Three-valued Computer System Diagnosis Implemented by Artificial Neural Network

Tinghuai Chen, Kaigui Wu, Yundi Wu

(Presenter)
Computer Institute, Chongqing University
Chongqing, Sichuan, 630044, P. R. China
Tel.: 0811- 966291, FAX: 0086- 811- 814656

Abstract: This paper studies how to construct a neural network which can be used to solve the three-valued computer system diagnosis problems. Simulation results by using program show that it is very effective for solving computer system diagnosis problems with neural networks.

1 Introduction

A neural network is a computational artifact which simulates the signal processing of a biological neural system. Recently, it is shown [1] that relatively simple nets of densely interconnected analog units can be efficiently used for solving difficult optimization problems such as integer linear programming problems, and thus many types of neural networks have been proposed. Computer system diagnosis problem can be transformed into an integer linear programming problem hence can be implemented by a neural network. In previous papers, the study of computer system diagnosis is stressed on the uniqueness of the solution, called the diagnosability, based on this result, this unique solution can be obtained by a neural network. In this paper, a three-valued computer system diagnosis problem studied in [2] is introduced in Section 2. The syndrome of the tests are transformed into constraints of an integer linear programming problem hence can be implemented by a neural network. In previous papers, the study of computer system diagnosis is stressed on the uniqueness of the solution, called the diagnosability, based on this result, this unique solution can be obtained by a neural network.

2. Three-valued computer system diagnosis

A computer system diagnosis problem can be represented by a system diagnostic graph \( D(V,A) \), where \( V \) is a set of vertices (computers or subsystems) and \( A \) is a set of test arcs \((v_i,v_j)\) representing the relation that the subsystem \( v_j \) can detect subsystem \( v_j \) via a direct communication channel, for example see Fig. 1. Here a subsystem \( v \) consists of two parts: a task processor and a communication processor. When a failure occurs to a subsystem, it may happen either to only one of these two parts or to both of them simultaneously. The faulty state of a subsystem \( v \) is an assignment of the value 0, 1/2, or 1 to \( v \) such that

\[
\begin{align*}
\text{if both of the processors of } v \text{ are fault-free,} & \quad v = 0 \\
\text{if only the task processor of } v \text{ is faulty,} & \quad v = 1/2 \\
\text{if the communication processor of } v \text{ is faulty,} & \quad v = 1
\end{align*}
\]

as shown in Fig. 2.

In performing test procedure according to the diagnostic graph, it is assumed that a fault-free subsystem has the capability of correctly recognizing the status of other subsystems tested by it, this outcome of the test is always believable. If a test is applied by a 1-faulty or 1/2-faulty subsystem, the outcome is always unreliable, either 0 or 1/2 or 1, because its task or communication
processor can not perform correctly. Therefore when a tester \( u \) detects a testee \( v \), the outcome \( T \) is shown in Fig. 3, where \( x \) may be any value of \( 0, 1/2 \) or \( 1 \).

\[
\begin{array}{ccccccc}
\text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}
\]

\[
\begin{array}{ccccccc}
\text{u} & \text{v} & \text{T} & \text{ } & \text{ } & \text{ } & \text{ } \\
\text{0} & \text{0} & \text{0} & \text{1/2} & \text{1/2} & \text{1} & \text{1} \\
\text{0} & \text{1/2} & \text{1} & \text{0} & \text{1/2} & \text{1} & \text{1} \\
\text{0} & \text{1/2} & \text{1} & \text{x} & \text{x} & \text{x} & \text{x} \\
\end{array}
\]

\( x = 0, 1/2 \) or \( 1 \)

**Fig. 3** Outcome of a test

Our diagnosis problem can be stated as that when all the outcomes of the tests, called syndrome, are given, how can we identify the status of all the subsystems?

