



## Contributed article

## Learning the parts of objects by auto-association

Xijin Ge, Shuichi Iwata\*

*Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, Komaba 4-6-1, Meguro-ku, Tokyo 153-8904, Japan*

Received 16 May 2001; accepted 19 November 2001

**Abstract**

Recognition-by-components is one of the possible strategies proposed for object recognition by the brain, but little is known about the low-level mechanism by which the parts of objects can be learned without a priori knowledge. Recent work by Lee and Seung (Nature 401 (1999) 788) shows the importance of non-negativity constraints in the building of such models. Here we propose a simple feedforward neural network that is able to learn the parts of objects by the auto-association of sensory stimuli. The network is trained to reproduce each input with only excitatory interactions. When applied to a database of facial images, the network extracts localized features that resemble intuitive notion of the parts of faces. This kind of localized, parts-based internal representation is very different from the holistic representation created by the unconstrained network, which emulates principal component analysis. Furthermore, the simple model has some ability to minimize the number of active hidden units for certain tasks and is robust when a mixture of different stimuli is presented. © 2002 Elsevier Science Ltd. All rights reserved.

*Keywords:* Neural networks; Auto-associators; Non-negativity constraints; Face recognition; Chinese characters; Feature detection; Whole-part relation; Pattern recognition

**1. Introduction**

Auto-associators are simple multilayered feedforward neural networks that regenerate a set of data vectors through a narrow hidden layer. The simplest auto-associator contains one hidden layer (Fig. 1A). Trained by the standard error back-propagation algorithm (Rumelhart, Hinton, & Williams 1986), these networks adapt to an efficient internal representation for the input dataset. Baldi and Hornik proved that the landscape of the quadratic error function of linear auto-associator has a unique minimum, and thus such networks are able to learn without local minima (Baldi & Hornik, 1989). A theorem due to (Hecht-Nielsen, 1995) states that a class of auto-associators with three hidden layers can carry out optimal data compression for arbitrary datasets. Auto-associators have been thought to have potential applications to image compression (Cottrell, Munro, & Zipser, 1989; Namphol, Chin, & Arozullah, 1996), speech processing (Ellman & Zipser, 1987; Gori, Lastucci, & Soda, 1996), modeling of the cognitive process of concept-formation (Gluck & Myers, 1993) and even the natural language grammar acquisition process (Hanson & Kegl, 1987).

Nonetheless, some studies suggest that linear auto-

associators emulate principal component analysis (PCA) (Baldi & Hornik, 1989; Broulard & Kamp, 1988), a linear algorithm for dimension-reduction, and cannot outperform PCA even in the presence of non-linearities in the hidden layers (Broulard & Kamp, 1988). Although some other works show that non-linear hidden units do play a role in certain tasks (Japkowicz, Hanson, & Gluck, 2000; Kramer, 1991), these networks are generally understood in the framework of PCA. The internal representation learned by auto-associators often consists of orthogonal basis functions that correspond to *holistic* features. Applied to a database of facial images, for instance, auto-associators learn ghostly looking ‘eigenfaces’ or ‘holons’ that resemble the eigenvectors of PCA (Cottrell & Fleming, 1990; Turk & Pentland, 1991; Valentin & Abadi, 1996; Valentin, Abadi, O’Tool, & Cottrell, 1994). Therefore, it is believed that auto-associators are subject to the limitations of PCA.

In this paper, we demonstrate that non-linear auto-associators, with certain constraints introduced, can learn internal representations that are fundamentally different from the holistic PCA representation. A recent work by Lee and Seung (1999) show that some constraints may have profound influences on the kind of internal representation that a computational model can learn. They developed an iterative algorithm for matrix factorization called non-negative matrix factorization (NMF). The algorithm approximates a database of facial images using basis

\* Corresponding author. Tel.: +81-3-5453-5884; fax: +81-3-3467-0648.  
E-mail address: iwata@race.u-tokyo.ac.jp (S. Iwata).

## Nomenclature

List of mathematical symbols

$\mathbf{X}$	matrix of raw data ( $n \times m$ )
$\mathbf{Y}$	matrix of basis vectors ( $n \times p$ )
$\mathbf{A}$	matrix of expansion coefficients ( $p \times m$ )
$Q(E)$	error function to be minimized in positive matrix factorization (PMF)
$E_{ij}$	residual value for factorization in PMF
$F$	objective function of non-negative matrix factorization (NMF)
$w(t)$	weights of connection in neural networks at the $t$ th iteration
$f(a)$	response function of the nodes in a neural network
$a$	sum of inputs to a node
$\theta$	bias term or threshold of a node
$\rho$	a parameter in the response function

functions and encodings that are both non-negative. It is demonstrated that NMF is able to learn the parts of faces that resemble intuitive notions. A whole face is represented by the combination of parts. The localized, parts-based representation is very different from holistic ‘eigenface’ of PCA. The non-negativity constraints force NMF to learn localized features that can be added together to regenerate the whole faces, since only additive combinations, not subtractive cancellations, are allowed in the regeneration. Inspired by Lee and Seung’s work, we introduce these constraints to auto-associators, and propose a ‘non-negative auto-associator’ (NA) model. Through several simulational experiments, we compare NA with the unconstrained network and NMF. Surprisingly, we found that NA can outperform NMF in certain tasks. For example, NA can minimize the number of basis functions for a given dataset.

