ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

Sammon Mapping Based Neural Pre-Processing Scheme for On-Line Classification of Gases/Odors

B. J. Singh[1], B.T.Vedanth[2], S. P. Krishna[3]
[123]*Assistant Professor, Department of ECE, MRIET, Secundrabad, A.P, India.*

Abstract - **Statistical methods have also been very efficient in resolving such classification problems although they are applicable only for off-line applications. It is with this motive that a novel pre-processing scheme implemented using ANNs has been in cascade with another ANN classifier, for processing the multi-element gas sensor array response. Such electronic systems are known as e-noses.**

Accordingly, Sammon Mapping, a very powerful yet statistical data transformation technique has been used to transform the sensor array response data to a two-dimensional data. In this two dimensional plot, respective gases/odors fall in clear clusters with good inter-cluster distance. This statistical process was then implemented in a neural network. Once trained, this ANN transformed any next unknown gas sample response to two dimensional response.

Now, another ANN has been used to classify the four considered gases/odors into four distinct classes using the two dimensional Sammon Mapped Data. To optimize the e-nose design for minimally required sensor elements, various combinations of the considered sensor array elements were tested for all correct classification results with least MSE. Finally, the Sammon transforming ANN, termed as 'Sammon Transformation Pre-processor' was cascaded prior to the 'Classifier ANN,' to complete the development of the e-nose which utilized only Two sensor elements for obtaining best classification results.

Keywords- ANN, Sammon mapping, Statistical data, PID, Inter-Cluster, MSE, Pre-Processor.

I. INTRODUCTION

Development of electronic nose (e-nose) has attracted major interest of researchers mainly for their applicability and necessity in variety of real life applications [1,2]. The ability of e-nose to precisely and continuously monitor and measure the presence of hazardous gases in real time ambience is crucial to prevent respective accidents and for accounting the short term and long term exposure of the workers to major industrial pollutants [3,5]. A simpler and effective e-nose with improved real time performance is therefore 'the need of the day'. For developing high performance e-noses, standard methodology consists of an array of gas sensors followed by stages for signal conditioning, pre-processing and pattern recognition for classifications of gases/odors [1,2]. A basic block schematic of such an e-nose is illustrated in as shown in fig.1.

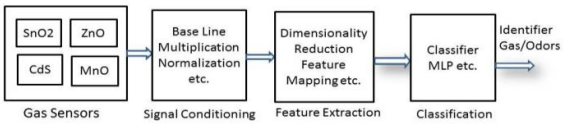

Fig.1. Basic schematic block of e-nose

Smell processing in humans is a very complex task which involves processing of many different categories of information which are both qualitative and quantitative in nature. e-nose technology strives to mimic the human system of smell processing and intelligent gas sensors (IGSs) are the foundation pillars of e-nose technology. Pattern Recognition (PR) techniques play a key role in IGS technology by improving the selectivity of poorly elective sensors. Artificial neural networks have emerged as one of the most sought after Pattern Recognition technique owing to their massively parallel nature and the ability to mimic human response to pattern recognition. Varying degrees of success have been achieved by applying ANN techniques for gas/odor discrimination

The sensor response is therefore processed for its feature enhancement by translating the sensor responses along mutually orthogonal principal axes (along which the transformed data is uncorrelated) using many a mapping methods such as principal component analysis (PCA), standardized principal component analysis (SPCA), linear discriminate analysis(LDA), exploratory projection pursuit etc. Typically, application of PCA/SPCA technique requires that the sensor response data must be first normalized either for zero mean (called PCA) or should be standardized for zero mean and unit variance (called SPCA). A feature vector is then obtained for transformation of this normalized or standardized data set. Finally, the normalized/standardized data set is transformed onto the principal component axes of PCA or SPCA, respectively, by multiplying the respective feature vector with the normalized/standardized data set. This transformed data set is partitioned to obtain training, validation and testing data sets.

