

論文96-33B-10-12

PCA를 이용한 다중 컴포넌트 신경망 구조설계 및 학습

(Multiple Component Neural Network Architecture Design and Learning by using PCA)

朴贊虎*, 李顯洙*

(Chan Ho Park and Hyon Soo Lee)

요 약

본 연구에서는 PCA(Principal Component Analysis)방법을 이용하여 입력패턴벡터의 차원을 감소시킨 후 다중의 컴포넌트 신경망에서 패턴들을 분할학습하는 새로운 신경망인 다중 컴포넌트 신경망(MCNN)을 제안한다. 제안한 신경망에서는 PCA의 역할을 하는 Oja's rule을 이용한 신경망을 사용하였으며, 이때의 출력패턴들은 CBP라고 불리는 다중의 소규모 컴포넌트 신경망에서 학습패턴으로 사용된다. 또한 학습이 용이하지 않은 패턴에 대해서는 새로운 CBP 신경망의 생성을 통하여 패턴의 분할학습이 동적으로 이루어 지게 한다. 학습 실험결과 기존 교사신호를 갖는 다층신경망인 EBP와 비교하여 전체 네트워크의 크기가 크게 감소함을 알 수 있었으며, 학습속도가 크게 향상되었다.

Abstract

In this paper, we propose Multiple Component Neural Network(MCNN) which learn partitioned patterns in each multiple component neural networks by reducing dimensions of input pattern vector using PCA(Principal Component Analysis). Proposed neural network use Oja's rule that has a role of PCA, output patterns are used as learning patterns on small component neural networks and we call it CBP. For simply not solved patterns in a network, we solves it by regenerating new CBP neural networks and by performing dynamic partitioned pattern learning. Simulation results shows that proposed MCNN neural networks are very small size networks and have very fast learning speed compared with multilayer neural network EBP.

I. 서 론

신경망 중 가장 널리 쓰이는 다층 퍼셉트론 신경망은 교사학습을 수행하는 전방향처리 학습알고리즘인 에러역전파 학습알고리즘을 사용하며, 일반화 능력과 비선형문제의 처리에 있어서 기존의 알고리즘에 의한 방법보다 우수한 능력을 가지고 있다. 다층 신경망을 이용한 응용분야로는 패턴 분류 및 인식, 이미지처리, 시스템제어, 로봇제어 등에서 매우 광범위하게 사용되

어진다.¹⁾⁻⁴⁾

신경망이 적용되어지는 분야에서의 응용문제는 매우 다양하여 종종 매우 큰 크기의 신경망을 필요로 하며 또한 복잡한 문제의 해결을 요구한다.

따라서 앞으로의 신경망은 이러한 문제들에 대해 유연하게 적응 할 수 있어야 한다. 그러나 다층 신경망의 경우에는 입력패턴과 교사신호패턴간의 차이에 의한 에러를 전체 연결가중치에 재부여함으로써 점진적으로 문제를 해결하는데 필요한 최적의 가중치를 결정하는 최급강화 방법을 이용하기 때문에 학습시간이 오래걸리고, 지역적인 국소점에 빠지는 경우에는 문제를 해결하지 못하는 단점을 가지고 있다. 또한 입력패턴과 출력패턴을 표현하는 각 입,출력층과 입력패턴에 대한 특

* 正會員, 慶熙大學校 電子計算工學科

(Dept. of Computer Engineering Kyunghee University)

接受日字:1996年7月31日, 수정완료일:1996年9月16日

정을 추출하는 역할을 담당하는 중간층이 완전연결을 가지고 있으므로 이때의 가중치들이 모든 입,출력 패턴에 대하여 공통적으로 반응해야 한다는 점은 문제의 복잡도가 높아질수록 학습을 더욱 어렵게 한다.¹²¹

이를 해결하기 위한 많은 연구가 진행되었으며 이에 대한 기존의 연구에서는 첫째, 학습알고리즘에 사용되는 각 학습파라미터의 최적값을 선택하기 위한 연구를 들 수 있다.¹⁵⁻⁷¹ 그러나 이 방법들 역시 문제자체의 복잡도가 크고 학습하고자 하는 패턴의 수가 증가할수록 최적의 파라미터의 값을 결정하기가 상당히 어려운 점을 가진다.