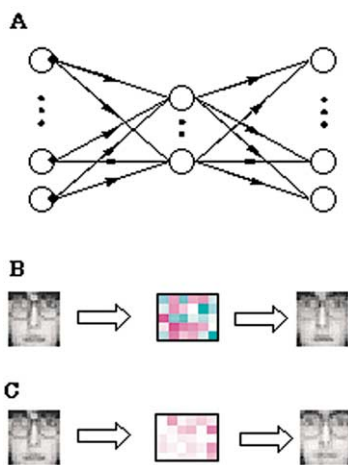


Fig. 1. (A) An auto-associator with one hidden layer. It reproduces input vectors using a relatively small number of hidden units. (B) Without any constraint, the network compresses an image in a way very similar to PCA. Note that the compressed form involves both positive (red pixels) and negative (blue pixels) coefficients. (C) In the non-negative auto-association model proposed in this paper, the synaptic connections and the activation values are confined to be non-negative. We are interested in how such constraints could influence the basis images, each of which is the information coded by a pixel in the compressed form.

The rest of this paper is organized as follows. Section 2 briefly introduces matrix factorization algorithms that incorporate the non-negativity constraints. Section 3 presents our new model. In Section 4, NA is applied to a database of facial images. The characteristics of its internal representation of faces are discussed in comparison with the unconstrained network. In Section 5, NA’s performance in the over-complete case is tested by a well-defined data set. This also shows the differences between the new model and the original NMF algorithm. In Section 6, NA is applied to a mixture of facial images and Chinese characters. Sections 7 and 8 include discussions and conclusions, respectively.

## 2. Origin of the non-negativity constraint

In physical sciences such as spectroscopy and chemometrics, there are often underlying physical models possessing the property of linear additivity. In such cases, the observed data is the result of additive superposition of several sources, thereby cancellations and negative values are often not meaningful. In environment monitoring, for example, measured densities of air borne particles come from various sources whose contributions are either positive or zero (Chueinta, Hopke, & Paatero, 2000). Other examples include astronomical images, each pixels of which is the result of additive superposition of the lights emitted from many stars. It turns out that there are a variety of such problems in which linear additivity holds. For the modeling of such systems, factor analysis using PCA arrives at negative values that are difficult to interpret. Recently, several algorithms have been developed for solving such problems.

One such method is called positive matrix factorization (PMF) (Paatero, 1997; Paatero, & Tapper, 1994). The basic goal is to construct an approximate factorization of the form

$$\mathbf{X} \approx \mathbf{Y}\mathbf{A}, \quad (1)$$

where  $\mathbf{X}$  is the  $n \times m$  data matrix, each column of which contains  $n$  features of one of the  $m$  instances of observation.

$\mathbf{A}$  is a  $p \times m$  matrix of expansion coefficients, and  $\mathbf{Y}$  contains  $p$  basis vectors that are believed to reveal internal regularities of the data source. Often  $p$  is much smaller than  $n$  and the process is referred to as dimension-reduction. Obviously, there is much freedom in the factorization as an algorithm can define  $\mathbf{A}$  and  $\mathbf{Y}$  at the same time in its own way. To meet different needs, algorithms often put various constraints on  $\mathbf{A}$  or  $\mathbf{Y}$ , or even both. In vector quantization (VQ), for example, each column of  $\mathbf{A}$  is confined to be a unary vector, with one element equal to unity and the other elements equal to zero; as a result, VQ finds prototypes in the raw data. In PCA, it is required that the basis vectors in  $\mathbf{Y}$  be orthogonal, which is important for linear dimension-reduction and convenient for calculation. In PMF all the elements in both  $\mathbf{A}$  and  $\mathbf{Y}$  are confined to be non-negative on account of the nature of environmental monitoring problem. Obviously the elements in  $\mathbf{X}$  must be non-negative at the first place. In most cases, they can be easily scaled to be so if the raw data is not.

The objective function of PMF is

$$Q(E) = \sum_{i=1}^m \sum_{j=1}^n (E_{ij}/\sigma_{ij})^2, \quad (2)$$

where

$$E_{ij} = X_{ij} - \sum_{k=1}^p A_{ik} Y_{kj} \quad (3)$$

is the residual value for factorization, and  $\sigma_{ij}$  is the standard deviation of the observed values  $x_{ij}$ . The PMF minimizes  $Q(E)$  with respect to  $\mathbf{A}$  and  $\mathbf{Y}$  under the constraint that all the elements of  $\mathbf{A}$  and  $\mathbf{Y}$  are non-negative. The iterative algorithm can be considered as a generalization of alternating least square approach. The mathematics of the algorithm can be found elsewhere (Paatero, 1997; Paatero & Tapper, 1994). This method has been used by researchers in the environmental sciences (see for example Chueinta et al., 2000).

Like PMF, the NMF algorithm of Lee and Seung (1999) uses non-negativity constraints on all the elements of  $\mathbf{A}$  and  $\mathbf{Y}$ . But NMF is developed to address very different problems like the learning of parts-based object representation. The following iterative algorithm is used to find  $\mathbf{A}$  and  $\mathbf{Y}$ :

$$Y_{ia} \leftarrow Y_{ia} \sum_{\mu} \frac{Y_{i\mu}}{(YA)_{i\mu}} A_{a\mu}, \quad (4)$$

$$Y_{ia} \leftarrow \frac{Y_{ia}}{\sum_j Y_{ja}}, \quad (5)$$

$$A_{a\mu} \leftarrow A_{a\mu} \sum_i Y_{ia} \frac{X_{i\mu}}{(YA)_{i\mu}}. \quad (6)$$

Starting from non-negative initial conditions for  $\mathbf{A}$  and  $\mathbf{Y}$ , iteration of these update rules finds an approximate factorization of  $\mathbf{X} \approx \mathbf{Y}\mathbf{A}$  by converging to a local maximum

of the objective function

$$F = \sum_{i=1}^n \sum_{\mu=1}^m [X_{i\mu} \log(YA)_{i\mu} - (YA)_{i\mu}]. \quad (7)$$

This objective function can be derived by interpreting NMF as a method for constructing a probabilistic model of image generation. In this model, an image pixel  $X_{i\mu}$  is generated by adding Poisson noise to the product  $(AY)_{i\mu}$ . The objective function in Eq. (7) is then related to the likelihood of generating the images in  $\mathbf{X}$  from the basis  $\mathbf{Y}$  and encoding  $\mathbf{A}$ . The update rule preserves the non-negativity of  $\mathbf{A}$  and  $\mathbf{Y}$  and also constrain the columns of  $\mathbf{Y}$  to sum to unity.