The two main components of an electronic nose are the sensing system and the automated pattern recognition system. The sensing system can be an array of several different sensing elements (e.g., chemical sensors), where each element measures a different property of the sensed chemical, or it can be a single sensing device (e.g., spectrometer) that produces an array of measurements for each chemical, or it can be a combination. Artificial neural net (ANN) implementation of Sammon mapping for use with neural classifiers for gas/odor classification, which have been trained with Sammon mapping transformed data. We propose and demonstrate a simple implementation of ANN classifier utilizing statistical mapping. This implementation has the capability to classify even those samples which are freshly generated from the gas sensor array. The proposed implementation has been demonstrated using Sammon mapping for the fact that the performance of mapping is usually better than SPCA and PCA data transformations [4]. It can accept a freshly generated sensor response for its correct classification.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

II. MATERIALS AND METHODS

a) The Data Selection

To test the proposed methodology, authentic, already published and accepted data was sought for. Accordingly, responses of a gas sensor array for acetone, carbon tetra-chloride, ethyl methyl ketone and xylene has been taken [6]. The sensor referred in their experiments consisted of an array of four sensor elements with an integrated heater on a substrate. The array was fabricated with SnO_2as a base material and doped with different materials, namely, ZnO (sensor A), undoped SnO2 (sensor B), MoO (sensor C) and CdS (sensor D). The sensor array was fabricated using thick film technology. The fabrication process is described in detail in Ref. [6]. Their test set up used to collect the sensor array response data and the experimental conditions in the laboratory has also been described in the reference. The four gases/odors used in the referred experiment are acetone (ace), ethyl methyl ketone (emk), carbon tetra-chloride (car) and xylene (xyl) and the outputs measured were percent change in resistance of different sensors when exposed to various concentrations of these gases. As mentioned earlier, data used in the present analyses are derived from the results of Ref.[6] and therefore, the sensor characteristics from Ref.[6] have been reproduced here for ready reference in fig.2(a)–(d).

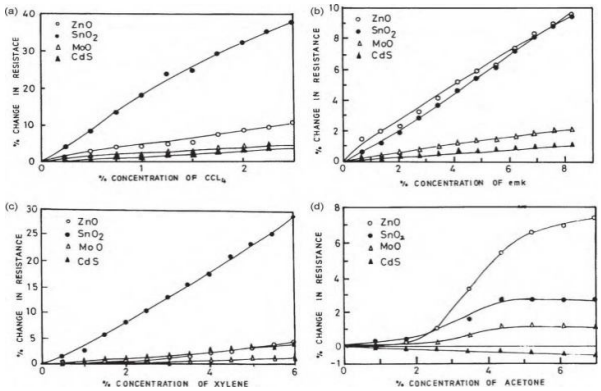

Fig.2. Sensor characteristics for (a) Carbon Tetra-chloride (b) Ethyl Methyl Ketone (c) Xylene and (d) Acetone exposures, in N_2ambience, under energized conditions

b) The Data Extraction

Two categories of data were extracted as Data Set I having 42 vectors (with 8 vectors for ace, 10 vectors for car, 12 vectors for emk and 12 vectors for xyl) and Data Set II consisting of 18 vectors (with 3 vectors for ace, 4 vectors for car, 6 vectors for emk and 5 vectors for xyl). The sample vectors were generally extracted with uniform spacing to cover the whole concentration axis of the graphs. The actual gas sensor array response data were then developed using (1)–(3), i.e. by shifting the origin of PID's window to match the origin of the graph and by scaling the pixel coordinates w.r.t. the graph's axes. An explanatory diagram for data extraction using PID is shown in Fig.3. For all data points, origin shifting and coordinate scaling was carried out using equations 1, 2, 3

$(x'_I, y_i') = (x - x0, y - y0)$, …… (1)

$$Conc.(x_i) = \frac{x_i'}{Xmax - X0/Xaxis\ rangofgraph} \text{……..(2)}$$

And

$$Response.(y_i) = \frac{y_i'}{Ymax - Y0/Yaxis\ rangofgraph} \text{……(3)}$$

Here, for all pixel coordinates (x, y) of various data points, x'_i, y_i are respective coordinates in terms of the graph parameters. Also, (x0,y0) is the pixel coordinate of the origin of the graph while x_{max} and y_{max} are the respective maximum values of x-axis and y-axis pixels on the graph axes. In fig.3. 'x-axis range of graph' and 'y-axis range of graph' are the represented ranges of change in percentage concentration and the range of percentage change in resistance, respectively; taken appropriately for each gas/odor. Finally, at the i_{th} data point x_i, y_i is the true coordinate while Conc.(x_i) and Response (y_i) is the true response of the sensor array on the graph.