둘째, 입,출력패턴들을 특성별로 분류하고 각각의 분류된 패턴들에 대한 적절한 파라미터를 부여하여 학습을 시키는 방법에 대한 연구를 들 수 있다. 이러한 분류의 대표적인 신경망으로는 RBF (Radial Basis Function)¹⁸¹과 Modular Network¹⁹¹등이 있다. RBF의 경우 빠른 속도로 학습을 할 수 있다는 장점이 있는 반면에 학습하고자하는 패턴의 수에 비례하여 중간층 감응역역의 수가 비례적으로 늘어나게 되어 신경망의 크기가 증가하고 일반화 능력이 떨어지는 단점이 있다. 또한 Modular network의 경우는 일반화 능력이 우수하고 학습속도가 향상되었으나 각 모듈의 수에 따라 부여되는 입력 및 출력뉴런의 수가 늘어나게 되므로 신경망의 크기에 있어서 문제점을 가지고 있다. 기존 전체 네트워크를 분할하여 학습하는 모듈 신경망 부류의 가장 큰 단점으로써, 각 모듈에 적용되는 입,출력뉴런의 수가 단일 네트워크의 입,출력 뉴런수와 동일하므로 전체 네트워크의 크기는 매우 커지게 된다는 점을 들 수 있다.

셋째, 입력 또는 출력 패턴의 변환에 의한 학습방법이 있다. 그 대표적인 학습방법으로는 비선형성격을 갖는 패턴을 선형의 성격을 갖는 패턴으로 변환시켜 학습시키는 Fao의 FUN (FUncional link Network)¹¹⁰¹가 있다. 이 네트워크는 단층 퍼셉트론의 구조를 가지므로 학습의 속도가 빠르고 네트워크의 구성이 간단하다는 장점이 있다. 그러나 입력패턴벡터의 차원이 증가할수록 부가되는 combinational logic unit의 수가 매우 크게 증가하므로 전체 네트워크의 크기는 실제문제에 적용하기 어려울 정도로 커지게 된다.

본 연구에서는 위에서 언급되어진 문제들을 해결하기 위하여 입력벡터의 차원을 감소시켜 다중의 컴포넌트 신경망인 CBP(Component BackPropagation)에

서 분할된 패턴을 독립적이고 병렬적으로 학습하는 새로운 다중 컴포넌트 신경망인 MCNN(Multiple Component Neural Network)를 제안한다. 제안한 신경망은 원 입력패턴간의 특성을 변화시키지 않고 적은 차원의 패턴으로 변화시켜주는 PCA방법을 신경망으로 구현한 Oja의 SGA(Stochastic Gradient Ascent)¹¹¹ 또는 GHA(Generalized Hebbian Algorithm)¹¹²를 사용하였으며, 차원이 감소된 입력패턴을 분할하여 에러역전과 학습을 수행하는 컴포넌트 신경망인 CBP로 전파시켜 학습시키는 새로운 신경망으로서 구조상의 유연성이 뛰어나고 학습시 우수한 성능을 가지는 새로운 신경망이다.

실험대상으로는 seven segment digit display 문제와 다층 신경망상에서 학습이 어려운 parity문제, 그리고 영문자 인식을 대상으로 하였으며 다층 신경망과 그 결과를 비교 검토하였다. 실험결과, 전체 신경망의 학습속도가 향상되었으며 신경망의 크기가 매우 감소함을 알 수 있었다. 특히 제안한 신경망은 문제의 크기가 크고 복잡한 문제에 있어서 매우 효과적임을 알 수 있었다.

