principal eigenvectors of the data correlation matrix as well as the associated eigenvalues. The corresponding learning algorithms are deduced for the unsupervised learning of one-layer linear neural networks. The part of the algorithm that estimates the principal eigenvectors turns out to be a version of the Sanger's generalized Hebbian algorithm (GHA) that enjoys adaptive learning rates and fast convergence. The proposed criterion differs with the standard PCA criteria, such as *Maximum Variance* and *Minimum MSE*, in that a) optimization of standard criteria results only in the principal eigenvectors, b) their corresponding learning algorithm, namely GHA algorithm, has a fixed learning rate. Simulation results illustrate the fast convergence of the derived algorithm.

1. Introduction

Principal component analysis is a widely used statistical technique for reducing the dimensionality of a data set, while retaining as much variation of the data as possible. In PCA, data are mapped via a linear transformation that packs most of the signal energy in the first few components and transforms correlated input data into a set of statistically decorrelated features usually ordered according to decreasing information content. It originally appeared in multivariate statistics literature and has then been encountered in a wide range of applications including data compression[1] and face recognition[2].

The past decade has shown an increasing effort to implement PCA by means of neural networks. The motivation behind such an effort is to relax the wide-sense stationarity assumption, which underlies optimality of the PCA method, and afford for an adaptive extraction of principal components (PC's) which is able to account for slow variations of source statistics.

In an influential paper[3] Oja introduced a linear neuron model with a constrained hebbian-type learning rule and proved the convergence of the neuron's weight vector to the first principal component. Sanger[4] extended the algorithm to the multi-neuron case, following a procedure similar to Gram-Schmidt orthonormalization, and showed the algorithm's ability to estimate principal components in a decreasing order of eigenvalues. Kung and Diamantaras developed an algorithm which recursively, rather than simultaneously, computes the principal components[5]. The motivation behind the algorithm is the need to extract the principal components when the number of required PC's is not known a priori. However, with respect to its adaptivity to the data's statistical variations, the algorithm is not as effective as PCA algorithms such as GHA which update all weight vectors simultaneously. Several other PCA algorithms have since been reported in the literature, among which we mention[6),7].

Recently, an optimization-based approach to PCA has been getting attention in the neural network society because it gives a mathematically sound formulation of the problem and helps to understand the properties of the corresponding learning algorithms. Indeed, many of the PCA learning rules and their modular versions[8] are iterative algorithms which estimate the principal eigenvectors by optimizing standard PCA criterions[9),10].

In this paper, we propose a PCA criterion, hereafter referred to as Entropy Likelihood (EL) criterion, which can be considered as a crossbreeding between the likelihood function and the differential entropy of the data's marginal probability density function (pdf). As compared to the *maximum variance* and *minimum MSE* criterions , optimization of the proposed criterion results in a couple of learning algorithms which extract not only the principal eigenvectors but also the associated eigenvalues. The derived algorithm is a variation of Sanger's GHA which enjoys adaptive learning rates. This feature, which takes root in the

† Telecommunications Research Laboratory

proposed criterion rather than being added in an ad hoc way, leads to a speed-up in the convergence of the GHA algorithm.

In Section 2, we first establish the correspondence between the solution optimizing EL and the PCA solution. After that follows the derivation of the stochastic gradient algorithms which optimize the criterion. Simulations in Section 3 illustrate the effectiveness of the adaptive learning rates in speeding up the convergence of the GHA algorithm. The paper finishes with concluding remarks in Section 4.

2. Entropy-Likelihood Learning

Let the input x be an N-dimensional random vector generated by the zero-mean multivariate Gaussian distribution

$$f(x;\Sigma_x) = \frac{1}{(2\pi)^{N/2}|\Sigma_x|^{1/2}} \exp\left\{-\frac{1}{2}x^T\Sigma_x^{-1}x\right\}$$

where Σ_x stands for the data covariance matrix which is assumed to be nonsingular with distinct eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. We believe that the assumption that the distribution is Gaussian does not impose a significant loss of generality because a) principal components are defined by the second-order statistics and do not depend on higher-order statistics, which characterize non-Gaussian data, b) the derived learning algorithm is identical, except for the multiplicative learning rate, to the Sanger GHA algorithm which does not assume Gaussianity.

2.1 Single-unit EL learning

A single linear neuron projects the N-dimensional input space into the one-dimensional subspace spanned by it's weight vector w. The probability density function of the neuron output $y = w^Tx$ can be written as

$$f(w^Tx;\nu_w) = \frac{1}{(2\pi\nu_w)^{1/2}} \exp\left\{-\frac{(w^Tx)^2}{2\nu_w}\right\} \tag{1}$$

where ν_w denotes the variance of the neuron output.

