Feature Extraction of Handwritten Characters Using Supervised and Unsupervised Independent Component Analysis

Seiichi OZAWA† Yoshinori SAKAGUCHI† Manabu KOTANI‡

† Graduate School of Science and Technology, Kobe University, Kobe 657-8501, JAPAN
‡ Faculty of Engineering, Kobe University, Kobe 657-8501, JAPAN

{ozawa, y-sakag}@eedept.kobe-u.ac.jp kotani@cs.kobe-u.ac.jp

ABSTRACT

Recently, Independent Component Analysis (ICA) has been applied to not only problems of blind signal separation, but also feature extraction of images and sounds. However, it is not easy to obtain high-performance features from real data by using conventional ICA algorithms. This might be originated in the fact that class information is not taken into consideration when feature extraction is conducted. It is considered that a remedy for this problem is to introduce a supervisor into ICA. Hence, in this paper, we shall study the effectiveness of Umeyama's Supervised ICA (SICA) for feature extraction of handwritten characters. Two types of control vectors (supervisor) are examined in SICA: (1) average patterns (Type-I) and (2) square/line patterns (Type-II). To demonstrate the usefulness of SICA, recognition performance is evaluated for handwritten digits that are included in the MNIST database. From the results of recognition experiments, we certify that SICA is effective for feature extraction if supervisor is designed properly. Furthermore, SICA features using Type-I control vectors are more effective than those using Type-II control vectors. Therefore, one can say that control vectors should be designed such that class information is reflected.

Keywords: Independent Component Analysis, Feature Extraction, Character Recognition, Supervised and Unsupervised Learning

1. INTRODUCTION

Recently, independent component analysis (ICA) has been widely known as a decorrelation technique based on high-order moment of input signals [1]. ICA has been so far applied to problems of blind signal separation such as sound/image separation and EEG signal separation. On the other hand, feature extraction of images and sounds has been also focused as one of prominent applications of ICA [2, 3, 4, 5]. Bartlett & Sejnowski extracted feature vectors from images of human faces using ICA, and showed that these feature vectors had greater viewpoint invariance for human faces as compared with Principal Component Analysis (PCA) ones [6]. (For notational convenience, we denote feature vectors obtained by ICA and PCA as ICA features and PCA features, respectively.) Since PCA decorrelates only the second order statistics of input signals, this result indicates that higher-order features are useful for capturing invariant features of face patterns as well as the second-order features. Such invariant characteristics of ICA features might be attractive for other pattern recognition problems.

In our previous works [7], we have presented two types of feature selection based on the cumulative proportion of eigenvalues and kurtosis. The former selection is carried out for principal components (PCs) of inputs and the latter is done for independent components (ICs). Through the recognition experiments, we have shown that a hybrid method, in which feature selection was carried out for ICs as well as for PCs, had attractive characteristics when low-dimensional feature vectors were used in recognition. However, the recognition performance was not always high from the practical point of view. It might be originated in the fact that class information is not taken into consideration when feature extraction is carried out.

Recently, Umeyama has proposed supervised ICA (SICA) [8], in which class information can be considered in the learning of a separation matrix. To overcome the above problem, we shall study the effectiveness of Umeyama's SICA for feature extraction of handwritten characters.

2. INDEPENDENT COMPONENT ANALYSIS (ICA)

Unsupervised ICA

Several ICA algorithms have been proposed so far, which are different in objective functions (or contrast functions) for statistical independence and how to derive ICA algorithms [1, 9, 10]. In general, estimated independent components obtained by these algorithms are different each other. However, it is difficult to discuss which algorithms are most appropriate for feature extraction. Therefore, we are not concerned here with the adequacy for ICA algorithms. In the followings, we shall adopt the bigradient algorithm proposed by Karhunen and Oja [11] because supervised ICA adopted here is a extended version of this algorithm.

