

복합시스템 고장진단을 위한 다중신경망 개발

배 용 환

안동대학교 기계공학교육과

(2000. 1. 15. 접수 / 2000. 5. 29. 채택)

Development of Multiple Neural Network for Fault Diagnosis of Complex System

Yong-Hwan Bae

Department of Mechanical Engineering Education, Andong National University

(Received January 15, 2000 / Accepted May 29, 2000)

Abstract : Automated production system is composed of many complicated techniques and it become a very difficult task to control, monitor and diagnose this compound system. Moreover, it is required to develop an effective diagnosing technique and reduce the diagnosing time while operating the system in parallel under many faults occurring concurrently. This study develops a Modular Artificial Neural Network(MANN) which can perform a diagnosing function of multiple faults with the following steps: 1) Modularizing a complicated system into subsystems. 2) Formulating a hierarchical structure by dividing the subsystem into many detailed elements. 3) Planting an artificial neural network into hierarchical module. The system developed is implemented on workstation platform with X-Windows[®] which provides multi-process, multi-tasking and IPC facilities for visualization of transaction, by applying the software written in ANSI-C[®] together with MOTIF[®] on the fault diagnosis of PI feedback controller reactor. It can be used as a simple stepping stone towards a perfect multiple diagnosing system covering with various industrial applications, and further provides an economical approach to prevent a disastrous failure of huge complicated systems.

1. 서 론

현대의 자동화 시스템은 어느 한 사람에 의하여 설계되기보다는 여러 사람의 기술이 집약된 지식의 총합체로 나타나고 있어서, 프로세스 제어 플랜트의 복잡함을 관리할 수 있고 유지 보수비용을 줄일 수 있는 제어, 감시, 그리고 진단을 위한 복잡한 시스템을 요구한다. 산업현장에 적용된 복잡한 자동화 시스템에서 발생하는 고장으로 인한 생산 손실은 더욱 커지고 있고, 그에 따르는 적절한 진단시스템은 제대로 구축되어지지 않고 있다. 따라서, 각 요소작업들이 오동작을 일으키거나 고장나기 전에 위험한 요소를 대체시킨다면, 고장이 일어났을 때 감지하

여 고장 요소의 오동작을 확인시키고, 시스템이 계속 작동할 수 있도록 여유 요소로 대체하는 시스템이 필요하다. 사고방지와 시스템의 빠른 복구를 위해서 진단시간의 단축과 효율적인 진단전략이 절실히 요구된다.

고장을 진단하기 위한 접근방법에는 모델베이스 기법과 패턴인식 기법이 적용되고 있는 것이 일반적이다. 모델베이스 기법은 수식모델이나 신경모델, 해석모델, 예측모델 등의 고장진단 모델을 사용하여 고장을 진단한다.¹⁾ 이것은 고장이 일어나는 원인을 그와 관련된 징후(symptom)에 의하여 진단하도록 되어 있다. 이 기법은 시스템 자체에서 요소의 결합이 전체 시스템 고장에 어느 정도 영향을 미치는 지는 정확히 알 수 없고, 제한된 범위에서만 적용할 수 있으며 계산량이 많다는 것이 문제로 지적되고 있다. 패턴인

yhbae@andong.ac.kr

식 기법은 인공신경망(Artificial Neural Networks)을 사용하여 지식을 학습하고 인간의 작업행동을 묘사해 고장진단을 한다.²⁾ 이 기법은 패턴을 인식하기 위한 알고리즘을 만들어야 하는 어려움이 있지만, 수학적 모델에 대한 지식이 없이 측정데이터에 따라서 고장을 분류할 수 있는 잇점이 있다.

인공신경망을 이용한 고장감지 및 진단은 여러 분야에서 연구 개발되고 있다.³⁻⁶⁾ 그러나 이전의 연구에서는 단일 인공신경망을 이용하여 고장진단을 수행하는 것이 대부분이다. 이렇게 되면 고장진단을 수행하는데 있어서 한 신경망에 의존하게 되므로 고장인식률이 낮아지게 되고, 차후에 시스템의 구성요소가 바뀌면 인공신경망을 다시 학습시켜야 하는 번거로움이 생긴다.

따라서 본 연구에서는 전체시스템을 계층적으로 분류하여 고장정보를 가지고 컴퓨터 시스템에 인공신경망을 이식시킴으로써 초기 고장진단에 착수하며, 다중고장진단을 하는데 효과적인 모듈형 인공신경망(Modular Artificial Neural Network, MANN)이라 명명한 시스템 개발에 중점을 둔다. 인공신경망이 계층적으로 모듈화되므로, 차후에 시스템의 구성요소가 바뀌더라도 해당하는 모듈만 수정하면 전체시스템을 사용할 수 있도록 구성한다. 시스템이 다중고장을 진단하기 위해서는 많은 프로세스를 수행시키며 관리를 해야하고, 또한 프로세스 상호간에 협조를 하면서 진단을 내려야 한다. 더구나 프로세스들 상호간의 통신이 순차적으로 발생하는 것이 아니기 때문에 프로세스들은 항상 통신로(프로세스간 메시지를 전달할 수 있는 통로)를 개방해 놓고 감시하고 있어야 한다. 이러한 조건들을 수행하기 위해서는 일반적으로 사용하는 개인용 컴퓨터의 순차적 제어로는 불가능하다. 따라서 본 연구에서는 다중프로세스(multi-process)기능과 다사용자(multi-user)기능, 그리고 프로세스간 통신기능인 IPC(Inter Process Communication)설비를 가지고 있는 워크스테이션을 사용하고, 운용되는 프로세스들의 수행사항을 가시화하기 위해서 X- Windows[Ⓢ] 환경을 바탕으로 하여 ANSI-C[Ⓢ] 언어와 MOTIF[Ⓢ]를 이용하여 프로그램을 개발하였다. MANN은 거대한 플랜트라는 전체시스템을 부속시스템이라는 요소단위로 모듈화하고, 다시 각 부속화된 부속시스템들을 그 부속시스템의 기계셀(cell)들

로 나누고, 또 더 작은 요소단위인 기계적 구성요소로 모듈화한 계층적 구조를 사용하여 각 모듈별로 인공신경망을 이식시켜서 고장진단을 수행했고, 이것을 이용하여 PI피이드백 제어반응기에서 시뮬레이션을 적용해 봄으로써 다중고장을 진단하였다.