Consider the EL criterion defined as

$$J(w,\nu_w) = E_X[\log f(w^Tx;\nu_w)]. \tag{2}$$

For a given normalized weight vector w, the EL criterion is asymptotically equivalent to the log-likelihood function associated with the marginal pdf. More specifically, for a training set $X = \{x_1, x_2, \cdots, x_n\}$ of statistically independent and identically distributed samples from the input density function $f(x;\Sigma_x)$ we have

$$J(w,\nu_w) = \lim_{n\to\infty} \frac{1}{n} \log \mathcal{L}(X_w;\nu_w). \tag{3}$$

Here $\mathcal{L}(X_w;\nu_w)$ denotes the likelihood function of the marginal pdf (1) and $X_w = \{w^Tx_1, w^Tx_2, \cdots, w^Tx_n\}$. The Maximum Likelihood (ML) estimate for the variance of the marginal pdf can thus be obtained by maximizing the EL criterion with respect to the parameter ν_w, i.e.,

$$\nu_w^{ML} = \arg\max_{\nu_w} J(w,\nu_w). \tag{4}$$

Replacing ν_w by $w^T\Sigma_x w$, we reduce EL to the negative of the differential entropy[11] of the marginal pdf, and its maximization yields the first principal eigenvector u_1 (see Lemma 1). However, unlike the differential entropy, optimization of the EL criterion, which has both w and ν_w as its arguments, yields not only u_1 but also the ML estimate of ν_{u_1}.

To optimize EL one can start with some initial estimate of these parameters and then update them iteratively along the gradient direction of the objective function. This results in the gradient algorithms

$$w(k+1) = w(k) - \eta^w(k)\{\nabla_w[J(w,\nu_w) + \xi(w^Tw-1)]\} \tag{5}$$

$$\nu_w(k+1) = \nu_w(k) + \eta^{\nu_w}(k)\left\{\frac{\partial}{\partial\nu_w}J(w,\nu_w)\right\} \tag{6}$$

with gradients given by

$$\nabla_w[J(w,\nu_w) + \xi(w^Tw-1)] = E_x[-\nu_w^{-1}(I - ww^T)xx^Tw] \tag{7}$$

$$\frac{\partial}{\partial\nu_w}J(w,\nu_w) = E_x\left[\frac{1}{2}\nu_w^{-2}\{(w^Tx)^2 - \nu_w\}\right]. \tag{8}$$

In accordance with the common practice, the expression for the Lagrangian multiplier at the optimum is substituted in the gradient (7).

Employing instantaneous estimates of the gradient vector (7) and the derivative function (8) yields the following stochastic gradient algorithms

$$w(k+1) = w(k) + \eta^w(k)\nu_w^{-1}(k)\{[I - w(k)w^T(k)]\}x_kx_k^Tw(k) \tag{9}$$

$$\nu_w(k+1) = \nu_w(k) + \eta^{\nu_w}(k)\nu_w^{-2}(k)\{[w^T(k)x_k]^2 - \nu_w(k)\} \tag{10}$$

which are on-line versions of the batch algo-

rithms given above. Stochastic gradient algorithms have the advantage of a low computational complexity at each iteration, and of a simple implementation in software and hardware, compared to algorithms using the true gradient[13]. We note that the learning algorithm (9) is identical to the Oja's single unit PCA algorithm[3], except for the multiplicative factor $\nu_w^{-1}(k)$. The learning algorithm (10) gives convergence to the variance of the data projected on the weight vector. This can be proved by rewriting Eq. (10) in the form of the Robbins-Monro algorithm

$$\nu(k+1) = \nu(k) - \eta(k)\xi_k^{w(k)}(\nu(k)),$$

where $\xi_k^{w(k)}(\nu(k)) = \nu_w(k) - [w^T(k)x_k]^2$, and using the fact that Robbins-Monro algorithm converges in the mean-square sense to the root θ of the regression function

$$\mu^w(\nu) = E_{\xi^w}[\xi^w(\nu)] = \int_{\Omega_{\xi^w}} \xi^w(\nu) f(\xi^w|\nu) d\xi^w,$$

provided that $\Sigma_{k=1}^{\infty}\eta(k) = \infty, \Sigma_{k=1}^{\infty}\eta(k)^2 < \infty$, and the regression function $\mu^w(\nu)$ satisfies the conditions

$$M_1(\nu-\theta)^2 \le \mu^w(\nu)(\nu-\theta) \le M_2(\nu-\theta)^2$$

and

$$Var[\xi^w(\nu)] = \int_{\Omega_{\xi^w}} [\xi^w - \mu^w(\nu)]^2 f(\xi^w|\nu) d\xi^w \le \alpha^2 < \infty,$$

where $0 < M_1 \le M_2 < \infty$[12].

