ELSEVIER

Sensors and Actuators B 30 (1996) 185–190

SENSORS AND ACTUATORS B CHEMICAL

Fuzzy neural computing of coffee and tainted-water data from an electronic nose

Sameer Singh [a], Evor L. Hines [b], Julian W. Gardner [b]

[a] *School of Computing, University of Plymouth, Plymouth PLY 8AA, UK*

[b] *Department of Engineering, University of Warwick, Coventry CV4 7AL, UK*

Received 28 November 1994; in revised form 24 July 1995; accepted 1 August 1995

Abstract

In this paper we compare the ability of a fuzzy neural network and a common back-propagation network to classify odour samples that were obtained by an electronic nose employing semiconducting oxide conductometric gas sensors. Two different sample sets have been analysed: first, the aroma of three blends of commercial coffee, and secondly, the headspace of six different tainted-water samples. The two experimental data sets provide an excellent opportunity to test the ability of a fuzzy neural network due to the high level of sensor variability often experienced with this type of sensor. Results are presented on the application of three-layer fuzzy neural networks to electronic nose data. They demonstrate a considerable improvement in performance compared to a common back-propagation network.

Keywords: Electronic noses; Fuzzy neural networks; Coffee; Tainted water

1. Introduction

Artificial neural networks (ANNs) have been the subject of research for over 20 years. However, it is during the last decade or so, that research interest has blossomed into commercial application, and they are now widely used as predictive classifiers, discriminators and in pattern recognition in general. Recent neural-network research has been directed towards the improvement of the ability of multi-layer perceptrons to generalize and classify data through the design of better training algorithms and superior networks. One important, yet neglected, aspect has been to understand the exact nature of the data. ANNs have been employed in the field of measurement where the nature of the data is highly diverse, ranging from digital pixel values from CCDs in vision systems to analogue d.c. conductance signals in a semiconducting oxide electronic nose. The uncertainty in the data comes in as a part of the real-world implementation itself, often attributed solely to the imprecision of the measurement. Conventional ANNs (e.g., multi-layer perceptrons) do not attempt to model precisely the vagueness or fuzziness of data. This often culminates in poorly trained networks where the problem becomes more significant as the uncertainty in the data increases and the size of the training set decreases. Fuzzy neural networks (FNNs) make use of fuzzy logic to model fuzzy data. FNNs have a relatively recent history but interest has increased through the application of fuzzy logic in non-linear control systems. In this paper we discuss FNNs and apply them to electronic nose data. We compare the performance of a FNN to that of a standard back-propagation (BP) network. We also consider how FNNs differ from their non-fuzzy counterparts and so the applications in which their performance should be better. More detailed discussions on FNNs can be found in Ref. [1].

2. Artificial neural networks

ANNs are mathematical constructs that try to mimic biological neural systems. Over the years, ANNs have become recognized as powerful non-linear pattern-recognition techniques. The networks are capable of recognizing spatial, temporal or other relationships and performing tasks like classification, prediction and function estimation. ANN development differs from classical programming in the fact that in modality the data variance is learnt over a number of iterations. One of the main problems of an ANN approach is knowing when optimal network parameters have been found. Further, as the data sets become less well behaved, the training typically becomes more difficult, and the class prediction less than satisfactory. It is generally accepted [2] that there are several advantages in applying ANNs as opposed to any other mathematical or statistical techniques. For instance, their generalization abilities are particularly useful since real-

0925-4005/96/$15.00 © 1996 Elsevier Science S.A. All rights reserved
SSDI 0925-4005(95)01800-6

world data are often noisy, distorted and incomplete. In addition, it is difficult to handle non-linear interactions mathematically. In many applications, the systems cannot be modelled by other approximate methods such as expert systems. In cases where the decision making is sensitive to small changes in the input, neural networks play an important role. Nevertheless, ANNs have some potential disadvantages as well, since the choice of the way in which the inputs are processed is often largely subjective and different results may be obtained for the same problem. Furthermore, deciding on the optimal architecture and training procedure is often difficult, as stated above. Many problems would need different subjective considerations, including speed, generalization and error minimization. ANNs have other potential disadvantages as well. For example, there is very little formal mathematical representation of their decisions and this has been a major hurdle in their application in high-integrity and safety-critical systems.