2 이론적 배경

2.1. 복합시스템의 계층구조

현대의 첨단시스템은 단순한 작업공정단위인 서브시스템들로 모듈화 할 수 있고, 각 공정단위는 세부 공정단위인 요소단위인 아이템(item)들로 모듈화 할 수 있다. 또, 각 세부 공정단위는 최소의 기계적 요소단위(component)로 분류할 수 있다. 이렇게 분류된 모듈간의 관계를 바탕으로 하여 계층구조를 Fig. 1과 같이 만들 수 있다. PI 피드백(feedback)제어반응기 시스템을 서브시스템으로 예를 들면, 아이템은 콘트롤러, 콤퓨러서, 반응기 등으로 구성될 수 있다. 그리고 하나의 아이템은 파이프, 센서 등과 같은 기계적 요소들로 구성 될 수 있다. 계층적 구조에 있어서 시스템을 구성하는 요소들을 바탕으로 하여 계층을 형성하는 것을 구조적 계층(structural hierarchy)이라 하고, 시스템이 올바르게 작동할 수 있도록 하기 위해서 서브시스템들의 기능적 분석에 바탕을 두어 계층을 형성하는 것을 기능적 계층(functional hierarchy)라 한다. 자동차 시스템의 엔진도 구조적 계층을 지니고 있다. 엔진은 시동서브시스템, 점화 서브시스템, 냉각서브시스템, 윤활서브시스템 등으로 구성되고, 점화서브시스템은 스파크 플러그, 점화코일, 분배기 등으로 구성되어 계층을 이루고 있음을 알 수 있다. 엔진의 기능적 계층측면을 고려할 때 엔진이 올바르게 작동하기 위해서는 시동서브시스템, 점화서브시스템, 연료서브시스템, 밸브의 기능과 윤활서브시스템의 기능과 냉각서브시스템의 기능이 유기적으로 작동해야 한다. 또한 연료 서브시스템이 제대로 작동하기 위해서는 연료탱크, 연료정화기, 연료펌프, 기화기의 기능과 공기정화기의 기능이 제대로 동작해야만 한다. 이러한 기능들이 계층적으로 조화롭게 돌아가야만 혼합가스가 분출될 수 있다. 이러한 구조적 계층과 기능적 계층은 고장의 원인을 찾는 데 많은 정보를

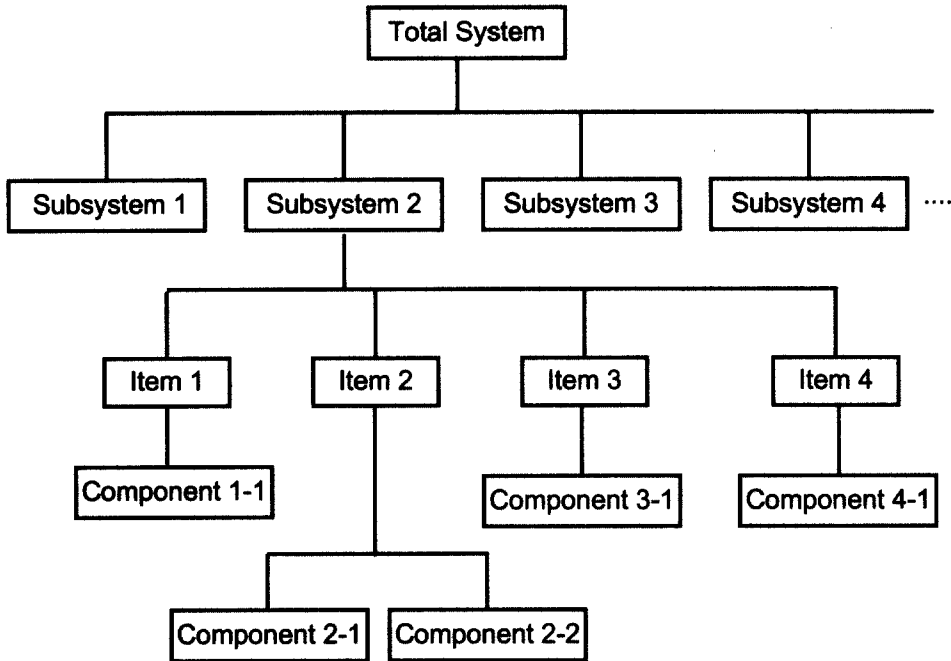


Fig. 1. Hierarchy of the complex system

제공한다. 즉 이 두 계층구조에 따라서 고장을 동시에 진단하여 조합한다면 진단시간이 훨씬 단축되고, 그 위치도 빨리 찾을 수 있다.

2.2. 다중프로세스간 통신제어기법

본 연구에서 복합시스템을 여러 개의 신경망을 이용하여 다중으로 고장진단하기 위해서 UNIX 체제의 IPC설비를 이용하는데, 이를 위하여 프로세스라는 개념을 이해해야 한다. 프로세스란 프로그램의 실행을 의미하며, 수행중인 프로그램 그 자체를 말한다. 또한 각계의 계층신경망의 통신을 위하여 메시지큐, 공유메모리, 세마포어 등의 IPC(Inter Process Communication) 설비를 사용한다. 메시지큐는 여러 개의 프로세스들간의 통신을 위하여 제공되는 것이다. 복합기계에서는 계층형 신경망들 사이에 특징패턴 데이터와 결과값을 각각의 신경망과 OR 프로세스에게 넘겨주기 위하여 사용한다. 하나의 서브시스템당 하나의 키값이 사용되고, 같은 메시지큐값을 사용한다.

본 연구에서 공유메모리는 특정의 관심 있는 데이터를 이것을 필요로 하는 여러 프로세스에게 동시에 제공하기 위하여 자신들의 가상주소

공간의 일부분(공유메모리)을 공유하면서, 공유메모리에 저장된 데이터를 읽고, 또 공유메모리에 데이터를 기록함으로써 상호 통신할 수 있다. 특히 신호 발생 시물레이터 터미널에서 들어온 이미지 데이터를 여러 개의 신경망에 전달하기 위해서 공유메모리를 사용하고, 또한 트리거 신호영역은 결과에이전트와 OR프로세스가 공동으로 사용하는 공유메모리이다. 공유메모리를 조작하기 위한 시스템 호출은 메세지큐를 위한 시스템 호출과 비슷하다.