Since $\mu^w(\nu) = E_{\xi^w}[\xi^w(\nu)] = (\nu - w^T\Sigma_x w)$, the first condition is obviously true for $M_1 = M_2 = 1$ and the second condition is true because $Var[\xi^w(\nu)] = Var[w^T x x^T w] = Var[y^2]$ and $Var[y^2]$ is a finite number due to the finite length constraint on w. Thus the algorithm converges in the mean-square sense to $u_1\Sigma_x u_1$ as $k \to \infty$.

2.2 Multi-unit EL learning

Consider a neural network with M linear neurons ($1 \le M \le N$). The network projects the N-dimensional input space into M one-dimensional subspaces spanned by the weight vectors $w_1, \cdots, w_M$ of the network. The EL criterion defined for the single-unit case can be employed for each output neuron in the multi-unit case. Regarding the variance of the ith neuron output, the ML estimate is given by

$$\nu_{w_i}^{ML} = \arg\max_{\nu_{w_i}} J(w_i, \nu_{w_i}). \quad (11)$$

Without imposing additional constraints, optimization of EL leads to the same solution for all the weight vectors. To induce the weight vectors to different solutions, hierarchical orthonormality constraints are imposed on the weight vectors: each weight vector is constrained to have unit length and to be orthogonal to the weight vectors of the neurons with lower indices. We then get the following result.

Lemma 1. The first M principal eigenvectors of the input data emerge as the optimal solution of the EL minimization problem with respect to w_i ($i = 1, \cdots, M$) subject to the hierarchical orthonormality constraints, i.e., $||w_i|| = 1$ and $w_i^T w_j = 0$ for $j < i$.

Proof: For w_i^* to be a local minimum of $J(w_i, \nu_{w_i})$ subject to the hierarchical orthonormality constraints, it is necessary that (w_i^*, ξ_i^*) bc a stationary point of the Lagrange function

$$L(w_i, \xi_i|\nu_{w_i}) = J(w_i, \nu_{w_i}) + \sum_{j=1}^{i} \xi_{ij}(w_i^T w_j - \delta_{ij})$$

where $\xi_i = (\xi_{i1}, \cdots, \xi_{ii})^T$ is the vector of Lagrange multipliers. This yields for $i = 1, \cdots, M$

$$\nabla_{w_i} L(w_i, \xi_i|\nu_{w_i})|_{(w_i^*, \xi_i^*)} = -\nu_{w_i}^{-1}\Sigma_x w_i^* + \sum_{j=1}^{i-1} \xi_{ij}^* w_j^* + 2\,\xi_{ii}^* w_i^* = 0 \quad (12)$$

$$\frac{\partial}{\partial \xi_{ij}} L(w_i, \xi_i|\nu_{w_i})|_{(w_i^*, \xi_i^*)} = {w_i^*}^T w_j^* - \delta_{ij} = 0 \quad \text{for} \quad j = 1, \cdots, i. \quad (13)$$

Multiplying Eq. (12) by ${w_j^*}^T$ from the left and using the constraint given in Eq. (13), we get $\xi_{ii}^* = \frac{1}{2}\nu_{w_i}^{-1}{w_i^*}^T\Sigma_x w_i^*$ and $\xi_{ij}^* = \nu_{w_i}^{-1}{w_j^*}^T\Sigma_x w_i^*$ for $j \ne i$. Substituting these back into Eq. (12), yields $\Sigma_x w_i^* = \sum_{j=1}^{i}({w_j^*}^T\Sigma_x w_i^*)w_j^*$.

Using the hierarchical orthonormality constraints, it can be shown by induction that $\Sigma_x w_i^* = ({w_i^*}^T\Sigma_x w_i^*)w_i^*$, for $i = 1, \cdots, M$. Therefore, w_i^* is a normalized eigenvector of Σ_x with the associated eigenvalue being equal to the variance of the data projected on it. Now, for a normalized eigenvector u_i, EL rewrites to $J(u_i, \nu_{u_i}) = -\frac{1}{2}\log(2\pi\lambda_i)$ where λ_i denotes the corresponding eigenvalue. Since eigenvalues of Σ_x are never negative, it is clear that w_1 is the eigenvector corresponding to the largest eigenvalue λ_1, w_2 is the eigenvector orthogonal to

w_1 and corresponding to the next largest eigenvalue λ_2, and so on.