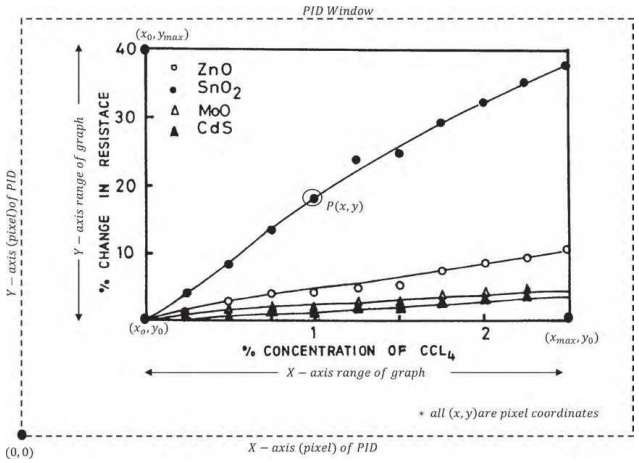

Fig.3. Graphical demonstration of data extraction using PID Tool

The true gas sensor array response data was hence intrinsically obtained for all the four gases/odors viz. ace, car, emk and xyl. The Data Set I was used for training and Data Set II was used for testing of the proposed e-nose development methodology. The experimentation has been carried out following the standard validation practices and recommendations, to ensure that the experiment is valid for any engineering application [2].

c) *Data Analysis and Clustering*

The cluster analysis is the classification of objects according to similarities among them, and organizing of data into groups. Clustering techniques are among the unsupervised methods, they do not use prior class identifiers. The main potential of clustering is to detect the underlying structure in data, not only for classification and pattern recognition, but for model reduction and optimization. Different classifications can be related to the algorithmic approach of the clustering techniques. Clustering techniques can be applied to data that is quantitative (numerical), qualitative (categories), or a mixture of both. In this thesis, the clustering of quantitative data is considered. The data are

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

typically observations of some physical process. Each observation consists of *n* measured variables, grouped into an *n*-dimensional row vector x*k* = [*xk*1*; xk*2*; : : : ; xkn*]*T ;* x*k 2* R*n*. A set of *N* observations is denoted by X = {x_k , *k* = 1, 2,….*N}*, and is represented as an *N*× *n* matrix:

In pattern recognition terminology, the rows of X are called *patterns* or objects, the columns are called the *features* or attributes, and X is called the *pattern matrix*.

$$X = \begin{bmatrix} x_{11} & x_{12} & . & . & . & x_{1n} \\ x_{21} & x_{22} & . & . & . & x_{2n} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_{N1} & x_{N2} & . & . & . & x_{Nn} \end{bmatrix}.$$

In this thesis, X is often referred to simply as the *data matrix.*
The base syntax of calling a clustering function of this

[Result] = *function* (data, param)
[Result] = *function* (result, data, param)

This means, the input arguments must be given in structure arrays, as one gets the output matrices in them as well. Some functions use the output matrices as inputs, these are described at their description.

The input data matrix data:X denotes the matrix of the N × n dimensional data set X, where N is the number of data points, and n is the number of samples.

To be specific, clustering is an unsupervised process of pattern association based on existing spatial relationship or similarities among multi-variant high dimensional feature space. The process of clustering involves three steps: (i) Defining a dissimilarity measure between examples, typically the distance, (ii) Defining a clustering criterion to be optimized, typically based on within and between cluster structure and (iii) Defining a search algorithm to find a "good" assignment of examples to clusters, since exhaustive enumeration of all possible clustering is clearly unfeasible.

Typically, the basic data used to form clusters is a table of measurements on several variables where each column represents a variable and row represents an object often referred to in statistics as a case.

Here we have different types of clustering algorithms, partition clustering, supervised clustering and k-means clustering. Cluster is a subset of the full data set; A large of techniques have been proposed for forming clusters from distance matrices.

The partition techniques usually produce clusters by minimizing a criterion function defined either locally or globally. In partition clustering, given a dataset and the number of clusters K, an algorithm organizes the objects into K partitions, in which each partition represents one cluster.

The supervised clustering is performed on attribute variables under the supervision of a target class variable. As a consequence, each generated cluster is labeled with only one specific class that has majority of data objects inside the cluster.

The k-means algorithm works unsupervised and is popular as it is easily implemented. If n is the number of patterns, k is the number of clusters, and l is the number of iterations taken, the time complexity is O(nlk). Typically l and k are fixed so the algorithm has linear time complexity to the size of the data set. A major drawback to this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function.

