

# ARMA 모형선정을 위한 통합된 신경망 시스템의 설계

지 원 철\* 송 성 현\*

## Design of An Integrated Neural Network System for ARMA Model Identification

*In this paper, our concern is the artificial neural network-based pattern classification, when can resolve the difficulties in the Autoregressive Moving Average (ARMA) model identification problem.*

*To effectively classify a time series into an appropriate ARMA model, we adopt the Multi-layered Backpropagation Network (MLBPN) as a pattern classifier, and Extended Sample Autocorrelation Function (ESACF) as a feature extractor.*

*To improve the classification power of MLBPN's, we suggest an integrated neural network system which consists of an AR Network and many small-sized MA Networks. The output of AR Network which will give the MA order.*

*A step-by-step training strategy is also suggested so that the learned MLBPN's can effectively ESACF patterns contaminated by the high level of noises. The experiment with the artificially generated test data and real world data showed the promising results.*

*Our approach, combined with a statistical parameter estimation method, will provide a way to the automation of ARMA modeling.*

### I. 서 론

예측기능이 포함되지 않은 경영활동은 생각할 수 없으므로, 효율적 경영계획의 수립 및

관리를 위해서는, 예측활동에 투입될 시간 및 비용과 예측결과의 정확성을 고려해 적절한 예측기법을 선택하는 것이 중요하다. 시계열분석은 예측의 주요 수단으로 인식되어 왔으며, 특

---

\* 홍익대학교 산업공학과

히, Box-Jenkins (B-J) (1976) 에 의해 체계화 된 자기회귀 이동평균 (ARMA) 모형은 통계이론에 의해 잘 뒷받침되어 있으며, 예측의 정확도에 있어서도 우수성을 인정받고 있다.

하지만 ARMA 모형의 수립에는 많은 시간과 노력이 요구되므로, 수백 혹은 수천 개의 시계열 자료로부터 제한된 시간내에 필요한 예측치를 얻어야하는 생산 재고 시스템과 같은 경영환경하에서는 실제 적용이 잘 이루어지지 못하고 있다.

만약 우리가 지수평활법과 같이 컴퓨터에 의한 완전 자동화가 가능한 ARMA 모형 수립 방법론을 개발한다면 변화가 심한 경영환경하에서의 예측에 있어 ARMA 모형의 유용성은 크게 개선될 것이다 (Montgomery & Johnson, 1976; Wheelwright & Makridakis, 1985).

시계열  $Z_t$ 에 대한 ARMA(p,q) 모형은 다음과 같이 표시된다.

$$\begin{aligned} \phi(B) Z_t &= C + \theta(B) a_t, \\ \phi(B) &= U(B) \Phi(B) = 1 - \phi_1(B) - \dots - \phi_p B^p, \quad U(B) = 1 - U_1 B - \dots - U_d B^d, \\ \Phi(B) &= 1 - \phi_1 B - \dots - \phi_{p-d} B^{p-d}, \quad \theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q \end{aligned}$$

$\theta_1 B - \theta_q B^q$ . B는 Backshift operator로서  $B^d Z_t = Z_{t-d}$ 을 의미하며,  $a_t$ 는 평균이 0이고 분산이  $\sigma^2$ 인 Gaussian white noise이다. 만약  $U(B) = (1-B)^d$ 이면 ARMA(p,q)는 B-J모형의 ARIMA(p-d, d,q)로 표시될 수 있다.

전형적인 ARMA 모형의 수립과정은 그림 1-A에서 볼 수 있듯이 모형선정, 모수추정 및 모형검증의 세단계를 반복적으로 거쳐 이루어진다.

이들 중 자기회귀 (AR)의 차수와 이동평균 (MA)의 차수, 즉 (p,q)를 결정하는 모형선정 단계는 매우 중요하다. B-J의 모형선정 단계에서는 ACF와 PACF를 사용하는데, 이 통계량들이 제공하는 정보량이 극히 제한적이어서, 모형수립 과정은 반복적 절차를 거치지 않을 수 없으며, 만족할 만한 ARMA 모형을 얻기 위해서는 잘 훈련된 전문가의 판단이 필수적이고, 많은 시간과 노력이 요구된다.

이러한 단점을 극복하고 보다 직접적인 ARMA 모형화가 가능하도록 하기 위해 많은 통계학적 접근들이 있었다.

첫째, 모형 선정의 어려움을 인정하고 모수추

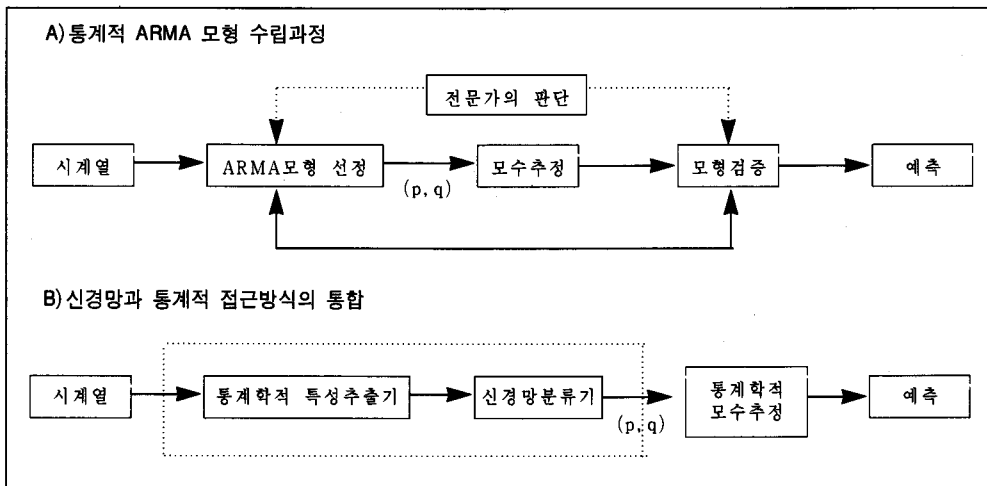


그림 1. ARMA 모형수립에 있어 통계적 방법과 신경망이 통합된 접근방법의 비교

정 후 모형의 적합도 검증에 치중하면서, 효율적인 모형 선정 전략을 모색하는 연구 (Akaike, 1974; Pandit & Wu, 1983; Parzen, 1982) 와 둘째, AFC와 PACF를 대체할 수 있는 새로운 통계량의 개발에 관한 연구(Benguin et al, 1980; Tsay & tiao, 1984; Woodward & Gray, 1981)로 크게 분류할 수 있다.

또 B-J의 방법과 Akaike's Information Criterion (Akaike, 1974)을 함께 사용하여 ARMA 모형 수립과정을 자동화하려는 노력도 있었다 (Hill & Woodworth, 1980; Kang et al, 1982). 하지만 ARMA 모형 수립에 여전히 많은 시간이 요구되고 있으며, ACF와 PACF를 대체한 통계량들도 불완전하여 전문가의 판단을 완전히 배제하기 어려워, 자동화된 ARMA 모형의 수립 방법으로 사용되기에는 부족하다 (Harvey, 1984). 최근 ARMA 모형수립과정에 패턴 인식 개념을 도입하여 의사결정지원시스템을 구성하려는 시도가 있었다 (Lee & Park, 1988).

본 연구에서는 통계적 접근방식이 갖는 한계를 극복하고 ARMA모형수립을 자동화하여 사용가능성을 높이기 위해 신경망과 통계적 접근방식의 통합사용을 제안한다. 즉 그림 1-A와 같은 반복적인 모형수립과정을 그림 1-B와 같이 전문가의 판단을 배제시킨 보다 직접적인 과정으로 개선하고자 한다.

최근 인공지능의 한 분야로써 새로이 재인식되고 있는 신경망은 과거 어려운 문제로 생각되었던 음성인식이나 자연어 처리등에 새로운 패턴인식이나 분류방법으로 사용되어 좋은 성과를 얻고 있다.

인간이 비교적 쉽게 해내는 문자등의 인식을 컴퓨터가 인지하기 어려운 것은, 주어진 입력에 대응하는 출력을 수학적 모형에 의해 구할 때, 입출력사이의 연관관계를 간단한 수식에

의해 표현하기 어렵기 때문이다.

신경망은 인간의 두뇌 신경 조직을 근간으로 하여 이러한 문제를 해결하고자 하는 것이다. 따라서, ARMA 모형 선정문제를 패턴분류 (Pattern Classification)문제로 생각하고, 신경망을 시계열 자료로부터 얻어지는 통계량에 의해 구성된 패턴들을 특정 ARMA 모형에 대응시키는 분류수단으로 사용함으로써, 모형수립과정에 필수불가결했던 전문가의 판단을 대체하고자 한다.

따라서, 본 연구에서는 신경망의 입력으로 사용될 패턴을 얻기위한 특성추출기 (feature extractor)로서 확장된 표본자기회귀함수 (ESACF)라는 통계량을 사용했고, 채택된 신경망 모형은 다층 Perceptron이며 학습은 Backpropagation 방법을 이용하였다. 앞으로 사용된 신경망 모형을 편의상 MLBPN (Multi-layered Backpropagation Network)이라 부른다.

본 논문의 나머지 부분은 다음과 같이 구성되었다. 장에서는 신경망에 대한 일반적 이론을 소개하고, 신경망을 이용한 시계열분석의 기존 연구와 본 연구를 비교한다. 장에서는 특성추출기로 사용될 ESACF에 대한 간단한 소개와 문제점을 제시한다.

장에서는 패턴분류기로서의 신경망 설계에 있어 세가지 대안을 설명한 후, 효과적인 ARMA 모형선정을 위해, ESACF 패턴에 대한 전처리 (Preprocessing) 과정 및 MLBPN의 설계 및 구현을 설명한다.

