Sensors and Actuators B 160 (2011) 760–770

Contents lists available at SciVerse ScienceDirect

Sensors and Actuators B: Chemical

journal homepage: www.elsevier.com/locate/snb

Gases concentration estimation using heuristics and bio-inspired optimization models for experimental chemical electronic nose

Lei Zhang*, Fengchun Tian, Chaibou Kadri, Guangshu Pei, Hongjuan Li, Lina Pan

College of Communication Engineering, Chongqing University, 174 ShaPingBa District, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 8 April 2011
Received in revised form 22 July 2011
Accepted 22 August 2011
Available online 10 September 2011

Keywords:
Particle swarm optimization
Adaptive genetic strategy
Back-propagation multilayer perceptron neural network
Electronic nose
Concentration estimation

ABSTRACT

By virtue of an electronic nose, detection and concentration estimation of harmful gases indoor become feasible by using a multi-sensor system. The estimation accuracy in actual application is constantly aspired by manufactures and researchers. This paper analyzes the application of different bio-inspired and heuristic techniques to the problem of concentration estimation in experimental electronic nose application. In this paper, seven different particle swarm optimization models are considered including six models used before in numerical function optimization, and a novel hybrid model of particle swarm optimization and adaptive genetic algorithm, for optimizing back-propagation multilayer perceptron neural network. We describe the performance of a particle swarm optimization technique, an adaptive genetic strategy and a back-propagation artificial neural network approach to perform concentration estimation of chemical gases and improve the intelligence of an E-nose.

1. Introduction

Electronic noses (E-noses) employ an array of chemical gas sensors and have been widely used for the analysis of volatile organic compounds [1] and vapor chemicals [2]. Pattern recognition provides a higher degree of selectivity and reversibility to the system, leading to an extensive range of applications. An E-nose is an instrument consisting of an array of reversible, but only semi-selective gas sensors coupled to a pattern recognition algorithm [3] and the formal structure of an electronic nose system has also been introduced. An excellent overview of the electronic nose technology is contained in Gardner and Bartlett [4] and techniques for processing the sensor responses were reviewed by Jurs [5] and Gutierrez-Osuna [6]. Data acquisition is the first step for data analysis. Sensors collect the data and convert it into an electrical signal pattern that is more suitable for computer analysis [3]. The output of each sensor is a pattern vector in the pattern space. Then, the pattern vector is passed into the second stage, feature selection. Feature selection is the process of identifying the most effective subset of the original features for obtaining the smallest classification error. The data produced by an electronic nose can be classified into two kinds: a set of semi-independent variables (the sensor array outputs) and a set of dependent variables (gases concentrations). Related applications of electronic nose technique have been researched [7–10]. The selectivity of the instrument is achieved through the application of pattern recognition methods to responses of the sensor array [11,12]. Regarding the neural network training, back-propagation neural network (BP-MLP-NN) which is a gradient based method, has been widely used to classify nonlinearly separable patterns in real application for its strong ability in recognition [13]. However, BP neural network still posses some inherent problems. First, BP model can easily get trapped in local minima for the problems of pattern recognition and complex functions approximation [14], so that a local optimal solution is obtained. Second, the solutions are different for every train with the random initial weights. Thus, neural network optimization algorithms have been proposed by researchers to improve the ability of finding global minima of BP by using genetic algorithm (GA) [15–18].

In this paper, we describe the application of several heuristics and bio-inspired optimization models for multidimensional nonlinear concentration estimation problems in experimental electronic nose applications. Similar heuristics and bio-inspired optimization model was also used for curve fitting in chemistry [19]. Particle swarm optimization (PSO) algorithm, which was originally developed by Kennedy and Eberhart, is an optimization method based on social behavior simulations [20], used to visualize the movements of a flock of birds which has the superior dynamic characteristics. Currently, a number of improved PSO models using different strategies have been investigated by researchers, such as inertia weight approach (IWA) [21], adaptive PSO (APSO) [13], attractive and repulsive PSO (ARPSO) [22], particle swarm optimization based on diffusion and repulsion (DRPSO) [23] and PSO

* Corresponding author. Tel.: +86 13 629788369; fax: +86 23 65103544.
E-mail address: leizhang@cqu.edu.cn (L. Zhang).

0925-4005/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.snb.2011.08.060

hybrid model based on bacterial chemotaxis (PSOBC) [24]. PSO has been widely used in many fields [25–28]. It has attracted attention in recent years because of its simplicity and high performance in searching global optima. Similar to genetic algorithm, the PSO algorithm is also an optimization tool based on population which is initialized with a population of random solutions and search for optimum by continuous updating of generations. With the introduction of inertia weights w first proposed, the performance of the PSO algorithm has been improved significantly.

The genetic algorithm (GA) is a global search procedure that searches from one population of points to another. Sexton and Gupta demonstrated that the GA significantly outperforms BP in optimizing neural network for a set of computer generated data [18]. Genetic algorithm is first used by choosing an objective function for optimizing the network. The objective function values are then used to assign probabilities for each point of the population, and points generating the lowest error are the most likely to be represented in the new population. The points forming this new population are then randomly paired for the crossover operation. This crossover operation results in each new point having information from both parent points. In addition, each point has a small probability of being replaced with a value randomly chosen from the parameter space. This operation is named as mutation. The genetic algorithm has been applied to particle swarm optimization for radial basic function (RBF) network [17] and order clustering [28].

In this paper, considering the fast convergence velocity with strong ability to find the global best solution of PSO and so many directions that genetic algorithm can also search in, an improved PSO optimization neural network based on adaptive genetic strategy (PSOAGS) has been proposed. We take three harmful gases (formaldehyde, CO and NO_2) for experiments, through analysis of simple approaches such as principle component regression (PCR), partial least square (PLS), a third-order polynomial fitting based nonlinear least square (NLS) and seven optimized neural network models, results demonstrate that the proposed heuristics and bio-inspired optimization neural network model can significantly improve the ability of global search for optimum, and it can more accurately estimate the concentrations of gases by an electronic nose.

2. Particle swarm optimization models considered

2.1. Standard particle swarm optimization (SPSO)

In this standard PSO system, a number of particles cooperate to search for the best solutions by simulating the movement and flocking of birds. These particles fly with a certain velocity and find the global best position after certain generations. At each generation t, the velocity is updated and the particle is moved to a new position. This new position is simply calculated as the sum of the previous position and the new velocity. The mathematical notation of PSO is defined as follows.

Supposing the dimension for a searching space is D, the total number of particles is N, the position of the ith particle can be expressed as vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$; the best position of the ith particle searching until now is $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$; the best position of all the particles searching until now is $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$; the velocity of the ith particle is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, then the standard PSO can be illustrated as:

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_{1i}^p(t) \cdot [p_{id}(t) - x_{id}(t)] + c_2 \cdot r_{2i}^p(t) \cdot [p_{gd}(t) - x_{id}(t)] \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad 1 \le i \le N, \quad 1 \le d \le D \quad (2)$$

where c_1, c_2 are the acceleration constants with positive values; w is called inertia factor.

$$r_{1i}^p(t) \leftarrow U(0,1), \qquad r_{2i}^p(t) \leftarrow U(0,1) \quad (3)$$

Noteworthy is that, placing a limit on the velocity v_{max} and adjusting the inertia weight w, the PSO can achieve better search performance.

$$v_p^i(t+1) = \begin{cases} v_{\max}^i, & v_p^i(t+1) > v_{\max}^i \\ -v_{\max}^i, & v_p^i(t+1) < -v_{\max}^i \\ v_p^i(k+1), & \text{otherwise} \end{cases} \quad (4)$$

Further, the inertia weight approach (IWA-PSO), in which a linear reduction from a large value to a small value during the search has been proposed [21]. The w is updated by:

$$w(t) = w_{start} - (w_{start} - w_{end}) \cdot \frac{t}{t_{\max}} \quad (5)$$

where t_{max} is the maximum generations, w_{start} is the initial value, and w_{end} is the terminal value.