Suppose that we observe a m-dimensional zero-mean input signal at time t, $\boldsymbol{v}(t) = \{v_1, \cdots, v_m\}'$, where $'$ means the transposition of matrices and vectors. Then the n-dimensional whitening signal, $\boldsymbol{x}(t)$, is given by the following equation:

$$\boldsymbol{x}(t) = \boldsymbol{M}\boldsymbol{v}(t) = \boldsymbol{D}^{-1/2}\boldsymbol{E}'\boldsymbol{v}(t), \tag{1}$$

where $\boldsymbol{M}$ means a $n \times m$ $(n \leq m)$ whitening matrix that is given by a matrix of eigenvalues, $\boldsymbol{D}$, and a matrix of eigenvectors, $\boldsymbol{E}$. Here, assume that $\boldsymbol{v}(t)$ is composed of n statistically independent components (ICs), $\boldsymbol{s}(t) = \{s_1(t), \cdots, s_n(t)\}'$. Then, the following linear transformation from $\boldsymbol{x}(t)$ to $\boldsymbol{s}(t)$ exists:

$$\boldsymbol{s}(t) = \boldsymbol{W}\boldsymbol{x}(t). \tag{2}$$

$\boldsymbol{W}=\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n\}'$ is often called a separation matrix, and it can be obtained through the training of a two-layer feedforward neural network. This neural network has n outputs denoted as $\tilde{\boldsymbol{s}}(t)$ $=\{\tilde{s}_1(t), \cdots, \tilde{s}_n(t)\}'$ and the ith row vector, $\boldsymbol{w}_i'(i = 1, \cdots, n)$, of $\boldsymbol{W}$ corresponds to a weight vector from inputs to the ith output, $\tilde{s}_i$.

The term 'independent' is used here according to the following definition in statistics:

$$p[s_1(t), \cdots, s_n(t)] = \prod_{i=1}^{n} p_i[s_i(t)], \tag{3}$$

where $p[\cdot]$ is a probability density function. Since the above probability density function is not preliminary unknown, suitable objective functions should be devised such that neural outputs, $\tilde{s}_i$, are satisfied with Eq. (3) as much as possible, i.e. $\tilde{\boldsymbol{s}}(t) \simeq \boldsymbol{s}(t)$. Karhunen and Oja have proposed the following contrast function [11], $J(\cdot)$, to be maximized in terms of output signals, $\tilde{\boldsymbol{s}}$:

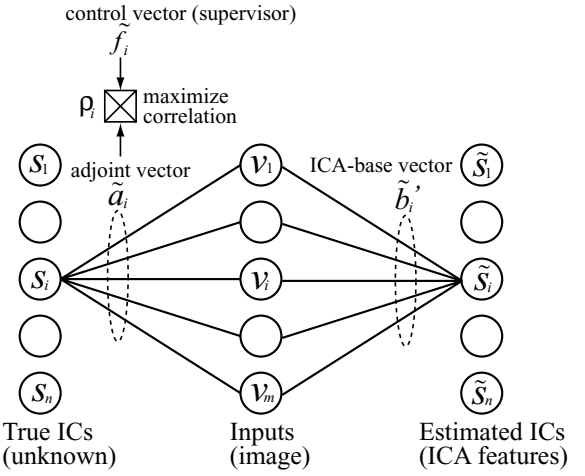

Figure 1: Schematic diagram of information processing in SICA.

$$J(\tilde{\boldsymbol{s}}) = \sum_{i=1}^{n} |E[\tilde{s}_i^4] - 3\{E[\tilde{s}_i^2]\}^2|, \tag{4}$$

where $E[\cdot]$ means expectation. As well known, Eq. (4) corresponds to the fourth-order cumulants of $\tilde{s}_i(t)$, called *kurtosis*. The following learning algorithms for a separation matrix, $\boldsymbol{W}$, are derived from the gradient of Eq. (4) and the orthonormality constraints of $\boldsymbol{W}$ [12]:

$$\begin{aligned}\boldsymbol{W}_{k+1} = \boldsymbol{W}_k &+ \mu(\tanh \tilde{\boldsymbol{s}}_k)\boldsymbol{x}_k' \\ &+ \gamma(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k')\boldsymbol{W}_k,\end{aligned} \tag{5}$$

where k means time step.

Supervised ICA

Umeyama has proposed a supervised version of ICA (SICA), in which a separation matrix is trained such that the contributions of ICs to input patterns could be controlled by *supervisor*. In other words, the training of SICA is carried out by maximizing correlations between each IC and specific sets of inputs as well as by strengthening independency of ICs.