특히, MLBPN이 갖는 한계를 극복하기 위해 작은 규모의 신경망들을 효과적으로 결합하는 방안을 제시한다. 장에서는 실험을 통하여 학습된 신경망들의 일반화 정도를 중점적으로 정리함으로써 설계된 MLBPN의 유용성을 보인다. 장에서는 본 논문의 내용을 정리하고 앞으

로의 연구방향을 기술한다.

## II. 신경망

### 1. 신경망 이론의 개요

신경망은 인간의 두뇌를 모방한 수학적 모형으로써, 간단한 계산만을 수행하는 처리기 (Processing Element, PE)의 집합으로 구성되는데, 이 PE들은 몇개의 계층(Layer)으로 조직될 수 있으며, 각 계층의 PE들은 상위계층의 PE들에 연결된다. 각 연결부위의 강도를 나타내는 가중치를 조정함으로써 패턴의 인식이나 필요한 지식의 획득이 이루어진다. 즉 많은수의 PE와 가중치 조정을 통해 인간과 같은 일반화나 패턴분류 능력을 가지게 된다.

신경망의 작동과정은 재생(Recall) 및 학습(Learning)의 두 단계로 나눌 수 있다 (Cruz, 1988; Pao, 1988). 재생단계는 주어진 입력패턴에 대한 신경망의 반응과정으로,  $h+1$ 번째 계층에 있는 임의의 PE,  $U_i(h+1)$ , 은 자신에 연결된 하위계층 PE들로부터의 출력  $a_j(h)$ 을 입력으로 받아 가중치  $W_{ij}$ 를 곱하여 합을 구한 후, 사전에 정의된 전이함수 (Activation Function or Transfer Function)  $f$ 를 사용하여 자신의 출력  $a_i(h+1)$ 을 내보낸다. 전형적인 feedforward신경망에서 이 과정은 다음 식에 의해 표현된다. (그림 2 참조)

$$U_i(h+1) = \sum_{j=1}^{N(h)} W_{ij}(h+1)a_j(h),$$

$$a_i(h+1) = f[U_i(h+1), \theta_i(h+1)], 1 \leq i \leq N(h), 0 \leq h \leq L-1.$$

$L-1$ 은 입력층을 제외한 계층의 수,  $N(h)$ 은 각 계층의 PE 개수를 나타내며,  $\theta_i$ 는 PE 고유

의 편차(bias)를 의미하고, 전이함수로는 그림 3에서와 같은 비선형 함수들이 주로 사용된다. 학습단계는 신경망이 주어진 입력패턴에 대해 원하는 반응을 보일 수 있도록, 재생단계에 의해 얻은 출력값들과 원하는 출력값사이의 오차를 줄이기 위해, 선택된 학습방법에 따라 가중치를 조정하는 것으로 다음과 같은 순환식에 의해 표현된다.

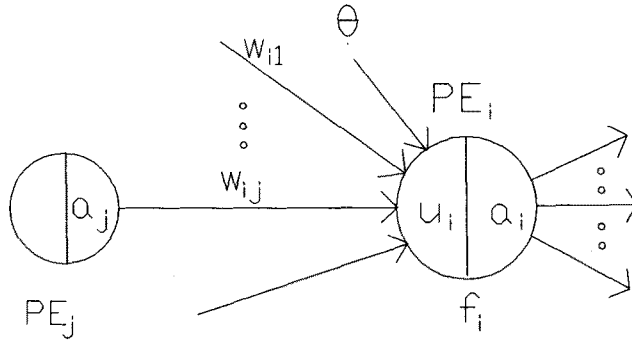
$$W_{ij}(l) \leftarrow W_{ij}(l) + \eta_j(l)\Delta W_{ij}(l)$$

즉, 새로운 가중치는 계산된 가중치의 변화량에 학습률을 곱하여 기존의 가중치에 더해 줌으로써 얻어진다. 여기서 학습률은 고정되거나 필요에 따라 변화될 수 있다 (Jacobs, 1988).

단순한 기능의 PE를 대규모로 연결시킨 구조를 갖는 신경망은 학습능력, 분산지식 표현, 병렬처리 능력 및 결합극복 능력 등의 특성을 갖고 있다.

현재 다양한 신경망 모형들이 개발되었는데 PE들간의 연결형태 즉 망의 구조와 학습 알고리즘의 차이에 따라 다양하게 분류되며, Hopfield Network, ART, Boltzmann Machine, 다층 Perceptron등이 대표적이다 (Anderson & Rosenfeld, 1988; Cruz, 1988; DARPA, 1987; Hopfield & Tank, 1985; Rumelhart et al, 1986). Rosenblatt (1962)의 Perceptron으로 대표되던 신경망 분야의 연구는 Minsky와 Papert (1969)에 의해 단일 계층 Perceptron은 선형함수에 의해 분리가능한 문제들만이 해결 가능하다는 것이 증명된 후 매우 침체되었다. 최근 PDP Group (Rumelhart et al, 1986)에 의해 다층 Perceptron의 학습방법 (Backpropagation 또는 Generalized Delta Rule)이 재정립되고 유용성이 입증됨으로써 다시 주목을 받게 되었다. 특히 은닉층의 사용은 입력 패턴에 대한 복잡한 분석을

처리기(Processing Element)  
의 구조



신경망의 일반적 구조

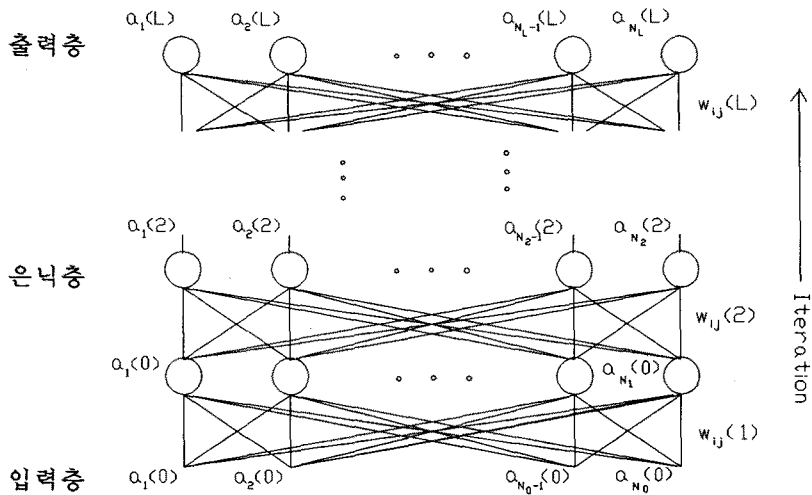


그림 2. 신경망의 기본구조

가능케 함으로 선형함수에 의해 분리될 수 없는 문제에 대해서도 좋은 결과를 기대케 한다. 각 신경망 모형 특히 MLBPN의 패턴분류기로서의 역할에 대해서는 Lippman의 논문 [1988]을 참조하기 바란다.

2. 신경망을 이용한 시계열 분석의 기존 연구

Lapedes와 Farber [1987]는 Glass-Mackey 비선형 미분방정식에 의해 생성된 시계열을 모형화하는 과정에서 MLBPN을 사용하여 좋은 결과

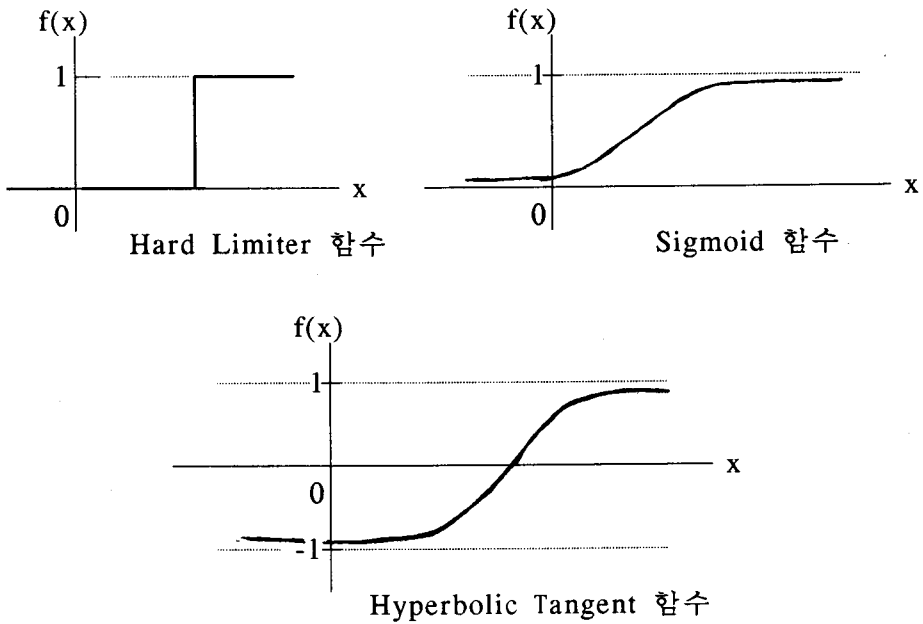


그림 3. 여러형태의 전이함수들

를 얻음으로써 신경망이 효과적인 예측수단이 될 수 있음을 보였다.

White [1988]는 주가 예측에 있어 신경망을 다중회귀분석모형처럼 사용한 연구를 보고했다. 이 연구결과들은 특정 시계열에 대해 좋은 예측치를 얻는 방법으로 유효하지만, 신경망을 학습시키는데 오랜 시간이 걸리는 점을 고려하면, 많은 수의 시계열 자료들을 제한된 시간 내에 처리해야 하는 생산-재고 시스템같은 경영 환경하에서의 접근방법으로는 적당하지 않다.

본 연구에서는 MLBPN을 시계열 생성시스템의 모형화 수단으로 사용하는 것이 아니라, ARMA 모형화 과정에서 사용되는 통계량을 정확히 해석해주는 패턴분류기로서 사용한다.

이러한 접근 방식은 Gorman & Sejnowski [1988]의 수증음파처리기에 의한 목표물분류 연구와 유사하다.

