TELKOMNIKA, Vol. 11, No. 6, June 2013, pp. 3304 ~ 3313
e-ISSN: 2087-278X

Sensor Temperature Compensation Technique Simulation based on BP Neural Network

Xiangwu Wei
Liaoning Provincial College of Communication, Shenyang, China
NO.102, Sheng North Road ,Shenyang. 110122
Corresponding author, e-mail: wxwwjq1963@163.com

Abstract

Innovatively, neural network function programming in the BPNN (BP neural network) tool boxes from MATLAB are applied, and data processing is done about CYJ-101 pressure sensor, and the problem of the sensor temperature compensation is solved. The paper has made the pressure sensors major sensors and temperature sensor assistant sensors, input the voltage signal from the two sensors into the established BP neural network model, and done the simulation under the NN Toolbox environment of MATLAB. From the compensation result, it has be found that the temperature interference variable effects on the pressure output identity has dropped from 22% to 1.1%, greatly improved the pressure sensor measurement precision and anti-interference ability.

Keywords: *sensor, identity compensation, temperature compensation, BP neural network*

1. Introduction

With the development of electronic technique and improve of automobile electronic, conventional mechanism hasn't already met the need of improving automobile function [1]. So, conventional mechanism of automobile has been replaced by electronic control system. A sensor is equipment which can exchange non-electronic signal, such as light, electricity, time, temperature, pressure and gas, to electronic signal and can offer it for other parts of the machine. A sensor is important component in the electronic control system of automobile, and its characteristic, including static one and dynamic one, works on the sensitivity of sensor and works on the automobile performance at last.

The engine control unit of automobile, which is typical control system, includes many sensors, such as pressure sensor, temperature sensor, liquid sensor, gas strength sensor, position sensor and knocking sensor, etc [2]. The all kinds of sensors mentioned above are the core of engine control system, and using them in engine control system can improve the dynamical behaviors of an engine, decrease producing exhaust gas, reduce consuming fuel and reflect which part of the automobile doesn't work, etc. Because they work in very bad environment where these are mud and dirty water ,vibrating caused by the engine and fuel vapor, etc, these sensors have to apply more advanced technique to overcome their bad environment than general sensors [3]. In its many performance targets, the reliability and measure precision of a sensor are the most important ones, for large measure error may cause engine control system to fail. In modern automobile, all sorts of sensors achieve all kinds of useful information and transfer them to the Micro-controller, which deals with the information and, according to the result of it, gives instructions to performing parts of the car. For example, engine ignition moment and spraying fuel moment are given by these ways. In fact, each sensor's failure will work on performance of the automobile, and, to a worse degree, will cause the engine not to run. Thus it can be seen, the study on sensor temperature compensation technique is important in both automobile and other fields [4].

2. BP Neural Network Theory and its Practical Application

BP network is currently the most widely used neural network model. It is a one-way transmission of multi-layer feed forward network. It can be seen as a highly nonlinear mapping

Received January 10, 2013; Revised April 13, 2013; Accepted April 24, 2013

from input to output. It is divided into the input layer, the hidden layer and the output layer, more than full interconnection between layers, there is no mutual connection between units of the same layer. As shown in Figure 1 is a typical BP network structure.

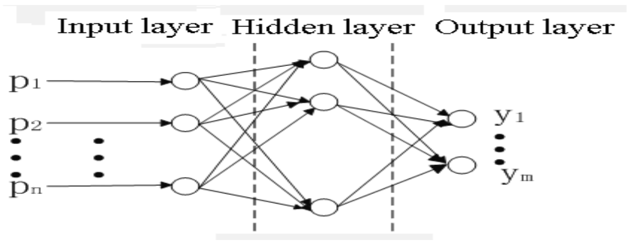

Figure 1. The Layer Network Structure

BP algorithm was developed by forward propagation of the information and the error reverse propagation, input mode processing and transmission layer by layer from the input layer through the hidden layer to the output layer, if the output layer does not get the desired result, then the error signal go back along the original path and modify the weights of each layer until the error is minimized, and ultimately achieve the desired target value [5]. In short, BP algorithm using gradient steepest descent method, weights along the negative gradient direction of the error function change, the error decreases gradually and approximate nonlinear functions.