When applied to a database of faces, NMF learns basis images that correspond to parts of faces, such as noses and eyes. This very interesting property is owing to the non-negativity constraints, which allow only additive, not subtractive, combinations to reconstruct images. As image pixels belonging to the same part of a face are coactivated when the part is emphasized, parts-based representation should be learnable from a set of examples.

As the essential factor that leads to parts-based representation in NMF is the non-negativity constraints, we wonder whether this will work in simple feedforward neural networks, especially the auto-associators.

### 3. Non-negative auto-associator

Fig. 1 shows the structure of a simple non-negative auto-associator (NA). The input and output layers contain the same number of units for the retrieval of patterns. The hidden layer has fewer units. All synaptic connections in NA are confined to be non-negative. During learning, connections are modified by back-propagation (Rumelhart et al., 1986) as long as the resultant values are positive; they remain zero even if the algorithm attempts to modify them into negative values. The modified learning rule can be written as

$$w_{ij}(t+1) = \begin{cases} 0, & \text{if } w_{ij}^*(t+1) < 0; \\ w_{ij}^*(t+1), & \text{otherwise;} \end{cases} \quad (8)$$

where  $w_{ij}^*(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$ , and  $\Delta w_{ij}(t)$  is the weight change required by back-propagation algorithm. This rule ensures that the connections will always be positive. In practical calculation, an upper limit of unity is also applied to the connection between hidden and output layer such that these connection weights are kept within the range  $[0, 1]$ , which is convenient for the interpretation of weight vectors.

Non-linear auto-associators often use the following response functions for hidden units (see, Cottrell et al., 1989, for example),

$$f(a_i) = \frac{2}{1 + e^{-(a_i - \theta_i)/\rho}} - 1, \quad (9)$$

where  $a$  is the sum of inputs to the  $i$ th node,  $\theta_i$  is a bias

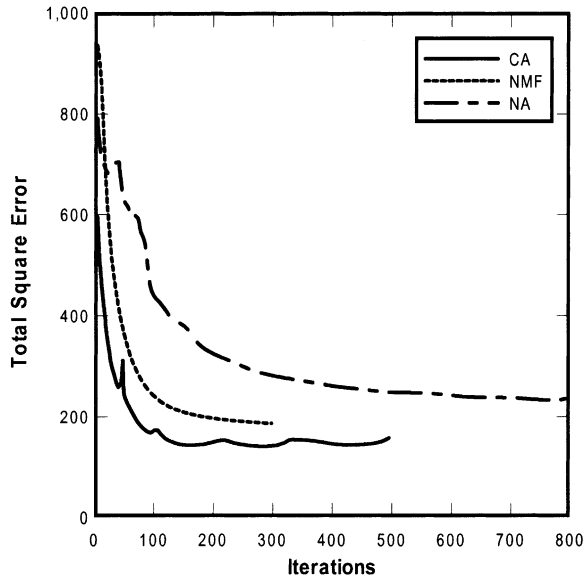


Fig. 2. Comparison of the convergence of three algorithms when applied to a database of facial images: CA that simulates PCA, non-NMF (Lee & Seung, 1999) and the present NA model.

term, and  $\rho$  is a control parameter. This function takes values in the range  $[-1, 1]$ . To honor the non-negative constraint on activation values, in the new model the conventional logistic activation function is used for hidden units, namely,

$$f(a_i) = \frac{1}{1 + e^{-(a_i - \theta_i)/\rho}}. \quad (10)$$

The activation function ensures that output values of hidden units will take the value in the range  $[0, 1]$ . Units in the output layer are linear and involve no bias, hence their activation values can only be positive. It is clear that the non-negativity of synaptic connections and activation values are guaranteed in NA.

Seen from the viewpoint of optimization, the synaptic weights are variables to be optimized for minimizing the objective function. Here the objective function is the square error of pattern retrieval. Error back-propagation algorithm is a gradient descent optimization of this error function with respect to the synaptic weights. If these weights are confined to be non-negative, the problem becomes a constrained optimization. Although this means a substantial reduction of the weight space, the learning process can still be characterized by gradient descent. Within the bounded weight space, gradient descent learning should still converge to a local minimum of the error function. In the following sections, we will show that the network can still find good solutions in practical applications.

#### 4. Learning the parts of objects

We first tested NA on a database of facial images. For

comparison, the conventional auto-associator (CA) and NMF are also applied to the same database. CA refers to auto-associators without the non-negativity constraints and whose response functions are given by Eq. (9). In short, it is the network that is believed to emulate PCA.

The subject facial images are obtained from several databases available on the world wide web, namely 39 from the Olivetti Research Ltd database (Samaria & Harter, 1994), 131 from the AR face database (Martinez & Benavente, 1998) and 21 from a database of University of Bern. So a total of 191 images with neutral or near neutral facial expressions are used. These images are hand-aligned in a  $19 \times 19$  grid. For each image, the gray-scale intensities were first linearly scaled so that the pixel mean and standard deviation were equal to 0.5 and 0.25, respectively. Finally, all pixel values are cut off to the range  $[0, 1]$ . Our dataset is available at: [www.race.u-tokyo.ac.jp/~xge/face/](http://www.race.u-tokyo.ac.jp/~xge/face/). One example is shown in the left side Fig. 1B and C.