본 논문의 2절에서는 PCA방법에 대한 고찰과 이 방법을 신경망에 적용시킨 oja's rule 및 신경망에 대하여 분석 고찰한다. 제3절에서는 제안한 다중 컴포넌트 신경망의 구조 및 학습방법에 대하여 설명하고, 제4절에서는 제안한 신경망에 대한 실험 및 그 결과를 분석 고찰한다. 마지막으로 제5절에서는 본 연구의 결론을 기술하고 향후 연구방향을 제시한다.

II. PCA (Principal Component Analysis)

1. PCA (Principal Component analysis) 방법

주성분 분석, 즉 PCA 방법은 Pearson에 의해서 1901년에 처음 소개된 방법으로써 다변량분석 및 특징 추출 그리고 패턴압축등에 널리 사용되고 있는 방법이며, 흔히 Karhunen-Loeve변환, 혹은 Hotelling 변환 등으로 불리워진다.¹¹³¹ 이 방법은 특히 높은 차원의 크기를 갖는 패턴들이 갖는 특징을 잃지 않으면서 선형변환 혹은 보다 낮은 차원의 패턴으로 변환시키는데 있어서 널리 쓰이는 방법이다. PCA 방법에 대하여 간단히 요약 설명하면 다음과 같다.

n 차원을 갖는 입력벡터 $X = [x_1, x_2, x_3, \dots, x_n]$ 에 대한 영의 평균 $\langle X \rangle$ 및 공분산 R_{xx} 을 나타내면 식(1),

식(2) 와 같다.

$$\langle X \rangle = EM \langle X \rangle \cong \frac{1}{N} \sum_{k=1}^N X_k \quad (1)$$

$$R_{XX} = E_M [XX^T] \quad (2)$$

$$\cong \frac{1}{N-1} \sum_{k=1}^N (X_k - \langle X \rangle)(X_k - \langle X \rangle)^T$$

공분산 R_{xx} 에 대한 직교정규화 고유벡터를 $W_1, W_2, W_3, \dots, W_m$ 이라고 가정하면 가장 큰 고유치 λ_1 값에 대한 고유벡터는 $W_1 = [w_{11}, w_{12}, \dots, w_{1n}]$, 다음으로 큰 고유치 λ_2 값에 대한 고유벡터는 $W_2 = [w_{21}, w_{22}, \dots, w_{2n}]$ 등으로 표현할 수 있다. 바꾸어 표현하면 공분산 R_{xx} 와 고유치 λ 와 고유벡터 W 의 관계는 다음과 같은 행렬식으로 나타낸다.

$$R_{xx} W_j = \lambda_j W_j \quad (j = 1, 2, 3, \dots, n) \quad (3)$$

이때의 고유치 λ_j 는 다음과 같은 조건을 만족한다.

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots, \lambda_n \geq 0 \quad (4)$$

이렇게 하여 주어진 입력 벡터 X 에 대한 처음의 m 차 주 성분은 식(5) 와 같은 선형변환식으로 표현된다.

$$Y = WX = \sum_{k=1}^m w_k x_k \quad (5)$$

$$(W = [w_1, w_2, w_3, \dots, w_m]^T \in R^{m \times n})$$

따라서 PCA방법은 위의 수식(5)에서 나타난 고유벡터 W 를 이용하여 선형변환하고 새로운 주성분 Y 를 구하는 것을 의미한다.

2. PCA를 이용한 신경회로망

1982년 Oja는 가장 첫 번째 주성분인 Y_1 의 값을 계산하기 위하여 일차선형처리 수식을 다음과 같이 정의하고 이를 신경망 학습알고리즘으로 표현하였다. 이를 Oja's rule 혹은 NHR(Normalized Hebbian Rule)로 부른다.^[14]

$$Y_1 = W_1^T X = \sum_{j=1}^n w_{1j} x_j \quad (6)$$

위의 식(6)에서 Y_1 은 첫 번째 주성분 또는 뉴런의 출력값을 의미하며 w 는 NHR신경망에서의 뉴런간의 연결가중치를, x 는 입력을 나타낸다.