2.1. BP Neural Network Transfer Function and Work Process

Commonly refer to the BP model, the error back propagation neural network is the most widely used neural network model. BP network basic processing unit (except for the input layer unit) is non-linear input-output relationship (transfer function is non-linear), generally used in the role of the S-type function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

BP neural network achieve the idea of multi-layer network learning. When an input pattern of a given network, it is transmitted by the input layer unit to the hidden layer unit, processed by hidden layer unit, and then sent to the output layer unit, after processing by the output layer unit produces an output mode, so called forward propagation. If the output response and the desired output mode error, and does not meet the requirements, then on into error backward propagation, error values along the connection path layer-by-layer backward transfer and fix connection weights of layers.

2.2. BP Neural Network Learning Algorithm

BP network learning simply is the promotion and development of δ-learning rule. Assuming BP network N processing units on each floor, transfer function, such as (2-1), the training set consists of M samples(x_k, y_k).The sum of p-th training sample (p=1,2, ..., M) unit j (activation function) as a_{Pj},the output as O_{Pj},the i-th input (the i-th neuron output) is O_{Pj}, then:

The sum of input j:

$$a_{Pj} = \sum_{i=0}^{N} W_{ji} O_{Pi} \tag{2}$$

The output of j:

$$O_{Pj} = f(a_{Pj}) = \frac{1}{1+e^{-a_{Pj}}} \tag{3}$$

If arbitrarily set the network initial value, for each input pattern, an error between the network output and the desired output is generally present. Define network error:

$$E = \sum_P E_P \tag{4}$$

$$E = \frac{1}{2}\sum_j (d_{Pj} - O_{Pj})^2 \tag{5}$$

Wherein, d_{Pj} represents the desired output to the output unit of the p -th input mode j . δ learning rule essentially is gradient steepest descent method, the weights along the negative gradient of the error function change. If the change amount of weight W_{ji} denoted ΔW_{ji} , then:

$$\Delta W_{ji} \propto -\frac{\partial E_P}{\partial W_{ji}} \tag{6}$$

And:

$$\frac{\partial E_P}{\partial W_{ji}} = \frac{\partial E_P}{\partial a_{Pj}} \frac{\partial a_{Pj}}{\partial W_{ji}} = \frac{\partial E_P}{\partial a_{Pj}} O_{Pj} = -\delta_{Pj} O_{Pj} \tag{7}$$

So that:

$$\delta_{Pj} = -\frac{\partial E_P}{\partial a_{Pj}} \tag{8}$$

So:

$$\Delta W_{ji} = \eta \delta_{Pj} O_{Pj}, \eta > 0 \tag{9}$$

This is commonly the learning rules. BP neural network in the learning process, the calculation for the error of the output layer and the hidden layer unit is different, and are discussed below.

When O_{Pj} represents the output of the output layer unit, the error is:

$$\delta_{Pj} = -\frac{\partial E_P}{\partial a_{Pj}} = -\frac{\partial E_P}{\partial O_{Pj}} \frac{\partial O_{Pj}}{\partial a_{Pj}} \tag{10}$$

From (3):

$$\frac{\partial O_{Pj}}{\partial a_{Pj}} = f'(a_{Pj}) \tag{11}$$

From (7):

$$\frac{\partial E_P}{\partial O_{Pj}} = -(d_{Pj} - O_{Pj}) \tag{12}$$

(12), (11) into (2-10):

$$\delta_{Pj} = f'(a_{Pj})(d_{Pj} - O_{Pj}) \tag{13}$$

Wherein, $(d_{Pj} - O_{Pj})$ reflecting the error amount of the output unit j ; the derivative terms of the action function $f'(a_{Pj})$ is the amount of the error decreases at ratio.