The input layer has 361 units to represent all the pixels of an image. In the hidden layer, a ‘bottleneck’ of 30 units is used. The learning rate is 0.03 and the coefficient for momentum term is 0.9. The control parameter  $\rho$  is chosen as 6. It is important to keep  $\rho$  within a reasonable range. In the training process, each image is used both as the input and desired output. In practical calculations, input patterns are linearly scaled to the range  $[0.1, 0.9]$  to serve as teaching patterns for achieving a faster convergence.

We found that learning in NA is slower than that in the unconstrained network (Fig. 2). As mentioned at the beginning of this paper, CA is able to learn without local minima and can converge quickly to the PCA solution. Modification of weight strengths is made according to gradient descent of squared error. But in NA such modifications required by gradient descent are often undermined by the non-negativity constraints. Although the solution space is smaller in NA than that of CA, the boundedness of solution space makes learning more slowly. Usually, training should stop at the point where sharp decrease of the total square error slows down. For example, for the unconstrained network it is about 120–180 iterations. Over-training beyond this point should often be avoided. But for NA, we find it is necessary to go beyond this point because over-training allows the network to better explore the bounded weight space. In our calculation of the faces, we repeatedly trained the network for some 2000 times using the same database until there is no systematic decrease of the square error. By ‘systematic decrease’, we mean the decrease that is not due to pure fluctuation. When converged, CA achieved the smallest retrieval error, hence giving the most accurate retrieval of the raw data. NMF approximates the database better than NA. Instead of the fidelity of representation, however, here we are more interested in the ways that information is coded.

When training is finished, the network learns to transform a face into a compressed form indicated by the output patterns in the hidden layer (Fig. 1B and C), from which a

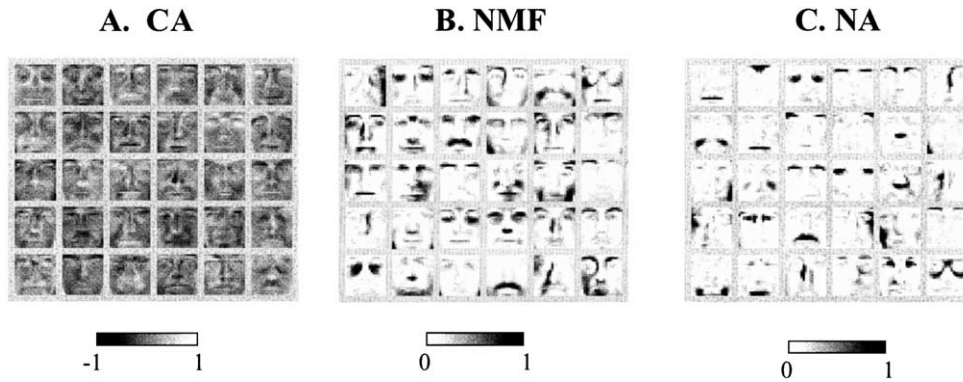


Fig. 3. The NA learns parts of faces, which are very similar to those of the non-NMF (Lee & Seung, 1999), whereas CA learns holistic representations. Each of the images is the fan-out weight vector of a hidden unit, and is the information that is represented by one pixel in the compressed form.

face can be approximately retrieved in the output layer. One pixel in the compressed representation influences multiple pixels on the raw image. For each hidden units there are 361 connections to the output layer. The weights of these connections can be plotted out as a  $19 \times 19$  image, which is the basis image discovered by the network. Fig. 3 shows the basis images created by NA, CA and NMF.

In contrast to the holistic representations of the unconstrained network, NA discovers localized features that resemble intuitive notions of facial parts. Without any a priori knowledge, some neurons in the hidden layer self-organized to detect eyes, while others learned to represent noses or mouths. During the learning process, the input weight vector to a hidden node adapts to detect a localized feature while the output weight vector adapts to regenerate the same feature. Eventually, the two weight vectors contain the same information: a facial part. There are different versions of the same facial part owing to the difference in location and lighting conditions. Interestingly, one neuron shown in the bottom right roughly represents the image of glasses. A whole face is generated by combining eyes, noses and other facial parts through selective firing of a subset of hidden units.

Detailed studies show that linear auto-associator without constraints automatically learns orthogonal bases, hence is indeed PCA-like. This is easy to understand because orthogonal bases are often the most efficient bases for approximation. The resultant basis images usually involve both positive and negative components. Furthermore, both additive and subtractive combinations of these bases are allowed to reproduce a face. In NA, however, orthogonality (and hence retrieval accuracy) is undermined by the non-negativity requirement. A face is represented by using additive, not subtractive, combinations of basis images. This has drastic influences on the basis images. As shown in Fig. 3C, the NA basis contain a large fraction of vanishing coefficients. These bases are localized features, in contrast to the holistic features discovered by the unconstrained network shown in Fig. 3A.

## 5. Learning a minimal set of parts

In Section 8, it is demonstrated that NA is able to learn the parts of faces and functions similarly as NMF. In this section, we address the difference between the two algorithms through a synthetic data set. The problem of extracting independent bars from visual images was previously addressed by Hinton, Dayan, Frey, and Neal (1995). The images in Fig. 4 are a part of 200 images that are generated by a computer program which randomly combines a set of predetermined basis images. In this case, 16 basis functions are used, corresponding to eight horizontal and eight vertical bars. The task is to discover those basis functions from examples.

Sixty-four units are used in input and output layer in accordance with the size of images. For the hidden layers, we use 20 basis functions in all three methods, slightly larger than the number of predetermined bases. The learning rate is chosen as 0.02, and the coefficient of the momentum term is 0.8.