이때 학습알고리즘의 목적은 식(6)에서의 가중치 $W_1 = [w_{11}, w_{12}, w_{13}, \dots, w_{1n}]^T$ 를 찾는 것으로 식(7)과 같은 가중치 갱신 방법에 의해서 표현되어진다.

$$w_{1j}(k+1) = w_{1j}(k) + \eta [y_1(k)x_j(k) - w_{1j}(k)y_1^2(k)] \quad (7)$$

이후 식(6)과 식(7)을 확장하여 하나의 주성분 뿐만 아니라 여러개의 주성분을 계산할 수 있는 학습 알고리즘들이 제안되었는데 이를 나타내면 1989년 Sanger에 의해 제안된 GHA^[12] 방법이 있으며 이를 가중치 갱신 수식으로 나타내면 식(8)과 같다. 식(8)과 식(9)에서 각 주 성분을 계산하는 수식은 식 (6)에서의 첨자 1을 i 로 변환한 것과 같으므로 생략하고 가중치 갱신 수식을 표현한다.

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w \quad (8)$$

$$\Delta w = \eta y_i(k) [y_j(k) - \sum_{h=1}^i w_{hj}(k) y_h(k)]$$

또한 SGA^[11] 방법은 1992년 Oja에 의해서 제안되었으며 GHA방법에 비하여 알고리즘의 차이는 거의 유사하나 작은 성분값을 추출하는데 있어서 보다 나은 성능을 보이고, 학습속도가 향상된 알고리즘이다. SGA 방법의 가중치 갱신에 대한 표현은 식(9)와 같다.

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w \quad (9)$$

$$\Delta w = \eta y_i(k) [y_j(k) - w_{ij}(k) y_i(k) - \alpha \sum_{h=1}^{i-1} w_{hj}(k) y_h(k)]$$

그림1에서는 식(8), 또는 식(9)에 의해서 비교사 학습을 수행하는 신경회로망의 구조를 나타낸다.

이때 입력으로는 n 차원의 입력패턴벡터가 사용되었으며, 출력으로는 m ($m \leq n$)차원의 출력패턴 벡터가 사용된다. 이때 전체 네트워크의 구조는 단층으로 입력층과 출력층간의 연결은 완전 연결 구조를 가진다.

III. 다중 컴포넌트 신경망(MCNN) 구조 및 학습

제안하는 다중 컴포넌트 신경망은 PCA학습 신경망과 다중의 컴포넌트 신경망인 CBP의 2개의 부분으로 나누어지며 PCA 신경망에서의 학습이 완료된 후 다시 다중의 컴포넌트 신경망에서 학습이 이루어진다. 다중 컴포넌트 신경망의 구조를 나타내면 그림 2와 같다.

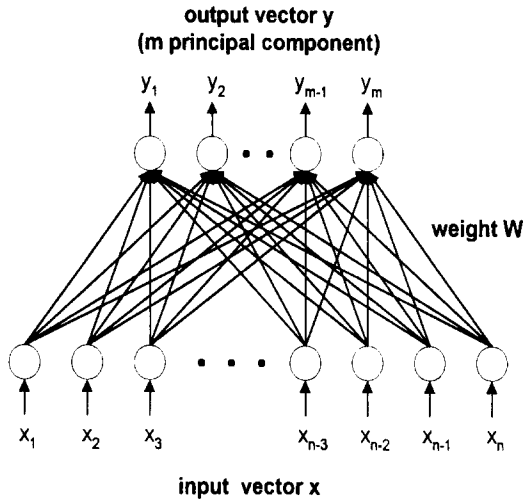


그림 1. PCA신경망 구조
Fig. 1. PCA Neural Network Architecture.