O_{Pj} represents the output of the hidden units, the error:

$$\delta_{Pj} = -\frac{\partial E_P}{\partial a_{Pj}}\frac{\partial O_{Pj}}{\partial a_{Pj}} = -\frac{\partial E_P}{\partial O_{Pj}} f'(a_{Pj}) \tag{14}$$

$$\frac{\partial E_P}{\partial a_{Pj}} = \sum_k \frac{\partial E_P}{\partial a_{Pk}}\frac{\partial a_{Pk}}{\partial O_{Pj}} \tag{15}$$

From (2):

$$\frac{\partial a_{Pk}}{\partial O_{Pj}} = W_{kj} \tag{16}$$

From (10):

$$\frac{\partial E_p}{\partial a_{pk}} = -\delta_{pk} \tag{17}$$

(16), (17) into (15):

$$\frac{\partial E_p}{\partial P_{Pj}} = -\sum \delta_{Pk} W_{kj} \tag{18}$$

So:

$$\delta_{Pj} = f'(a_{Pj})\sum_k \delta_{Pk} W_{kj} \tag{19}$$

So far, BP algorithm weights correction formula can be unified:

$$W_{ji}(t+1) = W_{ji}(t) + \eta\delta_{Pj} O_{Pj} \tag{20a}$$

$$\delta_{Pj} = \begin{matrix} f'(a_{Pj})(d_{Pj} - O_{Pj})OUTPUT \\ f'(a_{Pj})\sum_k \delta_{Pk} W_{kj} HIDDEN \end{matrix} \tag{20b}$$

In practical application, taking the convergence of the learning process into account, usually in order to make value of the learning factor Q is large enough, which without generating oscillation, then at the value correction formula (2-20), plus a momentum, so:

$$W_{ji}(t+1) = W_{ji}(t) + \eta\delta_{Pj} O_{Pj} + \alpha[W_{ji}(t) - W_{ji}(t-1)] \tag{21}$$

Wherein, a is a constant, called the momentum factor, and it determines the degree of influence on the last time learning weight update the learning weight this time.

In general, the BP network learning algorithm steps are described as follows:

1. Initialize the network and learning parameters, such as setting up a network initial matrix, the learning factor η, parameters α ;
2. Providing training samples, training network until meet the requirements;
3. Forward propagation process: given training mode input, calculate network output mode and compare the output with the expected pattern, if error, 4 is performed, otherwise, return 2;
4. Backward propagation process:

a. Calculate the error within the same layer.
b. Correction weights and threshold.

$$W_{ji}(t+1) = W_{ji}(t) + \eta\delta_{Pj}O_{Pj} + \alpha[W_{ji}(t) - W_{ji}(t-1)] \tag{22}$$

The threshold is the connection weight when i = 0.
c. Return □

BP network learning is achieved by using the given training. When network study is over, usually, with a root-mean-square (RMS) error quantitatively reflect the learning performance. This is defined as:

$$E_{RMS} = \sqrt{\frac{\sum_{p=1}^{m}\sum_{j=1}^{n}(d_{Pj} - y_{Pj})^2}{mn}} \tag{23}$$

m - The number of training mode set;
n - The number of network output layer unit.

Usually, when squared error of the network $E_{RMS} \langle 0.1$, it indicates that the requirements have been met for a given training set learning.

2.3. BP Neural Network Compensation Principle of Sensor Characteristics

The schematic of the BP neural network method to improve the characteristics of the sensor output by the two parts of the sensor model and the neural network model.(as shown in Figure 2)

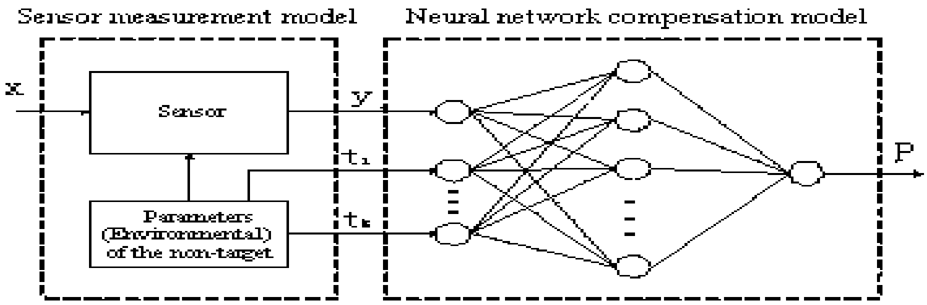

Figure 2. The Sensor Identity Compensation Principle Graph

Sensor model in the figure:

$$y = f(x, t_1, t_2 \ldots t_k) \tag{24}$$

x is the target parameter to be measured in the formula; $t_1, t_2 \cdots t_k$ are the k target parameters; y is the sensor output; p is the output after the BP network compensation.