Multi-layer perceptrons are the most commonly used ANN in pattern classification and typically comprise an input layer, an output layer and one or more hidden layers of nodes. Most of our electronic nose work has employed two-layer networks (excluding the input layer), since the addition of further hidden processing layers does not provide substantial increases in discrimination power [3]. We have used an advanced BP method called Silva's method [4] in order to train the neural networks in the conventional way on the electronic nose data (described later) and then compare the results with fuzzy neural models.

3. Experimental details

3.1. Fuzzy neural model

Fuzzy logic is a powerful technique for problem solving which has found widespread applicability in the areas of control and decision making. Fuzzy logic was invented by Zadeh in 1965 and has been applied over recent years to problems that are difficult to define by precise mathematical models. The approach is particularly attractive in the field of decision making, where information often has an element of uncertainty in it.

The theory of fuzzy logic in turn relates to the theory of fuzzy sets where an effort is made to distinguish between the theory of probability and possibility. There is more than one way in which fuzziness can be introduced into neural networks and hence different workers mean different things by the term 'fuzzy neural network'. Some researchers define these networks as having fuzzy inputs and fuzzy outputs and hence try to fuzzify (i.e., assign a membership value to data values within the range 0–1 using a possibility distribution) before data are presented to the ANN. This concept can obviously be further extended, as described, for example, by Zadeh [5], where the inputs and outputs are truly fuzzified by their transformation into linguistic terms. So rather than having a particular numerical value (e.g., in the input or output), we can describe values linguistically as very low, low, moderate, high, very high, etc. This kind of fuzzification, though tempting for some applications (e.g., classifying the quality of odours), would not be suitable for others in which the boundaries are hard to specify. Fuzzy logic attempts to distinguish between possibility and probability as two distinct theories governed by their own rules. Probability theory and Bayesian networks can be used where the events are repetitive and statistically distributed. The theory of possibility is more like a membership-class restriction imposed on a variable defining the set of values it can take. In the theory of probability, for any set A and its complement A^c, $A \cap A^c = \varnothing$ (null set), which is not true in the case of the theory of possibility. Possibility distributions are often triangular and so similar in shape to normal distributions with the mean value having the highest possibility of occurrence, which is one. Any value outside the min–max range has a possibility of occurrence of zero. Hence in mathematical terms, the possibility that a_j is a member of the fuzzy set $X = \{a_1, a_2, ..., a_n\}$ is denoted by its membership value $M(a_j)$. This membership value of a_j in X depends upon the mean, minimum and maximum of the set X. An introductory treatment to the theory of fuzzy logic is given by McNeill and Freiburger [6]. A more mathematical description of fuzzy sets and the theory of possibility is available in Dubois and Prade [7].

We have made use of the fuzzy neural model proposed initially by Gupta and Qi [8]. This model challenges the manner in which conventional networks are trained with random weights, because these random weights may be disadvantageous to the overall training process. Let us consider a $12 \times 3 \times 3$ neural network architecture. At the end of training we hope to have an optimal point in $51(12 \times 3 + 3 \times 3 + 3 + 3)$-dimensional space that describes the best set of weights with which to classify the training patterns, and also to predict unknown patterns. This optimal point is harder to achieve in practice as the data become more non-linear, additional difficulties being caused by noise in the data. The main problem with random weights is that we usually start the search from a poor point in space which either slowly, or perhaps never, takes us to the desired optimal point, i.e., a global minimum. A suitable starting point, preferably dependent on the kind of training data, is highly desirable. It can speed up training, reduce the likelihood of getting stuck in local minima and take us in the right direction, the direction for the global minimum. The result is a better set of weights that will better classify the test patterns. The fuzzy neural network (FNN) approach adopted here attempts to do exactly this. It makes use of possibility distributions [9], which helps in determining the initial set of weights. These weights themselves are fuzzy in nature and depend entirely on the training-set distribution. Here the neural network reads a file of weights before training. These weights are generated in advance by performing calculations on a possibility distribution function as shown in Fig. 1. Once the network is trained, the final weights are no longer fuzzy but can take any

