UC Berkeley

UC Berkeley Previously Published Works

Title

Pattern Recognition by a Distributed Neural Network: An Industrial Application

Permalink

<https://escholarship.org/uc/item/4gh1f4cz>

Journal

Neural Networks, 4(1)

Authors

Yao, Yong
Freeman, Walter J, III
Burke, Brian
[et al.](#)

Publication Date

1990-06-13

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/3.0/>

Peer reviewed

ORIGINAL CONTRIBUTION

Pattern Recognition by a Distributed Neural Network: An Industrial Application

YONG YAO, WALTER J. FREEMAN, BRIAN BURKE AND QING YANG

University of California at Berkeley

(Received 20 November 1989; revised and accepted 13 June 1990)

Abstract—In this report, a distributed neural network of coupled oscillators is applied to an industrial pattern recognition problem. The network stems from the study of the neurophysiology of the olfactory system. It is shown that the network serves as an associative memory, which possesses chaotic dynamics. The problem addressed is machine recognition of industrial screws, bolts, etc. in simulated real time in accordance with tolerated deviations from manufacturing specifications. After preprocessing, inputs are represented as 1×64 binary vectors. We show that our chaotic neural network can accomplish this pattern recognition task better than a standard Bayesian statistical method, a neural network binary autoassociator, a three-layer feedforward network under back propagation learning, and our earlier olfactory bulb model that relies on a Hopf bifurcation from equilibrium to limit cycle. The existence of the chaotic dynamics provides the network with its capability to suppress noise and irrelevant information with respect to the recognition task. The collective effectiveness of the “cell-assemblies” and the threshold function of each individual channel enhance the quality of the network as an associative memory. The network classifies an uninterrupted sequence of objects at 200 ms of simulated real time for each object. It reliably distinguishes the unacceptable objects (i.e., 100% correct classification), which is a crucial requirement for this specific application. The effectiveness of the chaotic dynamics may depend on the broad spectrum of the oscillations, which may force classification by spatial rather than temporal characteristics of the operation. Further study of this biologically derived model is needed to determine whether its chaotic dynamics rather than other as yet unidentified attributes is responsible for the superior performance, and, if so, how it contributes to that end.

Keywords—Autoassociator, Back propagation, Chaos, Collective effectiveness, Distributed neural network, Feature enhancement, Neural network industrial application, Minimum distance classifier, Nonlinear dynamics, Olfactory system, Pattern recognition.

1. INTRODUCTION

This is a continuation of an exploration of pattern recognition capabilities of the olfactory bulb model (Freeman, Yao, & Burke, 1988) and the distributed olfactory system model (Yao & Freeman, 1989, 1990).

Acknowledgement: Support from grants MH06686 from NIMH and AFOSR-87-0317 is gratefully acknowledged. The authors wish to thank Dr. Shawn Buckley from Cochlea Corporation in San Jose, California for kindly providing the data. The discussion with him was very stimulating and helpful. The computer simulation of this work is accomplished by using Cray X-MP/14 at the University of California at Berkeley. We are indebted to the 1988 and 1989 Cray Sponsored University Research and Development Grant Program. The current revised version has benefited from helpful comments and suggestions by an anonymous action editor.

Requests for reprints should be sent to Walter J. Freeman, The Division of Neurobiology, Department of Molecular and Cell Biology, University of California at Berkeley, CA 94720.

For many years, efforts have been made within a large research community to capture the principles by which biological systems accomplish pattern recognition (Amari, 1967, 1977; Anderson, 1972; Cooper, 1973; Freeman, 1960, 1975; Fukushima, Miyake & Ito, 1983; Grossberg, 1976, 1980; Hebb, 1949; Hopfield, 1982, 1984; Kohonen, 1972, 1988; Lippmann, 1987; Mead, 1989; Minsky & Papert, 1969; Pao, 1989; Pitts & McCulloch, 1947; Rosenblatt, 1958, 1962; Rumelhart & McClelland, 1986; Sejnowski & Rosenberg, 1986; Widrow & Hoff, 1961). In psychology, pattern recognition is defined as the process by which external signals arriving at the sense organs are converted into meaningful perceptual experiences (James, 1890). Our own perceptual experience and research tell us that the process of recognizing a pattern should at least include the following procedures. The stimulus pattern must be transduced by sense organs, and converted into a neural signal, which is enhanced by noise depression, information

compression, and amplification of the main features of the pattern. The prototype of the pattern must have been established by previous observation of multiple examples in the same class. This is most likely done by selective changes in synaptic strengths within the cortex. Finally, the altered synaptic connectivity gives rise to the association between each input pattern and its prototype. The association can be understood as a dynamic flow from an initial state to its "nearest" attractor or "wing" constituting the output, and the attractor or the wing is accessed by the input (Freeman, 1990a; Yao & Freeman, 1990).

A complete pattern recognition system usually consists of a transducer, preprocessor, feature extractor, and classifier (Duda and Hart, 1973). Pattern recognition systems based on the perceptron introduced by Frank Rosenblatt in 1958 (Rosenblatt, 1958, 1962; Minsky & Papert, 1969) operate by relaxation to one of a collection of equilibrium states, constituting the minimization of an energy function. Biological pattern recognition systems do not go to equilibrium and do not minimize an energy function. Instead, they maintain continuing oscillatory activity, sometimes nearly periodic but most commonly chaotic. This activity has the form of a spatial pattern of amplitude of a shared chaotic carrier wave over a distributed population of neurons (Freeman, 1990b). The function that is minimized is not an energy level; it is best described as a difference in pattern between an on-going event and a collection of prototypical spatial patterns, which we describe as lobes or wings of a global chaotic attractor (Yao & Freeman 1990). Except that we introduce a chaotic associative memory into our pattern recognition system, the system shares similar features with other systems based on neural networks. For example, since each value of the pattern is a scalar amplitude, each pattern can be expressed as a vector or point in the measurement space. The function that is minimized is a Euclidean distance between a sample point and one of the prototype points in hyperspace.

Our work has been directed toward simulating the pattern recognition capabilities of oscillatory systems, first using limit cycle attractors (Freeman et al., 1988) and now using chaotic attractors (Yao & Freeman, 1989, 1990). The approach is closely related to the analogy by Haken (1987) between pattern recognition and nonequilibrium phase transitions in fluids and lasers. According to his "slaving principle", when a chaotic system is raised in energy and brought close to a separatrix, a small fluctuation spreads rapidly and entrains the entire system into a coherent spatial pattern. Our experiments with chaotic dynamics were designed to test the hypothesis that small input signals fed into our chaotic system would rapidly switch it from one basin or wing to

another, depending on the sensitivities built into the connections by learning, thereby demonstrating amplification of the effect of the inputs by the chaotic process as an essential intermediary step in pattern recognition.

In this paper, we develop an algorithm that allows a machine to accept inputs in the form of a 64×1 vector, to enhance, compress, and amplify the signals, and to "learn" to classify objects as acceptable or not. We compare the efficacy of our model using chaotic attractors with our previous model using limit cycle attractors, with a binary autoassociator using point attractors, with standard statistical approaches, and with a three-layer back propagation network (Notice: The data sets are available on request on floppy disc in ASCII.)

2. PROBLEM FORMULATION

Figure 1 illustrates several examples of the objects which the machine is going to recognize. The two objects at the top are machine screws. The two objects in the middle are ball studs. The two objects at the bottom are Phillips tips. There are two classes of objects: acceptable and unacceptable. The key requirement for the machine in this application is to get rid of the unacceptable objects (i.e., the classification rate for the unacceptable objects has to be 100 percent). Given this constraint, we minimize the error rate for classifying acceptable objects.

There are four sets of data available for developing our new algorithm. Set 1 consists of data for

FIGURE 1. Samples of the recognition objects provided by Cochlea Co. through Dr. Shawn Buckley (see the text for the description).

20 unacceptable machine screws and 50 acceptable machine screws. Set 2 consists of data for 10 unacceptable ball studs and 50 acceptable ball studs. Set 3, and Set 4 are all composed of data for 50 unacceptable Phillips tips and 50 acceptable Phillips tips. The difference between Set 3 and Set 4 is that the former is the data set with the parts sliding tips first, and the latter is the data set with the parts sliding tip last. The data are provided by the Cochlea Corporation, and were obtained by using a unique sensing technique (Buckley, 1975; Buckley & Stelson, 1978; Tavormina & Buckley, 1978). After sensing, each object is represented by a real vector in a 64-dimensional space. The components of the vector are phase values (from $-\pi$ to π).

The task of recognizing these patterns is graded in complexity owing to intrinsic variability of the part under observation. The machine-screws in Figure 1 are nearly symmetrical. The acceptability is detected regardless of how the screw slides down the chute. The acceptable ball stud has a hole, but the hole can be in different rotational positions for each way it slides down the chute (in some positions it "looks" just like a bad one, i.e., one with no hole at all). The Phillips tips are most unsymmetric because there is no relation between the position of the hex shape to the Phillips tip. The angle between the Phillips tip and the chute is different for each part and unknown. It is easier to tell the fine from the coarse if the part slides tip first than if the part slides tip last because in the latter case the position of the tip differs when the part breaks the acoustic beam. Thus among these four data sets Set 4 is the most difficult, Set 3 is the second, Set 2 is the third, and Set 1 is the easiest.