세마포어는 특정의 임계영역(공유자원, 화일)속에서 한 순간에 존재하는 프로세스는 오직 하나로 제한하기 위하여 사용하는 것으로, 본 진단시스템에서는 시스템 성능을 체크하여 특정치 이하로 떨어지면 블랙보드 영역의 트리거에 메세지를 송신하여 세팅시킨다. 이후 이상부속시스템이 진단되고 있는 동안 성능체크모듈에서 계속 트리거가 작동될 수 있으므로 이 트리거영역을 임계영역으로 하여 진단이 끝날 때까지 트리거가 작동되지 않도록 하여야 한다.

또한 신호 발생 시물레이터 터미널에서 들어온 이미지 데이터를 120개 받으면 공유메모리상으로 보내 신호 전처리 모듈이 데이터를 공유메모리에

쓰고 있는 사이에 데이터의 완전성을 보장하기 위하여 공유메모리 영역에 대하여 작업을 하는 동안 다른 프로세스가 들어오지 못하게 임계영역으로 만드는데 이때 세마포어가 작동된다.

본 실시간 시스템에서는 성능체크에 의하여 부속시스템 고장진단을 위하여 트리거가 작동되는데 트리거는 어느 특정한 순간에 한번 존재한다. 다시 말해서 트리거는 진단이 되는 동안 임계영역속에 들어 있고 진단이 끝나면 시그날이 날아와 임계영역에서 빠져나오게 된다.

3. 시스템 구현

본 연구는 일반적인 복합시스템을 계층적 구조의 모듈로 분류하여, 각 모듈에 인공신경망을 이식시켜 시스템에서 발생하는 다중고장을 초기에 감지하는데 효과적인 계층형 인공신경망을 구현하는데 중점을 둔다. 이러한 시스템 구현을 위해서는 많은 프로세스를 관리할 수 있고, 가시화와 프로세스간에 통신이 가능한 유닉스 운영체제를 가진 워크스테이션의 X-Windows[®]에서 IPC설비와 ANSI-C[®] 프로그래밍 언어와 MOTIF[®]를 사용하여 PI피드백 제어반응기에 적용해 보고, 전체적인 고장진단 과정을 시뮬레이션해 본다.

3.1. 계층인공신경망(Hierarchical Artificial Neural Network)

모듈인공신경망(MANN)을 사용하는 기본적인 개념은 물리적 시스템의 계층적 구조를 사용하여 각 계층별로 인공신경망을 이식하여 다중고장을 진단하고, 좀더 빠르고, 쉽게 학습될 수 있는 인공신경망 구조를 개발하는데 있다. 단일 인공신경망을 사용할 때 구성요소나 학습패턴이 바뀌면, 전체 학습패턴을 다시 학습해야 하고 전체적인 알고리즘을 다시 프로그램 해야하는 단점이 있다. 본 연구에서는 새로운 구성요소가 추가 되더라도 추가된 구성요소에 대한 학습될 고장정보로 인공신경망을 만들고, 알고리즘에서 추가된 인공신경망을 호출하는 프로그램만 삽입함으로써 차후 수정이 용이하도록 구현한다.

학습 알고리즘으로는 다층 퍼셉트론에 적용되는 알고리즘 중 원하는 출력과 실제 출력사이의 에러의 사승평균을 최소화 하는 백프로파게이션

(back propagation) 알고리즘을 이용한다. MANN의 구조는 가장 단순한 형태로 두 개의 연결된 단계를 가진다. Fig. 2에 MANN의 기본적인 구조를 나타내었다. Net⁰라고 명명된 첫 번째 층에서는 신경망을 학습시키고 받아들이는 입력데이터를 처리한다. Netⁱ(i=1,2,...,n)로 명명된 두 번째 층에서는 첫 번째 층과 유사성을 가지고 있는데, 신경망을 학습한 상태에서 첫 번째 층의 활성화 신호를 기다린다. 첫 번째 층은 단일고장과 이중고장에 대한 학습패턴을 가지고 있고, 그 출력은 정상상태에 대한 출력을 고려함으로 분류된 고장의 수보다 하나 많은 수만큼의 노드가 존재하게 된다. 첫 번째 층의 출력노드는 두 번째 층에 있는 각 네트워크와 연결되어지고, 두 번째 층의 네트워크는 고장의 수만큼 존재한다. 각 네트워크에서는 자신의 고장을 포함한 이중고장에 대한 학습패턴을 학습한다. 전체적인 구조로 볼 때 n을 고장의 수라고 하면, n+1개의 네트워크가 존재하게 된다. n+1개의 학습패턴들은 Net⁰에서 학습되어지고, n개의 학습패턴은 Netⁱ(i=1,2,...,n)에서 학습되어진다. Netⁱ의 네트워크에는 이중고장에 대해 학습되었기 때문에 그 출력노드에는 1이 두 개씩 들어있다. 세 번째 층의 OR연산은 MANN의 출력이 되는데, 두 번째 단계의 출력들을 모아서 OR연산을 행하여 결과를 출력한다.

이러한 과정을 예를 들어 설명하자면 Fig. 2에서 고장 5,6,7을 가지고 있는 가공되지 않는 입력데이터를 가지고 단일고장과 이중고장만을 학습한 첫 번째 단계의 신경망(Net⁰)은 고장 5와 7을 감지한다. 따라서 두 번째 단계의 신경망중 고장⁵를 포함하는 이중고장만을 학습한 신경망(Net⁵)이 활성화된다. 마지막으로 OR프로세스는 Net⁵와 Net⁷의 수행결과를 받아서 OR연산하여 결과를 출력한다. 이러한 기본적인 MANN을 가지고 더욱 확장할 수 있다. 즉, 기본적인 MANN 아래에 다시 기본적인 MANN을 두 개 연결함으로써 4단계의 64개의 신경망을 구축할 수 있다. 이러한 형태로 계속 확장하여 계층을 형성해 가면 필요할 때마다 신경망을 추가할 수 있다. 전체시스템의 구성요소가 바뀔 때마다 전체 신경망의 형태에서 추가된 부분만 모듈로 만들어 삽입하면 된다. 기본적인 MANN을 두 개 연결하여 기계아이템과 기계적 구성요소까지 진단하는데 사용될 수 있는 확장된 MANN이 Fig. 3에 나타