d) Sammon mapping

Sammon's mapping has been designed and usually used to project high-dimensional data onto one to three dimensions in order to analyze the data structure or for classification based on two projections [3]. However, there is no obstacle that prevents extracting more than two or three projections, and hence the application of the mapping to feature extraction and classification. Recently, it has been suggested [7] to extract an *arbitrary* number of projections of Sammon's mapping and thereby exploit the 'classification potential' of the mapping. It was found that the classification accuracy based on Sammon's projections is comparable with, and in some cases even superior, to that based on other feature extractors[3,7]. Sammon's Non-Linear Mapping (NLM) projects data and tries to preserve inter-point distances in the low dimensional configuration. Suppose that we have *n* data points, **x***i*, *i* = 1, . . . , *n*, in *D*-space and, correspondingly, we define *n* points, **y***i*, *i* = 1, . . . , *n*, in *d*-space (*d* < *D*). Let *d*∗*ij* denote the distance between **x***i* and **x***j* and *dij* denote the distance between the corresponding points **y***i* and **y***j* . NLM works by randomly choosing an initial *d*-space configuration for **y***i*. The algorithm then computes all inter point distances *dij* in *d*-space, which are used to define an error measure *E* as

$$E = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(d_{ij}^{*} - d_{ij})^2}{d_{ij}^{*}}$$

E is commonly referred as Sammon's stress. It represents how well the inter-point distances in the current configuration in *d*-space fit the inter-point distances in *D*-space. The next step is to adjust the *d*-space configuration so as to decrease *E*. NLM uses a steepest descent procedure to search for a minimum of *E*. Sammon's stress is designed so that short distances contribute more to the value of *E*. In the process to minimize *E*, therefore, NLM puts more priority on the preservation of short distances rather than long ones. That is why NLM is capable of unfolding high dimensional data manifolds. Because the algorithm considers also long distances, however, it may fail to unfold strongly twisted patterns. One improvement over NLM is Curvilinear Component Analysis (CCA) [2] which uses a new error measure that totally ignores distances longer than a user-defined threshold.

A general mapping *f* transforms a pattern *X* of a *d*-dimensional input space to a pattern *Y* of an *m*-dimensional projected space, *m* , *d*, i.e. *Y* = *f*(*X*), such that a criterion *J* is optimized. The mapping *f* is determined from among all the transformations *g*, as the one that satisfies *J*{*f*(*X*)} =min *g* J{g(X)}. The mappings vary by the functional forms of *f* and the criteria they have to optimize. Although providing a very well established criterion (Eq. (1)), Sammon's algorithm does not provide an explicit mapping function, *f*; hence, the projection of a new pattern requires re-execution of the algorithm to the 'new' data set.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

e) Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. ANN cannot be programmed to perform a specific task.

A simple mathematical model of a single neuron was introduced in a landmark paper by McCulloch and Pitts in 1943 and it takes the form indicated in fig.4. McCulloch-Pitts neuron contains a number of simplifying assumptions that do not reflect true behaviour of biological neurons. It can be regarded as a non-linear function which transform a set of input variables x_i, (i=1, 2, ……., n) into an output variable z so that

$z = f(h);$ $\qquad z = f(\sum_{i=1}^{n} w_i x_i + w_0)$…… (4.1)

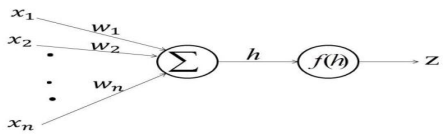

Fig.4. The McCulloch-Pitts model of a neuron.

Where output (z) and inputs (x_1, x_2, …….. x_n) model axon and dendrites. Weights (w_1, w_2,……...w_n) are analogous to the synaptic strength in a biological network and the offset parameter w_0 is called a bias corresponding to the firing threshold in a biological neuron. Bias can regarded as a special case of weight from an extra input whose value x_0 is permanently set to +1. The weights (and the bias) can be of either sign, corresponding to excitatory and inhibitory synapses. The output z of the unit, analogous to the average firing rate of a neuron, is then given by operating on 'h' with a non-linear activation function f(h), Eq. 4.1.

Learning is essential to most of these neural network architectures as shown in fig.5, hence the choice of a learning algorithm is a central issue in network development. A neural network learns about its environment through an interactive process of adjustments applied its synaptic weights and bias levels.

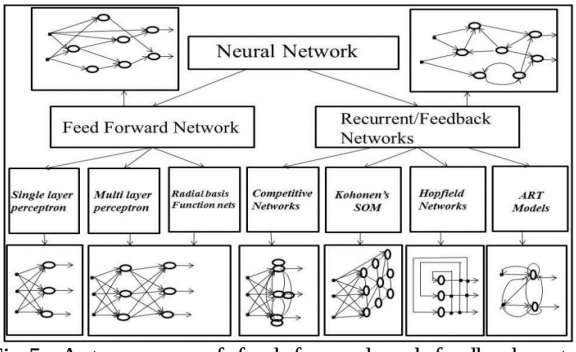

Fig.5. A taxonomy of feed forward and feedback network architectures.