When training is finished, the weight vectors are plotted the same way we did to facial images. The result is shown in Fig. 5. Most of the NMF bases correspond to horizontal or vertical bars as expected. However, since more basis images are used than really needed, some vertical bars are further divided arbitrarily into several pieces. This problem is more serious if 30 or more basis functions are used. This is the over-fitting phenomenon typical for many data analysis algorithms, and often there is no better solution than trial-and-error. When 16 basis functions are used, NMF can successfully find the right solution.

Surprisingly, we find that NA automatically learns a set of 16 bases containing exactly all the predetermined bars. The other four turn out to be blank with all their components nearly zero. The corresponding hidden units are inactive throughout the retrieval of all patterns in the dataset. How could NA eliminate unnecessary hidden units?

First, NMF basis functions are normalized, namely the following relation always holds for all hidden units during

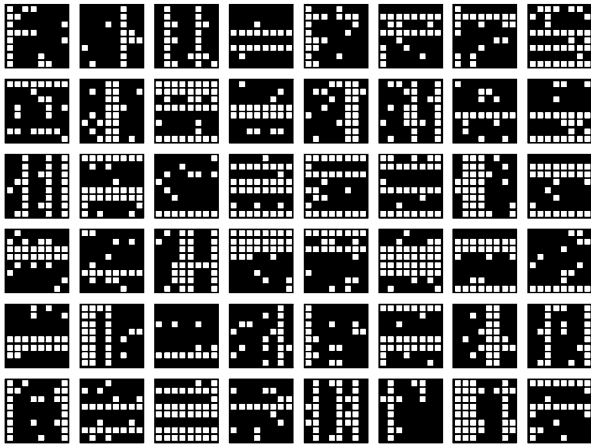


Fig. 4. Examples of training images produced by a computer program (Hinton et al., 1995). First, an orientation (i.e. horizontal or vertical) is randomly chosen with fair odds. Second, each bar of the chosen orientation is randomly instantiated with probability 0.25. Third, additive noise is introduced by randomly turning on with a probability of 0.2 each pixel that was previously off. All the images can be considered as the additive combination of 16 basis images that correspond to eight horizontal and eight vertical bars.

learning,

$$\sum_k Y_{jk} = 1, \tag{11}$$

where  $Y_{jk}$  can be considered as the connection between the  $j$ th hidden unit with the  $k$ th output unit (Lee & Seung, 1999). This is a consequence of the iteration rules given in Eq. (5). Clearly, this constraint immediately rules out the possibility of having all the components of a basis function near to zero. All basis functions must be involved, even though some of them may be unnecessary. On the other hand, NA does not have this normalization constraint, thus makes null basis function possible in principle.

Second, this property is partially attributed to the learning process determined by the error back-propagation algorithm. Learning in the network involves two phases. In the

first phase the input is presented and propagated forward to compute the output value for each unit, which is then compared with the target, resulting in an error signal for each output unit. The second phase involves a backward propagation of the error signal, according to which appropriate weight changes are made. To properly represent a ‘part’, a hidden unit must first detect the feature through the fan-in of connection weights, and then regenerate it through fan-out connections. Eventually these two weight vectors should resemble each other. The inter-dependence of these two sets of weights in the two-phase learning introduces feedback loops to the competition of hidden units for the error signal associated with a certain ‘part’. This mechanism makes it less probable for two hidden units to converge to smaller pieces of the same ‘part’. On the contrary, NMF has only one set of weight vectors used for both feature detection and regeneration. This increases the speed of convergence but leaves less room for robustness in the over-complete case.

Third, the non-linearity of hidden units and initial conditions are also essential for the robustness. When linear activation function is used, the network is often unable to find the right number of parts. Also it is important to assign small random values to all the weights in the network at the beginning of the learning, which is prohibited in NMF by the normalization constraint. It is helpful to examine the dynamics of learning process. As shown in Fig. 6, weight vectors are assigned a small random value in [0,0.1] before training. After 10 iterations, a few weight vectors gradually ‘grow up’. One by one, the horizontal and vertical bars are found. In the 120th iteration, all of 16 bars are learned. Four basis images remain to be null vectors. Those null weight vectors do not change significantly upon over-training since there is no longer any coherent error signal.

These mechanisms are responsible for NA’s ability to minimize the number of necessary bases. But we do not claim that NA will always find the optimal solution. Starting from different initial conditions, the network occasionally

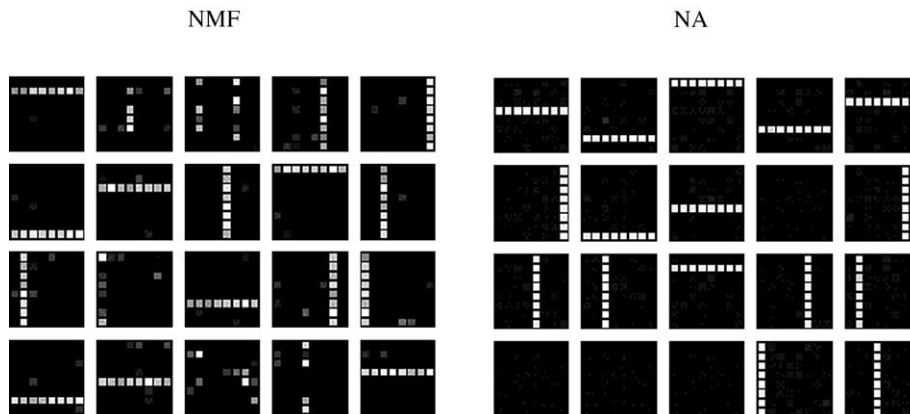


Fig. 5. NA finds exactly 16 basis images from the data set shown in Fig. 4. The other four are empty and the corresponding hidden units turn out to be inactive to all images in the training set. On the contrary, non-NMF uses all the 20 bases redundantly.