PCA에서 학습이 끝난 출력패턴들은 본 논문에서 제안한 패턴분할방법에 의하여 정해진 컴포넌트 신경망으로 분리되어져서 학습이 이루어 지게 되며, 만약 하나의 컴포넌트 신경망에서 주어진 패턴간의 학습이 어려운 경우 새로운 컴포넌트 신경망을 생성한 후 주어진 패턴을 분리하여 학습하게 된다. 따라서 제안한 네트워크는 학습시 동적으로 신경망의 구성이 이루어지게 되며 학습에 필요한 최적의 구조의 생성이 가능하다. 각 컴포넌트 신경망의 학습알고리즘으로는 에러역전파 학습알고리즘을 사용하며, 다중 신경망의 구조를 갖는다.

1. PCA 신경망 학습과 패턴의 분할방법

제안한 다중 컴포넌트 신경망을 구성하는 PCA신경망의 구조는 그림 1과 동일하며, 사용한 알고리즘으로는 입력값에 대한 작은 주성분값을 추출하는데 있어서 GHA보다 빠르고 효과적인 식 (8)의 SGA 알고리즘을 사용한다. 입력벡터에 대한 출력성분 즉 출력값이 하나인 경우에는 NHR과 동일하다. SGA 알고리즘을 이용한 학습의 경우에는 입력패턴간의 차이가 미세한 경우, 나타나는 패턴간의 출력값의 차이 또한 미세한 값으로 나타날 수 있다. 이러한 출력값을 그대로 작은 출력차원을 갖는 CBP 신경망으로 학습시킬 경우 학습속도는 증가할 수 있다. 이때 PCA신경망의 출력 차원을 크게 할 수록 패턴간의 출력값의 차이는 커지게 되나, 출력벡터의 차원이 너무 크게 되면 컴포넌트 신경망의 입

력벡터 차원이 커짐에 따라 신경망의 크기가 증가하게 된다. 따라서 이와 같은 단점을 극복하기 위해서는 출력차원의 결정 최적화가 필요하다.

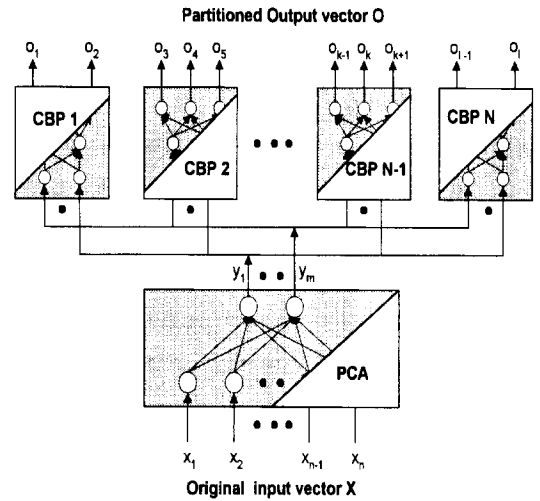


그림 2. 제안한 다중 컴포넌트 신경망 구조
Fig. 2. Proposed Multiple Component Neural Network(MCNN) Architecture.

본 연구에서는 PCA출력벡터의 차원이 원입력벡터의 차원보다 충분히 작게 적절히 구성하였다. 그리고 출력벡터의 각 원소의 값이 입력벡터에 대한 주성분이므로 전체 패턴들에 대한 주성분값의 최대값과 최소값을 구하고 각 컴포넌트의 수에 따라 선형변환된 각 주성분들을 분할하여 해당되는 영역의 패턴들을 하나의 CBP 신경망에서 분할 학습함으로써 학습성능을 개선시키고 전체 신경망의 크기를 감소시켰다. 이를 그림으로 표현하면 그림 3과 같다.