Let us describe the details of SICA. From Eqs. (1) and (2), the relation between inputs and estimated ICs is rewritten as follows:

$$\boldsymbol{v} = (\boldsymbol{E}\boldsymbol{D}^{1/2}\tilde{\boldsymbol{W}}^{-1})\tilde{\boldsymbol{s}} = (\boldsymbol{E}\boldsymbol{D}^{1/2}\tilde{\boldsymbol{W}}')\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{s}}, \tag{6}$$

where we should note that $\boldsymbol{W}$ is an orthogonal matrix. Here, $\tilde{\boldsymbol{A}}$ corresponds to an estimated mixture matrix. The ith column vector, $\tilde{\boldsymbol{a}}_i$ $(i = 1, \cdots, n)$, of $\tilde{\boldsymbol{A}}$ is called an *adjoint vector* whose element values mean the contribution of the ith IC, $\tilde{s}_i$, to an input pattern, $\boldsymbol{v}$ (see Fig. 1). Therefore, if we want to control the contributions of the ith IC, we should give desired signals to these adjoint vectors. In SICA, as shown in Fig. 1, a normalized control vector, $\boldsymbol{f}_i = \{f_{i1}, \cdots, f_{im}\}'$, is given to $\tilde{\boldsymbol{a}}_i$ as its desired signal, and the following correlation, ρ_i, between $\boldsymbol{f}_i$ and $\tilde{\boldsymbol{a}}_i$ is maximized:

$$\rho_i = \frac{\boldsymbol{f}_i' \tilde{\boldsymbol{a}}_i}{\| \tilde{\boldsymbol{a}}_i \|}. \tag{7}$$

The update rule of a separation matrix, $\boldsymbol{W}$, at time k is shown as follows:

$$\begin{aligned} \boldsymbol{W}_{k+1} &= \boldsymbol{W}_k + \mu(\tanh \tilde{\boldsymbol{s}}_k)\boldsymbol{x}_k' \\ &+ \gamma(\boldsymbol{I} - \boldsymbol{W}_k \boldsymbol{W}_k')\boldsymbol{W}_k + \boldsymbol{\Lambda}\boldsymbol{G}, \end{aligned} \tag{8}$$

where

$$\boldsymbol{G} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_p, 0, \cdots, 0]'. \tag{9}$$

Here, p is the number of ICs to be controlled. In Eq. (9), $\boldsymbol{g}_i$ is obtained from the derivative of ρ_i with $\boldsymbol{W}$ in Eq. (7), and $\boldsymbol{\Lambda}$ is a matrix of learning coefficients shown below:

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_p, 0, \cdots, 0), \tag{10}$$

where λ_i is given by

$$\lambda_i = \lambda_\mu' \frac{\| \tanh \tilde{s}_i \boldsymbol{x} \|}{\| \boldsymbol{g}_i \|}. \tag{11}$$

Here, λ_μ' is a negative constant that determines the balance between the independence term (the second term) and the correlation term (the fourth term) in the right hand side of Eq. (8) (see [8] for details).

3. FEATURE EXTRACTION OF HANDWRITTEN CHARACTERS

Feature Extraction Using Unsupervised ICA

As described in Section 2, ICA algorithms allow us to decompose input signals into their independent components such that they are satisfied with Eq. (3) as much as possible. Such characteristics of ICA can be applied to feature extraction of handwritten characters.

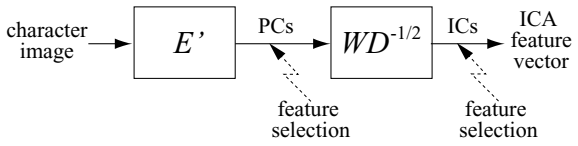

Figure 2: A block diagram of feature extraction of characters using ICA.