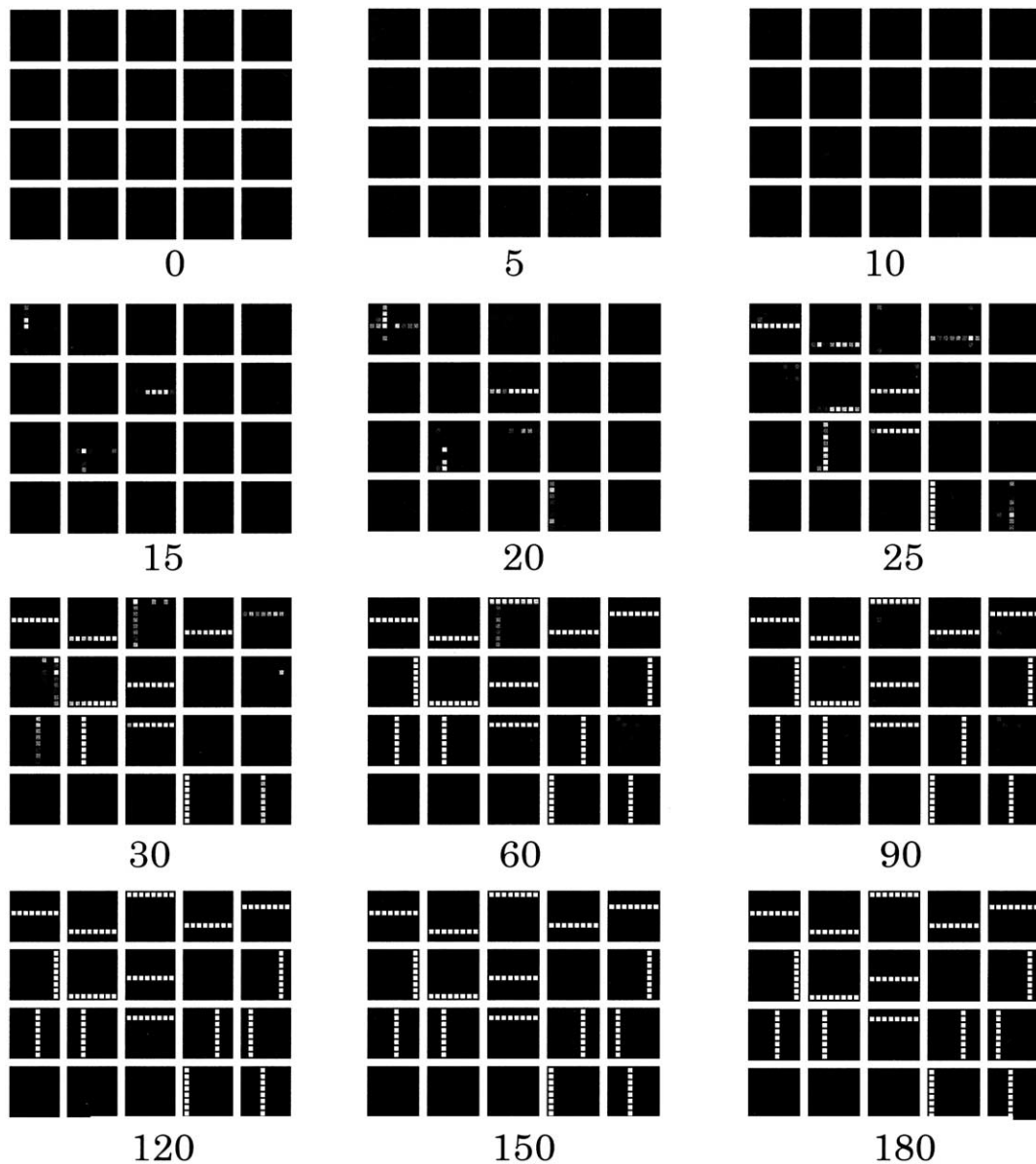


Fig. 6. Learning dynamics of the NA. The number at the bottom of each plot is the times of repeated training using the same set of data. Learning begins with small, random synaptic strengths. Gradually, connections are selectively strengthened. Error back-propagation leads to competition among hidden units for error signals, and makes it more likely for these units to learn independent bars.

divides one or more bars further into several pieces as NMF does. Also, the performance of NA depends on the training set. The training set of vision bars discussed before is a good one that contains enough instances of the combination. In most cases, there does not exist a well-defined set of basis functions. Nonetheless, it is clear that NA has some ability to minimize the necessary number of hidden nodes. Note that the ‘wake–sleep’ algorithm and the Helmholtz Machine are also able to eliminate unnecessary hidden units (Dayan, Hinton, Neal, & Zemel, 1995; Frey et al., 1997) in certain circumstance. Further study is needed to clarify the mechanism behind this property.

## 6. Mixing of different stimuli

As shown in Fig. 1C, not only the basis images but also the image encodings in NA are sparse, containing many vanishing coefficients. This means that the network develops sparse coding under the constraints. This is a new feature of NA and NMF in contrast to methods of PCA and alike by which almost all bases are used in the encoding of any given image. In this section, we will show that this property also has an important consequence when different types of stimuli are presented in the training set. Before going to this point, we will consider the problem of Chinese characters.