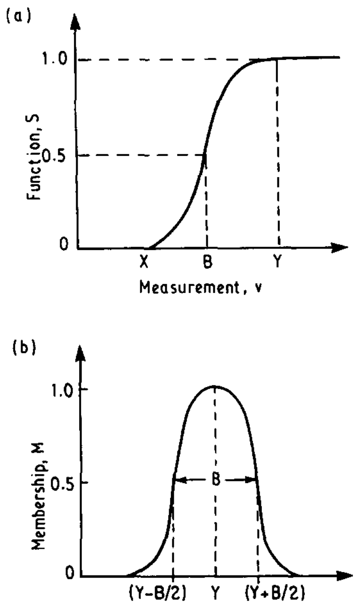

Fig. 1. (a) The possibility distribution $S(v: X, B, Y)$ is used to determine the membership value of a measurement v. $S(v: X, B, Y)$ is given by 0 when $v \leq X$, $2(v-X)^2/(Y-X)^2$ when $X<v\leq B$, $1-2(v-Y)^2/(Y-X)^2$ when $B<v\leq Y$ and 1 when $v>Y$. Note that Y is the mean value, X is the minimum value and B is the bandwidth in this possibility distribution. (b) The membership function M is related to the function S with $M=1-S(v: Y, Y+B/2, Y+B)$ when $v>Y$, and $M=S(v: B, Y-B/2, Y)$ when $v\leq Y$.

real value. These saved weights are then used with the test data for recognizing new patterns.

3.2. Electronic nose

The present work is concerned with the application of FNNs to electronic nose data. An electronic nose comprises of a set of odour sensors that exhibit differential response to a range of vapours and odours [10]. Previous work has been carried out in the Sensors Research Laboratory and the Intelligent Systems Engineering Laboratory at the University of Warwick to identify alcohols and tobaccos [11,12].

Here data were collected from an array of semiconducting oxide gas sensors ($i=1$ to n) in response x_{ij} to a measurand j in terms of a fractional change in steady-state sensor conductance G, namely,

$$x_{ij}=\frac{(G_{\mathrm{odour}}-G_{\mathrm{air}})}{G_{\mathrm{air}}} \tag{1}$$

This was chosen because it was found to reduce sample variance in earlier work on odours [10] and is recommended for use with semiconducting oxide gas sensors in which the resistance falls with increasing gas concentration.

The electronic nose comprised a set of either 12 or four commercially available Taguchi gas sensors (Figaro Engineering Inc., Japan); see Table 1 for the choice of sensors. The odour sensors have a sensitivity to certain gases at the ppm level. Measurements were made under constant ambient conditions (e.g., at 30 °C and 50% RH). We shall now briefly describe the implementation of two different neural network architectures for recognizing three different classes of coffee with 89 patterns and six different classes of water constituents with 60 patterns.

3.3. Coffee data

The coffee data set provides an interesting challenge for the fuzzy neural models. It consisted of 89 patterns for three different commercial coffees, 30 replicates of coffee A (a medium-roasted coffee blend of type I), 30 replicates of coffee B (a dark-roasted coffee blend, also of type I) and 29 replicates of coffee C (a dark-roasted coffee of a different blend, type II). Looking at the descriptive statistics for the individual sensor measurements, it was recognized that the nature of the variance in the sensor data would be difficult to model. It was soon realized that 100% recognition was unlikely to be achieved. The testing was performed using n-fold cross-validation.[1] The initial data set was segmented to give either a training set of 80 patterns and a test set of nine patterns for the first two coffees (this was done with nine fold), and then 81 patterns for training and eight patterns for testing the last coffee. This was necessary because the third class of coffee had one missing pattern. Each pattern consisted of 12 sensor values, x_{ij}. The patterns constituting the training and testing set were rotated so that in every fold we had a unique training and testing set. The $12\times3\times3$ architecture was trained using both Silva's method (a modification of the standard non-fuzzy back-propagation method) and its fuzzy counterpart. Although the weights for our fuzzy model were within the [0,1] range, the sensor data themselves were not coded in any particular way.

[1] A bootstrapping method could have been used to improve the true error prediction, but we wanted to compare the results with earlier work that used cross-validation [13].