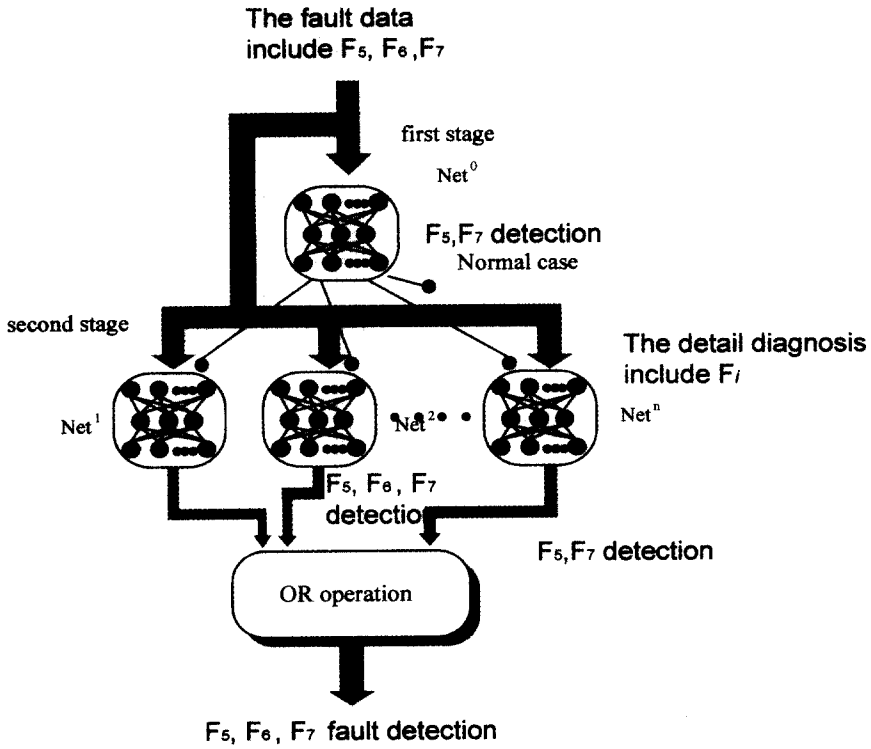


Fig. 2. The example of a fault diagnosis of a basic MANN

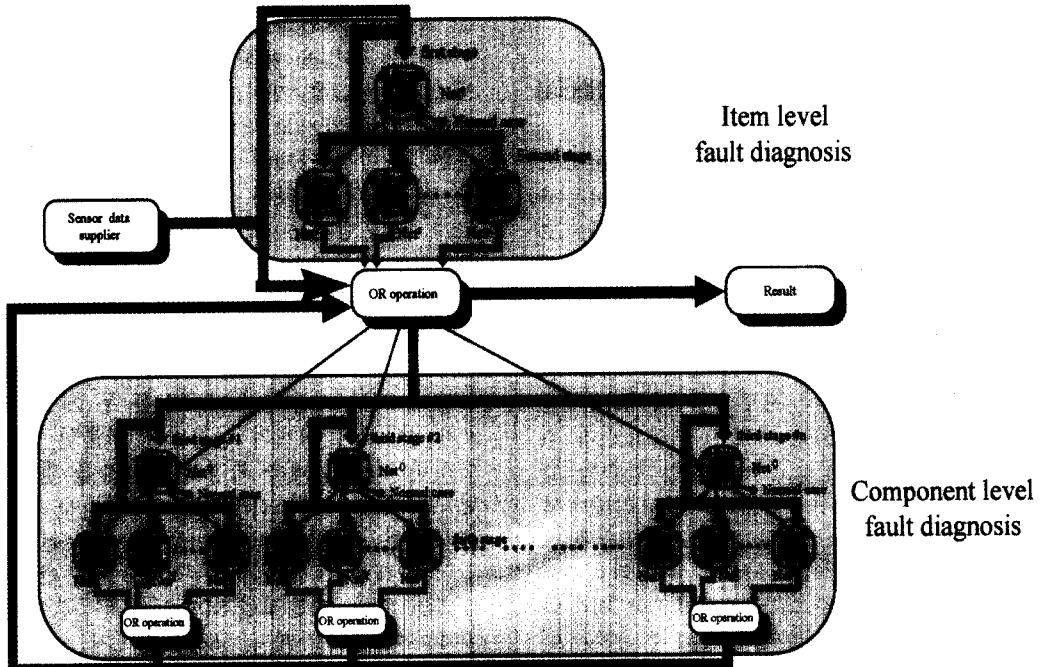


Fig. 3. The extended MANN architecture

나 있다.

각 신경망은 독립적으로 수행되는데 Net^0 가 Net^1 를 활성화시키는 방법은 유닉스 시스템이 제공하는 IPC설비중에 메시지큐를 사용한다. Net^0 의 각 출력노드와 Net^1 사이에 유일한 통신로를 만들어, Net^0 의 출력노드로 입력데이터를 보내면 Net^1 는 그 데이터를 받아서 결과를 산출하여 OR연산 프로세스로 보낸다. 물론, Net^1 의 출력노드는 OR 프로세스와 메시지큐라는 통신로를 가진다.

3.2. 운용 알고리즘

고장이 발생할 수 있는 시간과 그 수는 예측할 수 없기 때문에 프로그램은 고장이 발생할 때마다 감지할 수 있어야 하고, 그와 동시에 발생한 수많은 고장의 원인을 찾을 수 있어야 한다. 따라서, 여러 프로세스를 독립적으로 수행시키는 멀티프로세스 기능을 갖추고 있는 Work-

station에서 구현하는 것이 적합하다. 또한, 컴퓨터와 사용자간의 대화를 위한 사용자 인터페이스 기능을 제공하는 X-Windows에서 ANSI-C 프로그래밍언어와 MOTIF를 사용하여 프로그램을 작성했다.