비록, MLBPN의 사전훈련에 상당한 시간과 노력이 요구되지만, 일단 학습된 MLBPN을 이용한 ARMA 모형선정은 극히 짧은 시간 내에 이루어지므로 이미 잘 정립된 통계학의 모수추정방법과 통합하여 사용하는 경우, 제한된 시간 내에 많은 수의 시계열을 전문가의 개입없이 ARMA 모형화하여 예측치를 자동으로 얻을 수 있다. 따라서 앞으로의 토의는 ARMA 모형선정에 효과적인 신경망 패턴분류기의 설계 및 구현에만 중점을 둔다.

### Ⅲ. 특성추출기

패턴 분류는 분류대상 개체들로부터 얻은 입력패턴을 사전에 정의된 범주(Class)들의 하나에 대응시키는 매우 일반적인 정보처리의 한

형태로서 컴퓨터를 이용한 패턴분류작업은 다음의 세단계를 거친다 [Pao, 1988].

(1) 특성추출(feature extraction): 분류대상들을 적절한 표현수단을 사용하여 묘사하는 것으로, 각 class들의 특성을 잘 나타낼 수 있어야만 효과적인 분류작업이 가능해 진다.

(2) 대응함수(Mapping funtion)의 학습: 분류될 class들을 정의한 후, 특성추출에 의해 표현된 입력패턴과 요구되는 출력 class를 대응시켜서 구성된 학습패턴들을 사용하여, 의사결정규칙을 훈련을 통해 추천한다.

(3) 대응함수의 사용: 학습된 의사결정규칙을 이용하여 실제 분류작업을 수행한다.

ARMA모형선정을 패턴분류작업으로 간주하는 것은, 통계적 절차에 의해 얻어진 시계열의 특성치들을 입력패턴으로, 각 ARMA모형들을 분류될 class로 간주하고 MLBPN을 대응함수로서 사용함을 의미한다. 이 장에서는 특성추출기로서 사용될 ESACF에 대한 간략한 소개 및 ESACF에 대해 패턴분류개념을 적용시키는 이유를 설명한다.

Tsay와 Tiao [1984]에 의해 개발된 ESACF는 시계열이 안정적인 경우는 물론, 비안정한 경우에도 적용할 수 있는 ARMA모형 선정을 위한 통계량이다. 일반적인 시계열  $Z_t$ 에 대한  $k$ 번째 ESACF는 다음과 같이 정의된다.

$$r_1(k) = r_1[w_{k,t}^{(0)}], w_{k,t}^{(0)} = Z_t - \sum_{i=1}^k \hat{\phi}_{1(k)}^{(i)} Z_{t-i}$$

$\hat{\phi}_{1(k)}^{(i)}$  들은 AR차수가  $k$ 라는 가정하에 OLS 절차를  $k$ 번 반복시행하여 얻은 AR모수들에 대한 일치추정치이며,  $r_1(k)$  는 변형된 시계열  $w_{k,t}^{(0)}$  에 대한 시차  $j$ 에서의 표본자기회귀함수(Sample Autocorrelation Function)이다.

만약 어떤 시계열이 ARMA( $p, q$ )모형을 따른

다면, 변형된 시계열  $w_{p,t}^{(0)}$ 는 이론적으로 순수 이동평균모형인 MA( $q$ )를 따르게 되므로 B-J 방법론에서와 같이  $r_1(p), j > q$  는 0이 되는 Cut-off현상이 발생한다. 여기서 AR모형의 차수를  $k-p > 0$  로 가정할 경우 MA모형의 차수가 하나씩 증가한다. 이러한 현상을 이용하기 위해  $r_1(k)$  즉 ESACF 값들을 이차원표에 정리한 후, 0의 근사치들로 이루어진 역삼각형의 꼭지점 위치를 파악하여, AR과 MA의 차수를 결정한다.

| MA<br>AR | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---|---|---|---|---|---|---|---|---|---|
| 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1        | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2        | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3        | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4        | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5        | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 6        | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 7        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 9        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

그림 4. ARMA(1, 1)에 대한 이론적 ESACF표

본 연구에서는 ESACF표를 패턴분류에 이용하기 위해 이진값만을 갖도록 변형시킨다. 그림 4는 ARMA(1, 1)모형에 대한 이론적인 ESACF표를 나타낸 것으로, 각 ESACF값이 2배의 표준편차 내에 있을 경우에는 1을, 그 외는 0을 부여하여 변형시킨 것이다. ESACF를 이용한 ARMA 모형선정은 AR모수의 추정시 반복 OLS 절차의 각 단계에서 추정된 AR모수 사이의 순환관계를 이용함으로써 계산의 효율성이 높고, B-J 방법론과는 달리 비안정적 시계열에 대해서도 적용가능하므로 Overdifferencing의 문제가 발생되지 않는다는 장점이 있다.

하지만, ESACF도 완전한 방법은 아니다. B-J 방법보다는 직접적인 모형선정이 가능하지만 그림 4와 같은 원형적인 패턴(Prototypical

Pattern)은 실제로는 거의 존재하지 않는다. 특히 경영환경하의 시계열로부터 얻는 ESACF패턴들은 원인을 알 수 없는 많은 잡음(Noises)에 의해 오염되므로, 정확한 해석을 위해서는 여전히 전문가의 판단이 필요하다.

역삼각형의 경계에 위치한 ESACF 값들이 2배의 표준편차보다 약간 크거나 작을 경우, 시계열이 계절주기들을 내포하여 역삼각형이 여러 개 나타나는 경우, 정확한 해석이 어렵다. (그림 8,9의 (a) 참조) 이러한 사실은 자동화된 ARMA모형화를 위해서는 ESACF에 추가적인 의사결정수단을 적용해야함을 의미한다.

## IV. 통합된 신경망 시스템의 설계

### 1. 시스템의 구성

ARMA모형선정을 위한 MLBPN의 설계에 있어 다음과 같은 세가지 접근방식을 생각해 볼 수 있다.

#### 1.1. 대안 1

단순한 방법으로 그림 5와 같이 하나의 MLBPN을 이용하여, 모든 ARMA Class로부터의 ESACF패턴을 분류하는 것이다.

그림 5에서 사용된 입력패턴은 실제 화학공정으로 부터 얻은 시계열 자료 (Series A in Box & Jenkins, 1976)를 이용하여 ESACF 패턴을 구한 것이다. 이 시계열 자료는 ARMA(1,1)을 따르는 것으로 알려져 있는데 (Box & Jenkins,

1976; Tsay & Tiao, 1984) 얻어진 ESACF 패턴은 그림 4와 비교할 때 심한 잡음을 가지고 있음을 알 수 있다.

이 입력패턴에 대해 요구되는 출력 (Desired Output or Target Output) 은 7번째 출력PE만 1이고 나머지 출력PE들은 모두 0을 갖도록 하였다<sup>1)</sup>. 그림 5에 표현된 MLBPN은 두개의 은닉층을 가진 다계층Perceptron으로 fully-connected feedforward network이다.

이 경우, 하나의 MLBPN이 너무 많은 Class들을 취급해야 하므로 출력계층의 PE개수가 많아져, MLBPN내의 연결수가 기하급수적으로 증가하므로, MLBPN의 훈련에 오랜 시간을 요구하게 된다. 또, ESACF가 그림 4와 같이 분류하기 쉬운 원형(Prototype)을 가진다 해도 학습된 MLBPN을 현실의 시계열에 적용하기 위해서는 ESACF내의 많은 잡음에 대해서 정확히 반응해야 하는데, Class의 수가 많은 경우 학습 과정이 어렵고 학습이 되었다 하더라도 잡음의 간섭효과를 배제시키기 어려워 MLBPN이 훈련받지 않은 입력패턴들도 정확히 분류할 수 있는 일반화 능력을 갖추기 힘들다.

#### 1.2. 대안 2

하나의 MLBPN이 취급해야 할 Class의 수를 줄이기 위해 두 개의 MLBPN을 동시에 사용하는 것으로 ESACF의 특징을 고려한 것이다.

시계열로부터 ESACF표를 구하면, ARMA모형선정문제는 역삼각형의 꼭지점의 위치, 즉 꼭지점의 행과 열의 위치를 확인하는 작업으로 변화된다. 따라서 AR의 차수를 나타내는 열과

1) 7번째 출력 PE만 1을 가진 이유는 본 연구에서 ARMA(5,5)까지 고려하고 각 PE출력은 하나의 ARMA모형을 나타내도록 했기 때문에 35개의 출력 PE가 있게되고, 첫번째 출력 PE는 ARMA(0,1)를 두번째 출력PE는 ARMA(0,2)에 대응되는 형태로 원하는 출력값들을 구성했기 때문이다.