Let us consider Set 1 composed of 70 cases (20 cases of the unacceptable machine screws, and 50 cases of the acceptable machine screws). From a geometrical point of view, each case is a point in a 64-dimensional space. Thus the 70 cases may be denoted as $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{70}\}$. According to the terminology of the pattern recognition theory, the space is called a feature space, and the representation of each machine screw is 64 features. It is observed that *the phase values of each feature are normally distributed*. We assume that the number of machine screws is large enough to give reliable means and standard deviations for those 64 channels of the data. Owing to the nature of the phase values, the first consideration is to check each feature for a split distribution (values across $-\pi$ and π). Since $-\pi$ and π , in fact, are the same point with respect to the definition of phase, if a channel distribution is split at the point, then $-\pi$ is added to those values greater than zero and π is added to those values less than zero to correct this split. How can we know whether a feature is split at π or not? In the algorithm we check the

standard deviation for each feature. If the value of the deviation exceeds a prescribed number (here, 1.0 radians is adopted), we consider the feature is split at $-\pi$, and shift it.¹

For this particular recognition task, an additional procedure is also used to prune some irrelevant or redundant information caused by the sensing technique. The procedure is described as the following: Check the values of the standard deviation of the 64 features one by one among the training cases. If the value of the deviation is larger than a prescribed value (here, 0.5 radian is adopted), then we ignore the feature by simply setting the value of the feature at zero. The reason for doing this is that the high deviation of the corresponding sensing channel implies excessive noise. In other words, we assume the sensing channel does not provide useful information. This pruning procedure may be considered as a simplified realization of the habituation process in biological sensory systems. It is well known from statistical decision theory that the probability of classification error or risk should increase when fewer measurements are taken into consideration (Ferguson, 1967). However, this is true only for infinite sample sets for which the errors on estimates of the pattern recognition system parameters can be ignored. In this case only finite training cases are available and consequently estimation errors will no longer be negligible, and only those suspect measurements are pruned. Thus this additional procedure should lower the error rate. After the shifting and pruning procedures, the data are ready for all the methods tried below.

For the 70 cases in Set 1, suppose only the first five and the 21st to the 30th cases are labeled (i.e., the first five cases are known as unacceptable machine screws, and the cases from the 21st to the 30th are known as acceptable machine screws). The labels of the rest of the machine screws are unknown. Then the classification of the 70 cases in Set 1 becomes a supervised learning, or learning with a teacher problem since the data of those known screws can be used as training sets. Using the first five cases, we can calculate the centroid for the class of the unacceptable screws $\mathbf{x}^U$ by simply averaging them

$$\mathbf{x}_i^U = \frac{1}{5} \sum_{j=1}^5 \mathbf{x}_i^j, \quad i = 1, 2, \dots, 64. \quad (1)$$

¹ If the shifting does not minimize the standard deviation of the feature, then keep the original values, which implies that the wide distribution of the values has nothing to do with the splitting. It is the intrinsic variability of the feature. In addition, the prescribed number is not unique. For instance we may use 1.5 instead of 1.0. In this report, we are not going to discuss the optimal value of any particular parameter, and are mainly interested in principles.

The centroid for the acceptable screws $\mathbf{x}^A$ can be obtained by averaging the cases from the 21st to the 30th, too. Thus each class can be presented by its centroid. Since the distribution of the values of each feature is known to be Gaussian, a minimum distance classifier with respect to the two centroids implements a linear separation in the 64-dimensional feature space, and is optimal in the sense of minimizing the expected probability of misclassification according to decision theory. By using the minimum distance classifier under Euclidean distance metric, we obtain the classification result, which is presented in Table 1. The same procedure also yields the results for the other three data sets (see Table 1), where the first five and the 11th to 20th, the first ten and the 51st to the 60th, the first ten and the 51st to the 60th are used for centroids for the data sets Set 2, Set 3, and Set 4, respectively.

The results (see Table 1) are not satisfactory except the results for Set 1. The two classes within the data sets Set 2, Set 3, and Set 4 are not linearly separable. In order that the data be well separated by the classifier, the "quality" of the data should be enhanced in the sense of minimizing the within-class pattern variability while maximizing the between-class pattern variability. This is a so-called feature extraction problem. We will show that the distributed olfactory system model (see Figure 2) can solve the problem.

3. RELATED MODEL PROPERTIES (see Appendix for the Mathematical Description of the KIII Model)

Figure 2 illustrates a distributed neural network which is a K-model of the olfactory system derived from Freeman (1987). The subnetwork inside the dotted rectangular box serves as an olfactory bulb model. It is in the form of an array of 64 coupled oscillators forming an array with 64 input lines and 64 output lines in parallel. It is interconnected with lines for

mutual excitation, which are modifiable under a simplified Hebbian rule for learning. It is also interconnected with lines for mutual inhibition, which provide competitive contrast enhancement and controlled destabilization, and which are optimized and fixed. Output lines provide for read-out in parallel. The output lines also provide for spatial integration, coupling through subsidiary oscillators representing the target areas of cortical projection. There are multi-channel nonlinear feedbacks with delay to the array. Hereafter, the notation of the KIII model refers to the distributed network shown in Figure 2, and the notation of the KII model refers to the subnetwork inside the box in Figure 2 when the feedback pathways are cut off. The characteristics of those models have been described in detail (KII-Freeman et al., 1988; KIII-Yao & Freeman, 1990; and Freeman & Yao, 1990). Because of the scaling invariant property of both of the models, those characteristics found in low-dimensional cases are preserved when the number of channels of the models is extended up to 64.

The output of the models is retrieved and laundered by a process of spatial integration, which removes noncommon activity and enhances the co-operative activity at the common instantaneous frequency. Figure 3 shows two samples of the outputs. One is the impulse response from the KII model, another is from the KIII model under a smooth step input. The channels which receive the input or are associated with the templates respond with high amplitude oscillations (see Figure 3a) or high DC shifts (see Figure 3b). The learning rule used to make templates is a simplified Hebbian rule called input correlation (Freeman et al., 1988). The central idea of Hebbian learning is that a global network may learn by constructing internal representations of patterns in the form of "cell-assemblies" that learn to support one another's activities (Hebb, 1949). Pattern recognition in a biological system is a *collective effectiveness*. The strengthened Hebbian connections emphasize the ensemble. Let only two templates A

TABLE 1
Direct Classification of the Data by Means of the Minimum Distance Classifier Under Euclidean Distance Metric

Direct Classification for the Data Sets: Set 1, Set 2, Set 3, Set 4 After Shifting Split Features and Pruning Irrelevant Information								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-10	100%	100%	0.09	0.13	0.83	7.71
Set 2	1-5	1-10	80%	50%	0.02	0.10	0.11	1.52
Set 3	1-10	1-10	52%	78%	0.33	0.31	0.51	1.59
Set 4	1-10	1-10	56%	56%	0.33	0.48	0.72	1.74

Each centroid is found by averaging its corresponding training cases. The data are shifted if necessary to prevent split distributions around $+\pi$ and $-\pi$. Channels with excessive variance are set to zero. In Set 3, and Set 4, there are 10, and 11 channels are set to zero, respectively. Set 1 and Set 2 are not affected by this setting.

FIGURE 2. A topological diagram of the olfactory system with a distributed OB layer (N units), an lumped AON layer and a lumped PC layer (cited from Yao & Freeman, 1989). L_1 , L_2 , L_3 , L_4 simulate the global feedback pathways among the olfactory tissues. The top arrows come from the olfactory receptors, and the bottom arrow goes to the external capsule.

FIGURE 3. The examples of 64-array outputs from (a) the KII model with 100 ms time duration, (b) from the KIII model with 200 ms time duration.

and B be built in the subnetwork in Figure 2. Then an input covering more components of the template A move the output closer to point A than to point B.

In Figure 2, each circle denotes a pool of individual neurons, which is modeled as a second-order linear ordinary differential equation cascading with a threshold logic unit (i.e., sigmoid function). The threshold function devalues the role of single channels by bounding values between 0 and 1. For example, suppose all components of an input vector are exactly the same as those of the template A except input on one channel is far from the corresponding template channel for A. Thereby in the original feature space the input may be closer to the template B although the input is totally different from B. In this case the KII model moves the output closer to point A. The Hebbian learning in the mutually excitatory path and the threshold function of the neuronal pools together enhance the learned features of the input. In addition, the sigmoid curve is asymmetric, which distinguish the system from most other neural networks with symmetric sigmoid curves.

Since the convergence in the models is independent of their initial conditions, and the dynamics of the models is driven by their inputs, the models may be used to process an uninterrupted sequence of inputs. The simulated speed of process depends on the convergence time. In the KII model, the convergence time is about 50 ms. The next 50 ms series is used to get RMS values. In the KIII model, the convergence time is about 100 ms, and the next 100 ms are needed to get the means of the DC shifts (see Figure 3b). Thus the processes in the KII model, and in the KIII model require the simulated durations of 100 ms, and 200 ms per case, respectively. In other words, the models can be run at rates of five to ten patterns per second. The training of the systems is by presenting prototypes on one trial and does not require repeated runs to converge to optimized performance.