By the same argument as given in Section 2.1, the first M principal eigenvectors and the variances of the data projected on them can be estimated by the following algorithms

$$w_i(k+1) = w_i(k) + \eta^{w_i}(k)\nu_{w_i}^{-1}(k) \left\{ \left[I - \sum_{j=1}^{i} w_j(k) w_j^T(k) \right] \right\} x_k x_k^T w_i(k) \tag{14}$$

$$\nu_{w_i}(k+1) = \nu_{w_i}(k) + \eta^{\nu_{w_i}}(k)\nu_{w_i}^{-2}(k) \left\{ [w_i^T(k)x_k]^2 - \nu_{w_i}(k) \right\}. \tag{15}$$

Learning algorithm (14) is identical to Sanger's GHA learning algorithm[4], except for the learning rates. Compared with the GHA algorithm, which employs a fixed step size for the learning of all the principal eigenvectors, the derived algorithm adaptively adjusts the learning rates to the spread of the data experienced in the learning of the corresponding principal eigenvectors. In this way the algorithm tries to keep the same convergence rates for all the principal eigenvectors. This feature adds to the adaptive nature of the GHA algorithm and, as is shown by simulation studies, results in improved convergence speed. The second part of the algorithm provides the information needed by the first algorithm for adaptive adjustment of learning rates and in turn is provided with the directions along which data variances should be estimated.

3. Simulations

To evaluate the efficacy of the adaptive learning rate feature of the derived EL-based algorithm against the fixed learning rate of the GHA algorithm, we used both of these algorithms to extract the first 2, 3, 4, and 8 principal eigenvectors of a data set. The training data were generated by randomly (according to a uniform distribution over image indices) selecting blocks of size 8×8 from an image having a resolution of 512×512 pixels with a dynamic range of 8 bit or 256 grey levels. **Figure 1** shows the image used for training.

To insure convergence of the algorithm for samples from various image data, we normalized samples to have unit norm on average, i.e. $E[||X'||^2] = 1$. This was achieved by preprocessing samples according to $x_i' = (x_i - \mu)/(n \times \sigma)$ where μ, σ, n, x_i, and x_i' stand for the pixel mean, pixel variance, block size, ith sample, and the normalized sample, respectively. The weight vectors w_i's were also randomly initialized by drawing their elements from a uniform distribution over $(-1, 1)$.

Fig. 1 Image used for training

Preliminary experiments indicated that finding suitable learning rates was necessary to achieve acceptable convergence speed. The learning rate defined by $\eta(k) = 1/(k+b)$, b being a positive constant, is theoretically known to give convergence, but this convergence is generally slow due to the quick decrease of η's value. We slowed this decrease down by using a learning rate of the form $\eta(k) = 1/(c.k+b)$[7] with constant b being just large enough to induce stable learning for all experiments, and constant c being smaller than 1 to prevent a quick decrease of the learning rate. To find a suitable value for b we increased it in steps of 5 until a value giving stable behavior was found. We limited the possible choices for c to members of the set $S_c = \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, and set c to that member that gave fastest convergence. This way to set b and c resulted in $b = 35.0$, $c = 0$ for the proposed algorithm and $b = 15.0$, $c = 0$ for the GHA algorithm.

The improvement in convergence speed was evaluated through a comparison of MSE learning curves of the EL-based and GHA algorithms. These curves were obtained by plotting the mean square error, between the original image and the image reconstructed from the principal components, at every 200 samples

until convergence was achieved. At each epoch, the reconstructed image was obtained by dividing the image into nonoverlapping 8×8 blocks, transforming these blocks by the current estimates of the principal eigenvectors used in reconstruction, and then transforming them back into image blocks.

Figures 2–5 show the learning curves of the proposed and GHA algorithms for the first two, three, four, and eight principal components, respectively. Each point on these graphs represent an average MSE over 50 experiments which was enough to obtain a sufficient statistical significance. As can be seen from the learning curves, the EL-based algorithm results in fast convergence without sacrifying reconstruction accuracy. Also, the efficacy of the adaptive learning rates increases with the increase in the number of principal components employed in the reconstruction. Specifically, compared to Sanger's GHA algorithm the EL-based algorithm gives about 3.0, 3.3, 3.9 and 4.3 times faster convergence to respectively the first two, three, four and eight principal eigenvectors.