Based on Eqs. (1) and (2), the relation between inputs and outputs of ICA is given by

$$\tilde{\boldsymbol{s}}(t) = \tilde{\boldsymbol{W}}\boldsymbol{D}^{-1/2}\boldsymbol{E}'\boldsymbol{v}(t) = \tilde{\boldsymbol{B}}\boldsymbol{v}(t), \tag{12}$$

where $\tilde{\boldsymbol{W}}$ is a separation matrix trained by an ICA algorithm and $\tilde{\boldsymbol{B}} = \tilde{\boldsymbol{W}}\boldsymbol{D}^{-1/2}\boldsymbol{E}'$ is a $n \times m$ matrix. When an input, $\boldsymbol{v}(t)$, corresponds to the tth presentation of character images, the ICA output, $\tilde{\boldsymbol{s}}(t)$, can be considered as its feature vector (see Fig. 1). Here, the ith row vector, $\tilde{\boldsymbol{b}}_i'$ $(i = 1, \cdots, n)$, of $\tilde{\boldsymbol{B}}$ corresponds to a base vector spanning n-dimensional feature space (such base vectors are called ICA-bases for convenience). Since $\boldsymbol{E}'\boldsymbol{v}(t)$ corresponds to principal components (PCs) of $\boldsymbol{v}(t)$, one can say that an ICA feature vector is given with transformation $\tilde{\boldsymbol{W}}\boldsymbol{D}^{-1/2}$ of a PCA feature vector. Therefore, we can consider that the process of feature extraction using ICA consists of two types of transformations (see Fig. 2). One is the transformation from an input image to PCA features, and the other is the transformation from PCA features to ICA features.

Different feature selection (dimension reduction) can be applied to outputs of the above transformations: that is, we can reduce dimensions PCA features and/or ICA features. In our previous work [7], however, it is not easy to extract useful features by reducing dimensions of ICA features. Therefore, feature selection is carried out only for PCA features in this paper.

Cumulative proportion has been often used in feature selection for PCA features as a criterion of determining useful features. For convenience, eigenvalues of a covariance matrix of training samples are denoted in order of their magnitude: $\lambda_1 \geq \cdots \geq \lambda_m$. Then, the cumulative proportion, c_n, is defined as follows:

$$c_n = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{m} \lambda_i}, \tag{13}$$

where n is the number of large eigenvalues to be

selected. Let us introduce an upper bound of cumulative proportion, c_0, that gives a threshold value of determining what feature vectors should be adopted, then the largest value of n can be determined such that $c_n \leq c_0$ holds. We select n eigenvectors with the largest n eigenvalues as PCA-bases; that is, we consider a n-dimensional subspace spanned by eigenvectors with $\lambda_1, \cdots, \lambda_n$. After this feature selection is carried out, a n-dimensional vector of PCs, $\boldsymbol{E}'\boldsymbol{v}(t)$, is obtained, then a n-dimensional ICA feature vector, $\tilde{\boldsymbol{s}}(t)$, is calculated from Eq. (12).

Feature Extraction Using Supervised ICA

In pattern recognition problems, it is more desirable that extracted features belonging to different classes are mutually separated as much as possible in the feature space. Conventional ICA is, however, categorized in unsupervised learning; therefore, good separatability for extracted features is not always ensured. To overcome this problem, we should utilize class information (teacher signals) for extracting good features. Hence, supervised ICA (SICA) shown in the previous section is adopted here for feature extraction of handwritten characters.

As stated in Section 2, an adjoint vector, $\tilde{\boldsymbol{a}}_i$ $(i = 1, \cdots, n)$, in SICA indicates the contribution of the ith IC, $\tilde{s}_i$, to an input pattern (character image), $\boldsymbol{v}$. Through the learning of SICA, this contribution can be controlled by varying a control vector, $\boldsymbol{f}_i$. However, it is not clear how these control vectors should be designed in order to extract good independent features. As seen from Eq. (6), one can say that an input pattern is given by weighted sum of some adjoint vectors. Therefore, control vectors should be designed such that input patterns are approximated by weighted sum of the control vectors (note that adjoint vectors are trained so as to maximize the correlation with control vectors).