Very frequently, patterns in the same class differ a great deal owing to the natural variability, distortion, and noise of the data. The feedback pathways from the prepyriform cortex to the inner bulb and from the anterior olfactory nucleus to the inner as well as the outer bulb in the KIII model smooth irrelevant information and noise in the recognition task. It is the low-pass filter function of these feedback pathways that assists the biological system in pattern recognition in noisy, unpredictable and time-varying environments. On the other hand, the feedback pathways introduce chaos into the system. Chaos provides a way for a living organ to stay at an unpatterned state while there is not significant stimulus. The unpatterned state might be a "separatrix" of the system (analogous to a critical state between gas state and liquid state of water in physics) so that a small input could drive the system to a patterned state.

4. BUILDING PROTOTYPE PATTERNS

In this section we discuss how to build prototype patterns for this specific recognition problem. The prototype patterns are the basis of centroids in classification and the basis of establishing templates in the models. Let $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ be N data, which are shifted if their distributions at π are split, and certain features are already set at zero value if their standard deviations exceed 1.0. Suppose that we know the first m_1 data are acceptable and the second m_2 data are unacceptable. The rest of the data are unknown, and need to be classified. In this situation two prototypes (acceptable and unacceptable) need to be constructed one for each class

A. Statistical Approach

This method is based on the fact that the values of the 64 features are normally distributed, and the number (i.e., $m_1 + m_2$) of the training cases is large enough to determine the means and the standard deviations for the 64 features.

(i) *Normalizing.* For all the training cases $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{m_1+m_2}\}$, find the mean (mean_j) and the standard deviation (SD_j) for each feature (j). Then all the N vectors (or N cases) $\mathbf{x}^i = (x_1, x_2, \dots, x_{64})$ are converted into N normalized vectors $\mathbf{y}^i$, $i = 1, 2, \dots, N$ by the following transformation

$$y_j = \frac{x_j - \text{mean}_j}{\text{SD}_j}, \quad j = 1, 2, \dots, 64. \quad (2)$$

Thus $\{\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^N\}$ is a data set with a zero mean and a unit standard deviation.

(ii) *Averaging.* Pick the first m_1 normalized vectors, which are the acceptable objects, and the second m_2 normalized vectors, which are unacceptable objects. Find out the center point $\mathbf{y}^A$ of the acceptable objects by simply averaging them

$$y_j^A = \frac{1}{m_1} \sum_{k=1}^{m_1} y_j^k, \quad j = 1, 2, \dots, 64. \quad (3)$$

In the same way, the center point for the unacceptable objects, $\mathbf{y}^U$ can be found out, too.

(iii) *Binarization.* All of the normalized data $\{\mathbf{y}^i, i = 1, 2, \dots, N\}$ as well as $\mathbf{y}^A$ and $\mathbf{y}^U$ are transferred to 0–1 vectors denoted $\{\mathbf{z}^i, i = 1, 2, \dots, N, A, U\}$ by the following rule: If y_j^i is positive, $z_j^i = 1$; otherwise $z_j^i = 0$. Here $\mathbf{z}^A$ and $\mathbf{z}^U$ are the two prototype patterns.

B. Geometrical Approach

The prototype patterns built by the above statistical method rely on the statistical properties of the data.

These two patterns may have some overlap (i.e., both patterns have 1 at some channels). The following approach is purely geometrical.

(i) *Averaging.* Pick the first m_1 vectors $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{m_1}\}$, which are the training cases of the acceptable objects, and the second m_2 vectors $\{\mathbf{x}^{m_1+1}, \mathbf{x}^{m_1+2}, \dots, \mathbf{x}^{m_1+m_2}\}$, which are the training cases of the unacceptable objects. Find out the center point $\mathbf{x}^A$ of those acceptable objects, and the center point $\mathbf{x}^U$ of those unacceptable objects by means of the following equation

$$x_j^A = \frac{1}{m_1} \sum_{k=1}^{m_1} x_j^k, \quad x_j^U = \frac{1}{m_2} \sum_{k=m_1+1}^{m_1+m_2} x_j^k, \\ j = 1, 2, \dots, 64. \quad (4)$$

(ii) *Binarization.* All data $\{\mathbf{x}^i, i = 1, 2, \dots, N\}$ as well as $\mathbf{x}^A$ and $\mathbf{x}^U$ are transferred to 0–1 vectors denoted $\{\mathbf{z}^i, i = 1, 2, \dots, N, A, U\}$ by the following rule: If $x_j^i > (x_j^A + x_j^U)/2$, $z_j^i = 1$; otherwise, $z_j^i = 0$. Here $\mathbf{z}^A$ and $\mathbf{z}^U$ are the two prototype patterns. There is no overlap between these two patterns.

Obviously there are other ways to build these two prototype patterns. Here, however, our discussion is restricted to the above two methods. The proto-

type patterns are then used as centroids for a direct classification. By the direct classification we mean that there is no involvement of neural network models. In the model approach, the prototype patterns are used to make templates (i.e., altering the connection matrix $\mathbf{K}_{mm}$ in the KII or the KIII, $\mathbf{T}$ in a binary autoassociator). Our experience tell us that nonoverlapping templates are superior to overlapping templates. Thus it is expected that the models with the templates from the geometrical approach will have a better performance than those from the statistical approach. In a backprop network, one way to adjust weights is to use these prototype patterns as its training vectors.

5. PATTERN RECOGNITION ALGORITHM

A system to recognize the industrial objects might well have the form shown in Figure 4. The transducer takes the phase information of the acoustic scene of an object and passes information to the feature extractor, whose purpose is to represent the scene by measuring certain “features” or “properties” that distinguish the acceptable objects from the unacceptable objects. These features (or, more precisely the values of these features) are then passed to a

FIGURE 4. An algorithm for the industrial data recognition. The olfactory model is used to enhance the features. The method for building prototypes is the geometrical method (see the text for the detail).

classifier that evaluates the evidence presented and makes a final decision about the objects. The feature extractor has thus reduced each scene to a point or a feature vector in a 64-dimensional feature space. The aim of the feature extractor is to find out an appropriate set of features so that the classifier can partition the feature space into two regions, where all the points in one region correspond to the acceptable objects, and all points in another region correspond to the unacceptable objects. If the proper features have been extracted, points corresponding to objects coming from the same class will hopefully be close to each other in terms of geometrical distance.

An ideal feature extractor would make the job of the classifier trivial, and an omnipotent classifier would not need the help of the feature extractor. The question is: Is there such a feature extractor or such a classifier for this industrial data recognition? Here a minimum distance classifier is adopted. Table 1 suggests that a feature extractor is necessary for the data although the classifier adopted is optimal for this recognition problem according to the decision theory. In Figure 4 the KII model is introduced into the feature extractor to enhance the "quality" of the features. The feature extractor shown in Figure 4 consists of several steps. The input fed into the model is a 64-dimensional binary vector. In Figure 4 from the step of binarization to the step of feature enhancement, there is a subpath. The subpath denotes the learning procedure. This procedure is simulated by establishing templates in the model.

In short the transducer senses the input and converts it into a form suitable for machine processing. The feature extractor extracts presumably relevant information from the input data forming an appropriate feature space. The classifier uses this information to assign the input data to one of the two categories.

Since in Figure 4 the prototype patterns are built by means of the geometrical method, the whole algorithm is then purely geometrical. The geometrical approach focuses on finding data representations or organizations within the associated data structure. It is based on the following hypothesis: A slight distortion of a given pattern should produce a small displacement of its representation. Two different patterns should be assigned to the same class on the basis of their similarity, and to different classes on the basis of their dissimilarity. Here classification is done in 64-dimensional Euclidean space by calculating an "unacceptable centroid" and an "acceptable centroid". The outputs of the KII model under the inputs of the binarized training cases are used to build the two centroids. Each case is classified as belonging to the class for which its Euclidean distance is the least.

6. SIMULATION RESULTS

In order to make reasonable comparisons among different approaches within the limits of available computation time, we only use 20 cases for each data set. That is, the first ten unacceptable objects and the first ten acceptable objects are considered for each data set since in Set 2 only ten unacceptable cases are available. The prototype pattern, or the centroid of the unacceptable objects is built by using the first five unacceptable cases for each data set. In the same way, we may build the prototype pattern of the acceptable objects. The comparisons are made in two ways. One is to compare the results with respect to using a binary autoassociator (point attractor) model, or a three-layer backprop network, or using the KII (limit cycle) model, or using the KII (chaos) model, or with neither model. Another is to compare the results using the geometrical method with statistical method.² Figure 5 illustrates all the methods we have tried for recognizing these four data sets. The wording along each of the vertical arrows indicates the specific approach by which the corresponding table (or tables) is generated. For example, the middle arrow indicates how Table 1 is obtained. Notice: For 20 cases the results in Table 1 will be 100% for both classes in Set 1; 90% for the unacceptable and 80% for acceptable in Set 2; 50% for both classes in Set 3; 60% for the unacceptable and 70% for the acceptable in Set 4.