Table 1
Commercial semiconducting oxide gas sensors from Figaro Engineering Inc., Japan used to analyse the coffee and water samples

Sensor No.	Coffee samples	Water samples
TGS 800	√	√
TGS 815	×	√
TGS 816	×	√
TGS 821	×	√
TGS 823	×	√
TGS 824	×	√
TGS 825	√	√
TDS 830	√	√
TGS 831	×	√
TGS 842	×	√
TGS 880	√	×
TGS 881	×	√
TGS 882	×	√
TGS 883	×	√
Total	4	12

3.4. Water data

In this case the data set was collected using a smaller portable four-element electronic nose rather than the 12-element system used to collect the coffee data. There were in all 60 different patterns for six different types of water. The headspace of two vegetable-smelling waters types A and B, a musty water, a bakery water, a grassy water and a plastic water were analysed. Taking 10 folds again (rotating the patterns in training and testing sets), the network was trained with 54 patterns at any one time and tested with the remaining six patterns. Each pattern consisted of four sensor values. The neural network used had a $4\times6\times6$ architecture just like its fuzzy counterpart.

4. Data analysis using fuzzy neural model

In order to illustrate how a fuzzy neural model works, let us consider the above problem of discriminating between a set of different coffee samples. The first step is to define the training and testing sets. The training set can contain 27 patterns of each coffee (i.e., A, B, and C), a total of 81 patterns (about 90% of the patterns), and a testing set of two or three patterns of each type, a total of eight or nine (10% of all patterns). The next step is to obtain the starting weights, which are no longer random weights as in conventional networks. These will be obtained using possibility distribution functions (see Fig. 1). It is possible to use the permutations of different coffees with different sensors to yield many distributions (e.g., 36 different distributions can be drawn with three different coffees and 12 sensors). In order to find the weights, a choice must be made of which coffee patterns will be used to generate weights (since sensor values of coffees A, B and C differ significantly, only one coffee type can yield membership values). We chose coffee A data to assist in this process, since the sensors have registered higher values than in the case of coffees B and C (since medium-roasted coffees contain more volatile molecules than darker-roasted ones) and noise levels here are supposed to be higher. Out of the 27 patterns used for training, one pattern is taken out at random called **P**. The remaining 26 patterns are used to generate the distribution for each sensor (i.e., a total of 12). The formula used for such a process is described by Zadeh [5] as shown in Fig. 1. It may be seen that the possibility of occurrence of any measurement decreases quadratically as it gets further away from the mean value. The variable B in the formula is the measurement for which the possibility value is 0.5 and is also known as the 'cross-over' point. A further explanation of the details of the formula can also be found in Mamdani and Gaines [14]. Once all of the distributions have been generated ($D1, D2, ..., D12$), the membership of sensor values in pattern **P** ($s_1, s_2, ..., s_{12}$) is determined. This means we find the membership of s_i in distribution D_i (let us say it is m_i) for **P**.

Now let us describe the network mathematically. The inputs nodes can be defined by a vector $\boldsymbol{l}$, the hidden nodes by a vector $\boldsymbol{m}$ and the output nodes by a vector $\boldsymbol{n}$. The membership value m_i serves as a weight between l_i and all nodes of $\boldsymbol{m}$. Hence we can determine the weights of all the neurons connecting the input layer to the hidden layer.

A very similar approach is adopted for finding the weights connecting the hidden layer to the output layer, but rather than using the sensor value distributions, the hidden-node output distributions are used. In order to obtain these (if two-layer networks are being used), the network needs to be initially trained for a few iterations with random weights in the non-fuzzy mode. The hidden-node outputs can then be separately analysed following the steps given above.

4.1. Example

Let us see the role of possibility distribution in the Sensor 1 data for coffee A. We have chosen the first 26 values and found the following statistics:

$n=26$

Mean $(Y)=0.0706$

Min $(X)=0.0564$

$B=(X+Y)/2=0.0635$

Let us find the membership value of two measurements chosen at random, $v=0.076$ and $v=0.1011$. (Please refer to the formula in Fig. 1 for the following calculation. A membership value is the possibility that v is the member of the set of all 26 Sensor 1 values.)