아래의 그림 3에서 각 영역에 대한 경계 θ_i 는 CBP 신경망의 수를 n 라 하고 각 출력벡터의 원소를 y_i 라고 할때, 다음과 같은 식(10), 식(11)에 의해서 결정된다.

$$\theta = (\text{MAX}(Y_i) - \text{MIN}(Y_i)) / n \tag{10}$$

$$\theta_i = i \times \theta + \text{MIN}(Y_i) \quad (i = 1, 2, \dots, n-1) \tag{11}$$

따라서 각 영역에 대한 전체 표현은 다음과 같이 식 (12)로 나타낼 수 있다.

$$R: [\min(y_i) \leq \theta_1 \leq \theta_2 \leq \dots \leq \theta_{m-1} \leq \max(y_i)] \tag{12}$$

R : total region of component y_i

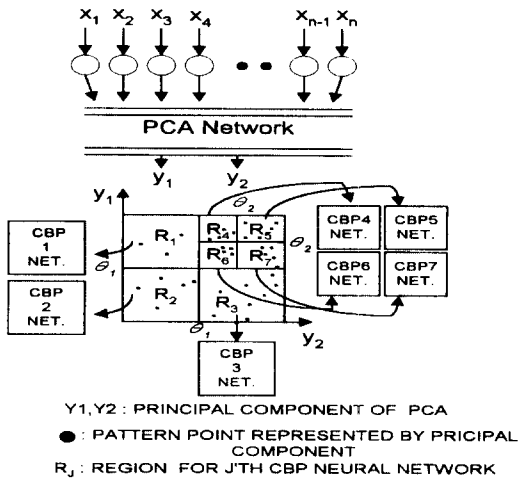


그림 3. PCA 신경망의 주성분 출력패턴 분할방법
 Fig. 3. Pricipal Components Partitioning Method of PCA Neural Network.

주어진 패턴의 학습이 어려운 CBP 신경망의 분할은 CBP 신경망에 할당된 패턴의 주성분값을 다시 식 (11) 그리고 식 (12)에 의해서 최대값과 최소값을 결정하고 영역 분할하여 학습한다. 이러한 분할방법은 기존 패턴 인식방법에서 decision tree 결정 방법과 유사하다^[15]. 그러나 PCA방법에 의해서 결정된 주성분 y_i의 값이 패턴간의 유사도(similarity)를 정확히 결정하지는 않으므로 분할방법에 대한 앞으로의 연구가 필요하다.

2. 컴포넌트 부분 신경망 CBP의 학습

컴포넌트 신경망인 CBP(Component Back Propagation)는 제안한 다중 컴포넌트 신경망상에서 PCA신경망을 제외한 부분으로, PCA신경망의 출력패턴을 입력패턴으로 받아서 동일한 영역에 속하는 패턴간의 학습을 수행하는 부분이다. 각 CBP의 구조는 다층 신경망의 구조와 동일하며, 이를 그림으로 나타내면 그림 4과 같다.

학습알고리즘으로는 에러역전파 학습알고리즘을 이용하여 교사학습을 수행한다. 에러역전파 학습알고리즘에 대한 식은 다음의 식(13)-(15)로 표현한다.

$$O_j^k = f \left(\sum_{i=1}^m w_{ij}^k x_i^{k-1} + \theta_j^k \right) = f(\text{net}_j) \quad (13)$$

이때 O_j^k는 k층의 j번째 출력값이고 w_{ij}^k는 k층의 j번째 뉴런과 k-1층의 i번째 뉴런의 연결가중치를 나타낸다. 또한 x_i^{k-1}은 k-1층의 출력을 나타낸다.

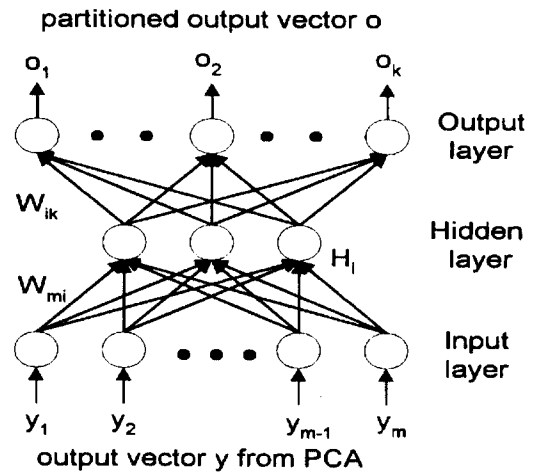


그림 4. 컴퍼넌트 에러역전파 신경망의 구조
 Fig. 4. CBP(Component Back Propagation) Neural Network Architecture.