1. The principles underlying this statement have become known as Hebbian Learning. Virtually most of the neural network learning techniques can be considered as a variant of the habbian learning rule. The basic idea is that if two neurons are active simultaneously, their interconnection must be strengthened. If we consider a single layer net, one of the interconnected neurons will be an input unit and one an output unit. If the data are represented in bipolar form, it is easy to express the desired weight update as,

$$w_i(new) = w_i(old) + x_i(old)$$

The training techniques used is called the perceptron learning rule. Perceptrons are especially suited for simple problems in pattern classification. Suppose we have a set of learning samples consisting of an input vector x and a desired output d(k). for a classification task, the d(k) is usually +1.

2. The perceptron-learning rule is very simple and can be stated as follows:

i. Start with random weights for the connections.
ii. Select an input vector x from the set of training samples.
iii. If output y(k) ≠ d(k) (The perceptron gives an incorrect response), modify all connections Wi according to δWi = η(dk - yk) xi ; (η-learning rate).

Go back to step.ii, Note that the procedure is very similar to the hebb rule: the only difference is that when the network responds correctly, no connection weights are modified.

3. Training basically involves feeding training samples as input vectors through a neural network, calculating the error of the output layer, and then adjusting the weights of the network to minimize the derivative easier. Once the error is computed, the weights can be updated one by one. In the batched mode variant, the descent is based on the gradient ΔE for the total training set,

$$\Delta W_{ij} = -\eta * \frac{\delta E}{\delta W_{ij}} + \alpha * \Delta W_{ij}(n-1)$$

When η and αare the learning rate and momentum respectively. The momentum term determines the effect of past weight changes on the current direction of movement in the weight space.

III. EXPERIMENT AND SIMULATION

a) Sensor Response and Data Extraction

As coined in the preceding chapters, Sammon mapping and clustering analysis has been performed through making dendrograms for the sensor responses (as presented). Further, once at each stage, a specific gas was identified which had natural tendency of getting separated out, a simpler neural network was trained at each stage for implementing.

Cluster analysis, also called segmentation analysis or taxonomy analysis, creates groups, or clusters, of data. Clusters are formed in such a way that objects in the same cluster are very similar and objects in different clusters are very distinct. Measures of similarity depend on the application.

Sammon mapping error E provides us with a measure of the quality of any given transformed dataset. However, we still need to determine the optimal such dataset, in terms of

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

minimizing E. Strictly speaking, this is an implementation detail and the Sammon mapping itself is simply defined as the optimal transformation, Sammon describes one method for performing the optimization. The transformed dataset Y_i is first initialized by performing PCA on the original data (an arbitrary, random initialization is sufficient, but using the principal components improves performance). Then, we repeatedly update the Y_i using steepest descent, considering the gradient of E with respect to the Yi, until satisfactory convergence is achieved.

The three categories of data were generated viz. data set I (with 8 vectors for ace, 10 vectors for car, 12 vectors for emk and 12 vectors for xyl). All of these vectors were uncorrelated and were generally uniformly spaced all over the concentration axis of the graphs at fig.2.(a) to (d). from raw data, normalized data were developed by shifting the origin to match the graphs origin and the data were scaled over the axes ranges of the graph, as the data points were in terms of respective pixel values of the software window. For all data points and graph windows, origin shifting and axes scaling was carried through (1), (2) and (3),

The normalized 'sensor response vs. gas concentration' data was hence intrinsically obtained for all the four gases/odors viz. ace, car, emk and xyl. These data were used for training and testing of the neural network thus conceived following standard validation practices and recommendation, making the experiment true for any engineering application.

The data as extracted from the graph using Precision Image Digitizer version 1.3.0.0 has been plotted for its scatter for three sensor responses to four, three and two gases, respectively in fig.6.(a), (b) and (c).

Now, from the scatter plots we can't classify the gases/odors. That is, the separability aspect of various gases is indicated by respective gases responses to gas sensor array as well as that is the data at that specific gas can be separated (clustered) out rather more easily clustering the data. Clustering is an unsupervised learning process that seeks to find spatial relationships or similarities among the data samples.