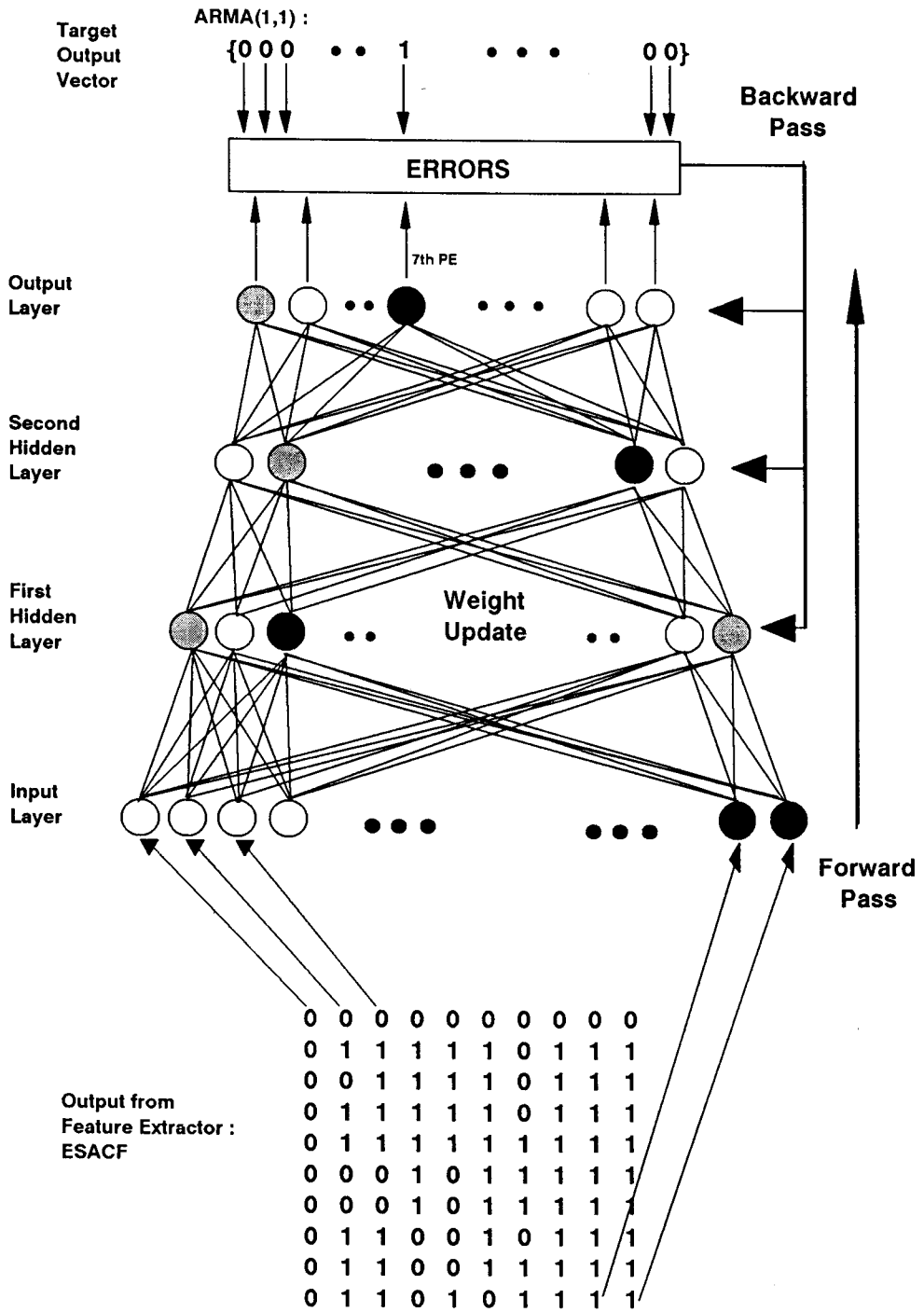


그림 5. 두개의 은닉층을 가진 MLBPN에 의한 ESACF 패턴의 분류

MA의 차수를 나타내는 행을 파악하는 작업은 별도의 두 작업으로 파악하여도 상관없다.

따라서 두 개의 MLBPN, 즉 AR의 차수를 파악하는 AR Network과 MA의 차수를 파악하는 MA Network, 즉 두개의 MLBPN으로 구성된 시스템을 생각할 수 있다.

이 경우 각 MLBPN이 취급해야 할 Class의 수는 크게 줄어든다.

예를 들어 ARMA(p,q)까지를 고려한다면, 하나의 MLBPN을 사용할 경우에는 (p+1)(q+1)-1개의 Class를 취급해야 되지만, 두 개의 MLBPN을 사용할 경우에는 AR Network은 p+1개, MA Network은 q+1개의 Class만을 취급하면 된다. 따라서 각 MLBPN의 구조가 간단해지며 학습시간을 대폭 절감할 수 있다.

반면에 역삼각형의 꼭지점의 위치를 파악하려면 역삼각형의 경계를 이루는 열과 사선을 정확히 파악해야 하는데, AR의 차수는 열에 의해 파악이 되므로 문제가 되지 않지만, MA의 차수파악을 위한 행은 역삼각형의 사선이 어느 열과 만나느냐 하는 것에 따라 변화하게 된다.

따라서 MA Network의 경우 상당히 다른 종류의 역삼각형 경계에 대한 정보가 같은 Class로 분류되어 입력패턴의 잡음이 심한 경우 혼란된 MA Network의 일반화 능력이 손상받기 쉽다는 단점이 있다.

### 1.3. 대안 3

마지막으로, 본 논문에서 제안하는 접근방식은 두번째 접근방식의 단점을 보완하기 위해 MA Network을 다시 여러개로 분할하여 AR Network과 유기적으로 통합시키는 것이다.

즉 하나의 AR Network에 의해 먼저 AR의 차수를 파악한 후, 각 AR차수에 대응되는 적은

규모의 MA network을 여러 개 사용하여 MA차수를 파악하려는 것이다.

이러한 접근방식은 많은 MLBPN을 다루어야 한다는 불편함이 있지만, 각 MLBPN이 다루어야 할 Class들은 동질적인 입력패턴들 만으로 구성되므로 비교적 간단한 구조의 MLBPN으로도 잡음이 심한 입력패턴에 효과적으로 대응할 수 있다.

그림 6은 통합된 신경망 시스템의 구조를 나타낸다. 이하의 각절에서는 통합된 신경망 시스템의 각 부분에 대해 상세히 설명하겠다.

## 2. 1차 전처리 (Preprocessing)

ESACF를 이용하여 효과적인 ARMA모형선정을 하기 위해서는 고려중인 ARMA모형들의 AR 및 MA 최대차수들보다 상당히 큰 크기의 ESACF표를 사용해야 한다.

특히 계절주기를 포함한 시계열의 분석시에는 ESACF표의 크기가 매우 커질 수밖에 없고 이에 비례해 MLBPN의 입력노드 수도 커진다.

이 같은 부담을 줄이기위해 입력노드 갯수를 효과적으로 줄일 수 있는 전처리과정을 다음과 같이 도입한다.

ESACF표의 역삼각형 경계에 대한 정보가 중요하므로 다음과 같은 전처리를 한다.

즉 mn의 크기를 갖는 ESACF표상의 한 요소 값을  $C_{ij}$ 라 하면, 전처리를 거친 입력  $\rho_k$ 는 다음과 같이 표시된다.

$$\rho_k = \begin{cases} \sum_{j=0}^{n-1} C_{j/m}, & i = k-1, \text{ for } k = 0, 1, \dots, m-1, \\ \sum_{i=0}^{n-1} C_{m-i+1, j/m}, & i = 2m-1-k, \text{ for } k = m, m+1, \dots, 2m-1, \\ \sum_{i=0}^{n-1} C_{i, j+1/m}, & j = k-2m, \text{ for } k = 2m, \dots, 2m+n-1. \end{cases}$$

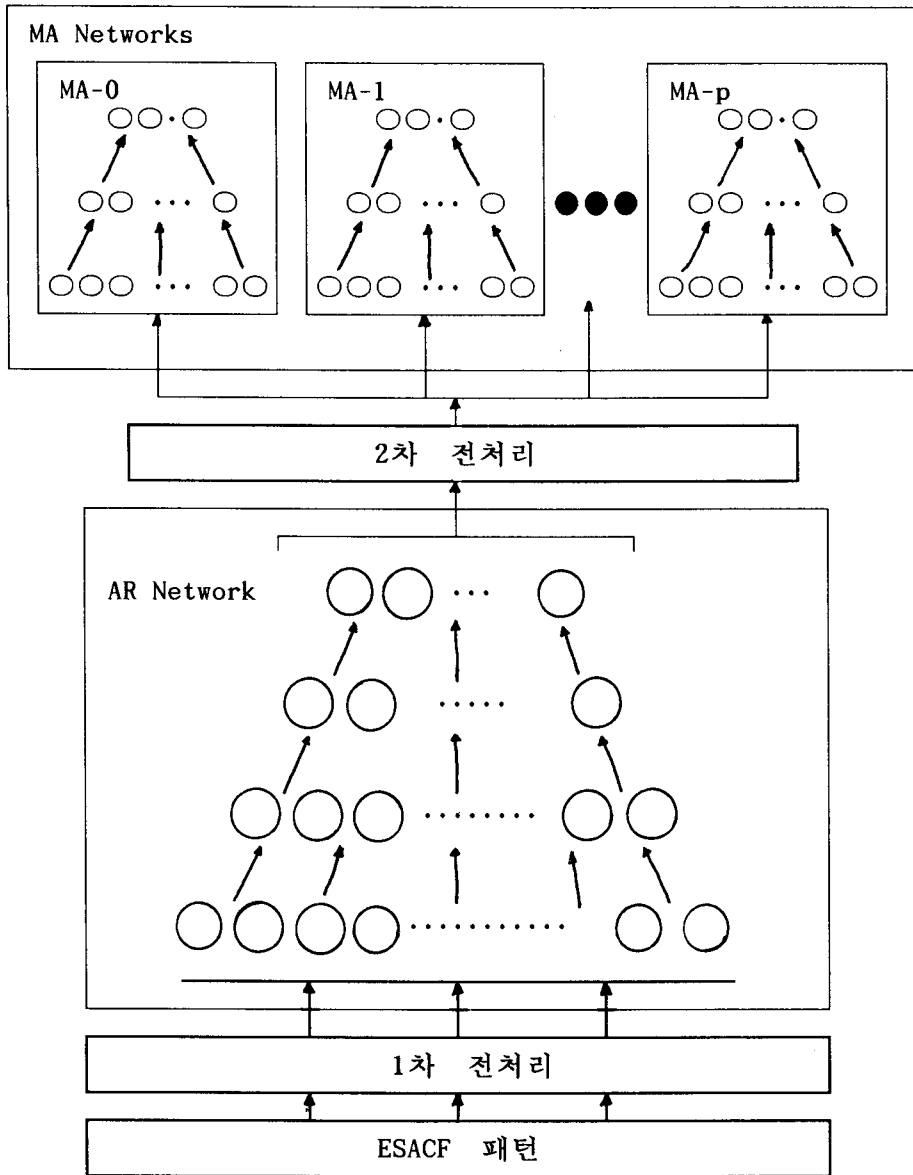


그림 6. ARMA 모형선정을 위한 통합 신경망 시스템의 구조

$\rho_k$ 를 얻기 위해 행 또는 열의 숫자로 나눈 것은 입력값들을 0과 1 사이에 존재시키기 위한 것이다.