The advantage of partitioning a subsystem into task and communication processors is that when a subsystem is diagnosed to be \( 1/2 \), then only the communication processor should be replaced, thus alleviating the maintenance work. Besides, if \( u \) detects \( v \) and the outcome is \( 1/2 \), then \( u \) can use the communication channel of \( v \) to transfer the test data to the subsystem \( z \) next to \( v \) and receive its response. It seems that there exists a test relation from \( u \) to \( z \). This test is called a virtue test as shown in Fig. 4 in dotted arc. When virtue tests are added to the test graph, maybe new \( 1/2 \) outcomes occur and this process goes on until no new \( 1/2 \) outcome occurs. The virtue test of Fig. 1 is shown in Fig. 5. The virtue tests will strengthen the diagnosing capability of the original diagnostic graph. Of course to obtain the virtue test outcomes consumes more time.

**3. Diagnostic algorithm**

**- Integer linear programming**

Since the syndrome is given, we are going to find the status of all subsystems. First, we consider all the possibilities of the status with respect to each value of the syndrome. Consider each column of the table of Fig. 3 as a point in the three dimensional 1-cube, see Fig. 6. Obviously the values of tester \( u \) and testee \( v \) should satisfy \( 0 \leq u, v \leq 1 \). For each value of \( T \), the possible status lie in the shaded areas. For \( T = 0 \), lie in the lower part of or on the straight line \( 2u - v = 0 \). That is to say

\[
\begin{align*}
T = 0 & \iff 2u - v = 0, \\
T = 1 & \iff 2u + v - 1 = 0, \\
T = 1/2 & \iff 2u + 2v - 1 = 0 \text{ and } 2u - 2v + 1 = 0
\end{align*}
\]

These inequalities corresponding to the syndrome of all tests are considered as the constraints and the three-valued system diagnostic algorithm is transformed into an integer linear programming. Since the case that all subsystems are \( 1 \)-faulty or \( 1/2 \)-faulty is of course compatible with any syndrome, we are going to find the solution with least number of "1"s and "1/2"s. Therefore the objective function is

\[
\alpha_0 \sum v_i + \alpha_1 \sum v_i = \min,
\]