2.2. Adaptive particle swarm optimization (APSO)

The adaptive particle swarm optimization (APSO) algorithm which is based on the standard PSO, was firstly proposed [21]. The APSO is called adaptive PSO for its new inertial weight improved by [13].

$$w = \begin{cases} w_{start} - \dfrac{w_{end}}{\max gen1} \cdot gen, & 1 \le gen \le \max gen1 \\ (w_{start} - w_{end}) \cdot \exp\left[\dfrac{\max gen1 - gen}{k}\right], & \max gen1 \le gen \le \max gen2 \end{cases} \quad (6)$$

where w_{start} is the initial inertial weight, w_{end} is the ending inertial weight of linear section, max $gen2$ is the total searching generations, max $gen1$ is the used generations that inertial weight reduced linearly, and k should be adjusted for the best solution. In the experiment, we set k to 2.

2.3. Attractive and repulsive particle swarm optimization (ARPSO)

The attractive and repulsive PSO (ARPSO) was introduced to overcome the problem of premature convergence [22]. It uses a diversity measure to control the swarm. This algorithm is based on the phases between attraction and repulsion. The velocity update formula of the particles based on the SPSO was described as:

$$v_{id}(t+1) = w \cdot v_{id}(t) + dir \cdot c_1 \cdot [p_{id}(t) - x_{id}(t)] + dir \cdot c_2 \cdot [p_{gd}(t) - x_{id}(t)] \quad (7)$$

where dir is the coefficient (with value of 1 or −1) which decides whether the particles attract or repel each other, c_1 and c_2 are the random number in the range of [0,2]. The dir can be determined by:

$$\text{if } (dir > 0 \text{ and } \text{diversity} < d_{low}), \quad \text{then } dir = -1 \quad (8a)$$

$$\text{if } (dir < 0 \text{ and } \text{diversity} > d_{high}), \quad \text{then } dir = 1 \quad (8b)$$

where d_{low} denotes the low diversity, d_{high} denotes the high diversity. The diversity can be calculated using the following formula [24]:

$$\text{diversity}(S) = \frac{1}{|S| \cdot |L|} \cdot \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^{n} (p_{ij} - \bar{p}_j)^2} \quad (9)$$

where S is the swarm, $|S|$ is the size of the swarm, $|L|$ is the length of the longest diagonal in the search space S, n is the dimensionality of the problem, p_{ij} is the jth value of the ith particle. $\bar{p}_j$ is the jth value of the average points $\bar{p}$. The attractive phase is also defined as the SPSO, while the individual particle is repelled by the best known particle position in the repulsion phase.

2.4. Particle swarm optimization based on diffusion and repulsion (DRPSO)

The diffusion repellent PSO model, in which particle swarm is not only attracted by the region of the best solution but also carry out diffusion repellent movement by the region of the worst solution for the potentially dangerous property premature of PSO, was proposed by [23].

The velocity updating formula of DRPSO based on the SPSO model is given by:

$$v_{id}(t+1) = -w \cdot f(v_{id}(t), \theta) - c_1 \cdot rand \cdot [W_{id}(t) - x_{id}(t)] - c_2 \cdot rand \cdot [W_{gd}(t) - x_{id}(t)] \quad (10)$$

where W_{id} is the worst previous particle, W_{gd} is the worst particle among all the particles, and $f(v_{id}(t), \theta)$ is $v_{id}(t)$ after clockwise rotation in a certain random angle θ [0°,360°].

2.5. Particle swarm optimization based on bacterial chemotaxis (PSOBC)

This PSO is a hybrid model based on bacterial chemotaxis, and proposed by [24]. In this model, the particle swarm is not only attracted by the regions where the best results were found, but also repelled by the regions where the worst results were found. In PSOBC model, the location of the worst point found so far and location of the worst point found by the total particles are added to the SPSO algorithm, the velocity updating formula is illustrated as follows:

$$v_{id}(t+1) = w \cdot v_{id}(t) - c_1 \cdot rand(\cdot) \cdot [W_{id}(t) - x_{id}(t)] - c_2 \cdot rand(\cdot) \cdot [W_{gd}(t) - x_{id}(t)] \quad (11)$$

where W_{id} is the worst previous particle, W_{gd} is the worst particle among the total particles. In this model, the same diversity guided method as the ARPSO and the DRPSO model was also employed and the individual particle was no longer attracted, but repelled by the worst known particle position and its own previous worst position.

3. Hybrid evolutionary algorithm

In this section, considering the fact that PSO has potential dangerous properties such as premature convergence and stagnation, a hybrid evolutionary algorithm based on an improved PSO and adaptive genetic strategy (PSOAGS) is presented. In actual applications, large number of samples and the characteristic of real time are necessary; thus, it is meaningful and important to find the best global solution in a fast converging velocity as soon as possible, aiming at reducing the search generations. This section describes the strategies of inertial weight, acceleration constant based on cosine mechanism. Also, an improved GA with adaptive probability variation is presented.

3.1. PSO algorithm using cosine mechanism

Through the analysis in Section 2, the inertial weight and acceleration constants influence the convergence of PSO in searching the global best solution. The linear reduction and exponent reduction of inertial weight can play a good role in finding the optimum, but it is easy to trap into the local minimum [13]; thus, a regular form of cosine wave is presented in this PSOAGS model. The strategies of inertial weight and acceleration constants based on the cosine mechanism are illustrated as follows:

$$w(t) = a + b \cdot \cos\left(\frac{t}{A} \times \pi\right) \quad (12)$$

where parameters a and b can be adjusted according to the boundary of w_{max} and w_{min}.

For parameters setup of inertial weight w, a suitable value for the inertia weight w usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of generations required to locate the optimum solution. We determine the w_{min} as 0.4 and the w_{max} as 0.9 which are recommended in PSO literatures [13,21–24]. After the fixed minimum and maximum of w, we can derive that $a = 0.65$, and $b = 0.25$ through Eq. (12). For the acceleration coefficients, $c_1 = c_2 = 2$ were proposed as default values in the previous literatures, but experimental results indicate that it can provide better results when acceleration coefficients c_1 and c_2 were described in a similar form with Eq. (12). We set the c_{min} to 0.6 and the c_{max} to 1.5 for both c_1 and c_2, so that $a = 1.05$ and $b = 0.45$. The value of A controls the frequency of cosine function, which also denotes the velocity of changing w; through experimental simulations, we finally provide A with 20 which can be a good choice.

3.2. Adaptive genetic strategy (AGS)

The crossover probability and mutation probability are the important parameters in genetic algorithm, for that the two probabilities decide the quality of population. It has been well established in GA literature that moderately large values of the crossover probability p_c ($0.5 < p_c < 1$) and small values of the mutation probability p_m ($0.001 < p_m < 0.05$) are essential for the successful working of GAs [29–31]. And GA has also been used to architecture optimization of neural network [32]. In this paper, we make the crossover and mutation probability varied adaptively with the Euclidean distances between individuals of the whole population. It is known that the Euclidean distances will be smaller and smaller with the continue search evolutions. We define the distance vector between individuals of each generation t as follows:

$$\vec{d} = (d_1^{(t)}, d_2^{(t)}, \ldots, d_J^{(t)}) \quad (13)$$

where J denotes the possible combinations among individuals, which can be calculated as:

$$J = PopSize \times \frac{PopSize - 1}{2} \quad (14)$$

where $PopSize$ denotes the number of the total particles ($PopSize = 50$, in this paper). Then the adaptive crossover probability of each generation t is shown as:

$$\text{probability}_{crossover} = \frac{\vec{d}_{\max}^{(t)} - \vec{d}_{average}^{(t)}}{\vec{d}_{\max}^{(t)} - \vec{d}_{\min}^{(t)}} \quad (15)$$

The adaptive mutation probability of each generation is shown as

$$\text{probability}_{mutation} = \frac{\vec{d}_{average}^{(t)}}{\vec{d}_{\max}^{(t)}} \quad (16)$$

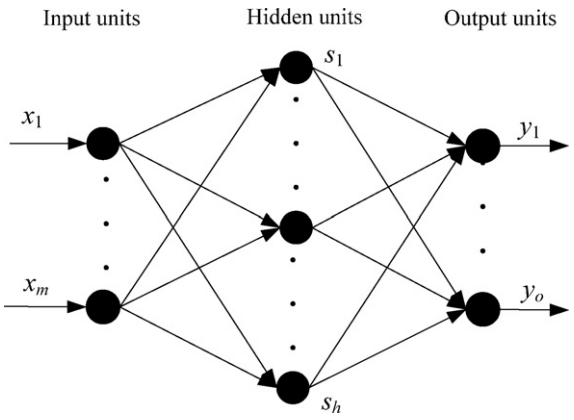

Fig. 1. Structure with two-layered feed-forward ANN.