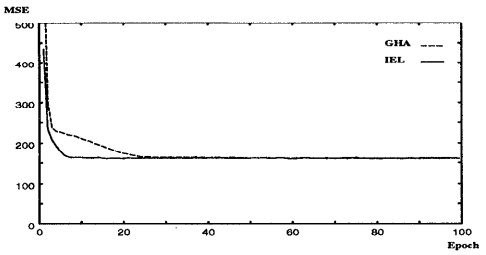


Fig. 2 Learning curves for the first two principal components.

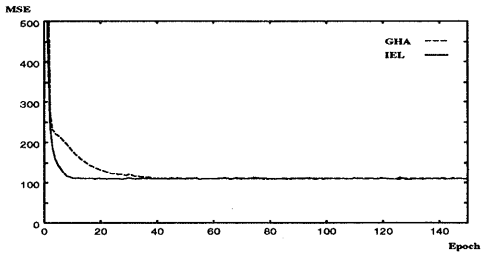


Fig. 3 Learning curves for the first three principal components.

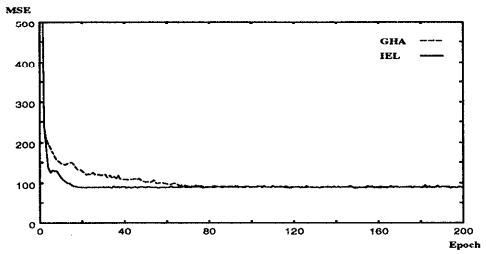


Fig. 4 Learning curves for the first four principal components.

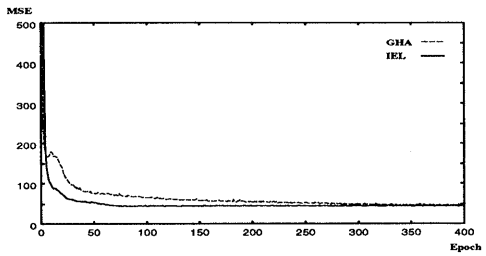


Fig. 5 Learning curves for the first eight principal components.

4. Concluding Remarks

In this paper, we proposed a PCA criterion and derived a learning algorithm that extracts data's principal eigenstructure. Unlike known standard PCA criterions, EL optimization leads to a variation of GHA algorithm which achieves high convergence speed by reflecting the dispersion of data experienced by each neuron to the corresponding learning rates. This feature overcomes the slow convergence of the GHA algorithm, particularly for low order principal components.

We note that the adaptive learning rate $\eta^{\omega_i}/\nu_{\omega_i}$, derived from optimizing the EL criterion, is the same as the optimal learning rate $1/M\nu_{\omega_i}$, which was proposed by Kung, et al.[5] in the context of the APEX algorithm, with $\eta^{\omega_i} = 1/M$. Here M denotes the number of input samples. They arrived at this optimal learning rate by analyzing the eigenmodes of the dynamic equations which governs the update of the coefficients appearing in the expansion of forward and lateral weight vectors in the eigenvector coordinate system. Because of the similarities in learning rates, both algorithms would have very similar convergence speed, were not it for APEX's mechanism of learning PC's one by one, which additionally speeds up its convergence, however, at the expense of its adaptiveness for nonstationary data. More precisely, for the recursive computation of each additional PC, GHA (and hence its variation derived here) requires $(m+1)n$ multiplications per iteration for the m-th neuron, as opposed to $2(m+n-1)$ multiplications per iteration in APEX[5], because APEX updates only the

weight vector corresponding to the last neuron and keeps the other weight vectors fixed.

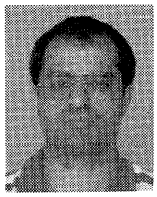
The parametric approach presented in this paper offers the possibility of merging the PCA technique with other parametric methods in pattern recognition in a uniform and seemless manner. Design of optimal classifiers in a reduced-dimensional space presents such a case where feature extraction and classifier design can be merged together.

Acknowledgments This work was financed by the Japan Ministry of Posts and Telecommunications as part of their Frontier Research Project in Telecommunications. To the paper were added the parts about the similarity of the proposed algorithm's adaptive learning rate with that of the APEX algorithm. We wish to thank the unknown referee who pointed this out to us.