if y and $t_1, t_2 \cdots t_k$ are a single-valued function of x, then the Equation (25)'s inverse function exists, is:

$$x = f^{-1}(y, t_1, t_2 \ldots t_k) \tag{25}$$

(24) expressed as a function of model is quite complex, it is difficult to describe with a specific function relationship, but you can use the BP neural network approximation (25) described the complex nonlinear relationship. The output of the target parameter measures sensor. And a variety of non-target parameters sensitive element as the BP network input, the output p after the BP network processes is x the target parameters to be measured, and eliminating the effects of a variety of non-target parameters [6].

2.4. BP neural network compensation principle characteristics of the pressure sensor

Shown in Figure 3 is the compensation schematic diagram of the neural network to a plurality of non-target (environmental) parameters, in order to study conveniently and easily describe the application of the actual example [7], can simplify the problem, then only study a non-target (temperature) parameter compensation issue. CYJ-101 pressure sensor, for example, can detect two parameters at the same time: the target parameter (pressure); the non-target parameters (temperature). CYJ-101 pressure sensor can detect simultaneously two parameters, so called for the two sensors. BP neural network's principle of compensation for the pressure sensor is shown in Figure 3.

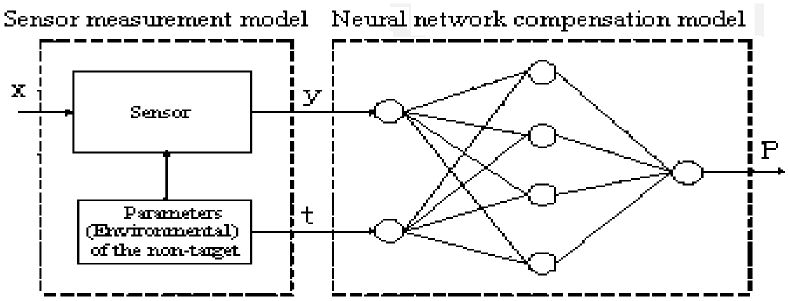

Figure 3. The Sensor Temperature Compensation Principle Graph

The pressure sensor model in the figure:

$$y = f(x,t) \tag{26}$$

x is the target parameter (pressure) to be measured in the formula; t is non-target temperature parameter; y is the output of sensor; P is the output(pressure) after the BP network compensation.

if y and t are a single-valued function of x, then the (4)'s inverse function exists, is:

$$x = f^{-1}(y,t) \tag{27}$$

Typically, the function model expressed by (27), although is not very complex, but solving the corresponding coefficient is complex, nonlinear relationship described by (27) can also use the BP neural network to approximate [8]. The output(pressure y) of the target parameter measuring sensor, and the output(temperature t) of non-target parameter temperature sensitive element as the BP network input, the output P after the BP network processes is x the target parameter to be measured, and eliminating the effects of a variety of non-target parameters.

3. Constructing a Sensor Temperature Compensation Model based on BPNN

In practice, sensor output characteristic always varies caused by many jamming factors, such as temperature, magnetic field and noise. In order to optimize sensor characteristic, the paper constructs a neural network in which one inputs quantity of the network is the sample data of output of sensor and the other input quantity is compensated quantity, and output of the network is the quantity which is going to be measured. Utilizing the obtained sample data to train the neural network can achieve the number of neurons and all parameters, which determine a neural network model. The model can be used to carry out sensor temperature compensation.

In order to construct a neural network, the number of hidden layers and the number of neurons in each hidden layer must be obtained [9]. The first number is obtained by the number of network input, and the second one is by result of simulation. This paper constructs a BP neutral network in which there is an output quantity and a nerve cell in the output layer. The sensor characteristic compensation model based on BP neutral network is showed as follow Figure 4 (Notice: each input, which number is R, is connected to each nerve cell.)

