The Application Research of Neural Network in Embedded Intelligent Detection

Xiaodong Liu[1], Dongzhou Ning[1], Hubin Deng[2], Jinhua Wang[1]

[1] Compute Center of Nanchang University, 330039,Nanchang, Jiangxi, china
{Xiaodong Liu, Dongzhou Ning, Jinhua Wang, ncliudong@sina.com.cn}
[2]East China Jiaotong University, 330039,Nanchang, Jiangxi, china
{ Hubin Deng, hubind66@yahoo.com.cn }

Abstract. Neural network which can adapt the sample data by training has good fault-tolerance and can be used in the field of intelligence widely. In the embedded system, restricted to the resources and the capacity of processor, the neural network application has a series of problems, such as losing timelines and the system could be collapsed easily. This article discusses how to use limited memory, processor and external equipment resources to achieve the neural network algorithm for improving the universality of detection system and adaptive ability in the embedded intelligent measuring system.

Key words: Neural Network, Embedded System, Intelligent Detection

1 Introduction

Most embedded detective devices are developed by using the fixed parameter method, that is, the mapping relationship between test data and test conclusion is fixed. However, in practical application, the input conditions of detection and the testing standards may need to be changed, which may make the existing detection system inadaptable. The neural network owns the ability of self-adaptation and self-learning, therefore, by studying on the sample data, it can determine the mapping relationship between input and output. And if the neural network is applied to the detection system, it may adjust the mapping relationship automatically by training samples, which can make the detection system more flexible and adaptable. Application of the algorithm is shown in the figure 1 [1].

Because of the constraints of the resources of embedded portable device and the capacity of processor, computational complexity should be minimized in the application of neural network algorithm, thus to improve the efficiency by centralizing resources to deal with necessary tasks.

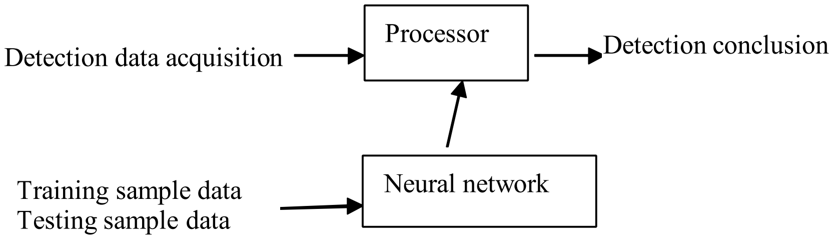

Fig. 1. application of neural network

2 Neural Network Platform Design

Great success has already been achieved in the research and application of neural network, however, there is no perfect theory as a guide in the aspect of network structure development and design, so network parameters are adjusted just only on the basis of former researchers' designing experience and experiment analyses. The functions of neural network platform development are network parameter settings, network training, network training analysis, network testing, network prediction etc. In the network structure designing and the training algorithm, because of the resources constraints in embedded system, we should use as simple and effective methods as possible. The following is the description of issues needed to be considered in the design:

2.1 The choice of network model

2-layer linear perceptron model and 3-layer BP network model are provided and the user can choose one of them in the network parameter settings in accordance with the complexity of the detection system. According to the Almighty Approaching Theorem, a 3-layer BP neural network with a hidden layer can approximate to any continuous function of bounded domain with arbitrary precision as long as there are enough hidden layer nodes[2]. Therefore in terms of the function, 3-layer BP neural network can meet most sophisticated detection systems' requirements, while for some detection systems which are simple mapping, linear perceptron model with a small amount of calculation and fast speed can be chosen so as to maximize the efficiency of the systems.

2.2 The choice of the number of input and output nodes

The number of the input nodes is determined by testing item. The number of the output nodes is generally determined by the number of the testing conclusions. But if there are a lot of testing conclusions, the number of the output nodes becomes great, then, the amount of calculation also increases. So when there are a lot of testing conclusions, the testing conclusions can be encoded to binary code, then output nodes take the number greater than or equal to $\log_2 N$, where N is the number of conclusions.