이와 같은 전처리 과정을 거침으로써,  $mn$ 개의 입력노드 수를  $2m+n-1$ 로 감소시킬 수 있으며, ESACF표상의 역삼각형 위치에 대한 정보

또 큰 손상없이 유지시킬 수 있다.

### 3. AR Network

MLBPN의 패턴분류능력은 각층의 PE들 사이의 가중치들에 의해 결정된다.

따라서 해당 응용분야에 적합한 MLBPN의 구조를 결정하기 위해서는 PE들간의 연결상태 및 관계를 결정하는 은닉층의 수, 각 계층의 PE수 및 PE에 사용된 전이함수에 대한 적절한 고려를 해야한다. AR Network는 다음과 같다.

첫째, 은닉층의 수는 MLBPN의 설계에 있어 매우 중요하는데, 분류대상작업의 복잡성을 감안하여 결정해야 한다.

AR Network은 모든 입력 패턴들이 반드시 거쳐야 하므로 강력한 분류능력을 지녀야 하며, 시계열로부터 얻어진 ESACF패턴이 입력 노드수를 줄이기 위한 전처리 과정만을 거쳐서 입력되므로 심한 잡음에도 정확히 반응할 수 있어야 한다. 따라서 MLBPN의 분류능력을 충분히 활용하기 위해 두개의 은닉층을 사용한다.

둘째, 출력층의 PE개수는 만약 ARMA(p, q)까지 고려한다면, P+1개가 되며, 입력층의 노드수는 1차 전처리과정의 결과에 의해 결정된다.

은닉층의 PE갯수는 MLBPN의 분류능력에 큰 영향을 미치는데, 입력층보다 많은 PE의 사용은 적은 회수에 의한 학습을 가능하게 하지만, 경우에 따라서는 MLBPN은 학습패턴 내의 불필요한 잡음들까지도 기억함으로써 학습된 MLBPN의 일반화 능력이 저해되는 경우가 많다.

지나치게 적은 수의 PE는 학습과정에 있어 수렴의 속도가 늦어지거나, 수렴의 실패를 초래할 수도 있고, 훈련된 MLBPN의 분류능력도 좋지 못하게 한다.

AR Network의 두 은닉층 PE수는 입력층의 갯수로부터 서서히 줄여 나갔는데 이는 ESACF표에 내재된 정보를 서서히 축약시켜서 출력층의 한 PE의 값으로 변화시켜야 하기 때문이다.

여기서 두번째 은닉층의 PE수는 출력층의 PE갯수보다 충분히 크도록 하여, 효과적인 분류

작업이 이루어지도록 하였다 (Lippmann, 1988).

마지막으로 입력층을 제외한 모든 PE에서 Sigmoid 전이함수를 사용하였는데 이 함수는 분류작업에 있어 MLBPN을 이용한 대부분의 연구에서 사용되었으며 다음과 같이 표현된다.

$$f(\text{net}_i + \xi_i) = \frac{1}{1 + \exp(-(\text{net}_i + \xi_i) / \beta)}$$

$\xi_i$  는 threshold 또는 bias로 불리며,  $\beta$  는 sigmoid함수의 형태를 결정하는 상수로써 1로 정했다. 특정패턴에 대한 AR Network의 출력값은 모든 PE에서 0과 1사이의 실수값을 갖게 되는데, 각 출력 PE의 값은 주어진 패턴이 각 PE가 나타내는 AR class에 속할 수 있는 정도에 대한 추정치로 해석할 수 있다.

따라서 AR Network의 출력들 중 가장 큰 값을 갖는 PE가 나타내는 AR class를 따라 해당 MA Network에 입력패턴을 전달하여 MA의 차수를 분류하도록 한다.

## 4. 2차 전처리

MA Network에 입력패턴을 전달하기 전에 AR Network에 의해 파악된 정보를 충분히 이용하기 위해 1차 전처리 전의 입력 ESACF표를 이용하여 다시 한번 더 전처리를 한다.

일단 AR 차수가 결정되면, ESACF표상의 첫 번째 열에서부터 해당 AR 차수를 나타내는 열의 직전까지는 MA차수를 결정하는데 도움이 되지 않으며, 필요없는 잡음들만 있을 뿐이다.

따라서, ESACF표의 해당 부분 값들을 모두 0으로 바꾼 후 1차 전처리와 동일한 절차를 거쳐 MA Network의 입력패턴을 구한다.

## 5. MA Networks

만약 ARMA( $p, q$ )까지의 모형을 고려한다면, MA Networks는 그림 5에서와 같이 MA=0, MA=1..., MA=P로 구성된 P+1개의 서로 독립적인 MLBPN들을 갖게 되며, 하나의 MA Network은 q+1개의 출력 PE를 갖게 된다.

각 MA Network에 입력되는 패턴들은 앞서 설명한 바와 같이 class들 내의 동질성이 잘보장 되도록 하였으므로, 비교적 간단한 구조의 MLBPN으로도 효과적인 분류작업이 가능하다. 따라서 MA Network에 사용된 은닉층의 수는 하나로 정했으며, 각 계층의 PE개수 결정은 AR Network에서 설명한 것과 같은 방식으로 결정하였다. 따라서 ARMA 모형의 선정은 AR 및 선택된 MA Network의 가장 큰 출력값을 각각 택함으로써 이루어 진다.

## V. 실험 및 결과 분석

### 1. 학습 데이터

MLBPN은 주어진 학습패턴들에 대해서만 가중치 조정을 통해 스스로 적응 학습하므로 (Learning by Examples) 일반화 능력을 갖도록 하기 위해서는 합리적인 학습데이터의 구성은 매우 중요하다. 물론, ARMA모형선정에 있어서도 그림4와 같은 이론적인 ESACF원형들만으로 훈련시킨 MLBPN으로 부터는 좋은 분류능력을 기대할 수 없다. 실험에 사용될 학습데이터를 준비하기 전에 많은 시계열로 부터 ESACF패턴을 구해 분석한 결과, 적어도 30%내외의 잡음에도 정확히 반응할 수 있도록 MLBPN을 훈련 시켜야만, 실제 적용이 가능하다는 결론을 얻

었다. 또, ESACF표상의 잡음의 분포상태를 살펴본 결과, 역삼각형 내부에는 비교적 적은 잡음만이 분포한다는 사실을 알게 되었다. 이러한 사실에 입각하여 원형 ESACF표 들에 잡음을 인위적으로 삽입하여 다음과 같이 학습데이터를 구성하였다.

1) ARMA(5,5)까지, 즉 35개의 class들이 고려되었으며, 입력패턴들은 10X10의 크기를 갖는 변형된 ESACF표이다.

2) 이진값으로 표현된 원형 ESACF표에서 임의로 0은 1로, 1은 0으로 바꾸어 잡음을 나타냈다. 하나의 ARMA모형에 대해 각기 10, 20 및 30개의 이진값들을 바꾸어 10%, 20% 및 30%의 세가지 잡음수준을 고려했다.

3) 하나의 학습패턴에 삽입된 잡음들 중, 역삼각형 내의 잡음은 ARMA( $p, q$ ) 모형의 경우 0과 12-p-q사이의 숫자 중에서 임의로 선택하였다.

4) ARMA( $p, q$ )모형을 가정하여 설명하면, 각 학습패턴에 원하는 출력값의 대응은 다음과 같다. 대안 1의 경우, 35개의 출력PE가 요구되고, 6p+q 번째 출력PE만 1을 갖고 나머지 출력PE들은 0을 갖는다. 대안 2와 3의 경우, AR network 및 모든 MA Network들은 각기 6개의 출력PE들을 가지며, AR network의 p+1번째 출력 PE와, 각 MA network의 q+1번째 출력 PE만이 1을 갖고, 나머지는 모두 0을 갖도록 했다.

5) 학습데이터는 각 잡음수준에 대하여 ARMA(0, 1), ARMA(0, 2), ..., ARMA(5, 5)의 순서로 하나의 ARMA모형에 대해 5개씩의 학습패턴들을 컴퓨터 프로그램을 사용하여 연속으로 생성하였다. 이때 난수발생에 사용된 seed number는 2137이었다.

6) 생성된 학습데이터는 다시 잡음수준에 따

라 4개의 집합으로 재구성하였다. 첫번째 학습 집합은 35개의 원형패턴들로만 구성하였으며, 둘째집합은 첫째 집합에 10%의 잡음을 가진 175개의 학습패턴들을 더하여 210개의 학습패턴으로 구성하였다. 셋째 학습집합은 원형패턴과 10% 잡음 및 20% 잡음을 가진 학습패턴들로 구성되었으며, 넷째 학습집합은 560개의 학습패턴들을 가지며 5)에서 생성된 모든 학습패턴을 포함하고 있다. 이와 같이 잡음수준에 따라 4개의 학습집합을 구성한 것은 MLBPN의 훈련에 상당한 시간이 소요되는 점을 감안함과 동시에, 잡음수준에 따른 MLBPN의 학습효과를 측정하기 위한 것이다.

7) 대안 2의 경우 하나의 ARMA 모형에 대해 AR Network과 MA Networks에서 요구되는 출력값들이 상이하므로 두개의 학습데이터 집단을 준비했으며, 대안 3의 경우 특히 MA Networks을 위한 학습데이터를 구성하기 위해 대안 2의 MA Network을 위한 학습데이터 집단을 AR차수에 따라 여섯 개로 분리하였다.

MLBPN의 구현 및 학습데이터의 생성에 사용된 프로그래밍 언어는 Turbo-C이며, 필요한 난수발생은 Turbo-C내의 라이브러리 함수를 이용하였다. 모든 실험은 386PC들을 사용했다.