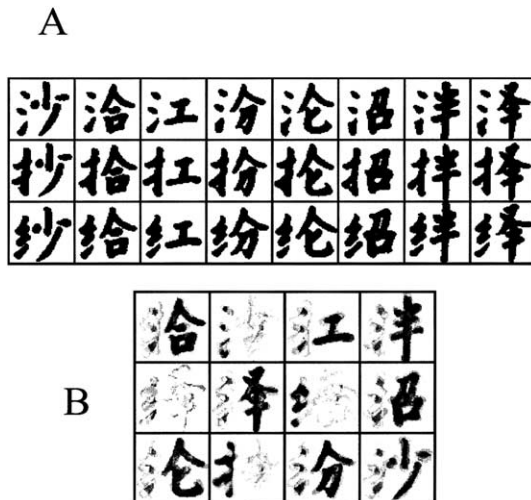


Fig. 7. (A) Two dozen Chinese characters carefully written by hands. (B) The basis images learned by NA from those examples. Each basis image roughly corresponds to an element, the building block of the Chinese writing system. The network's parts-based representation of characters is similar to that of human beings.

### 6.1. Parts-based approach for recognizing Chinese characters

The Chinese writing system is non-alphabetic. It applies a specific character to write each meaningful syllable. This makes the recognition of texts, even printed, by machines (Huang & Chung, 1987) much more difficult than that of alphabetic languages, where the syllables are one-dimensional combination of a small set of letters. However, Chinese characters are not drawings that completely independent of each other (see, Mackay, 2001; Feng, 2001, for recent discussion). In fact, many characters share smaller units called elements. Elements are often related to the meaning or pronunciation of the whole characters and appear frequently in related scripts. For example, those shown in the first row of Fig. 7A share an element in the left side, meaning 'water'. Most characters can be considered as the combination of such two-dimensional building blocks. The 24 characters in Fig. 7A can be derived from 11 elements. If these elements are recognized first and each character is represented in terms of combinations of elements, the problem can be made easier by a pseudo-alphabetic approach.

We suggest that techniques such as NMF and NA might have potential application in improving the efficiency in the recognition of Chinese characters. Fig. 7 gives some preliminary results with a very limited number of characters. Each is transformed into a bitmap image of  $60 \times 60$  pixels. A network with 3600 input units and 12 hidden units is applied. It approximately learns the 11 elements occurring in the training set, even though some of them overlap heavily with one another (non-orthogonal vectors).

Our purpose here is just to show that it is perhaps possible to use parts-based representations for the recognition of Chinese characters. The set of instances was chosen to make the learning easy. General Chinese characters are more complicated. Also, scripts were carefully written by hand to make the same element always appear in the same position in different characters, but elements are often transformed to keep the whole in balance for a better sense of beauty. For practical applications further study is needed.

### 6.2. Mixing faces with Chinese characters

Until now, independent neural networks are used for recognizing faces and Chinese characters. What will happen if we include these two different kinds of stimuli in one training set and present it together to one neural network? This can be easily tried, as both are databases of images. The Chinese characters in Fig. 7 are scaled into a  $19 \times 19$  grid in accordance with the format of facial images. These two databases are then mixed up. The resultant database consists of 191 faces and 24 Chinese characters. From this database, an NA with 40 hidden units learns an internal representation given in Fig. 8.

Surprisingly, the network still functions fairly well. Some hidden units learn to represent parts of faces while others adapt to elements of Chinese characters. Some characters are not decomposed as expected; the whole character is learned as a prototype. But this can be understood considering the relatively small sample size in the side of characters and the biased fine tuning of learning parameters in favor of faces. At least, one can say that the inclusion of Chinese characters do not interfere much with the learning of parts of faces.

The non-interference phenomenon is the consequence of the sparseness of the encoding. To represent a given image, NA requires only a small number of basis images. Therefore each basis image needs only to account for the localized features shared by a subset of instances in the training set and may be independent of the remaining instances. NMF and NA are able to extract features shared by some instances in the training set, not necessarily all of them. This makes it possible for NA to learn well from a mixture of different stimuli.

For PCA-like methods, such robustness is obviously impossible because principal components are those vectors that point to maximum co-variance and every instance counts in the calculation of co-variance. The non-inference property is another fundamental difference between non-negative approaches and orthogonal approaches. This property is interesting because when various kinds of stimuli are presented as input, only relevant hidden units will be active. The network is able to recall correctly according to the contents of the stimuli.



the non-negativity constraints should be encouraged. Therefore, we regard the constraints as one of the essential characteristics that lead to the localized representation.

But it is not proper to attribute all the properties of NA to constraints. The network architecture is also very important. Basically, NA can be understood as an information compression and retrieval network. In fact, this is also the design perspective of NMF and ‘wake–sleep’ algorithm. For the former, it is evident from our description in Section 2. In the latter, bottom-up recognition connections converts the input into representations in hidden layers and top-down generative connections then reconstruct the representation. The recognition and generative connections in the algorithm are similar to the input-to-hidden and hidden-to-output connections in auto-associators, respectively. So the three algorithms have very similar architecture for the information flow, which is important for the learning.

As an iterative matrix factorization algorithm, NMF should be distinguished the other two neural network models. The two network models differ in several aspects. First, ‘wake–sleep’ network uses stochastic neurons whereas neurons of NA are deterministic. Second, only the generative connections are constrained to be positive in the former; the recognition connections are still allowed to be negative. In NA both are positive. Finally, and most importantly, ‘wake–sleep’ algorithm is for unsupervised learning, whereas NA should still be considered as supervised learning because of the use of error back-propagation algorithm to communicate error signal to all connections. Therefore, ‘wake–sleep’ algorithm is more plausible in the biological basis.

## 8. Conclusions

It is found that the non-negativity constraints on neural activity and synaptic strengths can lead to sparsely distributed, parts-based representations in simple neural networks. In this context, a NA model successfully learns the parts of faces and Chinese characters.