A. Recognition Without Using the Models (Tables 2 and 3)

Let the data be binarized by the geometrical method. The results of classifying these binary data with the minimum distance classifier are presented in Table 2. In each data set, the first ten are ten unacceptable objects, while the second ten are ten acceptable objects. The classification rates and the D^* value give rise to the picture of the data structure. Here D^* -value is defined

$$D^*_{\text{-value}} = \frac{D}{\sqrt{\frac{1}{2}(SD_U^2 + SD_A^2)}} \quad (5)$$

where D is the distance between the two centroids, and SD_U (resp., SD_A) is the standard deviation of the ten unacceptable (resp., acceptable) objects. In other words, D^* -value is a measure of how "far"

² Since ten training cases are not enough to calculate the mean value and the standard deviation for each of the features, all of the available cases in each data set are used in the statistical method. In addition, in equation (2), dividing by SD_i does not affect the results of classification in this application. In the algorithm (2) is replaced by $y_i = x_i - \text{mean}_i$.

FIGURE 5. The block diagram of the procedures how each of the tables is generated. For instance, Table 1 is obtained by the procedures of sensing, numericalization, shifting, pruning, and classification.

the two classes are apart in units of the variability. In this case, D^* _value = 9.62. As a comparison to Table 1, the classification rates in Table 2 are improved to some extent.

By using the statistical method to binarize the data, we get quite similar results, which are listed in Table 3.

B. Recognition With the KII Model (Tables 4 and 5)

The KII model is actually composed of uniform oscillators, which are fully interconnected. It is known that the KII model can classify certain artificial data. Here we are very interested in understanding the

TABLE 2
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 Without Using Either the KII Model or the KIII Model (See Figure 5)

Recognition Without the Involvement of the Models for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of the Geometrical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D^* _value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	0.61	0.72	6.41	9.62
Set 2	1-5	1-5	80%	100%	0.64	0.35	3.21	6.22
Set 3	1-5	1-5	80%	50%	0.70	1.36	3.61	3.33
Set 4	1-5	1-5	60%	70%	0.68	0.68	3.19	4.69

In each data set, The first ten unacceptable objects and the first ten acceptable objects are considered. The two centroids are formed by using the first five cases in each class in terms of the geometrical method.

TABLE 3
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 Without Using the Models for the Feature Extraction (see Figure 5)

Recognition Without the Involvement of the Models for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of the Statistical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	0.56	1.00	6.05	7.47
Set 2	1-5	1-5	80%	90%	0.58	0.32	3.19	6.85
Set 3	1-5	1-5	80%	50%	0.78	1.33	3.46	3.18
Set 4	1-5	1-5	60%	70%	0.61	0.59	2.92	4.84

In each data set, the statistical method is adopted to form the two centroids, one for the unacceptable class and another for the acceptable class. The results in this table serve as a comparison with the results listed in Table 2, where the geometrical method is used.

capability of the KII model to improve this industrial data recognition. Feeding the data, which are binarized by the geometrical method, into the KII model, we obtain the results for Set 1, Set 2, Set 3, and Set 4 listed in Table 4. As we can see, the results about Set 2, Set 3, and Set 4 are not enhanced. If the statistical method is introduced, the results are a little bit worse (see Table 5). Here the comparisons are made between the classification results of the input data to the KII model and the output data from the model (i.e., between Tables 2, 3, and Tables 4, 5). The classification is based on the 64-array RMS values of the output from the KII model.

C. Recognition With the KIII Model (Tables 6-8)

The experiences of Buckley's colleagues at Cochlea Co. as well as the above study show the difficulties of recognizing the qualities of the ball studs (Set 2), the Phillips tips (Set 3, Set 4) in Figure 1 by the machine. However, it is easy for humans. There must be principles beyond the studies made so far. We

expect our chaotic network in Figure 2 to capture some of these principles. Let us consider the binarized data under the geometrical method. Table 6 show the results. The information of the distances listed in Table 7 tells us that the two classes of machine screws are well separated. For the data sets Set 2 and Set 3, we have 100% correct classification for the unacceptable objects, and a reasonably high classification rate (80%) for the acceptable objects. For Set 4, if we notice that SD_U is larger than SD_A, we can get 100% classification for the unacceptable objects (i.e., the 8th, 9th, and the 10th objects are clustered to the unacceptable class) and 80% for the acceptable objects (i.e., the 15th and the 20th are misclassified as unacceptable objects) by using the information of the standard deviations (see the discussion section for the details). In addition, this will yield 100% classification for both classes in Set 2, Set 3, where the values of SD_A are larger than the values of SD_U. The classification is based on the 64-array DC offset values of the output from the KIII model. (Notice: Due to the limitation of the

TABLE 4
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the KII Model for the Feature Extraction (see Figure 5), Which Serve as a Comparison with the Results Listed in Table 2

Recognition by Using the KII Model for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of the Geometrical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	1.07	0.92	12.07	12.07
Set 2	1-5	1-5	90%	60%	1.02	1.21	3.50	3.12
Set 3	1-5	1-5	50%	50%	1.34	3.89	5.84	2.01
Set 4	1-5	1-5	70%	70%	2.53	1.88	6.38	2.86

The cases considered are the same as in Table 2. In each data set, the first five unacceptable (resp., acceptable) cases are used to form the unacceptable (resp., acceptable) prototype by means of the geometrical method. The prototypes are then used to establish the templates in the KII model. The five RMS vectors from the KII model under the inputs of the first five unacceptable (resp., acceptable) normalized data are used to form the unacceptable (resp., acceptable) centroid for the classifier.

TABLE 5
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the Help of the KII Model for the Feature Extraction (see Figure 5).

Recognition by Using the KII Model for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of the Statistical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	1.18	1.96	11.26	6.97
Set 2	1-5	1-5	70%	90%	1.03	0.77	3.06	3.35
Set 3	1-5	1-5	50%	70%	1.24	3.70	5.04	1.83
Set 4	1-5	1-5	60%	70%	2.04	0.90	3.55	2.25

In each data set, the statistical method is adopted to form the two prototypes, one for the unacceptable class and another for the acceptable class.

space, distance information in regard to other results is not presented here.)

The results after binarizing by the statistical method are listed in Table 8. They are not as good as we would like. This implies that the way to make the two prototypes is important. If the templates are built according to inappropriate prototypes, some of the inputs will be pushed in a wrong direction.

D. Comparing With a Binary Autoassociator (Table 9 and 10)

This model (Amari, 1972; Anderson, 1972; Aplevich, 1968; Cowan, 1967; Grossberg, 1969; Hirsch, 1989; Hopfield, 1984; Kohonen, 1988; Malsburg, 1973; Sejnowski, 1977; Tank & Hopfield, 1986) now widely known as the Hopfield model has been studied for years. Therefore, comparing with it may give readers a view about the data as well as about the KIII model. We use the same notations as in Hopfield and Tank's paper. Let the input data and the prototype patterns be normalized by the geometrical method. Since only two prototype patterns are needed to be stored in

each of the data sets, the information capacity of the binary autoassociator with 64-channel is enough. In addition since the prototypes are 0-1 vectors, the connection matrix $[T_{ij}]_{64 \times 64}$ is set in the following way:

$$T_{ii} = 0, T_{ij} = \sum_{k=1}^2 [(2V_i^k - 1)(2V_j^k - 1)], \quad i \neq j.$$

The sigmoid curve is implemented by a three-segment piece-wise linear function: $h(x) = (x + u_0)/2u_0$, if $-u_0 < x < u_0$; $h(x) = 0$, if $x < -u_0$; $h(x) = 1$, if $x > u_0$. Here $u_0 = 0.1$. The step length of integration is 0.01. The starting point of the system is the origin. The 200th point of integration is taken as output of the system in response to the first input (the first object). The output is used for classification. For the second object the system needs to be restarted (i.e., to set the values of the 64 variables of the system at zero again to prevent the system from being trapped in the previous equilibrium). The next 200th iterative point is the output for the second object. By repeating the procedure, we can obtain 20 output vectors for each of the data sets. The clas-

TABLE 6
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the Help of the KIII Model for the Feature Extraction (see Figure 5)

Recognition by Using the KIII Model for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of the Geometrical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	3.71	4.70	19.95	4.71
Set 2	1-5	1-5	100%	80%	0.63	3.00	17.43	8.03
Set 3	1-5	1-5	100%	80%	2.23	3.34	23.45	8.26
Set 4	1-5	1-5	70%	100%	0.47	0.44	2.73	5.99

In each data set, the geometrical method is adopted to form the two prototypes. The results in this table serve as comparisons with the results listed in Tables 2 and 4.