Hence, we shall present two types of control vectors whose two-dimensional representation corresponds to (1) average patterns (Type-I) and (2) square/line patterns (Type-II). As shown in Fig. 3, Type-I control vectors are obtained by simply averaging training samples belonging to the same category; hence, the number of control vectors is equivalent to the number of categories. On the other hand, as shown in Fig. 4, Type-II control vectors are defined as square-shaped and line-shaped patterns. In the following recognition experiments, a 28x28-pixel character image is divided into 16 blocks (see Fig. 4), each of which is composed of 7x7 pixels. Twenty-four Type-II control vectors are constructed by combining these 16 blocks: 16 square fragments, 4 horizontal lines, and 4 vertical lines. In Fig. 4, all pixels included in a shaded block have +1 and others have 0. The above two types of control vectors are normalized such that their average and variance are equal to 0 and 1, respectively. We should notice that Type-II control vectors are defined regardless of digit classes, but each Type-I control vector includes class information.

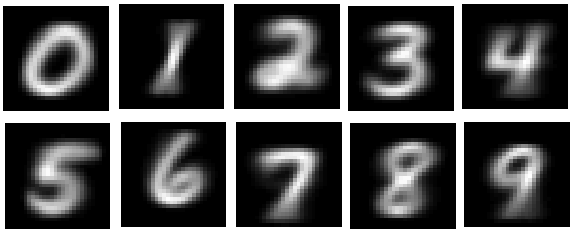

Figure 3: Examples of Type-I control vectors.

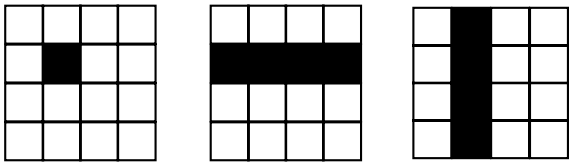

Figure 4: Examples of Type-II control vectors.

4. SIMULATIONS

To demonstrate the usefulness of SICA, recognition performance is evaluated for handwritten digits. A thousand of digit patterns in the MNIST database are used for training, and ten thousands of digit patterns are used for evaluation. Although the MNIST database originally includes 60,000 training samples, we use only 100 samples for each digit in order to reduce training time. Each image of handwritten digits is composed of 28x28 pixels and no preprocessing is carried out before feature extraction. Training samples are used for generating prototype vectors as well as learning ICA-bases. In classification, we adopt the similarity as a measure of distance between an input image and the pro-

totypes. After calculating similarities with all prototypes, recognition is conducted based on the k-nearest neighbor (k-NN) method, where k is set to 5 in the following simulations.

In SICA, the dimensions of feature vectors are the same as the number of control vectors; hence, the dimensions of feature vectors for Type-I and Type-II control vectors are 10 and 24, respectively. Dimension reduction is carried out only by sphering of input patterns. For comparative purpose, we evaluate the performance of feature vectors that are obtained by using conventional (unsupervised) ICA. In order to examine the independency of extracted features, $\tilde{\boldsymbol{s}} = \{\tilde{s}_1, \cdots, \tilde{s}_n\}$, the following absolute value of kurtosis is evaluated:

$$\mathrm{kurt}(\tilde{\boldsymbol{s}}) = \frac{1}{n}\sum_{i=1}^{n} |E[\tilde{s}_i^4] - 3\{E[\tilde{s}_i^2]^2\}|. \qquad (14)$$

If $\tilde{\boldsymbol{s}}$ has larger kurtosis, one can say that this feature vector is more statistically independent.

The results of recognition accuracy and absolute values of kurtosis for SICA with Type-I and Type-II control vectors are shown in Tables 1 and 2, respectively*. As you can see from Table 1, the performance of SICA with Type-I control vectors is higher than that of ICA and it improves with the decrease of λ'_μ, which determines the balance between the independence term and the correlation term (see Eqs. (8) $\sim$ (11)). If the absolute value of λ'_μ is large, the effect of supervisor becomes large in the training of ICA-bases. This means that the Type-I supervisor in SICA works effectively in feature extraction. Although absolute values of kurtosis for SICA features are slightly smaller than those for ICA features, one can say that the values are fairly high for any λ'_μ.

On the other hand, when Type-II control vectors are introduced into SICA, the extracted features have quite small kurtosis as compared with ICA (see Table 2); this means that Type-II control vectors are not suitable to maintain the independency of ICA features. Furthermore, the recognition performance of SICA with Type-II control vectors is worse than that of ICA.

The above results lead to the following conclusions:

1. If control vectors are designed properly in SICA, high-performance features could be extracted from handwritten digits.