다음은 Fig. 4, Fig. 5는 본 프로그램에서 가장 중요한 첫 단계 신경망과 두 번째 단계신경망을 수행하는 알고리즘을 나타낸다. Fig. 4는 MANN의 첫 번째 단계의 신경망으로서, 필요한 신경망의 파라미터를 읽어들이고, 메시지큐 ID를 얻어서 입력된 데이터로 신경망을 수행하여 출력값이 0.5보다 크거나 같을 때, 두 번째 단계의 프로세스들을 활성화시키는 신호를 송신한다. 그리고 다음 입력데이터가 있는지 확인한다. Fig. 5는 MANN의 두 번째 단계의 신경망으로써 메시지를 받아들일 큐형태와 메시지를 보낼 큐 형태를 얻고, 필요한 신경망의 파라미터를 읽어들이고,

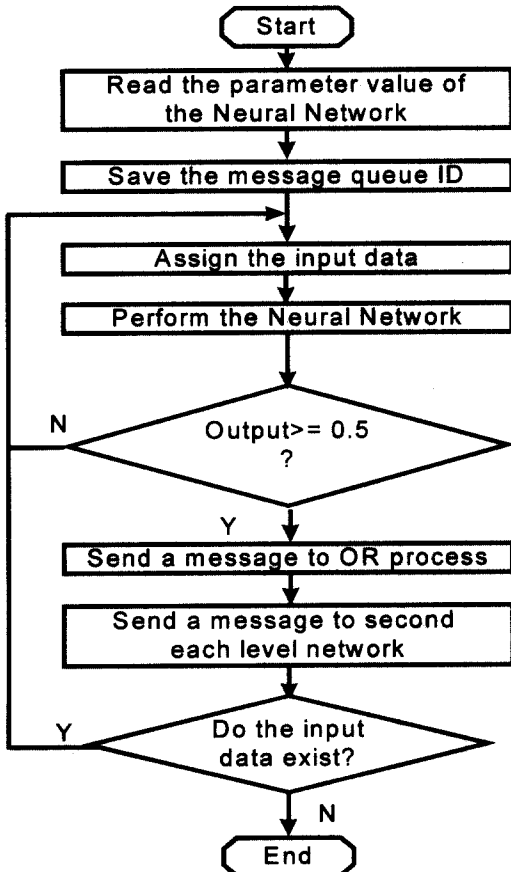


Fig. 4. The procedure of the first level network

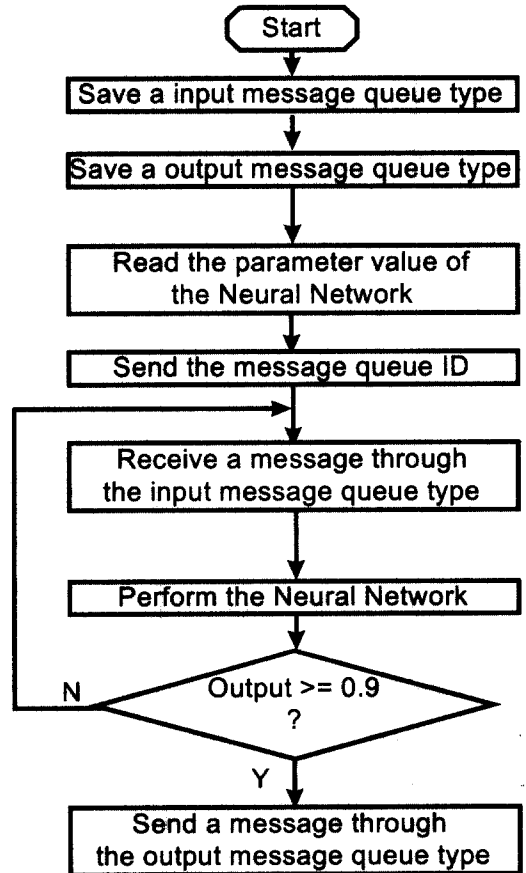


Fig. 5. The procedure of a second level network

메시지큐 ID를 얻어서 입력된 메시지들을 받아들이고 신경망을 수행하여 결과값이 0.9보다 크거나 같을 때, 출력메시지 큐로 결과는 전송한다. OR프로세스는 정해진 메시지큐로부터 두 번째 단계의 각 신경망 프로세스로부터 수행된 결과를 받아서 OR연산을 하여 그 결과를 처리한다. 그리고 종료를 원하면 주 프로세스에 종료신호를 보낸다.

3.3. 적용 예

본 연구의 적용대상으로는 Fig. 6과 같이 헵탄을 톨루엔으로 반응시켜 새로운 물질을 합성하는 PI feedback controlled reactor을 나타낸 것으로 거의 정상상태 조건하에서 작동한다. 아주 극단적인 돌발적인 사고는 일어나지 않는다고 가정한다. 또한, 초기 단계에서 포함된 프로세스 고장은 배제되고 PI 제어기에 의하여 완화되도록 한다. 탱크 속에 저장된 헵탄은 프로세스 펌프 1을 통하여 반응로로 보내지고, 반응로 속에 일어나는 작용으로서 반응비율은 반응로 속의 온도에 의하여 조절된다. 증기는 재생펌프 2를 통하여 열교환기로 공급된다. 열교환기의 출구값은 PI 제어기에 의하여 제어되는데, 제어되는 변수는 반응온도이다. 여기서 열교환기의 제어는 전자제어회로나 컴퓨터를 통하여 이루어진다고 가정한다. 그리고, 여기서는 작동유체의 상태에

만 관심이 있으므로, 제어기에서의 고장은 포함시키지 않는다. MANN의 적용에로써 7가지 가능한 고장 예를 고려하면 다음과 같다. 0.1% 감소 명령 신호 $u_0 (F_1)$, 2.0% 감소 주파수 요소 $h_0 (F_2)$, 0.1% 감소 활성화에너지 $Ea (F_3)$, 2.0% 감소 열교환기계인 $K(F_4)$, 2.0% 감소 전체 열전달 계수 $h (F_5)$, 1.0% 감소 입구에서의 헵탄 (F_6), 1.0% 감소 출구에서의 증기 반응온도 $T_i (F_7)$ 이라 정의하며, 파라미터속에서 이와 같은 변화들은 같은 정도(2.0%)의 프로세스 변수 속에 오차를 유발한다. 고장공간은 7차원($n=7$)이고, 상관관계 분석에 의해서 측정 공간 속에서 고장진단에 필요한 변수는 다음과 같이 7개만으로 가능해진다. 즉, 열교환기 출구 온도 $T_k(y_1)$, 반응기의 출구 온도 $T(y_2)$, PI 제어기의 출력 $s_i (y_3)$, 반응기 출구 물질 농도 1 (y_4), 반응기 출구 물질 농도 2 (y_5), 반응기 입구 물질 농도 (y_6), 반응기 입구 물질 농도 (y_7), 반응기 입구 온도 $T_i (y_9)$ 으로 정의한다.