출력함수로는 시그모이드 함수를 사용한다. 다음으로 에러역전파 학습의 가중치 갱신은 다음과 같은 수식을 따른다.

$$w(t+1) = w(t) + \Delta w = w(t) + \eta \delta x_i \quad (14)$$

여기서 η는 학습율(learning rate)을 나타내며 δ는 에러율을 나타낸다.

$$\begin{aligned} \delta_j &= (d_j - o_j) f'(\text{net}_j) \text{ if } j : \text{output unit} \\ \delta_j &= f'(\text{net}_j) \sum_i \delta_i w_{ij} \text{ if } j : \text{hidden unit} \end{aligned} \quad (15)$$

위의 식(15)에서 d_j는 교사신호를 나타낸다.

제안한 CBP 신경망의 크기는 주어진 영역의 패턴들을 학습할 수 있는 최소의 크기로 결정되어지며, 일반적으로 학습이 이루어 지지 않는 경우 중간 뉴런의 크기를 조정하여 학습을 용이하게 할 수 있다. 그러나 본 연구에서는 학습이 잘 이루어 지지 않는 경우 새로운 CBP 신경망을 동적으로 부가시켜 학습시킴으로써 학습의 효율을 높일 수 있다.

3. 제안한 MCNN 신경망 학습 알고리즘

제안한 MCNN 신경망의 학습은 크게 두가지의 단계에 의해서 학습이 이루어 진다. 첫 번째 단계에서는 원 입력벡터의 차원을 줄이고, 패턴들을 영역별로 분할하는 과정이며, 첫 번째 단계에서 신경망의 상태가 안정 상태에 이르렀을 때, 즉 가중치의 갱신이 거의 이루어

어지지 않는다고 판단한 경우, 두 번째 단계를 수행한다. 이때 PCA신경망의 안정 상태 SSV(Stable State Value)의 판단은 다음과 같은 식(16)에 의해서 결정된다.

$$SSV = \frac{1}{N} \left(\sum_{i=1}^N y_i(k+1) - \sum_{i=1}^N y_i(k) \right) < \gamma \quad (16)$$

위의 수식에서 k 는 반복횟수를 N 은 주성분의 수 즉 PCA의 출력벡터 크기를 나타낸다. 이때 γ 의 값은 초기화시에 결정되어지며 본 연구에서는 $10^{-3} \sim 10^{-4}$ 사이의 값으로 결정하였다. 두 번째 단계는 첫 번째 단계를 통하여 출력된 주성분 값을 입력패턴으로 하여서 각 CBP 신경망에서 학습하는 단계이다. 각 CBP의 학습은 독립적으로 이루어지며 병렬적으로 학습한다. 전체 입력패턴들이 분할되어 학습이 진행되므로 전체 원 출력벡터 또한 입력패턴의 영역별 분류에 따라서 분류되어지며, 이때 출력벡터의 결정방법은 주어진 문제의 성격에 따라서 다를 수 있다.

CBP 신경망의 학습시 학습이 잘 되지 않는 신경망의 패턴분할에 따른 새로운 CBP 신경망의 생성은 다음의 식 (17)을 따른다.

$$Error = \frac{1}{m} \sum_{i=1}^m Error[CBP_i]$$

$$\text{if } Error \times \beta \leq Error[CBP_i] \text{ then new } CBP_i \quad (17)$$