*We should note that dimensions of evaluated feature vectors are different in these two experiments. Therefore, we cannot simply compare the results in Tables 1 and 2.

Table 1: Experimental results for SICA with Type-I control vectors and conventional ICA. The dimensions of feature vectors and the number of control vectors are 10.

		accuracy [%]	kurtosis
SICA	$\lambda'_\mu = -1$	85.47	1.03
	$\lambda'_\mu = -10$	85.53	0.99
	$\lambda'_\mu = -50$	85.65	1.00
	$\lambda'_\mu = -100$	85.82	1.04
ICA		85.46	1.32

Table 2: Experimental results for SICA with Type-II control vectors and conventional ICA. The dimensions of feature vectors and the number of control vectors are 24.

		accuracy [%]	kurtosis
SICA	$\lambda'_\mu = -1$	88.86	0.305
	$\lambda'_\mu = -10$	89.04	0.341
	$\lambda'_\mu = -30$	89.05	0.328
	$\lambda'_\mu = -50$	89.09	0.311
ICA		89.84	3.08

2. SICA features using Type-I control vectors are more effective than those using Type-II control vectors. Therefore, one can say that control vectors should be designed such that class information is included in them.

3. In general, features with high independency are effective. However, as we can see from the recognition results for ICA features, features that simply increase their independency are not always effective. This suggests that increasing independency and introducing supervisor into ICA is a key to enhancing the performance of feature extraction.

5. CONCLUSIONS

We applied Supervised Independent Component Analysis (SICA) to feature extraction of handwritten digits. Two types of control vectors were introduced into SICA in order to examine the effectiveness of supervisor. From the results of recognition experiments, we certified that one of the control vectors worked effectively in SICA; that is, If control vectors are designed properly, increasing inde-

pendency and introducing supervisor into ICA can realize high-performance feature extraction.

REFERENCES

[1] A. Hyvärinen: "Survey on independent component analysis", *Neural Computing Surveys*, **2**, 94-128, 1999.

[2] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja: "Applications of neural blind separation to signal and image processing", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 131-134, 1997.

[3] M. Kotani, Y. Shirata, S. Maekawa, S. Ozawa, and K. Akazawa: "Application of independent component analysis to feature extraction of speech", *Proc. of Int. Joint Conf. on Neural Networks* (IJCNN99-Washington DC), CD-ROM #704, 1999.

[4] S. Ozawa, T. Tsujimoto, M. Kotani, and N. Baba: "Application of independent component analysis to hand-written Japanese character recognition", *Proc. of International Joint Conf. on Neural Networks* (IJCNN99-Washington DC), CD-ROM #462, 1999.

[5] Y. Watanabe, M. Hirahara, and T. Nagano: "Feature extraction of palm prints using supervised independent component analysis", *CD-ROM Proc. of 7th Int. Conf. on Neural Info. Processing*, 2000.

[6] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski: "Independent component representations for face recognition", *Proc. of the SPIE*, **3299**, 528-539, 1997.

[7] S. Ozawa, M. Kotani: "A study of feature extraction and selection using independent component analysis", *Proc. of 7th Int. Conf. on Neural Info. Processing*, **I**, 369-374, 2000.

[8] S. Umeyama, S. Akaho, Y. Sugase: "Supervised independent component analysis and its applications to face image analysis" (in Japanese), *Tech. Report of IEICE*, **NC99-2**, 9-16, 1999.

[9] A. J. Bell and T. J. Sejnowski: "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, **7**, 1129-1159, 1995.

[10] S. Amari, A. Chichocki, and H. Yang: "A new learning algorithm for blind signal separation", *Advances in Neural Information Processing Systems*, **8**, MIT Press, Cambridge, MA, 757-763, 1996.

[11] J. Karhunen and E. Oja: "A class of neural networks for independent component analysis", *IEEE Trans. on Neural Networks*, **8**, 3, 486-503, 1997.

[12] L. Wang, J. Karhunen, and E. Oja: "A bigradient optimization approach for robust PCA, MCA, and source separation", Proc. IEEE Int. Conf. on Neural Networks (ICANN95-Perth), 1684-1689, 1995.