Through Eqs. (15) and (16), we can see that the probability can be adaptively changed according to the whole population; when the GA converges to a local optimum, p_c and p_m have to be increased.

Combined with the adaptive GA operators such as selection, crossover, and mutation, the repetitive search guideline is used for improving the PSO. For each particle, the fitness values in solution searching are evaluated from the PSO and AGS, respectively. Note that, the genetic operators only perform on the positions of each particle. If the best fitness of PSO is superior to that of AGS, keep the best particle of PSO; otherwise, the best particle of AGS is used in the next generation instead. In addition, to prevent the best solution from trapping into the local optimal, we present the adaptive mutation of the best particle according to the best fitness value. If the best fitness is invariable with the maximum number of tolerated generations, the best particle so far will be randomly mutated according to the following formula:

$$Xbest = Xbest + R \times (Xworst - Xbest) \tag{17}$$

where $R \leftarrow U(0,1)$, *Xbest* is the best particle so far, *Xworst* is the worst particle so far; the purpose is to change *Xbest* in the range of *Xbest* and *Xworst*.

4. Concentration estimation algorithm

Through the analysis above, we have made a significant investigation on the heuristics and bio-inspired optimization algorithms considered. In this section, we aim to apply the PSO based techniques and genetic algorithm to optimize feed forward artificial neural network based on error back-propagation for gases concentration estimation by an electronic nose and verify the effectiveness of these evolutionary algorithms in concentration estimation.

4.1. Back-propagation multilayer perceptron neural network (BP-MLP-NN)

The back-propagation algorithm (BP) based on gradient descending was proposed by [33]. Rumelhart further formulated the standard back propagation algorithm (BPA) for multilayered perceptrons. The architecture of simple multilayered perceptron neural network is shown in Fig. 1 which presents a two layered neural network. BP algorithm has been improved by several adaptive back-propagation algorithms for its inherent disadvantages [13,17,18,34]. BP network based on a gradient descend, has been widely used to classify nonlinearly separable patterns in real application for its strong ability in recognition [13]. However, BP neural network still posses some inherent problems. First, BP model can easily get trapped in local minima for the problems of pattern recognition [33], and fail to find the global optimal solution. Second, the initial weight matrix *W* and bias vectors *B* of back-propagation neural network are randomly produced for training so that different weights and biases would produce different trained neural networks. Thus, obtaining the global minimum of regression error by only using back propagation neural network becomes little impractical.

4.2. Hybrid BP-MLP-NN and optimization algorithms

During weights optimization, the first step is to determine the encoding strategy. In this paper, each particle is encoded as a vector with real values. For the neural network structure, each particle should represent all weights of the whole network structure. According to the subsection mentioned above, the structure of BP-MLP-NN is *m*-*h*-*o* consisted of one hidden layer, then the total number of weights can be calculated as $m \times h + h + h \times o + o$; thus, the dimension of each particle is shown as

$$n = m \times h + h + h \times o + o \tag{18}$$

where *m*, *h*, and *o* denote the number of input neurons, hidden neurons and output neurons, respectively. To the BP-MLP-NN with three layers (one input layer, one hidden layer and one output layer), the weight matrix W_1 ($m \times h$), W_2 ($h \times o$) and the bias vectors B_1 ($h \times 1$), B_2 ($o \times 1$) can be obtained from each particle *x* using the following decoding rule:

$$\underbrace{x_1, x_2, \ldots, x_{m\times h}}_{W_1}, \underbrace{x_{m\times h+1}, \ldots, x_{m\times h+h}}_{B_1}, \underbrace{x_{m\times h+h+1}, \ldots, x_{m\times h+h+h\times o}}_{W_2}, \underbrace{x_{m\times h+h+h\times o+1}, \ldots, x_n}_{B_2} \tag{19}$$

The active functions of the hidden layer and output layer are selected as log sigmoid and pure linear function. The output of the *j*th hidden node is:

$$f(node_j) = \frac{1}{1 + e^{-\left(\sum_{i=1}^{m} w_{ij} \cdot s_i - b_j\right)}}, \quad j = 1, \ldots, h \tag{20}$$

where w_{ij} (one element of W_1) is the connection weight from the *i*th node of input layer to the *j*th node of hidden layer, and b_j (one element of B_1) is the bias of the *j*th hidden layer.

The output of the *k*th output node is:

$$y_k = \sum_{j=1}^{h} W_{kj} \cdot f(node_j) - b_k, \quad k = 1, \ldots, o \tag{21}$$

where w_{kj} (one element of W_2) is the connection weight from the *j*th hidden node to the *k*th output node, b_k (one element of B_2) is the bias of the *k*th output layer.

Considering the robustness of BP-MLP-NN, to avoid big difference between train error and test error and overfitting in training, the objective function of PSO is selected as the maximum between the relative validation errors and the relative training error to improve the robustness of NN, which is shown by:

$$F = \max\left\{ \frac{1}{N_1} \cdot \sum_{i=1}^{N_1} \left| \frac{Y_i^{tr} - T_i^{tr}}{T_i^{tr}} \right|, \frac{1}{N_2} \cdot \sum_{j=1}^{N_2} \left| \frac{Y_j^{cv} - T_j^{cv}}{T_j^{cv}} \right| \right\} \tag{22}$$

where N_1 denotes the number of training samples, N_2 denotes the number of cross-validation samples, Y^{tr} and T^{tr} represent the estimated and target concentrations of train samples, Y^{cv} and T^{cv} represent the estimated and target concentrations of

```
randomly initialize the positions and velocities of the total particles;
for i=1 to number of particles
    Evaluates each initialized particles using BP-MLP-NN train and test, through Eq.(22);
end for;
set Generation=0;
while Generation <Maximum generations
    Generation=Generation+1;
    Update the best fitness, the positions, and the velocities using Eq.(1, 2) under the strategies based
     on the cosine mechanism;
    for i=1 to number of particles
        evaluates each updated particle using BP-MLP-NN and obtain the best fitness PSO_best;
        update the positions using AGS;
        evaluates each updated particle from AGS using BP-MLP-NN and obtain the best fitness
       AGS_best;
    end for;
    if PSO_best < AGS_best
      keeps the best particle from PSO and the worst particle from AGS, go to the next generation;
    else the best particle from AGS and the worst particle from PSO are kept to the next generation;
    end if;
    if the best fitness is unchanged during the maximum tolerated generations
      Mutated the best particle using Eq.(17);
    end if;
end while;
```

Fig. 2. Pseudo-code implementation for the developed PSOAGS model.

cross-validation samples. The detailed evolutionary pseudo-codes of the PSOAGS model are illustrated as Fig. 2.

5. Applications to electronic nose

In this section, the presented heuristics and bio-inspired algorithms are applied to concentration estimation problems for an electronic nose. The experimental design by an electronic nose system is illustrated including the experimental platform, sensor array on the printed circuit board of an electronic nose. Three chemical gases are measured using a devised electronic nose system to estimate the unknown concentration of chemical analytes through the extrapolation ability of optimized BP-MLP-NN.

5.1. Experimental setup

Our sensor array in E-nose system consists of six different types of gas sensors: four from the TGS series (TGS2602, TGS2620, TGS2201A and TGS2201B), an oxygen sensor (O2A2, type of electrochemical), and one GSBT11 sensor of Ogam Technology in Korea. In addition, a module (SHT2230 of Sensirion in Switzerland) with two auxiliary sensors for the temperature (T) and humidity (H) are also used for compensation. The sensors were mounted on a custom designed printed circuit board (PCB), along with associated electrical components. An analog–digital converter is used as interface between the field programmable gate array (FPGA) processor and the sensors. FPGA can be used for data collection, storage and processing. The E-nose system is connected to a personal computer (PC) via a joint test action group (JTAG) port which can be used to transfer data and debug programs. An additional flash memory is used to save the embedded neural network as well as the weights and biases of the neural network trained on the PC. The data sets for these gases are made up of samples in R^8 space, it just means that an input vector with 8 variables can be obtained in each observation; the multidimensional response data set denotes the nonlinear relation with the gas concentration. The gases measurements are implemented by an E-nose in the constant temperature and humidity chamber whose type is LRH-150S in which the temperature and humidity can be effectively controlled in terms of the target temperatures and humidity.