본 연구에서는 최소의 측정으로 최소의 오분류를 가진 고장을 진단하는 것이다. 처음에는 문제속에 포함된 모든 변수를 전부 선택했다. 그러나 측정의 전처리에 의하여 네트워크의 입력을 최소로 할 필요가 있다. 측정을 적당하게 조절하

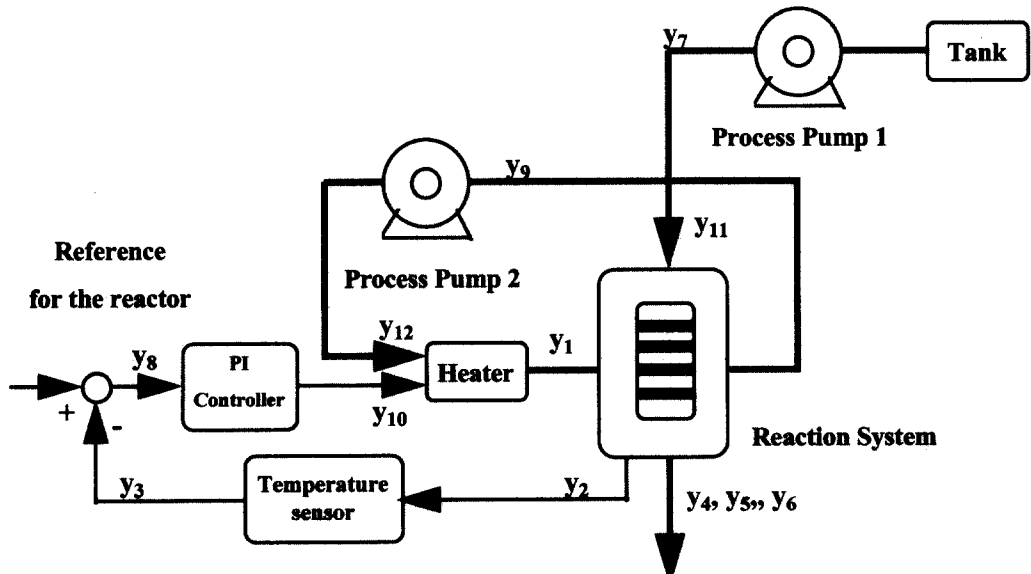


Fig. 6. The diagnosed object system

Table 1. The learning data for the single fault

Fault	Input Patterns							Output Patterns						
	y ₁	y ₂	y ₃	y ₄	y ₅	y ₇	y ₉	1	2	3	4	5	6	7
Norm	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
F ₁	-1.85	-0.74	-0.23	2.78	-2.78	0.00	0.00	1	0	0	0	0	0	0
F ₂	-0.83	0.00	-0.17	2.52	-2.52	0.00	0.00	0	1	0	0	0	0	0
F ₃	0.66	0.00	0.13	-2.03	2.03	0.00	0.00	0	0	1	0	0	0	0
F ₄	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0	0	0	1	0	0	0
F ₅	3.04	0.00	0.61	0.00	0.00	0.00	0.00	0	0	0	0	1	0	0
F ₆	-0.86	0.00	-0.17	-2.38	-2.59	-5.00	0.00	0	0	0	0	0	1	0
F ₇	0.43	0.00	0.09	0.00	0.00	0.00	3.00	0	0	0	0	0	0	1

Table 2. The learning data included the F₁ fault component

Fault	Input Patterns							Output Patterns						
	y ₁	y ₂	y ₃	y ₄	y ₅	y ₇	y ₉	1	2	3	4	5	6	7
F ₁	-1.85	-0.74	-0.23	2.78	-2.78	0.00	0.00	1	0	0	0	0	0	0
F ₁ F ₂	-2.68	-0.74	0.40	5.30	-5.30	0.00	0.00	1	1	0	0	0	0	0
F ₁ F ₃	-1.19	-0.74	-0.10	0.75	-0.75	0.00	0.00	1	0	1	0	0	0	0
F ₁ F ₄	-1.85	-0.74	0.68	2.78	-2.78	0.00	0.00	1	0	0	1	0	0	0
F ₁ F ₅	1.19	-0.74	0.38	2.78	-2.78	0.00	0.00	1	0	0	0	1	0	0
F ₁ F ₆	-2.71	-0.74	-0.40	0.40	-5.37	-5.00	0.00	1	0	0	0	0	1	0
F ₁ F ₇	-1.42	-0.74	-0.14	2.78	-2.78	0.00	-3.00	1	0	0	0	0	0	1

기 위하여 프로세스 측정값의 모든 오차 Δy_i 는 측정값의 2%로 하여 정규화된다. 변화계수는 2%이기 때문에 몇몇 노이즈가 더해진다면 분류는 올바르게 수행된다. Table 1에서 학습데이터는 프로세스 측정의 정규화와 데이터 축소(reduction)에 의하여 얻어진다. 이와 같은 데이터는 Net⁰를 사용된다. 고장 F₁이 존재할 때, 출력은 {1 0 0 0 0 0}이다. 7개의 고장에 대응한 7 학습하는 대개의 네트워크 Netⁱ(i=1,2,3,...,7)가 두 번째 레벨에 존재한다. Netⁱ를 위한 학습데이터는 앞에서 설명했듯이 두개의 고장을 위하여 Table 1에 더함으로써 얻을 수 있다. Table 2에서 F₁을 포함하는 모든 이중고장의 경우를 조사하자. 예를 들면 {F₁,F₂}를 위한 학습데이터는 Table 1에서 2행(F₁에 대응하는 고장)데이터를 3행(F₂에 대응고장)에 더함으로써 발생되고 출력은 {1 1 0 0 0 0}이 된다. 본 연구에서 사용된 신경망의 특성은 각 계층별 신경망의 입력은 7개, 은닉층은 30개, 출력층은 7개이고, $\eta = 0.1$, $\alpha = 0.9$ 로 세팅하고, 반복은 최소에러가 0.008보

다 적을 때까지 계속된다. 요구되는 반복수 평균은 6800이고, 학습패턴의 수는 29개였다 (normal case 1, single faults 7, double faults 21). 비록 학습은 단일, 이중고장에 기초하고 있지만, 테스트는 단일·이중·삼중고장들에 의하여 수행된다. 테스트할 패턴수는 정상 1개, 단일 7개, 이중 21개, 삼중 35개로 총 64개이다. output layer 속에 first level의 노드출력이 0.5보다 클 때 징후가 존재하는 것으로 고려하고, second level에서 출력이 0.9보다 큰 경우에 아이템 고장이 존재하는 것으로 하고, third level에서는 노드출력이 0.5 이상이면 component속에 고장징후가 존재하고, forth level에서는 출력노드가 0.9 이상이면 component에 고장이 존재하는 것으로 한다.