TABLE 7
Distance Information Corresponding to the Results Listed in Table 6

Distance Information Under the KIII Model with the Geometrical Preprocessing												
Input #	Set 1			Set 2			Set 3			Set 4		
	D_U	D_A	Labeled #	D_U	D_A	Labeled #	D_U	D_A	Labeled #	D_U	D_A	Labeled #
1	191.16	265.52	0	0.44	290.13	0	6.35	628.55	0	8.28	18.45	0
2	183.14	257.10	0	3.06	360.49	0	67.48	389.19	0	3.41	8.13	0
3	283.69	675.80	0	3.48	271.92	0	6.88	622.57	0	2.18	9.21	0
4	369.89	1124.2	0	1.55	325.20	0	7.14	621.40	0	2.69	8.64	0
5	304.86	999.41	0	0.76	279.80	0	2.72	578.83	0	7.47	16.87	0
6	161.63	576.84	0	7.16	391.98	0	52.21	401.00	0	2.62	4.47	0
7	243.72	953.28	0	3.40	254.79	0	11.55	670.96	0	2.87	5.08	0
8	226.36	666.23	0	1.83	270.53	0	5.38	617.93	0	3.62	3.46	1
9	323.71	614.38	0	1.27	274.96	0	6.77	618.99	0	4.54	3.33	1
10	649.50	927.20	0	5.29	243.38	0	7.72	583.80	0	4.19	3.97	1
11	862.45	378.49	1	1153.2	379.23	1	68.02	299.62	0	7.41	1.25	1
12	874.42	556.15	1	854.42	232.66	1	75.33	285.80	0	9.28	3.55	1
13	1790.1	798.00	1	159.60	123.28	1	1180.6	124.51	1	14.84	2.45	1
14	265.42	228.34	1	189.84	132.30	1	1356.3	194.72	1	18.19	5.58	1
15	644.99	487.12	1	173.87	145.16	1	1075.3	100.92	1	5.21	4.83	1
16	874.64	679.69	1	172.18	130.18	1	1015.7	90.85	1	16.07	6.20	1
17	1310.5	1011.5	1	227.32	71.24	1	1559.5	280.66	1	3.26	1.77	1
18	995.34	666.24	1	160.69	195.28	0	1091.2	107.37	1	10.05	4.23	1
19	831.64	542.06	1	163.14	181.33	0	1013.8	91.43	1	4.88	3.00	1
20	1035.9	726.22	1	188.35	125.05	1	1021.8	93.64	1	4.09	3.17	1

The first ten rows are for unacceptable objects, and the last ten rows are for acceptable objects.

sification results are listed in Table 9. Table 10 is the results in which the input data are normalized in terms of the statistical method. Like the KII model, we cannot get decent classification results by the autoassociator, either.

E. Comparison With a Feedforward Three-Layer Network Under Backprop Learning (Tables 11–13)

Feedforward multilayer networks under the backprop learning are most widely used as pattern recognizers in the field. This comparison might let readers gain further understanding about the KIII model and the nature of the industrial data. The

results presented here are by a specific three-layer network (see Figure 6) under the standard generalized delta rule (Rumelhart & McClelland, 1986). That is, during the learning phase, the weight matrix $\mathbf{w}$ is updated according to the following formula:

$$\Delta \mathbf{w}_j(n+1) = \eta \left(\sum_{k=1}^{T_{\text{train}}} \delta_k^j O_{\text{hidden}}[j] \right) / T_{\text{train}} + \alpha \Delta \mathbf{w}_j(n), \quad j = 1, 2, \dots, 64, \quad (6)$$

where η is a learning rate, α is a momentum constant, T_{train} is the total number of training cases, $O_{\text{hidden}}[j]$ is the output from the j th hidden unit, and $\delta_k^j = (t_k - O_k)O_k(1 - O_k)$, where t_k and O_k are the desired

TABLE 8
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the Help of the KIII Model for the Feature Extraction (see Figure 5)

Recognition by Using the KIII Model for the Data Sets: Set 1, Set 2, Set 3, Set 4 in Terms of Statistical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance Between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1–5	1–5	100%	100%	4.74	6.42	28.17	4.99
Set 2	1–5	1–5	80%	100%	3.60	0.93	15.48	5.89
Set 3	1–5	1–5	60%	50%	0.52	0.48	1.22	2.43
Set 4	1–5	1–5	70%	60%	0.93	0.76	2.83	3.34

In each data set, the statistical method is adopted to form the two prototypes. The results in this table serve as comparisons with the results listed in Table 6, where the geometrical method is used.

TABLE 9
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the Help of a Binary Autoassociator for the Feature Extraction (see Figure 5)

Recognition by Using a Binary Autoassociator for the Data Sets: Set 1, Set 2, Set 3, and Set 4 in Terms of the Geometrical Preprocessing								
Data Set	Cases for Centroids		Classification Rates		Standard Deviations		Distance Between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	0.39	0.46	4.28	9.96
Set 2	1-5	1-5	80%	100%	0.40	0.22	2.02	6.22
Set 3	1-5	1-5	80%	50%	0.43	0.84	2.22	3.34
Set 4	1-5	1-5	60%	70%	0.41	0.42	2.00	4.82

In each data set, the geometrical method is adopted to form the two prototypes. The results in this table serve as comparisons with the results in Tables 2, 4, and 6.

output, and the real output of the network for the k th training vector, respectively. The weight matrix $\mathbf{W}$ is updated according to the following formula

$$\Delta \mathbf{W}_j(n+1) = \eta \left(\sum_{k=1}^{t_{\text{train}}} \delta_{k_j}^n \text{Input}_{k_i} \right) / T_{\text{train}} + \alpha \Delta \mathbf{W}_j(n), \quad i, j = 1, 2, \dots, 64. \quad (7)$$

$\delta_{k_j}^n = O_{\text{hidden}}[j](1 - O_{\text{hidden}}[j])\delta_k^n \mathbf{w}_j(n)$, and Input_{k_i} is the i th component of the k th training vector.

Table 11 shows the random weights which are initially assigned to $\mathbf{w}$, and the weights $\mathbf{w}$ after the backprop learning for the data set Set 1 (Notice: Table 11 does not show the initial weights for $\mathbf{W}$, the learned weight $\mathbf{w}$ for Set 2-4 and the learned weight $\mathbf{W}$ for all the data sets because of the limitation of the space). Table 12 shows the classification results, where $\eta = 0.2$, $\alpha = 0.9$, and the stop condition of learning in the algorithm is that the total output error over the training cases, $T_{\text{error}} = \sum lu_p - O_p I$, is less than 0.01.

All the data presented are reprocessed by the geometrical approach. Firstly for each data set we use the two templates as the corresponding training cases

and use ten unacceptable cases and eight acceptable cases as the test cases. Secondly we use ten cases (five for each class) as the training cases and other ten cases (five for each class) as the test cases. Similarly, we can obtain results under the statistical approach (see Table 13).

The backprop network always gives 100% correct classification for the training cases, but the results for the test cases are not as good as the other methods tried. This does not mean that back propagation networks are not good since Figure 6 is just a special case. To solve this industrial data recognition, the backdrop networks can be more than three layers, the units in the hidden and in the output layers can be more than 64 and 1, respectively, although the size of the input layer is fixed by the particular problem. On the other hand, unlike an associative memory which has a good generalization, the backprop network requires more training cases than the associative memory. In addition, the values of the learning rate η and the momentum constant α are most likely not optimal. Through this study, we did find certain disadvantages of the backprop network. For example, (see Table 12 or 13), the learning time

TABLE 10
Recognition Results of the Data Sets Set 1, Set 2, Set 3, and Set 4 with the Help of the Binary Autoassociator for the Feature Extraction (see Figure 5)

Recognition by Using a Binary Autoassociator for the Data Sets: Set 1, Set 2, Set 3, and Set 4 in Terms of the Statistical Preprocessing								
Data Set	Cases for Centroids		Classification Test		Standard Deviations		Distance Between Two Centroids (D)	D*_value
	Unacceptable	Acceptable	Unacceptable	Acceptable	SD_U	SD_A		
Set 1	1-5	1-5	100%	100%	0.36	0.64	4.00	7.73
Set 2	1-5	1-5	80%	90%	0.36	0.20	2.01	6.85
Set 3	1-5	1-5	80%	50%	0.49	0.84	2.18	2.18
Set 4	1-5	1-5	60%	70%	88.02	80.24	171.59	2.04

In each data set, the statistical method is adopted to form the two prototypes. The results in this table serve as comparisons with the results listed in Table 9.