5.2. Gas data sets for train and test

In this paper, for validation of the proposed model, we measured three harmful gases: formaldehyde, carbon monoxide (CO), and nitrogen dioxide (NO_2). Each of the three candidates should be presented in the E-nose in many different concentrations, respectively, and the responses of the sensor array are saved on PC. Totally, 186 samples (dataset) include 68 formaldehyde samples, 47 CO samples and 71 NO_2 samples are measured within one month, and each sample corresponds to different concentrations and environment; the total measurement cycle time for one single measurement was set to 20 min, i.e. 2 min for reference air (baseline), 8 min for gas sampling and 10 min cleaning the chamber through injecting clean air before the next experiment begins. These samples are measured at the target temperatures of 15, 25, 30 and 35 °C and target humidity of 40%, 60%, 80% RH, through different combinations of these target temperatures and humidity. The precisions of the

chamber for temperature and humidity are ±0.1 °C and ±5% RH. And in each group of fixed temperature and humidity (e.g. 25 °C, 60% RH), the samples were measured at growing concentrations within desired range. In our experiments, there was little sensor drift in a short term of one month; thus, drift was not considered. Besides, we can find good reproducibility of the sensors through the repeated experiments under almost the same environment; the very little response difference of each sensor between almost the same two experiments can be neglected. The experimental sample collections were developed in the chamber. The basic experimental platform illustrated in Fig. 3 is the whole experimental process in this work. The part within the dashed lines shows the preparation for formaldehyde gas, and the part within the solid lines illustrates the preparation for CO or NO_2 gas. For the formaldehyde experiments, a gas header is necessary to store the formaldehyde gas and inject into the chamber by a pump, and the rough concentrations are controlled through the time interval of gas exhaust. For CO and NO_2 experiments, standard CO gas and NO_2 gas are prepared for the chamber with desired concentration range, and their concentrations depend on the volume of injected target gas. We can see that a reference meter which can measure the concentration of gases roughly is needed in the process for that we can have a concentration comparison with the reference meter. Note also that the true concentrations of formaldehyde were obtained by GC analysis using spectrophotometer.

To validate the effectiveness of all the models, the training and testing samples are randomly classified, 2/3 of the total samples are used as training and cross-validation samples, and the remaining 1/3 as test samples. The training dataset including cross-validation was constructed of 124 measurements. There are 47 formaldehyde samples, 27 CO samples and 50 NO_2 samples in which 10 formaldehyde samples, 10 CO samples and 10 NO_2 samples are recognized as cross-validation samples. The target concentrations of the training samples are shown in Table 1. The remaining 62 measurements including 21 formaldehyde samples, 20 CO samples and 21 NO_2 samples are used as test dataset, to evaluate the robust performance of the neural network optimization model. Table 2 presents the target concentrations of the testing samples.

Table 1
Target concentrations of 124 training samples.

Formaldehyde (ppm)			CO (ppm)		NO_2 (ppm)			
0.01	0.08	0.02	1	6	5	45	45	49
0.11	0.12	0.15	5.6	1	15	63	65	71
0.13	0.48	0.23	1	5	25	5	5	
0.9	1.03	0.55	5	1	45	30	15	
0.04	0.03	0.9	1	5	65	47	32	
0.05	0.13	0.08	5	1.4	5	65	66	
0.09	0.4	0.16	1	6	15	6	5	
0.92	0.53	1.5	1.5	1.6	30	15	15	
0.04	0.9	0.02	5	6	60	46	49	
0.1	0.04	0.1	1	0.8	3	66	71	
0.25	0.39	0.4	2	5	6	5	6	
0.79	0.9	1.01	3		23	32	17	
0.02	0.03	0.05	5		45	46	50	
0.1	0.15	0.13	1.3		65	61	68	
0.5	0.3	0.65	5		5	5	18	
0.64	0.99		1.3		15	15	34	

5.3. Concentration estimation and data analysis

In this work, the steady state values in the response of each sample as our features for algorithm analysis. Thus, for each sample, a vector of eight feature values (8 sensors) is selected as one observation. Assuming the number of samples is S, we will get a matrix X with 8 row vectors having S columns. This feature matrix X is the input matrix of BP-MLP-NN which will be analyzed using our proposed model. The targets of the BP-MLP-NN are the target concentrations of the corresponding gas samples. The matrix of sensor responses can be normalized as [0,1] using the following formula:

$$X'_{ij} = \frac{X_{ij}}{X_{\max}}, \quad i = 1, \ldots, 8; \quad j = 1, \ldots, S \tag{23}$$

where X_{max} is the maximum saturation value 4095 of each sensor and S is the number of total samples. Based on the input responses X and the target concentrations T, the nonlinear relation between sensors responses and the concentrations can be determined through W_1, W_2, B_1 and B_2 which are obtained by back-propagation algorithm learning based on heuristic and bio-inspired

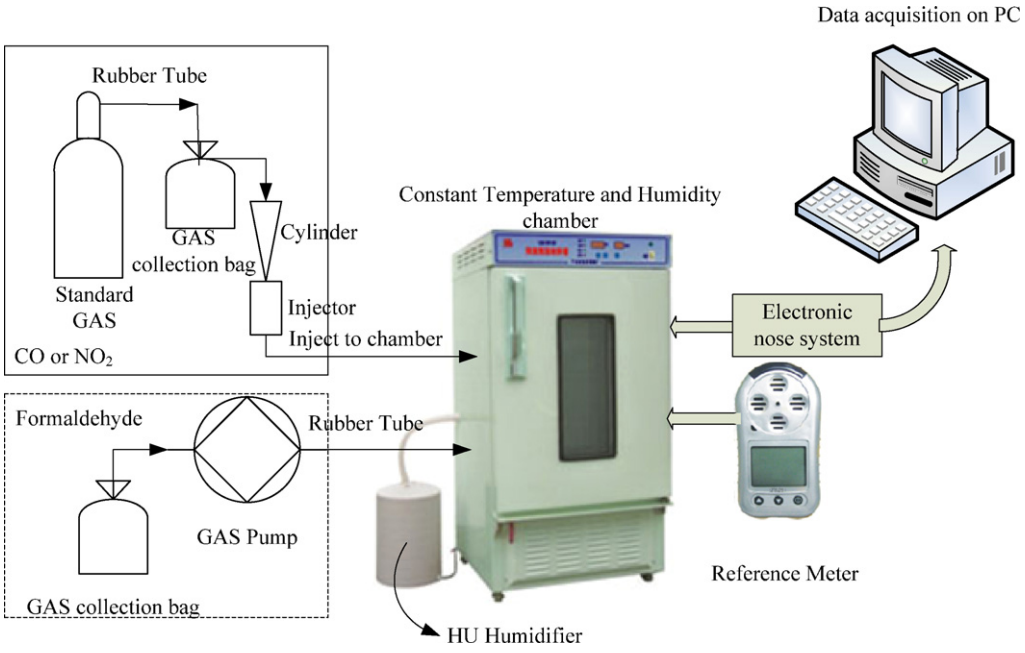

Fig. 3. Experimental platform for gases samples collection by an electronic nose system.

Table 2
Target concentrateions of 62 testing samples.