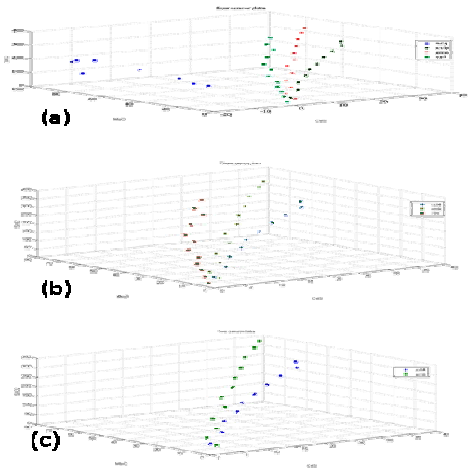

Fig.6.Scatter plot for (a) Four data (b) Three data and (c) Two data

b) Data analysis *and Sammon mapping*

The cluster analysis is the classification of objects according to similarities among them, and organizing of data into groups. Clustering techniques are among the unsupervised methods, they do not use prior class identifiers. The main potential of clustering is to detect the underlying structure in data, not only for classification and pattern recognition, but for model reduction and optimization. Different classifications can be related to the algorithmic approach of the clustering techniques.

Function Arguments
The base syntax of calling a clustering function of this

[Result] = *function* (data, param)
[Result] = *function* (result, data, param)

This means, the input arguments must be given in structure arrays, as one gets the output matrices in them as well. Some functions use the output matrices as inputs, these are described at their description.
The input data matrix *data:X* denotes the matrix of the $N \times n$ dimensional data set X, where N is the number of data points, and n is the number of samples.

Sammon Mapping

The sammon function calculates the original Sammon's Mapping. It uses *result.data.d* of the clustering as input and two parameters: the maximum iteration number (*param.max*, default value is 500.) and the step size of the gradient method (*param.alpha*, default value is 0.4.). The *proj:P* can be given either an initializing projected data matrix or the projection dimension. In latter case the function calculates with random initialized projected data matrix, hence it needs normalized clustering results. During calculation the Sammon function uses online drawing, where the projected data points are marked with different markers and the projected cluster centers with '*'. The online plotting can be disengaged by editing the code, if faster calculation wanted. The results are evaluated with *projeval* function.

The data set contains four categories of data were generated viz. data set I (with 8 vectors for ace, 10 vectors for car, 12 vectors for emk and 12 vectors for xyl), Where each class refers to a type of gas. One class is linearly separable from the other.

Sammon mapping for the four sensors (Cds, MoO, Sno_2 & ZnO) response with 42 samples can be mapped to two dimensional data with 42 samples, so that here we can reduced the dimensionality of data and projected to simple two dimensions. Like this we can project the verity of sammon mapping by using the different combinations of sensors responses. Then the gases can be easily classified from the cluster form of sammon mapping.
Fig.5.3. Result of Sammon's mapping projection by four gas sensors data.

We have consider the four gas sensors response and project into sammon mapping as shown below fig.7. The four gas sensors Cds, MoO, SnO_2 & ZnO response are projected into four cluster form, each cluster will have identical gas sensor response sammon mapping data.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

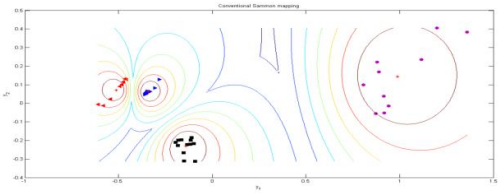

Fig.7. Result of Sammon's mapping projection by four gas sensors data.

Now, consider the three gas sensors for Sammon mapping and compare with the scatter plot of gas sensors response. Raw data (gas sensor responses) scatter plot is unable to classify the clusters or gasses, By using sammon mapping for those gas sensor responses then it will classify and projected to a clear form of clusters and two combinations of three gas sensors response of clusters as shown in fig.8. and fig.9. So that if we ignore any of the one gas sensor then also classification is possible by using sammon mapping from these combination of three gas sensors.

The firs three combinations of gas sensors CdS, MoO &SnO_2 responses analysis as shown

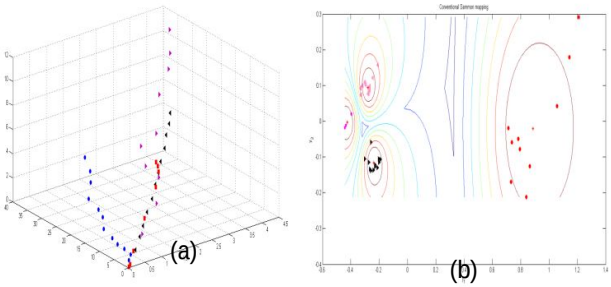

Fig.8. Three gas sensors CdS, MoO &SnO_2 for (a)Raw data scatter plot, (b)Sammon mapping.