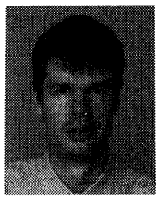
References

1) Jain, A.K.: *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs (1989).
2) Moghaddam, B. and Pentland, A.: Probabilistic, Visual Learning for Object Representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.696–710 (1997).
3) Oja, E.: A simplified neuron as a principal component analyzer, *J. Math. Biology*, Vol.15, pp.267–273 (1982).
4) Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks*, Vol.2, No.6, pp.459–473 (1989).
5) Kung, S.Y. and Diamantaras, K.I.: A neural network learning algorithm for adaptive principal component extraction, *ICASSP'90 Proceedings*, pp.861–864 (1990).
6) Xu, L.: Least Mean Square Error Reconstruction Principle for Self-Organizing Neural Nets, *Neural Networks*, Vol.6, No.5, pp.627–648 (1993).
7) Peper, F. and Noda, H.: A Symmetric Linear Neural Network that Learns Principal Components and their Variances, *IEEE Trans. Neural Networks*, Vol.7, No.4, pp.1042–1047 (1996).
8) Shirazi, M.N., Noda, H. and Sawai, H.: A Modular Realization of Adaptive PCA, *SMC'97 Proceedings*, Vol.4, pp.3053–3056 (1997).
9) Karhunen, J. and Joutsensalo, J.: Generalizations of Principal Component Analysis, Optimization Problems, and Neural Networks, *Neural Networks*, Vol.8, No.4, pp.549–562 (1995).
10) Peper F., Noda H. and Shirazi M.N.: Neural Networks Systems Techniques and Applications in the Determination of Principal Components in Data, *Expert Systems Techniques and Applications*, Gordon and Breach Science Publishers (1999).
11) Mansuripur, M.: *Introduction to Information Theory*, Prentice-Hall, Englewood Cliffs (1987).
12) Robbins H. and Monro S.: A Stochastic Approximation Method, *Ann. Math. Statistics*, Vol.22, pp.400–407 (1951).
13) Darken, C.J.: Stochastic Approximation and Neural Network Learning, *The Handbook of Brain Theory and Neural Networks*, Arbib, M.A. (Ed.), pp.941–945, MIT Press, Cambridge (1995).

(Received November 20, 1998)
(Accepted July 1, 1999)



Mahdad N. Shirazi was born in 1963 in Iran. He recieved his M.Sc. and Ph.D. degrees in Electrical Engineering from the Tottori University and Kobe University, respectively. In 1993 he became a Post-Doctoral Research Fellow at the Communications Research Laboratory, funded by the Science and Technology Agency (STA). Since 1995 he has been at the same laboratory, currently as Senior Research Scientist. His research interests include neural networks, pattern recognition, and image processing.



Ferdinand Peper was born in 1961 in the Netherlands. He received his M.Sc. and Ph.D. degrees (cum laude) from Delft University of Technology in the Netherlands. In 1990 he became a Post-Doctoral Research Fellow at the Communications Research Laboratory, funded by the Science and Technology Agency (STA). Since 1993 he has been at the same laboratory, currently as Senior Research Scientist. Also, from 1997 to 1998 he was a visiting researcher at the W.M. Keck Foundation Center for Integrative Neuroscience at the University of California, San Francisco, and since 1999 he is a Visiting Professor at the Graduate School of Himeji Institute of Technology. His research interests include neural networks, next-generation computers based on cellular automata, and evolutionary algorithms. He is a member of IEEE.

Hidefumi Sawai was born in 1954, in Kobe. He received his M.Sc. and Ph.D. degrees in Electrical Engineering from Keio University in 1977 and 1982, respectively. He joined Ricoh Company in 1983, where he engaged in Research & Development of Speech Recognition. From 1988 to 1991, he was a senior researcher at ATR (Advanced Telecommunications Research), where he was doing research on speech recognition using neural networks. From 1989 to 1990, he was an invited research scientist at Carnegie Mellon University, PA, USA. In 1995, he became the head of Auditory and Visual Informatics section at the Communications Research Laboratory. He is now the head of Human Neurosystem Science section of Intelligent Communications dihvision in Tokyo, and at the same time Professor of Graduate School in Kobe University. Dr. Sawai was a member of the Editorial Committee in IPSJ (from 1991 to 1995) and is a member of the IEICE, IPSJ, AI Society of Japan, JNNS, IEEE SP Socicty and Neural Network Council. His research interest includes intelligent information processing based on brain functions and evolutionary mechanisms, including neural networks, genetic algorithms, artificial life, evolutionary computation, pattern recognition, image and speech recognition.