Formaldehyde (ppm)		CO (ppm)		NO_2 (ppm)	
0.31	0.42	3	2.5	30	15
0.32	0.15	3	4	45	17
0.52	0.33	2.5	2.8	15	50
0.48	0.30	3.5	6	35	45
0.18	0.17	4	4	30	30
0.10	0.15	2.8	3.5	15	32
0.15	0.22	4	2.8	32	16
0.4	0.35	3	2.5	15	6
0.2	0.34	3		30	15
0.36		3		46	
0.49		3.5		30	
0.15		2.8		34	
0.34		3.5		6	

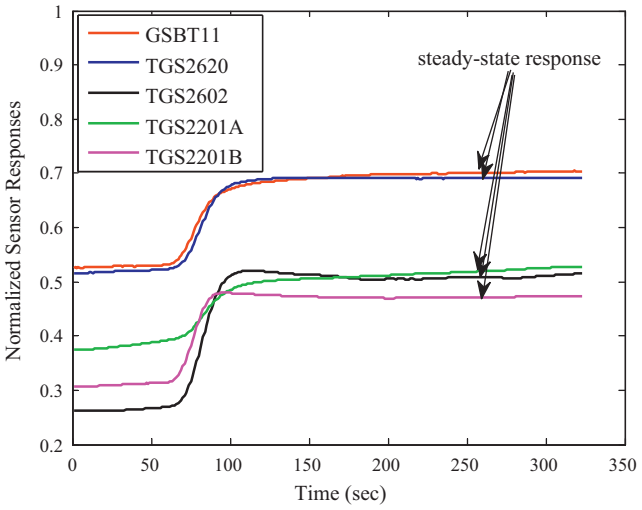

Fig. 4. Normalized gas sensor responses when exposed to formaldehyde with concentration 0.42 ppm under the conditions of temperature = 15 °C, and humidity = 60% RH.

optimization models. Assume the number of samples for each gas is set to n_1, n_2, and n_3, respectively. The training input data matrix P and training targets matrix T of neural network are described as the following forms:

$$P = \overbrace{\begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,n_1} & S_{1,n_1+1} & S_{1,n_1+2} & \cdots & S_{1,n_1+n_2} & S_{1,n_1+n_2+1} & \cdots & S_{1,n_1+n_2+n_3} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,n_1} & S_{2,n_1+1} & S_{2,n_1+2} & \cdots & S_{2,n_1+n_2} & S_{2,n_1+n_2+1} & \cdots & S_{2,n_1+n_2+n_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{8,1} & S_{8,2} & \cdots & S_{8,n_1} & S_{8,n_1+1} & S_{8,n_1+2} & \cdots & S_{8,n_1+n_2} & S_{8,n_1+n_2+1} & \cdots & S_{8,n_1+n_2+n_3} \end{bmatrix}}^{gas\ 1 \quad gas\ 2 \quad gas\ 3} \tag{24}$$

$$T = \overbrace{\begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n_1} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & t_{2,n_1+1} & t_{2,n_1+2} & \cdots & t_{2,n_1+n_2} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & t_{3,n_1+n_2+1} & \cdots & t_{3,n_1+n_2+n_3} \end{bmatrix}}^{gas\ 1 \quad gas\ 2 \quad gas\ 3} \tag{25}$$

where $S_{i,j}$ ($i = 1,\ldots,8$, $j = 1,\ldots,n_1+n_2+n_3$) is the selected feature at the steady state sensor response of one measurement, $t_{1,k}$ ($k = 1,\ldots,n_1$) are the actual concentrations of gas 1 for the n_1 times of measurements, and $t_{2,k}$ ($k = n_1+1,\ldots,n_1+n_2$) are actual concentrations of gas 2 for n_2 times of measurements; the same meaning to gas 3. The values of zero concentration (0 ppm) denote the absence of the corresponding gas components. The arrangement order of gas in the input data matrix P is formaldehyde (gas 1), CO (gas 2) and NO_2 (gas 3), respectively. In this case, the outputs of network are just the estimated concentrations. In one observation, the output is a vector with three values corresponding to concentrations of the three known gas components. The test samples are also arranged as the similar form (24) and (25). In this paper, $n_1 = 37$, $n_2 = 17$, $n_3 = 40$ for training samples; $n_1 = 10$, $n_2 = 10$, $n_3 = 10$ for cross-validation samples; $n_1 = 21$, $n_2 = 20$, $n_3 = 21$ for test samples.

Training was carried out with a sum squared output error goal of 0.005 up to a maximum of 2000 iterations. Sometimes the network did not reach the goal after 2000 iterations and the sum squared error was still higher than 0.005. The objective function used in PSO optimization procedure is shown in Eq. (22). The output matrix Y of neural network should be the predicted concentration matrix shown as:

$$Y = \overbrace{\begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,n_1} & y_{2,n_1+1} & - & \cdots & - & - & \cdots & - \\ - & - & \cdots & - & - & y_{2,n_1+2} & \cdots & y_{2,n_1+n_2} & - & \cdots & - \\ - & - & \cdots & - & - & - & \cdots & - & y_{3,n_1+n_2+1} & \cdots & y_{3,n_1+n_2+n_3} \end{bmatrix}}^{gas\ 1 \quad gas\ 2 \quad gas\ 3} \tag{26}$$

where the diagonal elements are the predicted concentrations of gas 1, gas 2 and gas 3; the values in the positions with '–' which are very close to zero, can be neglected.

In this work, to evaluate the final estimation model, the relative estimation error (train error and test error) of gas l ($l = 1, 2, 3$) are shown as follows:

$$\mathrm{TRE} = \frac{1}{N_l} \cdot \sum_{i=1}^{N_l} \left| \frac{Y^{tr}_{l,i+N_{l-1}+N_{l-2}} - T^{tr}_{l,i+N_{l-1}+N_{l-2}}}{T^{tr}_{l,i+N_{l-1}+N_{l-2}}} \right| \tag{27}$$

$$\mathrm{TEE} = \frac{1}{N_l} \cdot \sum_{j=1}^{M_l} \left| \frac{Y^{te}_{l,j+M_{l-1}+M_{l-2}} - T^{te}_{l,j+M_{l-1}+M_{l-2}}}{T^{te}_{l,j+M_{l-1}+M_{l-2}}} \right| \tag{28}$$

where Y^{tr} and T^{tr} denote the training output matrix and training target matrix; Y^{te} and T^{te} denote the test output matrix and test target matrix; N_l and M_l denote the number of training samples and test samples of gas l. Note that $N_{-1} = N_0 = 0$, $M_{-1} = M_0 = 0$. Then

Table 3
Relative estimation error of chemical gases concentration using PCR, PLS, NLS and optimized BP-MLP-NNs.

Concentration estimation methods	Relative estimation error of chemical gases (%)							
	Formaldehyde		CO		NO_2		Average	
	TRE	TEE	TRE	TEE	TRE	TEE	TRE	TEE
PCR	73.1578	105.4032	52.1022	66.3210	53.0121	62.2314	59.4240	77.9852
PLS	61.5431	79.5882	42.4425	50.5120	40.4178	51.2125	48.1345	60.4376
NLS	50.2152	58.0233	29.2412	38.5022	28.1586	36.5401	35.8717	44.3552
Single BP	35.7356	28.3328	10.7721	20.0285	12.9313	13.4460	19.813	20.6024
SPSO-BP	25.3224	19.7514	7.5381	9.4957	8.8106	11.8477	13.8904	13.6982
IWAPSO-BP	19.2493	22.8549	6.8687	10.1758	7.0175	7.7730	11.0602	13.6012
APSO-BP	37.8276	25.2891	7.2265	8.7764	6.3244	5.9902	17.1262	13.3519
DRPSO-BP	25.4639	22.1065	6.2033	9.1402	10.1575	10.3230	13.9416	13.8566
ARPSO-BP	20.5874	20.4023	5.8827	11.1103	11.3025	6.6240	12.5909	12.7122
SGA-BP	32.5937	18.8185	6.5593	8.4941	10.0178	6.6698	16.3903	11.3275
PSOBC-BP	31.3325	21.5227	6.1899	8.5397	8.2748	9.5581	15.2657	13.2068
PSOAGS-BP	16.3536	17.5681	5.0970	7.8017	5.1787	5.4962	8.8764	10.2887

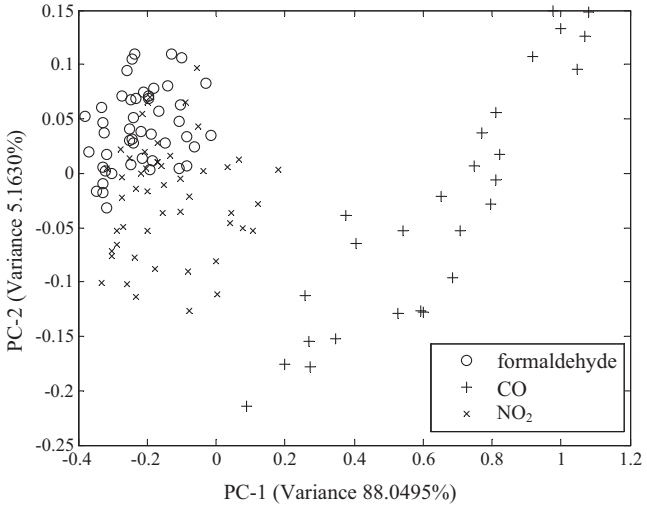

Fig. 5. Principal component score plots of original data set.