where \( v_1, v_2, ..., v_n \) are the subsystems in \( V \), \( \alpha_0 \) and \( \alpha_1 \) are weights of each part. Besides, if the number of 1-subsystems does not exceed the given integer \( t_1 \) and the number of 1/2-subsystems does not exceed the given integer \( t_{1/2} \), we will have additional constraints:

```plaintext
\[
\alpha_0 \sum v_i + \alpha_1 \sum v_i = \min,
\]

where \( v_1, v_2, ..., v_n \) are the subsystems in \( V \), \( \alpha_0 \) and \( \alpha_1 \) are weights of each part. Besides, if the number of 1-subsystems does not exceed the given integer \( t_1 \) and the number of 1/2-subsystems does not exceed the given integer \( t_{1/2} \), we will have additional constraints:
\[ \sum v_i \leq t_1 \quad \text{and} \quad \sum v_i \leq t_{1/2} \] respectively.

In [2] a necessary and sufficient condition of \( t_{1/2} \leq 1 \)-diagnosability (uniqueness of solution) for three-valued computer system diagnosis is introduced. For example, if \( t_1 = 1 \) and \( t_{1/2} = 1 \), the system in Fig. 1 is not \( t_{1/2} \) diagnosable, but by adding the virtue tests, the system in Fig. 5 is \( t_{1/2} \) diagnosable.

**Example 1:** Consider the system in Fig. 5, for \( t_1 = 1 \) and \( t_{1/2} = 1 \), the diagnosis problem is characterized by the following integer linear programming.

**Goal function:**
\[
\alpha_0 (v_1^1 + v_2^2 + v_3^3 + v_4^4 + v_5^5) - v_1^i = 1/2
\]

\[ + \alpha_1 (v_1^1 + v_2^2 + v_3^3 + v_4^4 + v_5^5) v_1^i = 1 - \text{min} \]

where \( \alpha_0 \) and \( \alpha_1 \) are weights usually \( \alpha_0 = \alpha_1 \).

**Constraints:**
\[
\begin{align*}
T_{12} &= 1, \quad 2v_1^1 + v_2^2 + v_3^3 \geq 0; \quad T_{15} = 0, \quad 2v_1^1 - v_5^5 \geq 0; \\
T_{24} &= 1, \quad 2v_2^2 - v_4^4 \geq 0; \quad T_{25} = 1, \quad 2v_2^2 + v_5^5 \geq 0; \\
T_{31} &= 0, \quad 2v_3^3 - v_1^1 \geq 0; \quad T_{32} = 1, \quad 2v_3^3 + v_2^2 \geq 0; \\
T_{41} &= 1, \quad 2v_4^4 + v_1^1 \geq 0; \quad T_{43} = 1, \quad 2v_4^4 - v_3^3 \geq 0; \\
T_{53} &= 0, \quad 2v_5^5 - v_3^3 \geq 0; \quad T_{51} = 0, \quad 2v_5^5 + v_1^1 \geq 0; \\
T_{54} &= 1/2, \quad 2v_5^5 + v_4^4 \geq 0; \quad 2v_5^5 - v_4^4 \geq 0; \\
\end{align*}
\]

Generally the m constraints (except the last one) of the system diagnosis can be expressed as

\[
\sum v_i \leq 1 \quad v_i = \text{"1"} \quad v_i = \text{"1/2"} \]

\[ i = 1, 2, 3, 4, 5 \quad i = 1, 2, 3, 4, 5 \]

\[ v_1^1, v_2^2, v_3^3, v_4^4, v_5^5 \geq 0, 1/2, 1. \]

4. **Energy Function**

An energy network consists of a set of neurons, the inputs of each neuron are the input signal \( I_i \) from outside and the feedback signals from other neurons as well as from itself. See Fig. 7. The output of a neuron is a monotone increasing function of the summation \( u \) of the weighted inputs (including the outside input \( I_i \)). The design of a network is simply to find an assignment of real values to the weights \( w_i \). Since each subsystem \( v_i \) of the system diagnosis has three states, these can be implemented by two neurons \( v_{10} \) and \( v_{11} \), both of them are either 1 or 0 as shown in Fig. 8. Thus \( v_i = a_{10} v_{10} + a_{11} v_{11} \), where \( a_{10} = 1/2 \) and \( a_{11} = 1 \).

The energy function to be minimized for the system is

\[
E = E_0 + E_1 + E_2 + E_3,
\]

where \( E_0 = \sum \sum a_{1j} v_{ij} \cdot (a_{1j} = 1/2, a_{11} = 1) \) stands for

\[
\sum v_{ij} \geq 0, \quad j = 1, 2, 3, 4, 5, \quad \text{for regulation, usually } a_{11} = a_{10}.
\]

The goal function which represents the total number of faulty subsystems to be minimized, in which \( v_{ij} \) correspond to the i-th subsystem \( v_i \), \( a_{1j} \) are coefficients for regulation, usually \( a_{1j} = a_{10} \):

\[
E_1 = \beta \left[ \sum F \left( \sum a_{1j} v_{ij} \cdot B_{ij} \right) + \sum F \left( a_{1j} \cdot v_{ij} \right) \right],
\]

\[ j = 1 \quad r = 0 \quad i = 1 \quad n \quad m \]

\[ r = 0 \quad i = 1 \quad n \quad m \]