3.4. 고장진단 결과 및 고찰

본 연구에서는 단일고장과 이중고장에 대해서만 학습을 수행하고, 단일고장과 이중고장, 그리고 삼중고장에 대해서 진단을 수행했다. 테스트될 패턴의 수는 정상상태는 한가지뿐이고, 단일

고장 7개, 이중고장 21개, 삼중고장 35개로 총 64개 상태이다. 고장 중에 감지될 고장요소의 개수를 조합으로 계산 (${}_{7}C_i, i=1,2,3$)해 낸 것이다.

본 연구에서는 첫 번째 단계의 출력층의 노드 값이 0.5보다 클 때, 그리고 두 번째 단계의 출력층 노드값이 0.9보다 클 때 고장이 존재한다고 가정했다. Table 3은 삼중고장에 대한 진단의 결과를 보여주고 있다. 이 삼중고장의 진단은 다중고장진단의 시작점으로 볼 수 있다.

이렇게 수집된 진단의 결과를 토대로 다음과 같이 진단성능을 말할 수 있다. 단일고장에서 7가지의 고장요소를 각각 포함하고 있는 입력데이터에 대해 모두(100%)감지할 수 있었고, 3중고장에서는 35가지 입력 데이터 중에 76% ($\frac{59}{(35)(3)}$)의 고장인식률을 가지고 있다. Table 3에서 이탤릭체로 진하게 표시된 F_2 는 모델링에

러나 상관관계함수의 오용 등으로 기인한 오분류이고, Impossible로 표시된 것은 존재하는 고장요소를 감지하지 못한 경우이다. 그러나 기존의 방법은 하나의 신경망에 여러 개의 단일고장, 이중고장, 삼중고장데이터를 동시에 입력시키고, 고장진단을 하기 때문에 고장진단효율이 매우 저조하고 또한 오진의 확률이 높다. Table 3에서 전통적인 하나의 단위 신경망을 이용하여 고장진단했을 때 고장결과와 MANN을 이용한 고장결과를 나타낸 것이다. 기존의 신경망에 1중, 2중 고장데이터를 학습시킨 후 3중고장 데이터를 35가지 입력했을 때, 출력노드값이 0.5 이상 정확성을 가지고 식별할 수 있는 3중고장 가지수는 0(0/35)인 반면, MANN에 의한 진단에서는 출력노드값이 0.9 이상의 정확성을 가지고 식별할 수 있는 3중고장 가지수는 12가지(12/35)이다. 또한 출력 노드값이 0.9보다 클 경우 기존의 신경

Table 3. Comparison diagnosis results of triple faults by traditional neural network and MANN

Fault	Traditional Mono Neural Network	After 1st Net	After 2nd Stage	After OR Operation	Classification*	Traditional Mono Neural Network Classification
$F_1F_2F_3$	F_1	F_1	F_1	F_1	1/3	1/3
$F_1F_2F_4$	F_1	F_1	F_1	F_1F_4	2/3	1/3
$F_1F_2F_5$	F_2F_4			Impossible	0/3	1/3
$F_1F_2F_6$	F_1	F_1	F_1F_2	F_1F_2	2/3	1/3
$F_1F_2F_7$	F_1F_7	F_1F_7	F_1F_7	F_1F_7	2/3	2/3
$F_1F_3F_4$	F_1F_4	F_4	F_4	F_4	1/3	2/3
$F_1F_3F_5$	F_2F_5	F_5	F_5	F_5	1/3	1/3
$F_1F_3F_6$	F_6	F_6	F_6	F_6	1/3	1/3
$F_1F_3F_7$	F_1F_7	F_1	F_1F_7	F_1F_7	2/3	2/3
$F_1F_4F_5$	F_4	F_4	F_2F_4	F_2F_4	2/3	1/3
$F_1F_4F_6$	F_1	F_6	F_1F_6	F_1F_6	2/3	1/3
$F_1F_4F_7$	F_1F_7	F_1	$F_1F_4F_7$	$F_1F_4F_7$	3/3	2/3
$F_1F_5F_6$	F_5			Impossible	0/3	1/3
$F_1F_5F_7$	F_7	F_2F_7	$F_2F_5 \cup F_2F_4F_7$	$F_2F_5F_7$	2/3	1/3
$F_1F_6F_7$	F_6	$F_1F_6F_7$	$F_1F_6F_7 \cup F_1F_6F_7 \cup F_1F_6F_7$	$F_1F_6F_7$	3/3	1/3
$F_2F_3F_4$	F_4	F_4	F_4	F_4	1/3	1/3
$F_2F_3F_5$	F_4F_5	F_5	F_5	F_5	1/3	1/3
$F_2F_3F_6$	F_6	F_6	F_6	F_6	1/3	1/3
$F_2F_3F_7$	F_7	F_7	F_7	F_7	1/3	1/3
$F_2F_4F_5$	F_2	$F_2F_4F_5$	$F_2F_4F_5 \cup F_2F_4F_5 \cup F_2F_4F_5$	$F_2F_4F_5$	3/3	1/3
$F_2F_4F_6$	F_6	F_4F_6	$F_4F_6 \cup F_4F_6$	F_4F_6	2/3	1/3
$F_2F_4F_7$	F_7	F_2F_4	$F_2F_4F_7 \cup F_2F_4F_7$	$F_2F_4F_7$	3/3	1/3
$F_2F_5F_6$	F_5	F_5	F_5F_6	F_5F_6	2/3	1/3
$F_2F_5F_7$	F_2F_7	F_2F_5	$F_2F_5F_7 \cup F_2F_5F_7$	$F_2F_5F_7$	3/3	2/3
$F_2F_6F_7$	F_6	F_6F_7	$F_2F_6F_7 \cup F_6F_7$	$F_2F_6F_7$	3/3	1/3
$F_3F_4F_5$	F_3F_5	$F_3F_4F_5$	$F_3F_4F_5 \cup F_3F_4F_5 \cup F_3F_4F_5$	$F_3F_4F_5$	3/3	2/3
$F_3F_4F_6$	Impossible	F_4F_6	$F_4F_6 \cup F_4F_6$	F_4F_6	3/3	0/3
$F_3F_4F_7$	F_3F_4	F_3F_4	$F_3F_4 \cup F_3F_4$	F_3F_4	2/3	2/3
$F_3F_5F_6$	F_5	F_5	F_5F_6	F_5F_6	2/3	1/3
$F_3F_5F_7$	F_3F_7	$F_3F_5F_7$	$F_3F_5F_7 \cup F_3F_5F_7 \cup F_3F_5F_7$	$F_3F_5F_7$	3/3	2/3
$F_3F_6F_7$	F_6F_7	F_7	F_6F_7	F_6F_7	2/3	2/3
$F_4F_5F_6$	F_5	F_4F_5	$F_4F_5 \cup F_4F_5F_6$	$F_4F_5F_6$	3/3	1/3
$F_4F_5F_7$	F_3F_7	F_4F_5	$F_4F_5F_7 \cup F_4F_5F_7$	$F_4F_5F_7$	3/3	2/3
$F_4F_6F_7$	F_6	F_4F_6	$F_4F_6F_7 \cup F_4F_6F_7$	$F_4F_6F_7$	3/3	1/3
$F_5F_6F_7$	F_5F_7	F_5F_7	$F_5F_6F_7 \cup F_5F_7$	F_5F_7	2/3	2/3