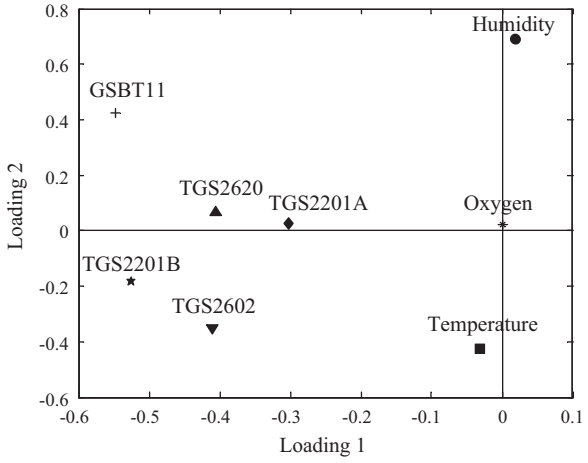

Fig. 6. Results of PCA for the 8-element sensor array showing the loadings of each sensor for principal components 1 and 2.

the correct estimation rates (CER) can be calculated as follows:

$$\mathrm{CER} = 100 - \frac{\mathrm{TRE} + \mathrm{TEE}}{2} \tag{29}$$

It is worthy noting that the zero concentration should be neglected when calculate TRE and TEE using Eqs. (27) and (28) for that infinite great would be obtained as the denominator is zero and the diagonal elements are only considered.

5.4. Results and discussion

5.4.1. Experimental results

The steady-state responses of sensors when exposed to analytes were used as features for concentration estimation. Fig. 4 illustrates the normalized gas sensor responses, and the selected steady-state response points with humidity, temperature and oxygen at the same position are recognized as features for each measurement. The structure of the BP-MLP-NN is consisted of three layers (one input layer, one hidden layer, and one output layer) which is set as 8-16-3 (8 input neurons, 16 hidden neurons and 3 output neurons); hence, the dimension of each particle is 303, calculated from Eq. (18). The eight input neurons denote the number of sensors and the three output neurons correspond to three measured chemical gases. For description of the responses of sensors in the original data set, a principle component analysis (PCA) result is obtained from the covariance of mean-centered data matrix. Fig. 5 shows PCA result of the original data set. The scores of three families are plotted for principal component 1st versus principal component 2nd. A PCA result can give the most information of raw response matrix. From this figure, the CO family can be separated quite well, but the other two families overlap. Besides, it can be seen that CO occupy relatively larger region of the PC space. In Fig. 6, the loadings of first two principal components are shown, indicating the contribution of each sensor to the analysis. We can obviously see that humidity sensor and temperature sensor, particularly the humidity, have a significant contribution to our problem and they are far from the gas sensors. And the distribution of the five gas sensors except the oxygen is dispersive so that they have no collinearity in the response matrix. Apparently, the oxygen sensor plays little role. Thus, we can speculate that the linear approaches would fail to solve the problem of concentration estimation effectively in our project because the responses of harmful gas sensors would be affected by both concentration and other environmental factors (e.g. temperature and humidity). However, for quantification comparisons of the problem, we first attempt to use two simple linear methods including PCR (principle component regression) [35] and PLS (partial least square) [35,36] to estimate concentrations of gases. Besides, a nonlinear least square (NLS) [37] third-order polynomial fitting is also used to estimate gas concentration.

Then, PSO optimization multilayer perceptron neural networks with back-propagation algorithm learning are used to validate the improved PSO model. Considering the instability of BP-MLP-NN resulted from randomly initialized particles, we run each PSO based concentration estimation program for 20 times, and 50 generations for each time. The average best fitness values including train error

Table 4
Correct estimation rates of chemical gases by using PCR, PLS, NLS and optimized BP-MLP-NNs.

Concentration	Correct estimation rates (CER) of chemical gases (%)			
	Estimation methods	CO	NO_2	Average
PCR	10.7195	40.7884	42.3783	31.2954
PLS	29.4343	53.5228	54.1848	45.7140
NLS	45.8870	66.1283	67.6506	59.8886
Single BP	67.9658	84.5997	86.8114	79.7923
SPSO-BP	77.4631	91.4831	89.6709	86.2057
IWAPSO-BP	78.9479	91.4778	92.6047	87.6693
APSO-BP	68.4417	91.9985	93.8427	84.7609
DRPSO-BP	76.2148	92.3282	89.7597	86.1009
ARPSO-BP	79.5052	91.5035	91.0367	87.3485
SGA-BP	74.2939	92.4733	91.6562	86.1411
PSOBC-BP	73.5724	92.6352	91.0836	85.7638
PSOAGS-BP	83.0392	93.5507	94.6625	90.4175

(TRE) and test error (TEE) of the 20 runs are recognized to evaluate each model.

Table 3 demonstrates the best train error and test error of each measured gas calculated by Eqs. (26) and (27). Table 4 presents the correct estimation rates (CER) calculated as Eq. (28).

Through Table 3, we can find that the simple linear methods including PCR and PLS obtain very big TREs and TEEs for each gas. Obviously, their correct estimation rates are also low with only average 31.2954% and 45.714% (see Table 4). When the NLS method by a third-order polynomial fitting is used, the concentration estimation performance has been improved a little and average 59.8886% of correct estimation rate is obtained.

When compared with neural network methods, we can see that the linear methods are far inferior to single BP approach. This also demonstrates that the responses of multidimensional MOS sensor array are strongly nonlinear with gas concentrations, because of their high sensitivities to temperature and humidity. For comparing the PSO based optimization models, we analyze the results for each gas separately. For formaldehyde gas, compared with single BP, the estimation errors including TRE (35.7356%) and TEE (28.3328%) are larger than the BP with the best PSOAGS optimization algorithm (16.3536% of TRE and 17.5681% of TEE). For CO gas, 10.7721% of TRE and 20.0285% TEE with single BP, while 5.0970% and 7.8017% are obtained using the best PSOAGS. For NO_2 gas, 12.9313% train error and 13.4460% test error were obtained by single BP, 5.1787% train error and 5.4962% test error were obtained by the best PSOAGS model. The CER of optimized methods shown in Table 4 are evidently higher than single BP. Further, the performance of E-nose has been improved 10% of the average CER compared the PSOAGS model (90.4175%) and single BP algorithm (79.7923%). Other PSO based optimization models present similar CERs according to Table 4. IWAPSO-BP and ARPSO-BP give higher average CERs (87.6693%, 87.3485%) than SPSO-BP and DRPSO-BP (86.2057%, 86.1009%). While APSO-BP and PSOBC-BP present relatively lower average CERs (84.7609%, 85.7638%). In visual, for viewing the process during searching for the best solution using optimization algorithms, we present Fig. 7 which describes the convergence curves with generations of the average concentration estimation error of the three chemical gases by using each optimization model. At the 8th generations, the best solution has been found out using PSOAGS model while PSOBC and ARPSO obtain the solutions at the 33rd generations. It can be seen from Tables 3 and 4 that the optimization algorithms are indispensable in E-nose application for accurate concentration estimation. Note that the estimation error is taken into consideration as the average best fitness (see Eq. (22)). The proposed PSOAGS model can find the best solution in a fastest converging velocity and reduce the run time of optimization algorithm. The SGA model is inferior to the method in this work. Note also that each curve is obtained from the average results of 20 runs.