TABLE 11
(a) The Random Initial Weights Assigned to w. (b) The Weight Matrix w After the Back Propagation Learning for Set 1 Under the Geometrical PreProcessing

Random Initial Weights $w = (w_1, w_2, \dots, w_{64})$							
0.317	0.514	0.631	0.521	0.551	0.429	0.169	0.612
0.895	0.896	0.450	0.460	0.593	0.459	0.351	0.023
0.668	0.859	0.714	0.664	0.242	0.338	0.860	0.066
0.139	0.155	0.247	0.565	0.399	0.536	0.256	0.955
0.964	0.176	0.483	0.634	0.400	0.129	0.811	0.747
0.217	0.087	0.455	0.272	0.052	0.114	0.103	0.756
0.819	0.963	0.986	0.851	0.400	0.493	0.995	0.020
0.664	0.901	0.840	0.692	0.596	0.803	0.293	0.936
Weights w after the Backprop Training							
0.317	4.306	0.591	0.481	0.511	0.388	0.129	0.572
0.855	0.856	0.410	0.420	0.553	0.419	0.310	0.018
0.628	0.818	0.673	0.624	0.202	0.298	0.820	0.025
0.099	0.114	0.207	0.525	0.358	0.496	0.216	0.914
0.923	0.135	0.442	0.594	0.359	0.089	0.770	0.706
0.176	0.046	0.415	0.231	0.012	0.073	0.063	0.716
0.779	0.922	0.945	0.811	0.360	0.453	0.955	0.021
0.624	0.860	0.800	0.652	0.556	0.763	0.252	0.896

increases exponentially as the number of the training cases increases. Usually, the greater the number of training cases, the better the performance of the network. But for this recognition task, the network takes about 3–4 min to learn ten cases in Cray X/MP computer. Another example is that the classification results depend on the initial weights. Here we list the results for other four different random initial weights: 80 and 50% ($T_{\text{iteration}} = 533$); 80 and 100% ($T_{\text{iteration}} = 216$); 70 and 50% ($T_{\text{iteration}} = 533$); and 80 and 100% ($T_{\text{iteration}} = 518$), where the data set is Set 1, and the two templates by the geometrical preprocessing are used to train the network.

In conclusion, the KIII model under the geomet-

rical preprocessing is an appropriate way to enhance the qualities of the data for this industrial data recognition task. In other words, the algorithm shown in Figure 4 is suitable for this application.

7. DISCUSSION

In the above study, we treat the industrial data recognition as a standard supervised learning problem. Of the methods tried here, KIII is the most successful. The reasons why our chaotic olfactory neural network is superior are presented. That is, the chaotic dynamics introduced by the feedback pathways removes noise and irrelevant information with re-

TABLE 12
Recognition Results of the Data Sets Set 1–4 by Means of the Backprop Network in Figure 6 Under the Geometrical Preprocessing

A(64,64,1) Backprop Network by the Geometrical Preprocessing $\eta = 0.2$, $\alpha = 0.9$, and $T_{\text{error}} < 0.001$.				
Classification Rates				
Data Set	$T_{\text{iteration}}$	T_{error}	Unacceptable	Acceptable
a: U_train = 1, A_train = 1, U_test = 10, A_test = 8; 100% for the Training Cases				
Set 1	539	0.00998371	80%(A U U U U U U U A)	100%(A A A A A A A)
Set 2	134	0.00997246	100%(U U U U U U U U U)	12.5%(A U U U U U U U)
Set 3	209	0.00990856	80%(U U U U U U U A A)	37.5%(A A A U U U U U)
Set 4	580	0.00997655	50%(U A U U A A U A A U)	75%(A A A A U A U A)
b: U_train = 5, A_train = 5, U_test = 5; 100% for the Training Cases				
Set 1	17125	0.00999982	80%(U U U U A)	100%(A A A A A)
Set 2	22106	0.01	40%(U A A U A)	60%(A U U A A)
Set 3	25827	0.00999985	60%(A U U U A)	20%(U A U U U)
Set 4	50595	0.00999995	60%(U U A A U)	60%(U U A A A)

TABLE 13
Recognition Results of the Data Sets Set 1–4 by Means of the Backprop Network in Figure 6 Under the Statistical Preprocessing

A(64,64,1) Backprop Network by the Statistical Preprocessing. $\eta = 0.2$, $\alpha = 0.9$ and $T_{error} < 0.01$.				
Data Set	$T_{iteration}$	T_{error}	Classification rates	
			Unacceptable	Acceptable
a: U_train = 1, A_train = 1, U_test = 10, A_test = 8; 100% for the Training Cases				
Set 1	516	0.00998782	90%(U U U U U U U U A)	75%(U U A A A A A)
Set 2	243	0.00997809	70%(A U U U U U A U U A)	37.5%(A U U A A U U U)
Set 3	71	0.00729945	40%(U A U A A A U U A A)	62.5%(A A A A U U A U)
Set 4	584	0.00998102	60%(U A U U U A U A A U)	75%(A A A A U A U A)
b: U_train = 5, A_train = 5, U_test = 5, A_test = 5; 100% for the Training Cases				
Set 1	12272	0.00999991	80%(U U U U A)	80%(A A A A U)
Set 2	47468	0.00999975	60%(U A U U A)	100%(A A A A A)
Set 3	24696	0.00999988	60%(A U U U A)	20%(U A U U U)
Set 4	The Program Does Not Converge Within 1000 s CPU Time of Cray X/MP			

spect to the recognition task. The basal activity of the KII system, which replicates the unstimulated basal state of cortex, holds the system in a critical state or “a separatrix” of energy level so that the system can rapidly go into a patterned state when a related input is presented no matter its current location. The role of each individual channel is depressed by the collective effectiveness of Hebb’s “cell assemblies” and the sigmoid function of the neuronal pools. Elsewhere we have shown (Freeman, 1990a,b) that the bulbar “signal” to the prepyriform cortex must be oscillatory (rather than asymptotically steady) at a spatially common instantaneous frequency, in order that spatially incoherent activity

(“noise”) be removed by spatial averaging. We speculate that the broad spectrum provided by the aperiodic (chaotic) carrier wave of the signal prevents the concentration of activity into narrow spectral bands (limit cycles), and thereby it prevents the system from locking onto spurious resonances at particular single frequencies that are not intrinsically related to the spatial attributes of the cell assemblies controlling the classificatory state transitions. These important characteristics make it possible for the chaotic network to work in heavily “polluted” environments.

The results presented here show the principles of the algorithms. The data samples are too small to assert exactly how good the individual algorithms are. In addition, for this specific industrial data recognition, there are ways in which we may improve the performance of the pattern recognition algorithm.

(i) First, we may make better use of the deviation information in the minimum distance classifier by introducing a *weighted distance metric*. Let SD_A and SD_U be the standard deviations of the acceptable objects and the unacceptable objects, respectively. The weighted distance metric is defined by

$$WD_A = \alpha \frac{SD_U}{SD_A} * \text{Euclidean Distance between the point and the acceptable centroid,}$$

$$WD_U = \alpha \frac{SD_A}{SD_U} * \text{Euclidean distance between the point and the unacceptable centroid,}$$

(8)

where α is a positive number, WD_A (resp., WD_U) is the distance of a point to the acceptable (resp., unacceptable) centroid in the weighted distance metric space.

FIGURE 6. A three-layer feedforward neural network, which is composed of 64 units in the input layer, 64 units in the hidden layer, and a single unit in the output layer. $w = (w_1, w_2, \dots, w_{64})$ is the connection weights from the hidden layer to the output layer. $W = (W_{ij})_{64 \times 64}$ is the connection weights from the input layer to the output layer.

(ii) Second, the idea of the weighted distance metric can also be applied to the geometrical method to form the center point. This yields the center point closest to the class in which the training cases cluster more compactly.

(iii) Third, since the cost of error for the unacceptable objects is more than that for the acceptable objects, taking into account this information may be worthwhile. For instance, we may build the centroid for the unacceptable objects by means of the statistical method, and find the radius from the centroid such that all of the unacceptable objects in the training set can be included into the circle. If the number of the training cases is large enough, we may get 100% classification for the unacceptable objects with probability 1. Then no unacceptable object will be misclassified as acceptable.

(iv) Our estimate of the run time speed of the process of convergence relies on the hardware implementation of the KII model in comparison to the software version (Eisenberg, Freeman, & Burke, 1989). We say the algorithm in Figure 4 can recognize an uninterrupted sequence of objects at a speed of 200 ms per each. Here the time referred is simulated run time, which is different from elapsed CPU time in computers. The CPU time for recognizing a single object is about 10 s (about 1 min in off-line batch processing) on the Cray X-MP, which involves solving a 658th-order nonlinear differential equation. The CPU time spent on other computation may be negligible comparing with the CPU time spent on solving the equation. Thus the simulated run time is the time needed for the KIII model to reach a steady state from its basal state and to hold it long enough for us to get the mean value of the DC shift. Further exploration will require hardware embodiment of the KIII set to give the requisite speed.

Our KIII model demonstrates that a fully parallel distributed system operates most effectively in cooperative modes analogous to nonequilibrium phase transitions in nonlinear dynamic systems, and it demonstrates the utility of chaotic dynamics for the amplification of small input fluctuations, provided that the sensitivity of the system has been shaped by prior learning. The amplification is done primarily by the local strengthened associational mutually excitatory synapses (Edelman & Freeman, 1990) in the cell assemblies, and also by the global rise in sensitivity of populations under mutual excitation. We suggest that the broad spectrum of the carrier may facilitate the transition to that basin selected by the stimulus-activated cell assembly, by maintaining a high level of activity at all times, and by preventing collapse of the system into a deep energy well that corresponds to an equilibrium or to a limit cycle. We therefore

suggest that our KIII model not only clarifies a key role for chaos in biological pattern recognition; it illuminates certain other properties, which we introduced into the model for convenience or to improve its performance, and which we now re-interpret in terms of their biological significance.