## 2. 학습방법

Backpropagation 알고리즘은 본 연구에서 사용한 fully-connected feed forward multi-layered Perceptron을 학습시키는데 가장 많이 이용된 방법으로 전체 학습패턴들에 대한 오차값을 최소화하는 가중치의 집합을 찾는다. 그림 5에 나타낸 것처럼 Backpropagation 알고리즘은 크게 Forward Pass, Backward Pass 및 Weight Update의 세 부분으로 나눌 수 있으며, 학습된 ML

BPN의 재생과정은 Forward Pass만으로 이루어진다 (Rumelhart et al, 1986).

그림 7은 본 연구에서 ESACF패턴 분류를 위해 Backpropagation 알고리즘을 어떻게 사용했는가를 pseudo-code 형태로 표현한 것이다.

즉, 학습률(Learning Rate)과 관성항(Momentum Term)계수로  $\eta = 0.1$ 과  $\alpha = 0.9$ 를 사용했으며, 모든 가중치들은  $-0.1$ 과  $0.1$  사이의 임의의 값들로 초기화 시킨후, 각 패턴의 입력 시마다 가중치를 조정하는 Pattern Recursion방식을 사용하여 MLBPN들을 학습시켰다.

MLBPN의 학습을 원활히 하기 위해 본래 0과 1사이에 값들로 구성된 학습패턴들의 입력력값들은 모두 0.1과 0.9사이의 값을 갖도록 변환시켰으며, 전체 학습패턴들에 대한 오차값이 사전에 정한 임계치보다 작아질때 학습을 종료시켰다.

학습된 MLBPN이 잡음에 심하게 오염된 ESACF 패턴들에 대해서도 잘 반응하기 위해서는 체계적인 학습전략이 필요하다.

채택된 학습전략은 앞절에서 설명한 4개의 학습집합들을 단계적으로 MLBPN들에 학습시키는 것이다.

먼저 원형 ESACF들로만 구성된 첫번째 학습집합을 가지고 해당 MLBPN을 학습시킨후, 학습된 MLBPN에 대해 두번째 학습집합을 학습시켰다.

같은 방식이 셋째, 넷째 학습집합에 대해서도 적용되었는데, 이같은 학습전략은 각 잡음수준에 대해 훈련된 MLBPN의 일반화 정도를 측정하기 위해 고안하였지만, 전체 학습데이터를 훈련시키는데 필요한 시간을 상당히 감소시켜 주었으며, 심한 잡음이 있는 학습자료에 대해서도 균형있는 학습을 가능하게 하는 효과가 있었다.

### Procedure: Backpropagation Algorithm for ESACF pattern Classification

Set  $\eta = 0.1$ ,  $\alpha = 0.9$ ,  $\epsilon = \infty$ ;  
 Initialize weights to small random numbers between -0.1 and 0.1;  
 Do while  $\epsilon$  is greater than predetermined error bound  
   Repeat until all training input/output pairs are presented to MLP  
     Select an input/output pair from the given training set;  
     Use subprocedure:Forward to compute actual outputs;  
     Compute  $E$  by summing differences between target and actual outputs;  
     Use subprocedure:Backward to compute errors in all PEs;  
     Use subprocedure:WeightUpdate to adjust weights;  
   endrepeat  
 Compute  $E$  as an averaged  $E$  over all input/output pairs;  
 Increase epochs by 1;  
 enddo  
endprocedure

/\*  $(x_1, x_2, \dots, x_n)$  and  $(d_1, d_2, \dots, d_m)$  is an input and target output pair.  
 $L$  is a number of layers.  $a_j(s)$  is an activated value at PE  $j$  in layer  $s$   
 and  $s = 1, 2, \dots, L$ .  $w_{ij}(t)$  is a weight between PE  $i$  in layer  $s-1$  and PE  
 $j$  in layer  $s$ .  $t$  is an index for recursions between weight updates \*/

#### Subprocedure: Forward

Repeat until actual outputs,  $a_j(L)$ 's are calculated  
 If a PE in input layer, then  
    $a_j(1) = x_j$ ;  
 else  $a_j(s) = f(\sum_i w_{ij}(t) \cdot a_i(s-1))$ ;  
 endif  
endrepeat  
endsubprocedure

#### Subprocedure: Backward

Repeat until error term  $\delta$  for each PE is calculated  
 If the PE is in L-th layer (output layer), then  
    $\delta_j = a_j(L) \cdot (1 - a_j(L)) \cdot (d_j - a_j(L))$ ;  
 else (in hidden layers)  
    $\delta_j = a_j(s) \cdot (1 - a_j(s)) \cdot \sum_k \delta_k \cdot w_{jk}(t)$ ;  
 endif  
endrepeat  
endsubprocedure

#### Subprocedure: WeightUpdate

Repeat until all weights are updated  
 $w_{ij}(t+1) = w_{ij}(t) + \eta \cdot \delta_i \cdot a_j(s) + \alpha \cdot (w_{ij}(t) - w_{ij}(t-1))$ ;  
endrepeat  
endsubprocedure

그림7. Pseudo-code로 표현한 Backpropagation 알고리즘

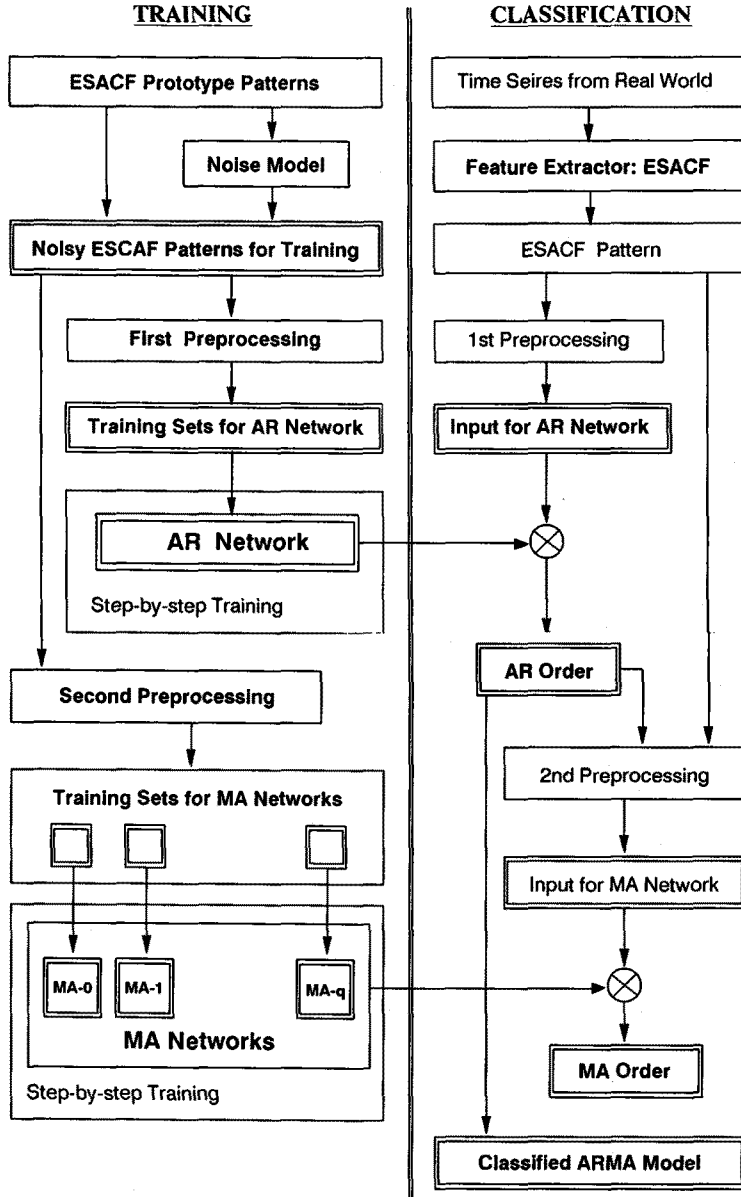


그림8. ARMA 모형선정을 위한 통합된 신경망 시스템의 학습 및 활용과정



### 3. 실험결과와 분석

그림 8은 ARMA모형선정을 위해 설계한 통합된 신경망시스템을 학습시키고, 학습된 MLBPN들을 가지고 분류작업을 수행하는 절차를 도표화한 것이다. 훈련된 MLBPN들의 분류능력을 검증하기 위해 검증데이터를 준비했는데, 원하는 출력값이 없다는 것 외에는 5.1절에서 설명한 학습패턴의 생성과 같은 방식이 적용되었다. 한 ARMA모형에 대해 각 잡음수준마다 2개씩의 검증패턴(test pattern)을 생성하여, 즉 각기 70개의 검증패턴을 갖는 잡음수준 10%, 20% 및 30%의 세개의 검증집합으로 만들었다.

seed number는 16087을 사용하였다. 학습과정의 각 단계에서 학습을 마친 MLBPN들에 세개의 검증집합들을 모두 적용하여 잡음수준에 따른 분류능력의 변화를 알 수 있도록 했다.

본 연구에서 제안한 통합된 신경망 시스템, 즉 대안 3의 경우, 실험에 사용된 MLBPN의 각 계층의 PE개수는 AR Network의 경우 29-25-18-6이었으며 MA Network들의 경우는 29-18-6이었다. 표 1과 2는 각 학습집합에 학습된 MLBPN들의 분류능력을 검증데이터를 사용하여 분석한 결과를 요약한 것으로, 가운데 칸들은 잘못 분류한 검증패턴의 수를 잡음수준에 따라 정리한 것이며, 마지막 칸은 전체 검증데이터에 대한 적중률을 나타낸 것이다.