* Number of faults diagnosed correctly / number of faults occurring.
 F_2 = misclassification

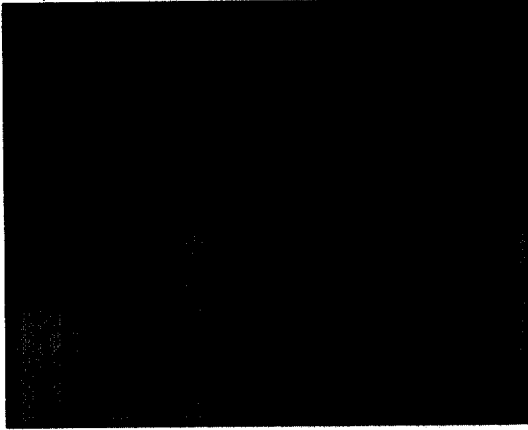


Fig. 7. The visualization of a implemented MANN

망은 35가지 3중고장 입력데이터 중에 43.8% ($\frac{46}{(35)(3)}$)의 고장인식을 할 수 있었다. 결과에서 알 수 있듯이 기존의 신경망에 대하여 MANN은 고장진단율이 1.7배 정도의 정확도를 가지고 있다. 또한 기존의 신경망은 삼중고장에 대하여 고장진단에서 부분적인 고장요소는 찾을 수 있지만 정확한 삼중진단은 거의 불가능함을 알 수 있다. 반면 MANN은 12가지의 3중고장을 정확히 인식할 수 있었다. 이것은 기존의 하나의 단위 신경망을 이용하여 고장진단을 할 경우 단일, 이중, 삼중고장을 한꺼번에 학습시킴으로써 신경망으로 하여금 고장인식률을 매우 떨어뜨리게 하는 반면, 본 연구에서 MANN에 의한 학습방법은 각각에 맞는 고장을 선별하여 학습함으로써 다중고장인식률을 높일 수 있었다. Fig. 7은 워크스테이션의 모니터상에서 MANN의 진단과정수행의 일부를 보여주는 사진이다.

4. 결론

본 연구에서는 동시 다발적으로 발생하는 다중고장을 진단하기 위하여 사용될 수 있는 유닉스 운영체제하에서 멀티프로세싱으로 IPC설비를 이용하여 MANN이라 명명한 새로운 계층형 인공신경망 형태의 거시구조를 제안했다. MANN은 신경망의 많은 수의 패턴을 학습과 고장분류가 좀더 효율적으로 수행될 수 있도록 매우 많고 훨씬 작은 서브들로 분류한다. PI 피이드백 제어반응기의 MANN을 이용한 고장진단의 예로

부터 결론은 다음과 같다.

첫째, 기존의 신경망이 단일 및 이중, 삼중고장에 대한 데이터를 모두 학습하는 반면, 본 연구에서 제시된 MANN은 각각에 맞는 고장을 선별하여 학습함으로써 다중고장인식률을 높일 수 있다.

둘째, MANN은 신경망을 계층화하여 표현함으로써 각 프로세스는 독립적으로 학습될 수 있고, 다음에 수정할 때 필요한 부분만 수정함으로써 개선이 가능하다.

셋째, 본 연구의 MANN은 전문가 시스템이나 다른 모델베이스 고장진단과 연결운용이 가능할 것으로 사료된다.

감사의 글 : 본 연구는 1998년도 안동대학교 학술연구조성비 지원에 의하여 연구되었습니다. 이에 관계자 여러분들께 감사드립니다.

참고문헌

- 1) Iserman, R., "Process Fault Diagnosis based on Dynamic Models And Parameter Estimation Methods", *Fault Diagnosis in Dynamic Systems : Theory and Applications*, Prentice Hall, pp. 253~291, 1989.
- 2) Timo, S., Seikki, N. K., Hannu, K., "Neural Networks in Process Fault Diagnosis", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 4, July/August, pp. 815~825, 1991.
- 3) Decker, D. K., "Some Proposed Applications of Artificial Neural Systems to Spacecraft Power Subsystem", *Proc. IJCNN*, San Diego, pp. 219~229, 1989.
- 4) Dietz, W. E., Kiech, E. L., Ali, M., "Jet and Rocket Engine Fault Diagnosis in Real Time", *J. Neural Network Comput.*, pp. 5~14, 1989.
- 5) McDuff, R. J., and P. K. Simpson, "An Adaptive Resonance Diagnostic System", *J. Neural Network Comput.*, No. 2, pp. 19~29, 1990.
- 6) Marks, K., "Neural Network Applications in the Automotive Industry : Diagnosis and Control of Vehicle Systems", *Conf. on Neural Network Application for Manufacturing*, 1991.
- 7) 홍릉과학출판사 편집부, UNIX 시스템 프로그래밍, 홍릉과학출판사, pp. 238~247, 1991.
- 8) 원유현, 백정현, UNIX 시스템 프로그래밍, 정익사, pp. 317~340, 1995.