The second three combinations of gas sensors MoO, SnO_2 & ZnO responses analysis as shown fig

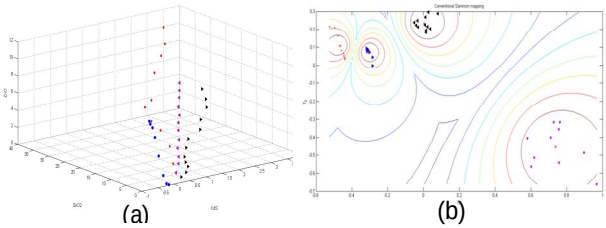

Fig.9.Three gas sensors CdS, SnO_2 & ZnO for (a)Raw data scatter plot, (b)Sammon mapping.

c) *Neural Network training*

The following architecture of the neural net with MLP's has been chosen to be trained with LMS back propagation algorithm.

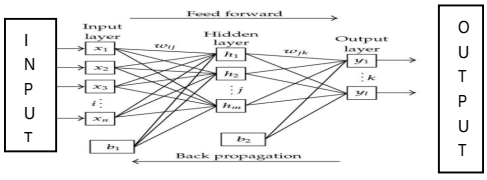

Fig.10. General Architecture of FFBP network.

Therefore even a very simple neural net should be able to classify the gases from the all gas sensors responses. A two layer neural network with varying number of neurons with hyperbolic ten-sigmoid function in the hidden layer and only one or two neurons with linear function in the output layer is shown in fig.10. Training data and testing data were used to train the aforesaid classifier for known classes of gas/odor using LM algorithm with performance function as MSE and four versions of data were taken for the designing of the weights and biases of the classifier network initialized with random numbers between 0 to 1 for 100 epochs with MSE goal to be 0.0001. Optimally train set of weights and biases was obtained when weights and biases update starts to raise MSE in little successive iteration, initialized with a value of 0.001, which was incremented or decremented by multiplying with 10 or 0.1 respectively. The learning rate as in, is multiplied with the negative of the gradient to determine the changes to the weights and biases. For this work when minimum gradient of update decent is reached, the training stops.

Now, consider the gas sensor responses as input and the target output is the mathematically calculated sammon mapping data , then train to the neural net-I (NN-I) for computing the sammon mapping, the training neural net-I is shown in fig.11, from this trained NN-I we get the sammon mapping data to the input of neural net-II for classification of gases/odors.

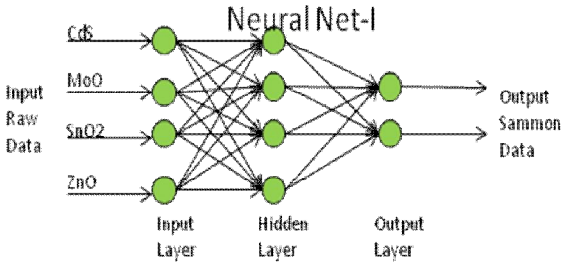

Fig.11. Trained Neural Net-I for Sammon mapping.

The raw testing data set-III as input to the trained NN-I to produce the output as sammon mapping data as output with varying the number of neurons in the hidden layer, the output neurons are not changed. This trained neural net-I simulated output is Sammon mapping data set-IV. These trained neural Network output is given as to the input for the next neural network which is a classifier neural net-II as shown in fig.12.

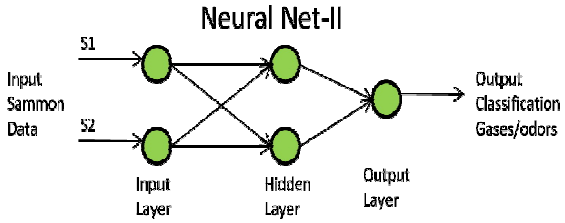

Fig.12.Trained Neural Net-II for Classification of gases/odors.

The trained Neural Network-II (NN-II) is a classifier, taking input as sammon mapping data and classify to the respective gases/odors.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 105– 111 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

IV. RESULTS AND PERFORMANCE

Now we consider the sensor responses for classification of gases/odors. First we consider the all four gas sensors response The trained Neural Network-II(NN-II) is a classifier, taking input as sammon mapping data and classify to the respective gases/odors with performance and error as shown in fig.13.