AR Network의 경우 학습집합의 잡음수준이 증가함에 따라 적중률이 급격히 증가하지만, MA Networks의 경우 그 증가율이 비교적 완만하다. 이는 생성된 학습데이터를 다시 각 MA Network들에 적합하도록 분리하였으므로 하나의 MA Network에 사용된 학습패턴의 수가 상대적으로 적었기 때문이다. 따라서, MA Networks의 학습패턴 수를 증가시키면 분류

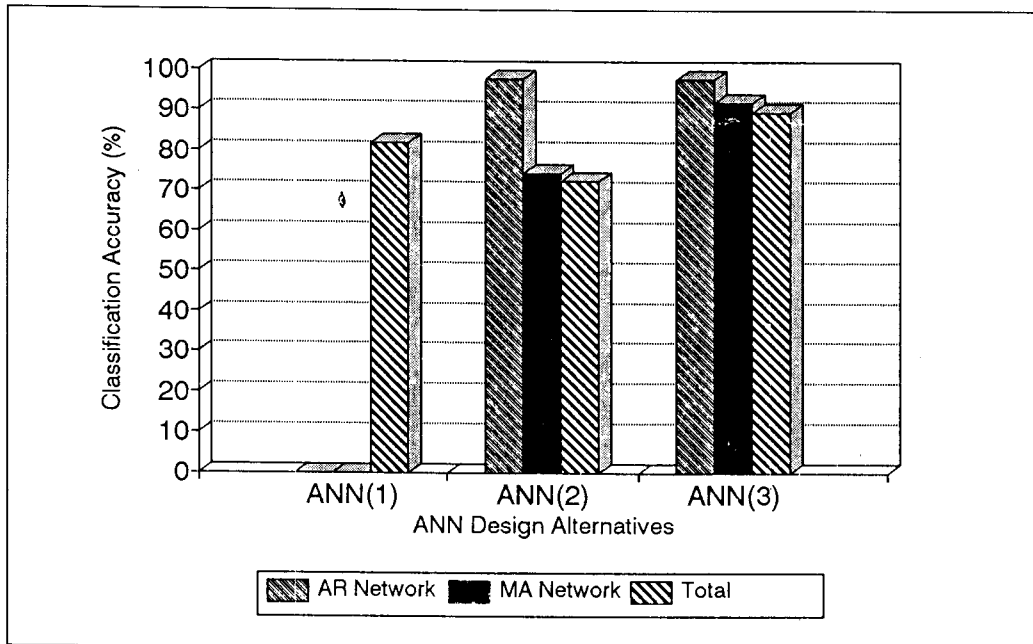


그림9. Classification Performance of Alternative Designs

능력이 개선될 가능성이 높다.

각 MA Network들의 분류능력은 큰 편차를 보이지 않았으며, 그림 9에서와 같이 분류능력이 상당히 개선됐음을 알 수 있다. 그림 9은 4.1절에서 설명한 세 대안에 대해 본 연구에서 설명한 학습자료 및 검증자료를 동일하게 적용하여 얻은 결과를 요약한 것이다. ANN(1)은 대안 1의 경우로 각계층에 사용된 PE 수자는 100-70-50-35 개이며, ANN(2)는 대안 2의 경우로 4.2절에서 설명한 1차 전처리만을 적용하여 입력 자료를 사용하였기 때문에 AR 및 MA Network 모두 29-25-18-6 개의 PE들이 사용되었으며, ANN(3)는 대안 3을 의미한다. 대안 1과 2에 대해서는 Lee & Jhee [1990]의 실험결과를 인용한 것이다.

실험결과를 요약하면, 첫째, 제안한 신경망 시스템은 잡음이 많은 경우에도 매우 효과적이었다. 특히 ANN(1)의 경우 30%의 잡음수준에 학습시키면 오히려 분류능력이 저하되는 현상이 있었지만, 표 1과 2에서 볼 수 있듯이 ANN(3)에서는 이와 같은 현상을 피하면서 효과적인 학습이 가능했다. 둘째, 단계적 학습방법은 전체학습시간의 절감효과 뿐만아니라, ESACF의 이론적 원형이 갖는 정보를 충분히 이용할 수 있게 함으로써 효율적인 학습과정이 이루어지도록 해주었다. 셋째, 도입된 전처리 방식들

은 계절주기를 갖는 큰 규모의 ESACF패턴들도 쉽게 처리할 수 있게 해주며, 2차 전처리는 MA Network들의 분류능력을 향상시키는데 크게 기여하였다. 넷째, 임의로 생성된 학습데이터를 사용하였는데, 만약 현실의 많은 시계열로부터 ESACF패턴을 구해 이용한다면 더 좋은 분류능력을 기대할 수 있을 것이다.

또, 제안된 통합된 신경망 시스템의 실제 이용가능성을 검증하기 위해 현실의 시계열 자료를 이용하였다. 그림 10의 (a)는 미국 소비자물가지수(Pandit & Wu, 1983) 자료로부터 ESACF패턴을 구한 것이다. 이 시계열은 ARMA(4,3)을 따르는데, 구해진 패턴은 심한 잡음이 있어 정확한 ARMA모형의 선정이 어렵다. 그림 10의 (b)는 (a)를 AR Network의 입력으로 사용하기 위해 1차 전처리를 한 결과이다. AR Network의 출력값은 (d)에 표시했는데 다섯번째 PE의 값이 0.681로 가장 크며, 이에 해당하는 AR의 차수는 4이다.

(c)는 이 시계열의 AR차수가 4라는 사실을 이용하여 2차 전처리를 한 결과로 MA-4 Network의 입력으로 사용된다. (d)에 표시된 MA-4 Network의 출력값들 중 네번째 값이 0.872로 가장 크므로 MA의 차수는 3이 된다. 따라서 이 시계열에 적합한 모형은 ARMA(4,3)으로 정확히 판정되었다. 그림 11은 커피원두의 카페인

표 1. ARNetwork의 검증데이터 분류 결과  
(은닉층의 PE개수 : 25-18)

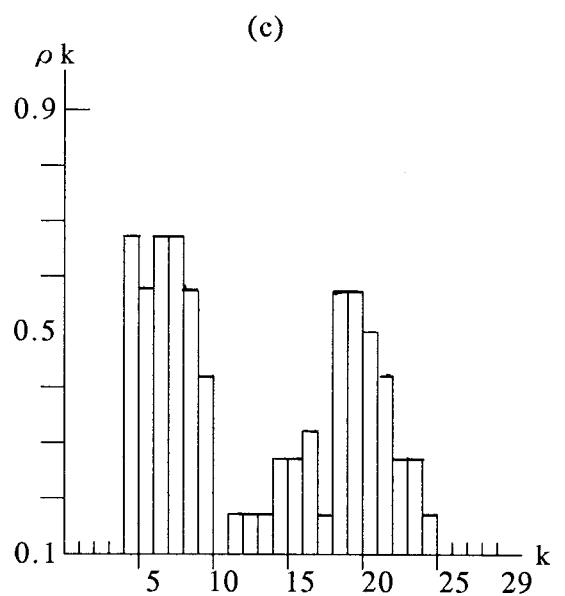
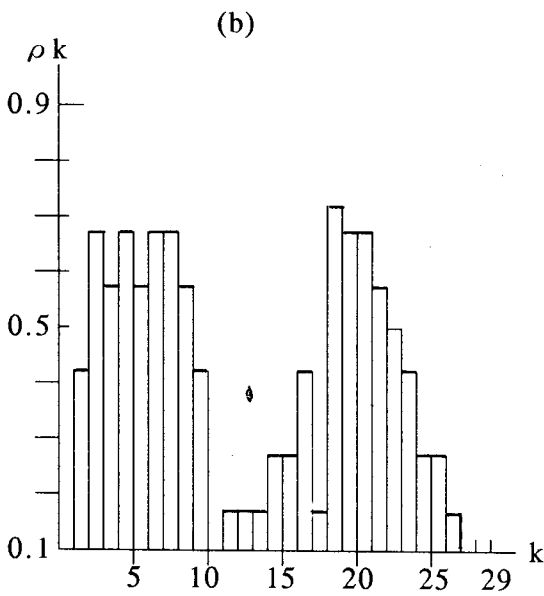
| 학습<br>집합 | 잡음 수준 |     |     | 적중률<br>(%) |
|----------|-------|-----|-----|------------|
|          | 10%   | 20% | 30% |            |
| 1        | 14    | 23  | 38  | 64.3       |
| 2        | 0     | 4   | 18  | 89.5       |
| 3        | 1     | 2   | 7   | 95.2       |
| 4        | 0     | 3   | 2   | 97.6       |

표 2. MANetwork의 검증데이터 분류 결과  
(은닉층의 PE개수 : 18)

| 학습<br>집합 | 잡음 수준 |     |     | 적중률<br>(%) |
|----------|-------|-----|-----|------------|
|          | 10%   | 20% | 30% |            |
| 1        | 0     | 8   | 24  | 84.8       |
| 2        | 3     | 12  | 18  | 86.7       |
| 3        | 2     | 8   | 16  | 89.5       |
| 4        | 0     | 6   | 13  | 91.9       |

(a)