where \( \sum \sum a_{ij} v_{ir} - B_j \geq 0, j=1, \ldots, m \) are constraints corresponding to the syndrome of all the tests, \( F \) is a special function explained later, \( s_t = t_1/2 (t_1) \) for \( r = 0 \) (1),

\[
\sum_{i=1}^{n} v_{i0}^x t_{1/2} = \sum_{i=1}^{n} v_{i1}^x t_1
\]

are the constraints of the upper bound for the faulty subsystems.

\[
E_2 = - \sum_{i=1}^{n} \int_{r=0}^{t_{1/2}} g^{-1} (v) dv,
\]

this term is to guarantee the global stability of the system,

\[
E_3 = \delta \left( \sum_{i=1}^{n} v_{ir} (1-v_{ir}) + \sum_{i=1}^{n} v_{i0} v_{i1} \right),
\]

this term is to guarantee that each subsystem \( v_i \) is three-valued.

5. Design of the network

From the energy function we can obtain the dynamic equation of the neuron system. Note that

\[
\frac{d u_{ir}}{d t} = - \frac{\partial E}{\partial v_{ir}}
\]

\[
= - \alpha_\gamma a_\gamma \beta \left[ \sum_{j=1}^{m} a_{ij} D_{ji} \left( \sum_{r=0}^{t_{1/2}} a_{ij} D_{ji} y_{ir} - B_j \right) \right]
\]

\[
= - \gamma - 2 \delta v_{ir} - \delta v_{i0}^{t_{1/2}} - (\alpha_\gamma a_\gamma + \delta) \sum_{j=1}^{m} a_{ij} D_{ji}
\]

Here \( f(z) = dF(z)/dz \) is a penalty function, if \( z < 0 \), then \( f(z) = -z^2 \), which causes penalty proportional to \( z^2 \). In comparison with the typical dynamic equation of Hopfield Network, we can obtain the values of weights for mutual connection

\[
w_{ir,j} = \begin{cases} \gamma & i = j, r = s \geq 1 \\ 0 & \text{otherwise.} \end{cases}
\]

The outputs of neurons are the inputs to the constraints represented by penalty function \( f \). The weights which connects the \( j \)-th output of the penalty function \( f \) to the \( i \)-th neuron \( (v_{i0}, v_{i1}) \) are \( -\beta a_{ij} D_{ji} \) respectively, \( j=1,2,\ldots,m \), but for the last two constraints the weights are \( \beta \) as shown in Fig. 9. The input to the neuron \( v_{ir} \) from outside is \( \mathbf{I}_{ir} = -(\alpha_\gamma a_\gamma + \delta) \), \( i = 0,1,2,\ldots,n \) and that to the penalty function \( f \) are \( -B_j \) \( (j=1,2,\ldots,m) \) and \( t_{1/2}, t_1 \) respectively. The entire neural network is shown in Fig. 9.
6. Stability of the Network

\[
dE = \sum_{i=1}^{n} \sum_{r=0}^{1} \frac{dE_{ir}}{dt} - \sum_{i=0}^{n} \frac{dE_{ir}}{dt}
\]

Since \( E \) is bounded, by Lyapunov's Theorem, it is stable. The uniqueness of the solution comes from [2].

7. An simulation example

The three-valued computer system diagnosis problem shown in Fig. 5 is taken as an example to verify the above theory by using the proposed network simulated by computer program. The parameters are taken as follows: \( C=R=1, t_0=1/2, a_1=1, \alpha_0-5, \alpha_1=20, \beta=\gamma=300, t_1=1, t_1/2=1 \). The initial state of the neurons are generated randomly near 1/2. When the difference of energy \( E \) between two successive iterations \( \Delta E \leq 10^{-6} \), the algorithm is considered to be convergent. For any such initial state, we obtain after 74 to 150 iterations (depend upon initial state) the unique solution:

\[
\begin{align*}
\nu_{10} &= 0, \quad \nu_{20} = 0, \quad \nu_{30} = 0, \quad \nu_{40} = 1, \quad \nu_{50} = 0, \\
\nu_{11} &= 0, \quad \nu_{21} = 1, \quad \nu_{31} = 0, \quad \nu_{41} = 0, \quad \nu_{51} = 0.
\end{align*}
\]

Thus the status of the subsystems diagnosed is \( v_1=0, \quad v_2=1, \quad v_3=0, \quad v_4=1/2, \quad v_5=0 \), which is correct.

8. Conclusion

We have transformed a three-valued computer system diagnosis into an integer linear programming problem, which can be implemented by a neural network modified from that of Tank and Hopfield [1]. The main modification is to use continuous neural variables to implement three-state discrete variables. The solution is unique, stable and the algorithm converges fast. It seems successful to obtain the solution of computer system diagnosis by using neural network and this technique will be extended to other system diagnosis models in the future works.

References