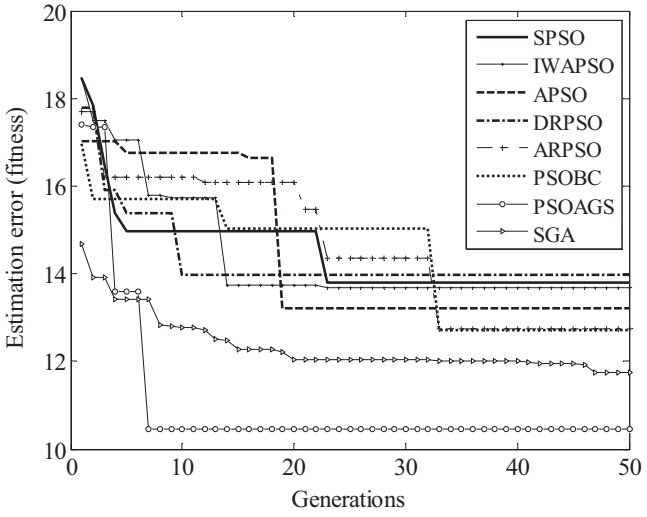

Fig. 7. The convergence curves of the average concentration estimation error of the three pollutant gases using BP-MLP-NN with different optimization models.

5.4.2. Evaluation of complexity

We have developed these algorithms to run on programs written by us in MATLAB 2009a, operating on a laboratory computer equipped with Inter Core (TM) i3 CPU 530, 2.93 GHz processors and 2 GB of RAM. Through experimental results, we can find that the problem of concentration estimation in this paper cannot be solved by simple linear methods or nonlinear least square polynomial fitting due to that the MOS gas sensors would be disturbed by temperature and humidity. This also demonstrates the strong nonlinear relation between responses of sensors and gas concentration. When we use optimized ANN to build the multidimensional nonlinear relation, the complexity of algorithm should be considered. First, we should note that the neural network training is employed on PC but not the E-nose itself. Second, the well trained hyper-parameters (weight matrices and biases) should be saved on the E-nose system for real time concentration estimation. At the first step, the neural network optimization process by heuristic and bio-inspired algorithm would increase the calculation complexity because of the maximum evolving generations and populations (*PopSize*) in optimization. In our optimization, the optimization time will depend on the maximum generations and populations. In this paper, we set both the maximum generations and the number of populations as 50, then the total training times should be 2550 according to the pseudo-codes shown in Fig. 2, and about average 4.899 s were consumed for each training. Thus, the whole neural network optimization process would consume about 3.47 h. Fast convergence velocity for the best solution would reduce the optimization time; while it will take several minutes even less than 1 min to find the estimation parameters of PCR, PLS, NLS. At the second step for real time concentration estimation using the well trained neural network parameters obtained at the first step, it only takes 0.0138 s to calculate the concentration of one sampling and it can meet the property of real time. Similarly, 0.0020, 0.0206 and 0.0093 s would be consumed for PCR, PLS and NLS, respectively. So, for accurate concentration estimation of harmful gases by an E-nose, the CPU time for searching a better estimation model is worthy sacrificing on PC with the fast development of computer.

5.4.3. Discussion

This paper describes many methods that can be used to estimate gas concentration by an E-nose. First, we present the simple

methods (PCR, PLS and NLS) with low complexity during the process of parameters searching. But the results demonstrate the failure of these simple approaches. As we know, a relation of logarithm or exponent function between the responses of MOS gas sensors and the corresponding concentrations exists under a fixed temperature and humidity. So, many published literatures have used simple linear or nonlinear method to solve the problem of concentration estimation. However, it is reasonable to believe the logarithm or exponent relation may be lost when the temperature and humidity changed and the relations should become a family of logarithm or exponent functions with many combinations of temperature and humidity. In this case, artificial neural network would be a better choice for nonlinear regression with high dimension of data set. The temperature and humidity are recognized as two inputs of ANN for compensations.

In PSO, the number of variables in each particle is calculated using Eq. (18), which is directly decided by the architecture of the BP neural network. Thus, to problems with high dimension, such as prediction in multi-sensor system (E-nose), study of nonlinear optimized models becomes necessary. Compared with genetic algorithm, we can find the superior performance of particle swarm optimization in nonlinear system. All the considered optimization algorithms adopt the same real-value encoding strategy for neural network weights and bias optimization. Also, PSO can optimize the sensor array or selection of sample subsets or the structure and related parameters of neural network for improving the performance of an E-nose. In the improvement of PSO, the convergence performance would be paid more attention for actual engineering applications. The TER, TRE and CER analysis of each optimization model in E-nose application demonstrate the PSOAGS model can find the best optimum in a fastest convergence velocity. Therefore, the time complexity can be reduced significantly on PC.

To avoid obtaining local solution, an adaptive genetic algorithm is presented for improving the solutions searching ability of particle swarm optimization. This adaptive characteristic of crossover and mutation probability improves the performance of particles in searching according to the distances among all the particles. It also prevents the particles from being trapped into local optimal. The improved strategy of PSO which is based on the cosine function still contains the adaptive characteristic. However, this adaptive cosine strategy will not make the particles diffused without any limitation because of the boundary [−1, 1] of cosine function.

Through the comparisons with simple linear approaches and polynomial regression, we believe that the heuristic and bio-inspired optimization neural network models are necessary for solution of concentration estimation problem in the future E-nose researches. The apparent advantages of optimized neural network are better robustness, ability of anti-noise and long-term stability compared with those simple approaches. As we know, the sensor drift and the reproducibility should also be considered for long term cases when using simple methods. However, they can be neglected when using the proposed methods in this paper for slight drifts and response variations. Even though complex algorithm would increase complexity, it is only wasted in the hyper-parameters learning of neural network on PC but not in our E-nose system. With the fast development of computer, the complexity of algorithm may be tolerated and accepted. The proposed PSOAGS model would consume a little more computer resource for considering the crossover and mutation operations of genetic algorithm. However, in current electronic nose application, especially the harmful gases detection indoor, high correct prediction rates and real-time property are necessary. From the angle of optimization, almost all optimization models cannot assure the global optimum so that the potential to optimize multi-sensor system for nonlinear problems becomes endless.

6. Conclusions

In this paper, we addressed the issues of gases concentration estimation in electronic nose. Through the analysis of the simple linear methods (PCR, PLS) and a polynomial based nonlinear least square (NLS), we failed to obtain an accurate concentration estimation model using approaches of low complexity for the multidimensional nonlinear problem in E-nose. Then, for development of the estimation problem in E-nose, we have analyzed the performance of several heuristics and bio-inspired optimization algorithms in the problems of concentration estimation. Specifically, we have studied the performance of SPSO, IWA-PSO, APSO, DRPSO, ARPSO, PSOBC and standard genetic algorithm (SGA), optimizing the back-propagation neural network for gases quantification. Moreover, a hybrid evolutionary algorithm PSOAGS is also proposed for improving the intelligence of electronic nose further. A large number of experiments have also been carried out for traversing concentrations in different environments (temperature and humidity) by an electronic nose. We have shown that all the compared approaches provide better estimation accuracy than single BP-MLP-NN, but the PSOAGS outperforms other considered algorithms. In this work, we demonstrated that the heuristics and bio-inspired optimization algorithms are robust and effective approaches in terms of the intelligence of an electronic nose.

Acknowledgments

This work was supported by the Key Science and Technology Research Program (CSTC2010AB2002), Central University Postgraduate' Science and Innovation Funds of China (CDJXS10161114, 200911B1A0100326), Central University Postgraduate' Innovation Team Building Project (200909C1016).