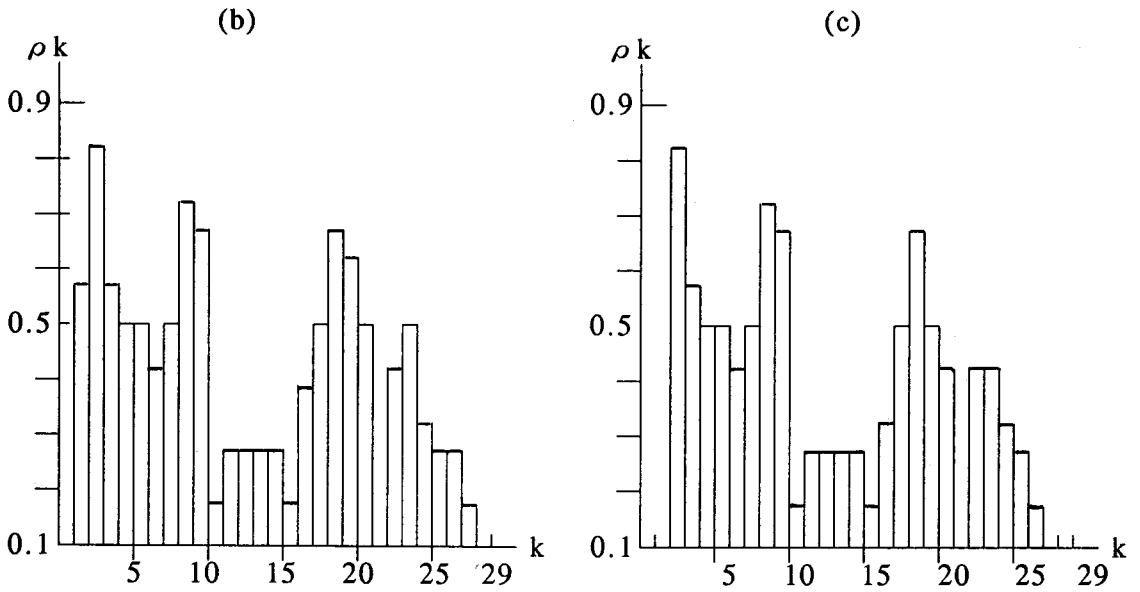
|   |    |   |   |   |   |   |   |   |   |   |   |
|---|----|---|---|---|---|---|---|---|---|---|---|
|   | MA | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|   | AR |   |   |   |   |   |   |   |   |   |   |
| 0 |    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 |    | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2 |    | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 3 |    | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 4 |    | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 5 |    | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 |    | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 |    | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 8 |    | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 9 |    | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |



(d) AR Network : 0.097 0.086 0.592 0.103 0.681 0.099  
 MA-4 Network : 0.096 0.089 0.107 0.872 0.077 0.094

그림 10. KConsumer Price Index : ARMA(4,3)  
 (a) ESACF 패턴 (b) 1차 전처리 후  
 (c) 2차 전처리 후 (d) 출력결과

|         |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|
| MA \ AR | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1       | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 2       | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 3       | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4       | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5       | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 6       | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 7       | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 8       | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 9       | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |



(d) AR Network : 0.098 0.591 0.829 0.106 0.109 0.085  
 MA-2 Network : 0.231 0.491 0.086 0.130 0.095 0.822

그림 11. Caffeine Data : ARMA(2,5)

- (a) ESACF 패턴
- (b) 1차 전처리 후
- (c) 2차 전처리 후
- (d) 출력결과

함유량 (Hamilton & Watt, 1978) 에 관한 시계열에 제안한 접근방법을 적용한것으로 앞서 설명한 것과 같은 방식으로 보면 된다.

이 시계열은 계절주기를 갖고 있는 것으로 알려져 있는데, (d)에서 보는 바와 같이 ARMA (2,5)로 정확히 분류하였다. 이는 통합된 신경망 시스템이 현실의 ARMA 모형화에 있어서 모형선정의 어려움을 극복하는데 이용될 수 있음을 보여주는 것이다.

## VI. 맺음말

본 연구에서는 전문가의 개입없이 ARMA 모형화가 가능하도록 신경망을 사용한 ARMA 모형선정문제를 다루었다. 잡음에 오염된 ESACF 패턴들을 효과적으로 분류하기 위해 하나의 AR Network과 작은 규모의 여러 MA Networks들로 구성된 통합된 신경망 시스템이 제안되었다.

AR Network은 AR 차수를 결정하고 어느 MA Network이 MA 차수 파악에 이용될 것인지를 지정한다. 제안한 신경망 시스템에 의한 실험 결과는 검증자료에 대해 90%정도의 정확성을 보였다. 따라서, 신경망의 출력,  $(p, q)$ , 즉 선

정된 ARMA모형의 형태를 바탕으로 하여, 이미 이론적 체계가 확립되어 있는 통계적 모수추정 방법을 사용하여 예측치를 구할 경우, ARMA 모형에 의한 예측치의 자동생성이 가능해진다. 동시에 이론적 우수성에도 불구하고 ARMA 모형선정의 어려움으로 인하여 발생하는 예측치의 질저하 문제도 방지할 수 있다.

앞으로의 연구방향으로는 MLBPN의 훈련에 많은 시간이 소요되므로 더 많은 ARMA모형들을 분류해낼 수 있는 신경망 시스템을 학습시키기 위해 Backpropagation 알고리즘의 수렴속도를 개선할 수 있는 방법을 모색하는 것과 본 연구에서 제안된 접근방법에 기초하여 시계열 모형화를 위한 의사결정지원시스템(DSS)의 개발이다.

이 DSS는 일종의 DSS Generator로서 시계열의 그래픽처리, 통계적 모수 추정 프로그램 및 추정된 모형의 적합도 검증을 위한 통계처리 등을 포함하게 될것이다. 개발에 있어 중요한 문제의 하나는 본 연구의 접근방식에 의한 해에 사용자가 만족하지 못할 경우 어떻게 대체 ARMA 모형을 제시할 것인가하는 문제가 될 것이다.

## 참 고 문 헌

Akaike, H. A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*. AC-19, 716-723.

Anderson, J. A., & Rosenfeld, E. *Neurocomputing: Foundations of Research*. (ed.). Cambridge MA:MIT Press.

Benguin, J. M., Gourioux, C., & Monfort, A. (1980). *Identification of a Mixed Autoregressive Moving Average*

Process: The Coner Method, in *Time Series*, Anderson, O. D., (ed.). Amsterdam:North-Holland, 423-436

- Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis - Forecasting*. San Francisco:Holden-Day.
- Cruz, C. A. (1988). *Understanding Neural Networks: A Primer*. Amherst: Graeme.
- DARPA. (1987). *DARPA Neural Network Study*. AFCEA International Press.
- Gorman, R. P., & Sejnowski, T. J. (1988). Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. *Neural Network*, 1, 75-89.
- Gray, H. L., Kelly, A. G., & McIntire, D. D. (1978). A New Approach to ARMA Modeling. *Communications in Statistics, B*, 7, 1-77.
- Hamilton, D. C., & Watt, D. G. (1978). Interpreting Partial Autocorrelation Functions of Seasonal Time Series Models. *Biometrika*, 65, 135-140.
- Harvey, A. C. (1984). A Unified View of Statistical Forecasting. *Journal of Forecasting*, 3(3), 245-275.
- HILL, G. W., & Woodworth, D. (1980) Automatic Box-Jenkins Forecasting. *Journal of Operational Research Society*, 31, 413-422.
- Hopfield, J. J., & Tank, D. W. (1985). Neural Computation of Decision in Optimization Problems. *Biological Cybernetics*, 52, 141-152.
- Jacobs, R. A. (1988). Increased Rates of Convergence through Learning Rate Adaptation. *Neural Networks*, 1, 285-307.
- Lee, J. K., & Jhee, W. C. (1990). Two-staged Neural Network Approach for the ARMA Model Identification. *KAIST Working Paper*.
- Kang, C. A., Bedworth, D. D., Rollier, D. A. (1982). Automatic Identification of ARIMA Time Series. *IIE Transactions*, 14, 156-166.
- Lapedes, A., & Farber, R. (1987). Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling. *Los Alamos National Laboratory Report, LA-UR-87-2662*.
- Lee, K. C., & Park, S. J. (1988). Decision Support in Time Series Modeling by Pattern Recognition. *Decision Support Systems*, 4, 199-207.
- Lippmann, R. P. (1988). An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4, April, 4-22.
- Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.
- Montgomery, D. C., & Johnson, L. A. (1976). *Forecasting and Time Series Analysis*. New York:McGraw-Hill.
- Pandit, S. M., & Wu, S. M. (1983). *Time Series and System Analysis with*

- Applications*, New York:John Wiley & Sons.
- Pao, Y. H. (1988). *Adaptive Pattern Recognition and Neural Networks*. New York:Addison-Wesley.
- Parzen, E. (1982). ARMA Models for Time Series Analysis and Forecasting. *Journal of Forecasting*, 1982, 1, 67-82.
- Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. New York: MI Spartan.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Internal Representations by Error Propagation, in Rumelhart, D. E. & McClelland, J. L. (Eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I: Foundations*, MA:MIT Press.
- Tsay, R. S., & Tiao, G. C. (1984). Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models. *Journal of American Statistical Association*, 79, 84-96.
- Wheelwright, S. C., & Makridakis, S. (1985). *Forecasting Methods for Management, 4th eds*. New York:John Wiley & Sons.
- White, H. (1988). Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. *Proceedings of International Joint Conference on Neural Networks*, San Diego, California, Vol. II, 451-458.
- Woodward, W. A., & Gray, H. L. (1981). On the Relationship between S-array and Box-Jenkins Method of ARMA Model Identification. *Journal of American Statistical Association*, 76, 579-587.

## ◆ 저자소개 ◆

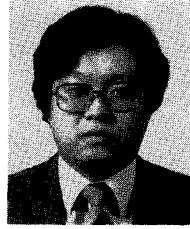


공동저자 지원철은 현재 홍익대학교 산업공학과 조교수로 재직중이다.

그는 서울대 경영학과를 졸업하고, 과학기술원 산업공학 석사과정을 이

수하고, (주)대림산업 전산실에서 근무한 후 한국과학기술원 경영과학과에서박사학위(MIS)를 취득하였다.

주요 관심분야는 신경망, 전문가시스템 및 지식기반 시뮬레이션 등의 인공지능기법을 이용한 의결정지원시스템이다. 응용분야로는 시계열예측 및 CIM과 재무분석시스템 등이다.



공동저자 송성현은 현재 홍익대학교 산업공학과 교수로 재직중이다.

그는 서울공대 산업공학 전공으로 학사과정을 이수하고, (주)금성사에

서 생산관리의 실무경험을 쌓은 후, 서울대 대학원에서 경영과학(OR)으로 석사, 박사학위를 취득하였으며, Georgia Tech의 Material Handling Research Center에서 연구원으로 근무한바 있다. 그의 주 관심사는 Nonlinear Optimization, 생산계획, 물류시스템 설계 및 운영방법, 컴퓨터 응용 등이다.