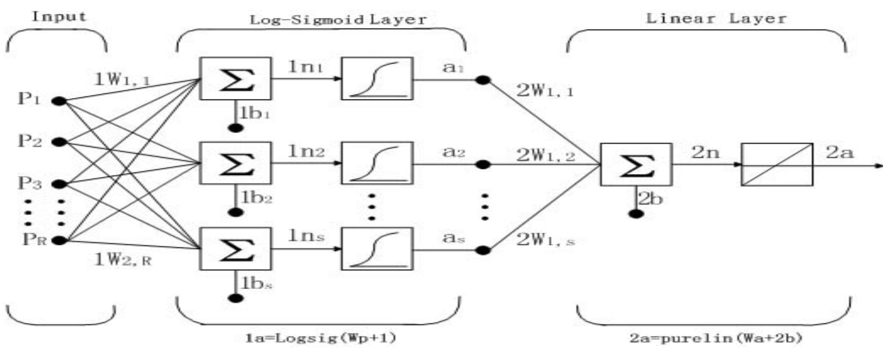

Figure 4. The Sensor Identity Compensation Model on the base of BP Neural Network

From Figure 4, we can know that the output of neared cell in hidden layer in this sensor temperature compensation model based on Bp neural network is as follow:

$$\vec{a} = f(\vec{W}\vec{P} + \vec{b}) \tag{28}$$

In this equation, input vector is $\vec{P} = [p_1, p_2, p_3 \cdots p_R]^T$; output vector is $\vec{a} = [a_1, a_2, a_3 \cdots a_S]^T$; offset value vector is $\vec{b} = [b_1, b_2, b_3 \cdots b_S]^T$;

The weight value vector is:

$$\vec{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \vdots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

; f is transfer function which may be different in the same layer or in various layers according to the peculiarity of a network.

The transfer functions f_1^1, f_2^1 of two neural cells in hidden layer are logarithm-s function (log-sigmoid function), and the transmission function of output layer is linearity function, so the output equation of neural network is showed as follow.

$$a^2 = f^2([W_{1,1}^2, W_{1,2}^2][\begin{matrix} f_1^1(W_{1,1}^1 p + b_1^1) \\ f_2^1(W_{2,1}^1 p + b_2^1) \end{matrix}] + b^2) = f^2(W^2 p^2 + b^2) \tag{29}$$

4. Simulation of Sensor Temperature Compensation based on BPNN

4.1. Two-dimension demarcating experiment

According to need in practice, the input and output value of a pressure sensor which model is CYJ-101 are demarcated at various surrounding temperature (such as T=22□, 44.0□, 70.0□). By this way, we can obtain demarcated value showed in the table 1, which comes from correlative referable information [5, 6].

4.2. Normalizing Sample Data for Training the BP Neural Network

Because the transfer function of the hidden layers is logarithm S model and hyperbolic secant S model, output value of the two function will respectively be limited to (0,1) and (-1,1). During training the network, in order to avoid entering saturation state quickly, which will interrupt the learning process of the nodes; we must, first of all, normalize the input value and the output value of the neural network, and furthermore normalize also the sample data before used to train the network [10].

For the output value of transfer function in the neural network is limited to (0, 1) and (-1, 1), after normalizing, the sample data is limited to (-1, 1). So according to the following formula:

$$\overline{X}_{im} = \frac{X_{im} - X_{i\min}}{X_{i\max} - X_{i\min}} \tag{30}$$

$$\overline{Y}_m = \frac{0.9(Y_m - Y_{\min})}{Y_{\max} - Y_{\min}} + 0.05 \tag{31}$$

In this formula:

$\overline{X}_{im}$, $\overline{Y}_m$ —— normalized input value and output value of the m-th sample data;

X_{im} , Y_m ——demarcated input value and output value of the m-th sample data and the i-th sensor;

$X_{i\max}$, $X_{i\min}$ ——maximal output demarcated value and minimal output demarcated value of the i-th sensor (it is the input of the neural network);

$Y_{\max}$, $Y_{\min}$ ——maximal input pressure p value and minimal input pressure p value of the sensor(it is the expected output value of the neural network).

Now we is going to normalize these demarcated experiment data, and we end up to build up the criterion sample data store by employing the demarcated experiment data (Table 2).