When $v=0.076$,

$M=1-S\ (0.0706, 0.10235, 0.1341)$

$=1-0.0144$

$=0.985$

(This is expected since the membership value of any value very close to the mean is nearly 1.)

When $v=0.1011$,

$M=1-S\ (0.1011, 0.0706, 0.10235, 0.134)$

$=1-0.4628$

$=0.537$

5. Results

It was evident that the sensor outputs were non-linear in concentration and contained significant errors attributable to systematic noise. Initially, after trying several different training algorithms and architectures on a non-fuzzy neural network, the success rate was no better than 86% on the coffee data and no better than 75% on the water data.

Tables 2 and 3 summarize the results of our data analysis, and show the superior performance of the fuzzy neural model when compared to the BP technique. Note that when the difference in the final output value and the desired value of any output-layer node was above the error tolerance limit, it was tagged as *misclassified.* If more than half of the nodes in

Table 2
Results of analysing the coffee data. 81 patterns were used for training with nine patterns tested in each fold

Fold	Patterns misclassified by FNN	Nodes misclassified by FNN	Patterns misclassified by BP	Nodes misclassified by BP
1	1	2	1	2
2	0	0	1	2
3	1	2	2	4
4	1	2	3	6
5	1	2	1	2
6	1	2	1	2
7	0	0	1	1
8	0	0	1	2
9	1	2	3	5
10	1	1	2	2
Total	7	13	16	28

Table 3
Results of analysing the tainted-water data. 54 patterns were used for training with six patterns tested in each fold

Fold	Patterns misclassified by FNN	Nodes misclassified by FNN	Patterns misclassified by BP	Nodes misclassified by BP
1	1	3	3	5
2	1	2	2	3
3	0	0	1	1
4	2	3	2	3
5	1	2	2	2
6	0	0	1	2
7	1	2	3	4
8	1	2	3	4
9	1	2	1	2
10	1	2	1	2
Total	9	18	19	28

a pattern were misclassified, the pattern itself was described as *misclassified*.

The fuzzy neural models had about half the number of misclassified patterns compared to their non-fuzzy counterparts. In addition, they converged in less time and with a much reduced error. It should also be stressed that better results were not simply obtained because of a relatively smaller training set compared to other applications, because the non-fuzzy models were gauged with their best start of random weights. For this, the best training performance of the first 10 starts was taken for comparison. The accuracy had now improved to 93% on coffees and 85% on water data by making use of the fuzzy neural model compared to the figures of 86% and 75% before. [2] This is a significant increase in terms of the total number of patterns correctly classified. A t-test was done on the coffee and water data shown in Tables 2 and 3. The null hypothesis H_0 demonstrated that there was no significant difference between the mean number of misclassified nodes and patterns using the FNN model and the back-propagation model for the coffee and water data. In the case of coffee data, the hypothesis H_0 was comfortably rejected at 5% significance level ($t=-3.86$, $p=0.002$ for patterns [3] and $t=-3.50$, $p=0.0034$ for nodes). The same results were obtained for the water data ($t=-5.01$, $p=0.0004$ for patterns and $t=-3.35$, $p=0.0042$ for nodes). This shows that our FNN is a significantly better technique than the conventional BP network.

6. Conclusions

Fuzzy neural networks (FNNs) have been shown to manage uncertainty in real-world sensor data. Their performance on electronic nose data was found to be superior to that of their non-fuzzy neural counterparts. We believe that this was due to the possibility distribution for weight determination averaging out the uneven uncertainty found in the poor semiconducting oxide gas sensors. This is especially important when there is a huge search space and a good starting point is required. The performance given by non-fuzzy networks depends on the initial set of random weights or other training parameters. In our comparison we used a good non-fuzzy back-propagation network and so our FNN results would be even more favourable if compared to a 'vanilla' back-propagation network. FNNs are generic and so may be applied to areas in which standard neural networks are currently employed. In conclusion, the introduction of fuzzy parameters into conventional neural networks can offer a significant advantage when solving difficult classification problems such as that presented by electronic nose instrumentation.

Acknowledgements

The authors wish to thank Mr T. Tan and Miss I. Ene who gathered the coffee and water electronic nose data, respectively. We also thank Mr John Davies of Severn Trent Water for providing us with the water samples.