First, in our model the modifiable mutually excitatory connections are two-valued, either low (naïve) or high (learned). In a distributed system with large numbers of connections, a graded change is less useful than an all-or-none change in strength while learning. We suggest that physiologists bear this property in mind when attempting to determine the nature of the synaptic change with true associative learning in cortex.

Second, we found it useful to suppress noisy channels with high variance, and to suppress overlapped channels belonging to more than one template. We suggest that neural mechanisms be sought by which central habituation performs these tasks in cortex.

Third, the input to our model was preprocessed by normalization, binarization, and (for template formation) averaging. There is clear experimental evidence for normalization of inputs in both the olfactory and visual systems (Freeman, 1975; Grossberg, 1988). The performance of olfactory receptor neurons and visual "feature detector" neurons in response to their favored inputs suggests that they, too, "binarize" input, in the sense that they fire strongly in a burst or not at all. Again, in a massively parallel distributed system, "graded" changes in firings of single cells may be irrelevant to the whole.

Finally, our model works with examples of classes and not with prescribed features. The input is "visual" in the sense that energy is reflected from the surfaces of the objects and is detected by an array of sensors over time. However, the wavelength of the sound at 40 KHz (8.5 mm) is much larger than the sizes of the differences in shapes (less than 1 mm) that are detected and used by the system for recognition. We do not know, nor need we describe, the "features" in the sound interference patterns which the system relies upon to do its task.

REFERENCES

- Amari, S. (1967). A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, **EC-16**, 279–307.
- Amari, S. (1972). Characteristics of random nets of analog neuron-like elements. *IEEE Transactions on Systems, Man and Cybernetics*, **2**, 643–653.
- Amari, S. (1977). Neural theory of association and concept formation. *Biological Cybernetics*, **26**, 175–185.
- Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, **14**, 197–220.
- Aplevich, J. D. (1968). Models of certain nonlinear systems. In Caianiello E. R. (Ed.), *Neural Networks*. Berlin: Springer-Verlag.
- Buckley, S. (1975). Phase monitored inspection. *Proceedings of*

- the Third North American Metalworking Research Conference. (pp. 701–712).
- Buckley, S. & Stelson, S. (1978). Phase monitoring for automated inspection, positioning and assembly. *Transactions of the Society of Manufacturing Engineering*, **6**, 56–63.
- Cooper, L. N. (1973). A possible organization of animal memory and learning. In B. Lundquist & S. Lundquist (Eds). *Proceedings of the Nobel Symposium on Collective Properties of Physical Systems* (pp. 252–264). New York: Academic Press.
- Cowan, J. D. (1967). *A mathematical theory of central nervous activity*. Unpublished dissertation, Imperial College, University of London.
- Duda, R. O. & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: John Wiley.
- Edelman, J. A., & Freeman, W. J. (1990). Simulation and analysis of a model of mitral/granule cell population interactions in the mammalian olfactory bulb. *Proceedings International Joint Conference on Neural Networks*, **1**, 62–65.
- Eisenberg, J., Freeman, W. J. & Burke, B. (1989). Hardware architecture of a neural network model simulating pattern recognition by the olfactory bulb. *Neural Networks*, **2**(4), 315–325.
- Ferguson, T. S. (1967). *Mathematical statistics: A decision theory approach*. New York: Academic Press.
- Freeman, W. J. (1960). Correlation of electrical activity of prepyriform cortex and behavior in cat. *Journal of Neurophysiology*, **23**, 111–131.
- Freeman, W. J. (1975). *Mass action in the nervous system*. New York: Academic Press.
- Freeman, W. J. (1987). Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biological Cybernetics*, **56**, 139–150.
- Freeman, W. J. (1990a). Searching for signal and noise in the chaos of brain waves. Chapter 5 in S. Krasner (Ed). *The Ubiquity of Chaos*. Washington, DC, AAAS, pp. 47–55.
- Freeman, W. J. (1990b). On the problem of anomalous dispersion of a in chaotic-chaotic phase transitions of neural masses, and its significance for the management of perceptual information in brains. In H. Haken, & M. Stadler (Eds.), *Synergetics of cognition* (Springer Series in Synergetics Vol. 45, pp. 126–143). Berlin: Springer-Verlag.
- Freeman, W. J., & Yao, Y. (1990). *Chaos in the biodynamics of pattern recognition by neural networks*. An oral presentation at the International Joint Conference on Neural Networks, 1990 Winter, Washington, D.C.
- Freeman, W. J., Yao, Y., & Burke, B. (1988). Central pattern generating and recognizing in olfactory bulb: A correlation learning rule. *Neural Networks*, **1**, 277–288.
- Fukushima, K., Miyake, S. & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, **13**, 826–834.
- Grossberg, S. (1969). On learning and energy–entropy dependence in recurrent and nonrecurrent signed networks. *Journal of Statistical Physics*, **30**, 73–80.
- Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.
- Grossberg, S. (1980). How does a brain build a cognitive code. *Psychological Review*, **87**, 1–51.
- Grossberg, S. (Ed.). (1988). *Neural networks and natural intelligence*. Cambridge, MA: MIT Press.
- Haken, H. (1987) *Advanced Synergetics*. Second printing, Berlin: Springer-Verlag.
- Hebb, D. O. (1949). *The Organization of behavior*. New York: John Wiley.
- Hirsch, M. W. (1989). Convergent activation dynamics in continuous time networks. *Neural Networks*, **2**, (5), 331–349.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, **79**, 2554–2558.
- Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, **81**, 3088–3092.
- James, W. (1890). *Psychology* (Briefer Course). New York: Holt.
- Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers*, **21**, 353–359.
- Kohonen, T. (1988). *Self-organization and associative memory*. Berlin, Springer-Verlag.
- Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, **4**, 4–22.
- Malsburg, C. van der (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, **14**, 85–100.
- Mead C. (1989). *Analog VLSI and neural systems*. Reading, MA: Addison-Wesley.
- Minsky, M. L., & Papert, S. A. (1969). *Perceptron: An introduction to computational geometry*. Cambridge, MA: MIT Press.
- Pao, Y.-H. (1989). *Adaptive pattern recognition and neural networks*. Reading, MA: Addison-Wesley.
- Pitts, W., & McCulloch, W. (1947). How we know universals: The perception of auditory and visual forms. *Bulletin of Mathematical Biophysics*, **9**, 127–147.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–408.
- Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. New York: Spartan Books.
- Rumelhart, S. E., & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: MIT Press.
- Sejnowski, T. J. (1977). Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology*, **4**, 303–321.
- Sejnowski, T. & Rosenberg, C. R. (1986). *NETalk: A parallel network that learns to read aloud*. (Johns Hopkins University, Electrical Engineering and Computer Science Technical Report, JHU/EECS-36/01).
- Skarda, C. A., & Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. *Brain and Behavioral Science*, **10**, 16–195.
- Tank, D. W., & Hopfield, J. J. (1986). Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, **5**, 553–541.
- Tavormina, J. J., & Buckley, S. (1978). Automated positioning and assembly under minicomputer control via acoustic phase monitoring. *Transactions of the Society of Manufacturing Engineering*, **6**, 448–454.
- Widrow, B., & Hoff, M. E. (1961). *Adaptive switching networks* (IRE Wescon Convention Record).
- Yao, Y., & Freeman, W. J. (1989). *Pattern recognition in olfactory systems: Modeling and simulation*. An oral presentation in the International Joint Conference on Neural Networks, Washington, D.C.
- Yao, Y., & Freeman, W. J. (1990). Model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, **3**(2):153–170.

APPENDIX: A MATHEMATICAL DESCRIPTION OF THE OLFACTORY SYSTEMS

The purpose of this appendix is to include all equations, working parameters, and enough detail to allow readers to replicate the simulation results although listing all the data is impossible within the text. For the neurophysiology and the detail of the equations,

please refer to Freeman, 1975; Yao & Freeman, 1990. Let L be a linear operator, and

$$L = \frac{d^2}{dt^2} + a \frac{d}{dt} + b,$$

where $a = 220/\text{sec}$ and $b = 720/\text{sec}$.