Table 1. The Sensor Input and Output Mark Data under the Different Work Temperature

Demarcating Value	T=22℃		T=44℃		T=70℃	
P/10⁴Pa	U/mV	Ut/mV	U/mV	Ut/mV	U/mV	Ut/mV
0.00	0.00	290.5	0.00	542.5	0.00	826.1
1.00	25.27	268.8	16.12	517.8	11.10	800.2
2.00	44.00	247.2	33.25	493.0	28.10	769.1
3.00	62.72	224.5	50.42	465.3	44.93	740.0
4.00	81.40	206.0	67.62	435.6	61.38	706.2
5.00	100.2	184.4	84.73	410.8	78.57	669.3

Table 2. The Neural Network Input and Output Standard Sample Pool

serial number of sample	Normalized Output value	T=22℃ Normalized Input value		serial number of sample	Normalized Output value	T=44℃ Normalized Input value		serial number of sample	Normalized Output value	T=70℃ Normalized Input value	
m	$\overline{Y}_m$	$\overline{U}$	$\overline{U}_t$	m	$\overline{Y}_m$	$\overline{U}$	$\overline{U}_t$	m	$\overline{Y}_m$	$\overline{U}$	$\overline{U}_t$
1	0.05	0.000	1.000	7	0.05	0.000	1.000	13	0.05	0.000	1.000
2	0.23	0.252	0.795	8	0.23	0.190	0.815	14	0.23	0.141	0.835
3	0.41	0.439	0.592	9	0.41	0.392	0.626	15	0.41	0.358	0.637
4	0.59	0.626	0.378	10	0.59	0.595	0.415	16	0.59	0.572	0.451
5	0.77	0.813	0.204	11	0.77	0.798	0.189	17	0.77	0.781	0.235
6	0.95	1.000	0.000	12	0.95	1.000	0.000	18	0.95	1.000	0.000

4.3. Analyzing the Result of Simulation

With synthesizing the various factors into the neural network model, two nerve cells, which stand for the input nodes, are included to the input layer. And an input node is connected to the output voltage port of a sensor and the other one is fed with the non-wanted voltage value obtained by measuring the temperature in the environment in which the sensor works. There is a neural cell, which stands for the output node, in the output layer, and the output of the neural cell is the normalized pressure value which has been compensated by the BP neural network.

For some papers, the pressure output value of the output node is regarded as one that BP neural network gives after forcing the wanted pressure value and the temperature value to amalgamate [11]. The correct number of node is obtained by programming with MATLAB, or calculating with MATLAB Tool-Box.

The data is used to training the neural network with the BP method, and x is the input data and it is the wanted output value, which is expected output value, during the process of learning. From this input x, we can know that these are two units in it, and for example 0.00 and 1.00 is couple of normalized sample data. And one input sample data is the wanted quantity; output voltage value U, of the sensor, and the other input sample data is non-wanted quantity, temperature measure voltage value U_t, which is interfering factor. The output is a unit, which is expected to give pressure value [12].

In the command window of MATLAB, by running simulation program, a value 'a', which is the simulation result of 'x1', can be obtained, and it is showed as follows:

a = 0.0501 0.2513 0.4334 0.6219
0.7794 0.9503 0.0501 0.2387
0.4110 0.5970 0.7895 0.9503
0.0501 0.2261 0.4038 0.5729
0.7586 0.9503

The max value of pressure and the min value of pressure are listed as:

$P_{\max} = 5 \times 10^4 Pa$, $P_{\min} = 0$

From the following formula, $\overline{Y}_m = \dfrac{0.9(Y_m - Y_{\min})}{Y_{\max} - Y_{\min}} + 0.05 \Rightarrow$

We can get these following equations: $Y_m = \dfrac{(\overline{Y}_m - 0.05)\times(Y_{\max} - Y_{\min})}{0.9}$

$$P_m = \frac{(\overline{P}_m - 0.05)\times(P_{\max} - P_{\min})}{0.9} = \frac{(\overline{P}_m - 0.05)\times(5\times10^4 - 0)}{0.9} = 55556(\overline{P}_m - 0.05)$$

After applying neural network to compensation the output of sensor, the pressure value of the sensor and the temperature which is correlated to the pressure value, are recorded as the Table 3:

Table 3. The Merge Compensating Result through BP Neural Network

Jamming Quantity (non-wanted quantity)	Measured Pressure P/10^4(wanted quantity)					
T/□	demarcated value	Merged value	demarcated value	Merged value	demarcated value	Merged value
22		1.0883		3.0661		5.0017
44		1.0483		3.0389		5.0017
70	1.0	0.9783	3.0	2.9050	5.0	5.0017
max $\lvert\Delta P\rvert$ the maximal offset		0.110		0.024		0.000