Quite unexpectedly, NA trained by error back-propagation algorithm has some ability to minimize the number of parts necessary for representing a given data set. In the experiments of recognizing independent bars from images, the network finds the exact solution in over-complete case. The mechanisms for this robustness are discussed in terms of the normalization constraint and learning dynamics.

It is demonstrated also that the network still works well when a mixture of faces and Chinese characters are presented. This is possible because the basis images learned under the non-negativity constraints are localized features shared by some examples in the training set, not necessarily all of the examples. The non-interference property is thus related to the sparseness in encoding.

For further study, we would consider the application of the parts-based representations of faces discovered by NA in various tasks such as identification and sex distinction.

## Acknowledgements

We would like to thank Masaharu Nakazawa, Kazuo Furuta, Shinobu Yoshimura, Jerome Piat, Yiming Mi, Shin-ichi Yonamine, and Naohiro Shichijo for very stimulating discussions. We would also like to thank Alex Martinez and Robert Benavente of Purdue University, F.S. Samaria of AT & T Laboratories, Cambridge, and Bernard Achermann of University of Bern for kindly make available their databases of face images. Finally, we are indebted to the anonymous reviewers for many suggestions that improved the formation of our manuscript.

## References

- Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2, 53–58.
- Barlow, H. B. (1959). Sensory mechanisms, the reduction of redundancy, and intelligence. *National Physical Laboratory Symposium No. 10, The Mechanisation of Thought Processes*. Her Majesty’s Stationery Office, London.
- Barlow, H. B. (1989). Unsupervised learning. *Neural Computation*, 1, 295–311.
- Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychology Review*, 94, 115–147.
- Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59, 291–294.
- Chueinta, W., Hopke, P. K., & Paatero, P. (2000). Investigation of sources of atmospheric aerosol at urban and suburban residential areas in Thailand by positive matrix factorization. *Atmospheric Environment*, 34, 3319–3329.
- Cottrell, G.W. & Fleming, M. (1990). Face recognition using unsupervised feature extractor. *Proceedings of the International Neural Network Conference* (pp. 322–325). Paris: Kluwer.
- Cottrell, G. W., Munro, P., & Zipser, D. (1989). Image compression by back propagation: An example of extensional programming. In N. E. Sharkey, *Models of cognition: A review of cognitive science* (pp. 208–240), Vol. 1. Norwood, NJ: Ablex Publishing.
- Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz machine. *Neural Computation*, 7, 889–904.
- Ellman, J. L. & Zipser, D. (1987). *Learning the hidden structure of speech*. Technical Report No. 8701. San Diego: Institute for Cognitive Science, University of California.
- Feng, J. (2001). Entropy illustrates the flexibility of Chinese. *Nature*, 410, 1021.
- Frey, B. J. (1997). A simple algorithm that discovers efficient perceptual codes. In M. Jenkin & L. R. Harris, *Computational and psychophysical mechanisms of visual coding*. New York: Cambridge University Press.
- Ge, X. (2000). PhD Thesis. *Extracting knowledge from databases by redundancy reduction*. The University of Tokyo.
- Gluck, M. A., & Myers, C. E. (1993). Hippocampal mediation of stimulus representation: A computational theory. *Hippocampus*, 3, 491–516.
- Gori, M., Lastrucci, L., & Soda, G. (1996). Autoassociator-based models for speaker verification. *Pattern Recognition Letters*, 17, 241–250.
- Hanson, S. J. & Kegl, J. (1987). Parsnip: A connectionist network that learns natural language grammar from exposure to natural language sentences. *Proceedings of the Ninth Annual Conference on Cognitive Science*, 1987.
- Hecht-Nielsen, R. (1995). Replicator neural networks for universal optimal source coding. *Science*, 269, 1860–1863.

- Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The wake–sleep algorithm for unsupervised neural networks. *Science*, *268*, 1158–1161.
- Huang, J. S., & Chung, M. -L. (1987). Separating similar complex Chinese characters by Walsh transform. *Pattern Recognition*, *20*, 634–647.
- Japkowicz, N., Hanson, S. J., & Gluck, M. A. (2000). Nonlinear auto-association is not equivalent to PCA. *Neural Computation*, *12*, 531–545.
- Kramer, M. A. (1991). Nonlinear principal component analysis using auto-associative neural networks. *AIChE Journal*, *37*, 233–243.
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.
- Mackay, A. L. (2001). Character-building. *Nature*, *410*, 19.
- Martinez, A. M. & Benavente, R. (1998). *The AR Face Database*. CVC Technical Report #24, June 1998.
- Namphol, A., Chin, S. H., & Arozullah, M. (1996). Image compression with a hierarchical neural network. *IEEE Transactions of Aerospace and Electronic Systems*, *32*, 326–338.
- Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, *37*, 23–35.
- Paatero, P., & Tapper, U. (1994). Positive matrix factorization: a nonnegative factor model with optimal utilization of error-estimates of data values. *Environmetrics*, *5*, 111–126.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). In D. E. Rumelhart & J. L. McClelland, *Parallel distributed processing* (p. 318), Vol. 1. Cambridge, MA: MIT Press.
- Samaria, F. S. & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. *Proceedings of the Second IEEE Workshop on Applications of Computer vision*. December 1994, Sarasota, FL.
- Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *The Journal of Cognitive Neuroscience*, *3*, 71–86.
- Valentin, D., & Abdi, H. (1996). Can a linear autoassociator recognize faces from new orientations? *Journal of Optical Society of America A*, *13*, 717–724.
- Valentin, D., Abdi, H., O’Toole, A. J., & Cottrell, G. W. (1994). Connectionist models of face processing: A survey. *Pattern Recognition*, *27*, 1209–1230.

### Further reading

<ftp://iamftp.unibe.ch/pub/Images/FaceImages>. Copyright 1995, University of Bern.