2.3 The choice of the number of hidden layer nodes

If the network model is BP network, the number of the hidden layer nodes needs to be set. Generally speaking, if there are too few hidden layer nodes, the network can not study well, and much more training will be needed and, meantime the training will not be highly precise; while too many nodes will bring issues such as a large amount of calculation, long training time and reduced network fault-tolerance capacity. Because there is no corresponding theoretical guidance, most designs are determined by combining the experience and trial calculation. System uses the default settings as the empirical formula (1).

$$n1 = \sqrt{n+m} + 2 \quad . \tag{1}$$

In the formula (1), "n1" is the number of the hidden layer nodes; "n" is the number of the input nodes; "m" is the number of the output nodes.

If the training effect of experience value is not good, you can re-set the number of the hidden layer nodes in the parameter settings and do many experiments in order to achieve faster convergence rate. In order to know the training efficiency, you should output the training number in training process, changes of the network errors, changes of network weights and the training time and so on, so as to a make a report for the network analyses.

2.4 The choice of transfer function

In the application of detection, most of the outputs are the field data and test conclusions. The field data are collected by the acquisition system; detection conclusions are obtained from the neural network algorithm. The detection conclusions are indicated with two-value, so the transfer function of output layer generally uses the sigmoid function or the hard limit function. The sigmoid function formula is shown as following .

$$f(x) = \frac{1}{1+e^{-x}} \, . \tag{2}$$

2.5 The choice of initial weight

Initial weight is generated randomly by the random function, and its value range is (-0.5, 0.5).

2.6 The choice of the learning rules

Because of the particularity of embedded systems, we should try to select the learning algorithm with simple calculation and low memory consumption. The perceptron network model structure is simple, and the overall amount of calculation is not large, so the weight correction algorithm of BP model is mainly considered. Conventional BP learning algorithm has large amount of calculation and long training time, and the slope of sigmoid function is close to 0 when the input is large, therefore the increase amplitude of gradient of learning algorithm is very small ,which may make the modification process of network weights almost at a standstill and the training very inefficient.

To reduce the amount of calculation and to improve the efficiency, we may adopt a flexible BP learning algorithm. We only need to consider the symbol of gradient to the error function, rather than the increase amplitude of gradient. The symbol of gradient determines the direction of weight updating, and the weight size change is determined by an independent “update value” (based on the former “update value”)[2].The iterative process of weight correction example is shown below:

$$\omega(t+1)=\omega(t)+\alpha\Delta\omega(t)\times \mathrm{sgn}(\frac{\partial E_t}{\partial\omega(t)}) \quad . \qquad (3)$$

In the formula (3)[2], Δω (t) is the previous update value, and the initial value Δω (0) is set by the actual application. α is the learning parameter, using variable step-size learning, that is, if it is in two successive iterations, the symbol of the partial derivative of error function to a weight remains unchangeable, then α is 1.2; otherwise α is 0.8[2]. It is faster to converge by using this algorithm, which can also be achieved effectively in embedded system.

2.7 The preprocessing of input and output data in training sample

In the network learning process, because the transfer function of neuron is a bounded function, while in the detection, as the dimension and unit of input test item data may be different, some values are very large, while some values are very small, which has a great influence on the network.. To prevent some neurons from reaching saturation state, and meantime make the larger input fall in the region where gradient of neuron activation function is large, input and output vectors need to be normalized before the training. Namely:

$$\mathbf{x'i}=\frac{\mathbf{xi}-\mathbf{xi}_{\mathbf{min}}}{\mathbf{xi}_{\mathbf{max}}-\mathbf{xi}_{\mathbf{min}}} \quad . \qquad (4)$$

In the formula (3): xi_{max}, xi_{min} are the maximum value and minimum value of each input component of number i neuron; xi, x'i are respectively the former and later input component after pretreatment of number i neuron. The input data values after normalization processing range between 0 and 1. Normalization processing requires a large number of mathematical operations. But if there is no huge difference among the input data values, the normalization is not required to reduce the computational complexity. Therefore, we may use whether to normalize the input data or not as the network training option parameter and make choice before the training.