References

[1] J.E. Haugen, K. Kvaal, Electronic nose and artificial neural network, Meat Science 49 (1998) S273–S286.
[2] L. Carmel, N. Sever, D. Lancet, D. Harel, An eNose algorithm for identifying chemicals and determining their concentration, Sensors and Actuators B 93 (2003) 77–83.
[3] M.S. Simon, D. James, Z. Ali, Data analysis for electronic nose systems, Microchimica Acta 156 (2007) 183–207.
[4] J.W. Gardner, P.N. Bartlett, Electronic Noses: Principles and Applications, Oxford University Press, Oxford, 1999.
[5] P.C. Jurs, G.A. Bakken, H.F. McClelland, Computational methods for the analysis of chemical sensor array data from volatile analytes, Chemical Reviews 100 (2000) 2649–2678.
[6] R. Gutierrez-Osuna, Pattern analysis for machine olfaction: a review, IEEE Sensors Journal 2 (3) (2002) 189–202.
[7] G.C. Green, A.D.C. Chan, H.H. Dan, M. Lin, Using a metal oxide sensor (MOS)-based electronic nose for discrimination of bacteria based on individual colonies in suspension, Sensors and Actuators B 152 (2011) 21–28.
[8] C.H Shih, Y.J. Lin, K.F. Lee, P.Y. Chien, P. Drake, Real-time electronic nose based pathogen detection for respiratory intensive care patients, Sensors and Actuators B 148 (2010) 153–157.
[9] C. Wongchoosuk, A. Wisitsoraat, A. Tuantranont, T. Kerdcharoen, Portable electronic nose based on carbon nanotube-SnO_2 gas sensors and its application for detection of methanol contamination in whiskeys, Sensors and Actuators B 147 (2010) 392–399.
[10] B.A. Botre, D.C. Gharpure, A.D. Shaligram, Embedded electronic nose and supporting software tool for its parameter optimization, Sensors and Actuators B 146 (2010) 453–459.
[11] P. Wang, T. Yi, A novel method for diabetes diagnosis based on electronic nose, Biosensors and Bioelectronics 12 (9–10) (1997) 1031–1036.
[12] P. Wang, J. Xie, A novel recognition method for electronic nose using artificial neural network and fuzzy recognition, Sensors and Actuators B: Chemical 37 (3) (1996) 169–174.
[13] J.R. Zhang, J. Zhang, A hybrid particle swarm optimization back-propagation algorithm for feed-forward neural network training, Applied Mathematics and Computation 185 (2007) 1026–1037.
[14] M. Gori, A. Tesi, On the problem of local minima in back-propagation, IEEE Transactions on Pattern Analysis and Machine Intelligence (1992) 76–86.
[15] V. Maniezzo, Genetic evolution of the topology and weight distribution of neural networks, IEEE Transactions on Neural Networks 5 (1) (2002) 39–53.

[16] J.N.D. Gupta, R.S. Sexton, Comparing back-propagation with a genetic algorithm for neural network training, Omega 27 (6) (1999) 679–684.
[17] C.F. Juang, Y.C. Liou, On the hybrid of genetic algorithm and particle swarm optimization for evolving recurrent neural network, in: Proc. IEEE. International Joint Conference on Neural Networks, 2005.
[18] R.S. Sexton, J.N.D. Gupta, Comparative evaluation of genetic algorithm and back-propagation for training neural networks, Information Sciences 129 (1–4) (2000) 45–59.
[19] M.J. Polo-Corpa, S. Salcedo-Sanz, A.M. Perez-Bellido, P. Lopez-Espi, R. Benavente, E. Perez, Curve fitting using heuristics and bio-inspired optimization algorithms for experimental data processing in chemistry, Chemometrics and Intelligent Laboratory Systems 96 (2009) 34–42.
[20] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[21] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, IEEE. Evolutionary Computation 1 (2002) 84–88.
[22] J. Riget, J.S. Vesterstrom, A diversity-guided particle swarm optimizer – the ARPSO Technical Report 2-2002, Department of Computer Science, 2002.
[23] W. Jiang, Y. Zhang, A particle swarm optimization algorithm based on diffusion–repulsion and application to portfolio selection, IEEE, International Symposium on Information Science and Engineering (2008).
[24] B. Niu, Y. Zhu, An improved particle swarm optimization based on bacterial Chemotaxis, IEEE, Intelligent Control and Automation 1 (2006) 3193–3197.
[25] C.H. Hsu, W.J. Shyr, K.H. Kuo, Optimizing Multiple Interference Cancellations of Linear Phase Array Based on Particle Swarm Optimization, Journal of Information Hiding and Multimedia Signal Processing 1 (2010) 292–300.
[26] J.F. Chang, S.C. Chu, J.F. Roddick, J.S. Pan, A parallel particle swarm optimization algorithm with communication strategies, Journal of Information Science and Engineering 21 (2005) 809–818.
[27] M.F. Horng, Y.T. Chen, S.C. Chu, J.S. Pan, B.Y. Liao, An extensible particle swarm optimization for energy effective cluster management of underwater sensor networks, ICCCI2010, LNAI 6421 (2010) 109–116.
[28] R.J. Kuo, L.M. Lin, Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering, Decision Support Systems 49 (2010) 451–462.
[29] J. Horn, N. Nafpliotis, An enriched Pareto genetic algorithm for multi-objective optimization, IEEE (2002).
[30] K.A. DeJong, An Analysis of the Behavior of a Class of Genetic Adaptive System, 1975.
[31] D.E. Goldberg, Genetic algorithm in search, Optimization and Machine Learning (1989).
[32] D. Ballabio, M. Vasighi, V. Consonni, M.K. Zareh, Genetic algorithms for architecture optimisation of counter-propagation artificial neural networks, Chemometrics and Intelligent Laboratory Systems 105 (2011) 56–64.
[33] W. Wu, J. Wang, M.S. Cheng, Z.X. Li, Convergence analysis of online gradient method for BP neural networks, Neural Networks 24 (2011) 91–98.
[34] O. Kisi, Multi-layer perceptrons with Levenberg–Marquardt training algorithm for suspended sediment concentration prediction and estimation, Hydrological Sciences Journal 49 (6) (2004) 1025–1040.
[35] J. Getino, M.C. Horrillo, J. Gutiérrez, L. Arés, J.I. Robla, C. García, I. Sayago, Analysis of VOCs with a tin oxide sensor array, Sensors and Actuators B 43 (1997) 200–205.
[36] S. de Jong, SIMPLS: an alternative approach to partial least squares regression, Chemometrics and Intelligent Laboratory Systems 18 (1993) 251–263.
[37] H. Zhou, M.L. Homer, A.V. Shevade, M.A. Ryan, Nonlinear least-squares based on method for identifying and quantifying single and mixed contaminants in air with an electronic nose, Sensors 6 (2006) 1–18.

Biographies

Lei Zhang received his Bachelor degree in electrical/electronics engineering in 2009 from the Nanyang Institute of Technology, China; from September 2009 to December 2010, he studied for a master degree in signal and information processing. He is presently with Chongqing University, pursuing his Ph.D. degree in Circuits and Systems. His research interests include computational intelligence, artificial olfactory system, and nonlinear signal processing in electronic nose.

Fengchun Tian received Ph.D. degree in 1997 in electrical engineering from Chongqing University. He is currently a professor with the College of Communication Engineering of Chongqing University. His research interests include electronic nose technology, artificial olfactory systems, pattern recognition, chemical sensors, signal/image processing, wavelet, and computational intelligence. In 2006 and 2007, he was recognized as a part-time professor of GUELPH University, Canada.

Chaibou Kadri received his Bachelor degree in electrical/electronics engineering in 2001 from the Federal University of Technology Bauchi, Nigeria; his master's degree in communication and information system in 2009, from Chongqing University China. He is presently with Chongqing University, pursuing his Ph.D. degree in Circuits and Systems. His research interests include signal processing for gas sensors array instruments, and machine learning.

Guangshu Pei received her Bachelor degree in electronic and information engineering institute, Shaanxi University of Technology; from September 2009 to December 2011, she studied for a M.S. degree in signal and information processing in College of Communication Engineering, Chongqing University, China; her research interests include cognitive radio technology and signal processing.

Hongjuan Li received her Bachelor degree in college of communication engineering in 2009 from the Chongqing University, China; from September 2009 to December 2011, she studied for a Master degree in circuits and system. Her research interests include circuits and system design in electronic nose technology.

Lina Pan received her Bachelor degree in college of communication engineering in 2009 from the Chongqing University, China; from September 2009 to December 2011, she studied for a master degree in signal and information processing. Her research interests include artificial olfactory system and electronic nose technology.