Table.1. Target output and Neural Network classifier output with error

Perf	0.8988	1.0466	1.0709	1.9999	2	2	2	3.0222	3.0999	3.1033	3.1203	3.1171	3.1571	3.9955	3.9955	3.9933	3.9944	3.9955
T	1	1	1	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4
error	0.0101	0.0464	0.0703	0.0004	0	0	0	0.0219	0.0993	0.1036	0.1202	0.1171	0.1571	0.0045	0.0046	0.0062	0.0056	0.0047

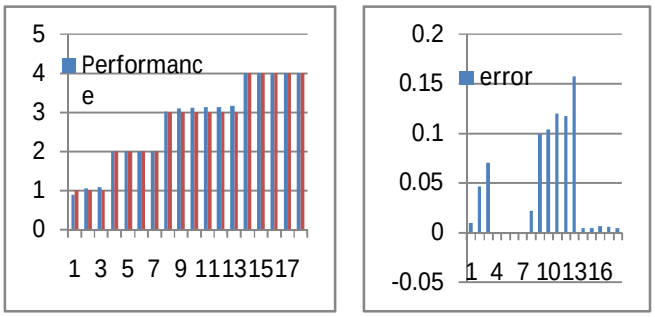

Fig.13. Performance and error for four Sensors Classification

Optimally weights and biases is the condition for early stopping, if the error in successive iterations typically begins to rise, for predetermined number of iterations (few, as described earlier) and the set of weight and biases at the minimum of the error is retained. As previously described, the network is trained with LMS algorithm. In here, we have initialized the parameter δ with a value of 0.001. It is multiplied by 0.1, whenever the performance function is reduced by a step where as it is multiplied by 10, whenever a step would increase the performance function. When δ becomes larger than 109 the training is stopped. Once, the network was trained, it was exported and saved. Multiples of such classifiers were created taking variety of training and learning criteria.

V. Conclusion

Statistical techniques, no matter how very powerful, may NOT be directly usable in real-time e-nose designing. However, the statistical characteristics of the considered data set can be learnt by an Artificial Neural Network and the transformation of newer data can be carried out by this ANN, on real-time basis. A very powerful statistical technique named Sammon Mapping has been implemented successfully in a Sammon mapping transformation Neural Network. This ANN has been used as a Sammon Mapping Pre-Processor, for transforming raw multi-sensor array response to the Sammon Mapping based analysis space. Using Sammon Mapping, real-time e-nose could be designed using two sensing elements only. All the mutually independent test sensor response data were classified correctly, which confirmed the superiority of the e-nose system we have designed.

REFERENCES

[1] K.K. Shukla, R.R. Das, R. Dwivedi, Adaptive resonance neural classifier for identification of gases/odors using an integrated sensor array, Sens. Actuators B 50(1998) pg. 194–203, and references therein.

[2] D.S. Lee, D.D. Lee, S.W. Ban, M. Lee, Y.T. Kim, SNO2 gas sensing array for combustible and explosive gas leakage recognition, IEEE Sens. J. 2 (**2002**) 140–146.

[3] Sammon JW Jr. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 1969; 18:401–409.

[4] Ravi Kumar, R. R. Das, V. N. Mishra and R. Dwivedi, "A Neuro-Fuzzy Classifier-Cum-Quantifier for Analysis of Alcohols and Alcoholic Beverages Using Responses of Thick-Film Tin Oxide Gas Sensor Array", IEEE SENSORS JOURNAL.

[5] N. S. Rajput, R.R. Das, V. N. Mishra, K. P. Singh, R. Dwivedi, "A neural net implementation of SPCA pre-processor for gas/odor classification using the responses of thick film gas sensor array", Sens. Actuators B 148(2010); 550-558.

[6] J. W. Sammon, Jr, "A nonlinear mapping for data structure analysis," IEEE Transactions on Computers, vol. C-18, no. 5, pp. 401–409, 1969.

[7] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 296–317, 1995.

[8] Dick de Ridder, Robert P.W. Duin, "Sammon's mapping using neural networks: A comparison", Pattern Recognition Letters 18 Ž1997. 1307–1316.

[9] Balazs Balasko, Janos Abonyi and Balazs Feil, "Fuzzy clustering and data analysis toolbox".

[10] Jun Fua, Guang Li, Yuqi Qin, Walter J. Freeman "A pattern recognition method for electronic noses based on an olfactory neural network" Sensors and Actuators B 125 (2007) 489–497

[11] Amine Bermak, So_ane Brahim Belhouari, Minghua Shi, and Dominique Martinez, "Pattern Recognition Techniques for Odor Discrimination in Gas Sensor Array"

[12] K. K. Shukla a, R. R. Das, R. Dwivedi, "Adaptive resonance neural classifier for identification of gases/odors using an integrated sensor array", Sensors and Actuators B 50 (1998) 194–203.