If the neural network training is unsuccessful, or if the training time is too long, we need to analyze the process of training and to train after readjusting the network parameters. If the training results are satisfactory, we may use the testing sample data to test the function of the network.. Successful testing demonstrates that the construction of the network is completed, and then the network parameters and the weights of each Layer of the network are saved as files, ready for testing procedures.

3 Input and Output Platform Design

In the application of neural network, we need to input a large number of training data and output the training process and the training results, which requires a higher demand of input and output. To reduce the cost of portable detect device and the burden of volume, we may use two kinds of solution methods. One method is training on a host computer independently, and we download the network weights and network parameters of success training to the target computer in form of files for the detection program to recover the network;The other method is sharing the rich resources of input and output device of remote computer by using the remote training mode. In the above two methods, the training process of the former method is conducted on PC machine, which doesn't take up resources of embedded device, and its advantage is that the high-speed PC-CPU and the large capacity memory can greatly enhance the training speed, while the disadvantage is that the neural network parameter settings and the training are independent of detection system, which is inconvenient for users to modify and update the system; the second method integrates network training and testing, completed by the target machine, with remote computer only serving as sharing input and output devices. Because of limitations of embedded CPU and memory, the training becomes slow, and even collapses when the network structure is complex, or the amount of calculation is huge. But the system is flexible, and more convenient to be maintained and updated.

Because in neural network design, requirements of application have already been considered, and we have adopted more effective and simpler methods in the design of network structure and training algorithm; besides, the network training is the preliminary work of establishing detection system, and it will not be run after the network is built successfully, so its speed does not affect the work efficiency of the detection system. For these reasons, we use the second method to design.

To reduce the development, maintenance and use costs, we use B / S structure in the system , embedding a small WEB server in device, all of the user interface using static or dynamic web pages and being viewed through remote browser, and the data interaction and generation of dynamic web pages being realized by using high efficient CGI programming. The overall structure of the system is shown in figure 2:

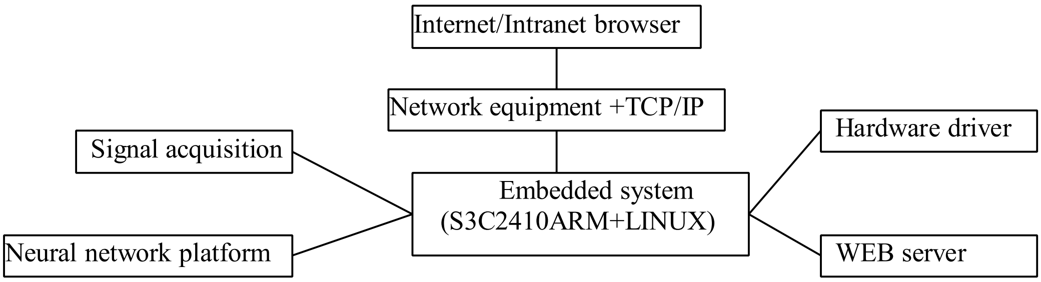

Fig. 2. Overall system structure diagrams

4 Conclusion

In the detection system, using neural network algorithms can increase the system flexibility and self-adaptability. When testing standards or project data change, we can use new training data to train the network again and get the new network parameters for updating system rapidly. Besides, the system is based on remote detection technology, making the system updating more convenient. And that also reduces the requirements of input and output of portable detect device and the cost of hardware and software development; if the test site environment is poor, the remote detection could show more on its advantages.

But the small embedded WEB server in the system may accordingly occupy more resources .Moreover, due to the inherent delay of network communication, the display of testing data and testing results have a certain delay relative to field data, so it is only suitable to the situation in which on-site environmental data are stable. If the on-site environment changes fast, you can add display components in the embedded device, which can have the devices test offline, and neural networks building and the system maintaining and updating could be conducted remotely.

References

1. Bernard, W., Samuel, D.S.: Adaptive Signal Processing(自适应信号处理). Machinery Industry Press, Beijing(2008)
2. Yubo, Tian.: Hybrid Neural Network Technology(混合神经网络技术). Science Press, Beijing(2009)