The N -channel olfactory system (see Figure 2) may mathematically be described by the following differential equations. The dimension of the system is $10N + 18$ (e.g., if $N = 64$, the dimension is equal to 658).

$$L(X_p[i]) = \frac{1}{N} \sum_{j=1, j \neq i}^N K_{pp} Q(X_p[j]) + K_{pp} I[i] + \frac{K_s}{T_{\text{on}} - T_{\text{off}}} \sum_{k=1}^{t_{\text{on}}} Q(X_{\text{on}}(k)). \quad (\text{P})$$

$$L(X_{M1}[i]) = K_{mm} Q(X_{M2}[i]) - K_{mc} (Q(X_{C1}[i]) + Q(X_{C2}[i])) + K_{mp} Q(X_p[i]) + \frac{1}{N} \sum_{j=1, j \neq i}^N K_{mm} Q(X_{M1}[j]) + K_{mi} I[i],$$

$$L(X_{M2}[i]) = K_{mm} Q(X_{M1}[i]) - K_{mc} Q(X_{C1}[i]),$$

$$L(X_{C1}[i]) = -K_{cc} Q(X_{C2}[i]) + K_{cm} (Q(X_{M1}[i]) + Q(X_{M2}[i])) - \frac{1}{N} \sum_{j=1, j \neq i}^N K_{cc} Q(X_{C1}[j]) + \frac{K_i}{T_{\text{on}} - T_{\text{off}}} \sum_{k=1}^{t_{\text{on}}} Q(X_{\text{on}}(k)) + \frac{K_c}{T_{\text{on}} - T_{\text{off}}} \sum_{k=1}^{t_{\text{on}}} Q(X_{\text{c}}(k)),$$

$$L(X_{C2}[i]) = K_{cm} Q(X_{M1}[i]) - K_{cc} (Q(X_{C1}[i])), \quad (\text{OB})$$

TABLE 14
Testing Data for the Kill Model

a: Template (A) for the Unacceptable Class Under Geometrical Preprocessing							
1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	1.000	1.000	1.000	0.000	0.000	1.000
0.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000
1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	1.000	1.000	0.000	1.000	0.000	1.000
1.000	1.000	0.000	0.000	1.000	1.000	1.000	1.000
0.000	1.000	0.000	1.000	0.000	1.000	0.000	0.000
1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000
b: Template (B) for the Acceptable Class Under Geometrical Preprocessing							
0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	0.000	0.000	0.000	1.000	1.000	0.000
1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000
1.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
0.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000
1.000	0.000	1.000	0.000	1.000	0.000	1.000	1.000
0.000	1.000	0.000	0.000	1.000	1.000	1.000	1.000
c: Output of DC Offsets of the Kill Model when Template A as an Input							
4.802	4.802	4.802	-1.271	-1.271	-1.271	-1.271	-1.271
-1.271	-1.271	4.802	4.802	4.802	-1.271	-1.271	4.802
-1.271	4.802	4.802	4.802	4.802	4.802	-1.271	4.802
4.802	4.802	4.802	-1.271	-1.271	-1.271	-1.271	-1.271
-1.271	4.802	4.802	4.802	-1.271	4.802	-1.271	4.802
4.802	4.802	-1.271	-1.271	4.802	4.802	4.802	4.802
-1.271	4.802	-1.271	4.802	-1.271	4.802	-1.271	-1.271
4.802	-1.271	4.802	4.802	-1.271	-1.271	-1.271	-1.271
d: Output of DC Offsets of the Kill Model when Template B as an Input							
-1.092	-1.092	-1.092	4.998	4.998	4.998	4.998	4.998
4.998	4.998	-1.092	-1.092	-1.092	4.998	4.998	-1.092
4.998	-1.092	-1.092	-1.092	-1.092	-1.092	4.998	-1.092
-1.092	-1.092	-1.092	4.998	4.998	4.998	4.998	4.998
4.998	-1.092	-1.092	-1.092	4.998	-1.092	4.998	-1.092
-1.092	-1.092	4.998	4.998	-1.092	-1.092	-1.092	-1.092
4.998	-1.092	4.998	-1.092	4.998	-1.092	4.998	4.998
-1.092	4.998	-1.092	-1.092	4.998	4.998	4.998	4.998

First using all the data in Set 1, we can build two templates A and B, which are listed in (a) and (b), respectively, by means of the geometrical preprocessing. There is no overlap between these two templates. Interested readers may use A and B to create the connection matrix K_{mm} . Then given A as an input, it will yield the output of the DC offsets of the Kill model similar to the vector shown in (c) (i.e. an "on" channel has high value, and an "off" channel has a low value). The parameters used are listed in the text. The input signal used is as follows: during [0, 400 ms], $\text{Input}[i] = 2$; during [400 ms, 800 ms], $\text{Input}[i] = 8$ if the i th channel is on, otherwise keeping on 2. The DC offsets are obtained by taking the mean of the states of the 64 mitral cells in the Kill model within [600 ms, 800 ms]. (d) Output of the DC offsets of the Kill model when B is presented as an input.

$$\begin{aligned}
L(X_{t1}) &= K_{\alpha}Q(X_{t2}) - K_{\alpha}(Q(X_{t1}) + Q(X_{t2})) \\
&\quad + \frac{1}{N} \sum_{j=1}^N K_{\alpha m}Q(X_{M1}[j]), \\
L(X_{t2}) &= K_{\alpha}Q(X_{t1}) - K_{\alpha}Q(X_{t1}), \\
L(X_{t3}) &= -K_{\alpha}Q(X_{t2}) + K_{\alpha}(Q(X_{t1}) + Q(X_{t2})) \\
&\quad + \frac{K_3}{T_{i3} - T_{v3}} \sum_{k=t-T_{i3}}^{t-T_{v3}} Q(X_{A1}(k)), \\
L(X_{t2}) &= K_{\alpha}Q(X_{t1}) - K_{\alpha}Q(X_{t1}), \quad (\text{AON}) \\
L(X_{A1}) &= K_{\alpha\alpha}Q(X_{A2}) - K_{\alpha\beta}(Q(X_{B1}) \\
&\quad + Q(X_{B2})) + \frac{1}{N} \sum_{j=1}^N K_{\alpha m}Q(X_{M1}[j]), \\
L(X_{A2}) &= K_{\alpha\alpha}Q(X_{A1}) - K_{\alpha\beta}Q(X_{B1}), \\
L(X_{B1}) &= -K_{\beta\beta}Q(X_{B2}) + K_{\beta\alpha}(Q(X_{A1}) + Q(X_{A2})) + K_{\beta\alpha}Q(X_{\alpha}), \\
L(X_{B2}) &= K_{\beta\alpha}Q(X_{A1}) - K_{\beta\beta}Q(X_{B1}), \quad (\text{PC}) \\
L(X_{\alpha}) &= -K_{\alpha\beta}Q(X_{B2}), \quad (\text{C})
\end{aligned}$$

where $i = 1, 2, \dots, N$, $A = a + b$, $B = ab$, K_1, K_2, K_3, K_4 are feedback gains, $t - T_{vj}$, $T_{vj} - T_{vi}$ ($j = 1, 2, 3, 4$) are the delay starting time and the terms of the delay, $I[i]$ is the input signal

to the i th channel, and $Q(\cdot)$ is a sigmoid function (i.e., the non-linearity of each neuron pool). $Q(v) = Q_m[1 - e^{-v^{1/(1+Q_m)}}, \text{ if } v > -u_0, \text{ otherwise } Q(v) = -1, u_0 = -\ln[1 - Q_m \ln(1 + 1/Q_m)]]$. Here a and b are open loop rate constants.

The working set of the parameters used in this paper is listed below. Structural parameters: $Q_m = 5.0$, $a = 0.2/\text{ms}$, $b = 0.72/\text{ms}$; global parameters: $K_{mm, \text{low}} = 0.5$, $K_{mm, \text{hi}} = 4.0$, $K_{\text{eg-link}} = 1.5$, $K_{\text{eg}} = 0.2$; feed forward parameters: $K_{\alpha} = 1.0$, $K_{\beta} = 1.0$, $K_{mp} = 0.5$, $K_{mf} = 1.0$, $K_{am} = 1.0$, $K_{\beta} = 1.0$; feedback parameters: $K_1 = 1.6$, $T_{v1} = 11$, $T_{i1} = 20$; $K_2 = 0.25$; $T_{i2} = 15$, $T_{i2} = 26$; $K_3 = 2.0$, $T_{v3} = 12$, $T_{i3} = 25$; $K_4 = 1.5$, $T_{v4} = 24$, $T_{i4} = 39$. Internal parameters ($K_{mm}, K_{mg}, K_{gm}, K_{gg}$) = (0.25, 1.5, 1.5, 1.8), ($K_{\alpha}, K_{\alpha\alpha}, K_{\beta}, K_{\beta\beta}$) = (1.5, 1.5, 1.5, 1.8), ($K_{\alpha\alpha}, K_{\alpha\beta}, K_{\beta\alpha}, K_{\beta\beta}$) = (0.25, 1.4, 1.4, 1.8); input parameters: $IN_{\text{min}} = 2$ (or 0), $IN_{\text{max}} = 8$, $T_i = 50$ ms (or 0 ms). The integral method used is Runge-Kutta method, the time step is 0.5 ms.

For those readers who wish to understand the KIII model, Table 14 shows two templates of Set 1 under the geometrical preprocessing. The readers may use them to build the connection matrix K_{mm} . Then using these two templates as inputs, the readers should get outputs close to those shown in the table (Notice: Do not expect your outputs to be exactly the same as the outputs in Table 14 since KIII is a chaotic system. Since our simulation has shown that it is stable, you should be able to obtain close results). Notice: Let $K_{mp} = K_1 = K_3 = 0$. Then equations $L(X_{M1})$, $L(X_{M2})$, $L(X_{A1})$, $L(X_{A2})$ are the mathematical model of the olfactory bulb (i.e., the KII model).