By analyzing the data in the Table 3, we can end up to get the compensated pressure output value of the sensor [13]. Furthermore, the Equation (31) can give the fluctuating pressure value on condition that the pressure value of the sensor has been compensated by the network.

$$\alpha_p = \frac{\max\lvert\Delta P\rvert}{P_{FS}} \tag{32}$$

In the above equation:

P_{FS} =5.0 $\times 10^4$ is the fixed pressure value at full measuring range of sensor (In another word, it is the maximal pressure output value of sensor at T=22℃)

$\max |\Delta P|$ is the maximal absolute value of fluctuating pressure (from the Table 3, we can get this $\max |\Delta P|$ =0.11 $\times 10^4$).

And then:

$$\alpha_P = \frac{\max |\Delta P|}{P_{FS}} = \frac{0.11 \times 10^4}{5 \times 10^4} = 0.022 = 2.2\%$$

According to the data in the Table 1, as P= $5.0 \times 10^4 Pa$, temperature T will rise from 22℃ to 70℃.

$\max |\Delta y|$=100.12-78.57=21.55mv, using this value in Equation (32), we will get $\alpha_p = \frac{21.55}{97.12} = 0.22 = 22\%$

From the above experiment, analysis and result, the output value of a sensor has to be compensated because it will be affected by non-wanted factors. By the above calculating result, we can see that, at the same variety in temperature, the stability of output of a sensor in which these is a BP neural network is better than that without network in it.

By the above calculating result, we can see that, at the same temperature variety, the stability of output of a sensor in which these is a BP neural network is as 11 times as that of a senor in which these isn't a BP neural network.

5. Conclusion

From analyzing practical example, we can know that by applying BP neural network technique, an ideal compensating effect to banish the error caused by the surrounding temperature can be gotten depending on rather less data. And this method can enhance remarkably the stability of CYJ-101 model pressure sensor. Furthermore, calculation process is much simpler than that in fitting curve which is used before. And simultaneity applying this method has many advantages such as being easy to build up model and high calculation efficiency, etc. With the further development of applying BP neural network technique to improve the output characteristic of a sensor, it can be believed that applying widely BP neural network technique will raise further the measure precision of a sensor and the working stabilization of a sensor, and bring much more intellectuality to a sensor.

References

[1] Hanaa T El-Madany, Faten H Fahmy, Ninet MA El-Rahman, Hassen T Dorrah. Design of FPGA Based Neural Network Controller for Earth Station Power System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 281-289.
[2] Xiaohui Chen. A Rough Neural Network Algorithm for multisensory Information Fusion. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1235-1241.
[3] Budi Rahmani. Early Model of Traffic Sign Reminder Based on Neural Network. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(4) 713-722.
[4] Hui Dong. *Sensors in Automobile*. Beijing: Beijing Institute of Technology Press. 2000.
[5] Junhua Liu. *Intelligence Sensors System*. Xi'an:Xi'an the University of Science and Technology Press. 1999.
[6] Xiaodong Wang. Synthesis Modifying Static Error Method of Sensor Based on Neural Network. *Process Automation Instrumentation*. 1997; 18(3): 310-313.
[7] Huihai Wang. *Nonlinear compensating Technology and Applying in Sensor Based on Neural Network.* Central South University. 2004.
[8] Bangwen Yang. *The Latest Practicality Manual of Sensor*. Beijing: Beijing Posts and Telecommunications Press. 2004.
[9] Kejie Li. *New Editing Technology Manual of Sensor*. National Defence Industry Press.2005.
[10] Hongrun Zhang, Yafan Zhang. *Tutorial on Senor Technology and Applying*. TSinghua University Press. 2005:312-313.
[11] Guoyu Li. *Designing Intelligence Pressure Sensors*. Hebei Technology University. 2004.
[12] Jun Peng. *Sensor and Measure Technology*. Xi'an the University of Science and Technology Press. 2003: 3-4.
[13] Chung WYJW, et al. Sensors and Actuators B. 2000; 111(64): 118-123.