References

[1] B. Kosko, *Neural Networks and Fuzzy Systems — A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, 1992, pp. 263–270.
[2] D. Hammerstrom, Neural networks at work, *IEEE Spectrum, 30* (1993) 26–33.
[3] S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, CA, USA, 1991.
[4] A.A. Fekadu, Multilayer neural networks, genetic algorithms and neural tree networks, *M.Sc. Dissertation*, University of Warwick, UK, 1992.

[2] Note that linear discriminant function analysis yielded a value of only 80%, see Ref. [13].

[3] The critical t value at 5% significance level and 9 degrees of freedom is 1.83.

[5] L.A. Zadeh, *Fuzzy Logic and Its Applications*, Academic Press, New York, 1965, pp. 29–33.
[6] D. McNeill and P. Freibeger, *Fuzzy Logic*, Touchstone Books, New York, 1993.
[7] D. Dubois and H. Prade, *Fuzzy Sets and Systems*, Vol. 144, Academic Press, New York, 1980.
[8] M.M. Gupta and J. Qi, On fuzzy neuron models, in L.A. Zadeh and J. Kacprzyk (eds.), *Fuzzy Logic for the Management of Uncertainty*, John Wiley, New York, 1992, pp. 479–490.
[9] S. Singh, Fuzzy neural networks for managing uncertainty, *M.Sc. Dissertation*, University of Warwick, UK, 1993.
[10] J.W. Gardner and P.N. Bartlett, A brief history of electronic noses, *Sensors and Actuators B, 18–19* (1995) 211–220.
[11] J.W. Gardner, E.L. Hines and M. Wilkinson, Application of artificial neural networks in an electronic nose, *Meas. Sci. Technol., 1* (1990) 446–451.
[12] H.V. Shurmer, J.W. Gardner and H.T. Chan, The application of discrimination techniques to alcohols and tobaccos using tin oxide sensors, *Sensors and Actuators, 18* (1989) 361–371.
[13] J.W. Gardner, H.V. Shurmer and T.T. Tan, Application of an electronic nose to the discrimination of coffees, *Sensors and Actuators B, 6* (1992) 71–75.
[14] E.H. Mamdani and B.R. Gaines (eds.), *Fuzzy Reasoning and its Applications*, Academic Press, New York, 1981.

Biographies

Sameer Singh was born in New Delhi, India in 1970. After receiving his BE in computer engineering with distinction from BIT, India, he got his M.Sc. in information technology for manufacturing from Warwick University in 1993. He has recently completed his Ph.D. in the application of artificially intelligent methods for quantifying recovery in speech and language disorders. He is now a lecturer in the School of Computing, University of Plymouth, UK. He is a fellow of Royal Statistical Society (UK) and associate member of the Institution of Electrical Engineers (UK). His main research interests include neural networks, fuzzy logic, expert systems and linguistic computing.

Evor Hines is a senior lecturer in electronics in the Department of Engineering at the University of Warwick. His research interests include intelligent systems engineering areas such as artificial neural networks, genetic algorithm and fuzzy logic. Over the last seven or so years he has been involved in work applying these techniques in areas such as sensor data processing (e.g., electronic nose), medical data processing (e.g., spectral imaging, impedance imaging), business data processing (e.g., stock prediction, sales forecasting), amongst others. He has been involved in the publication of more than 70 papers.

Julian W. Gardner was born in Oxford, UK in 1958. He received his B.Sc. with highest honours in 1979 from Birmingham University and the Ph.D. degree from Cambridge University in 1983. His dissertation focused on electron conduction in thin-film devices. From 1983 to 1987 he was in industry, where he worked on instrumentation and sensors. He is currently a reader in microengineering in the Department of Engineering at Warwick University. His research interests include the modelling of electron devices, silicon microsensors, chemical sensor array devices and electronic noses. He is author or co-author of over 150 technical papers and has recently published a book on microsensors. Dr Gardner is a member of the Institution of Electrical Engineers (UK) and the Institute of Electrical and Electronic Engineers (US). In 1989 he received an Esso Centenary Education Award from the Royal Society and Fellowship of Engineering, London and was, in 1994, an Alexander von Humboldt Research Fellow in Germany.