$$\text{else old } CBP_i$$

위의 수식에서 전체 신경망의 에러값 $Error$ 는 각 신경망 CBP_i 의 에러값을 더하여 전체 CBP 신경망의 수 m 으로 나눈값이다, 또한 β 의 값은 상수값으로 문제에 따라 달라질 수 있으며, 문제의 복잡도가 높을수록 β 의 값은 감소한다. 학습시 초기의 에러값은 크게 변동되어질 수 있으므로 위의 수식을 초기에 적용할 경우 매우 많은 CBP 신경망의 생성을 필요로 한다. 따라서 본 연구에서 새로운 CBP 신경망의 생성은 전체 신경망의 에러가 어느정도 감소된 이후에 생성하는 것으로 제한하였다. 새로운 CBP 신경망의 생성방법은 문제의 복잡도에 따라 전체 신경망의 크기를 학습시 동적으로 변화시킬 수 있으며, 초기에 가장 작은 크기의 신경망으로부터 출발한다는 점에서 문제의 복잡도에 따른 최적의 신경망을 구성할 수 있다는 장점을 가진다. 본 연구에서 제안한 다중 컴포넌트 신경망의 학습알고리즘을 나타내면 다음과 같다.

단계1) 초기화 과정

PCA 신경망 초기화

(가중치 w 초기화, 학습파라미터 초기화)

CBP 신경망 초기화

(가중치 초기화 w , 학습파라미터 초기화)

단계2) PCA 학습과정

전방향 처리 : [수식 6.]에 의한 주성분 출력

후방향 처리 : [수식 8.] 또는 [수식9.]에 의한 가중치 갱신

단계3) PCA 신경망의 안정 상태 판단

[수식16.]에 의한 안정상태 판단

만약 $SSV < \gamma$ 이면 단계4)로,

아니면 단계2)로 가서 PCA 재 학습수행

단계4) PCA 신경망의 출력값에 대한 패턴의

분할과정 : [수식 10-12.]

단계5) CBP 학습과정

전방향 처리 : [수식 13.]

새로운 CBP 신경망의 생성 결정: [수식 17.]

후방향 처리 : [수식14-15.]

단계6) 학습 완료조건 판단과정

허용 iteratin 판단 : 만족하면 단계7)로,

아니면 단계 5)로 가서 CBP 재 학습

허용 에러의 판단 : [수식17.]의 $Error$ 가

허용에러보다 작으면 단계7)로

아니면 단계5)로 가서 CBP 재 학습

단계7) 학습완료

IV. 모의 실험 및 결과

본 연구에서 제안한 MCNN (Multiple Component Neural Network)의 우수성을 평가하기 위해서 Seven segment 문제, 일반 EBP신경망에서 학습이 어려운 Parity 문제 그리고 일반적인 영문자 인식문제를 대상으로 실험하였으며, 각 대상문제에 대한 학습속도 및 전체 신경망의 크기를 비교하였다. 실험은 Pentium-120 PC상에서 C로 구현된 simulator를 통하여 이루어졌으며 신경망의 크기를 나타내는 평가 파라미터로는 전체 신경망이 갖는 연결가중치의 수를 비교하였다. 또한 학습속도의 평가를 위해서는 전체 학습iteration과 학습시간(sec)을 분리하여 비교하였다. 본 연구에서 제안한 신경망은 각 컴포넌트 신경망의 병렬처리를 기본으로 하기 때문에, 각 컴포넌트 신경망의 학습시간은

각각 다르다. 따라서 MCNN의 전체 학습시간은 PCA 신경망의 학습시간과 컴포넌트 신경망인 CBP의 학습시간을 합친 것으로 계산하였으며, 전체 MCNN의 학습 iteration은 PCA의 학습 iteration을 제외한 컴포넌트 신경망의 학습 iteration만을 고려하였다. 왜냐하면 PCA신경망은 단층신경망으로 구성되며 학습수식이 매우 간단하여 매 학습iteration당 소요되는 시간이 컴포넌트 신경망의 학습iteration과 비교하여 매우 빠르기 때문에 같은 iteration으로 계산하기 어렵기 때문이다. 따라서 전체 학습iteration은 PCA신경망의 입력을 받아서 학습처리하는 컴포넌트 신경망중 가장 학습시간이 오래걸린 신경망의 학습iteration을 기존 EBP학습 iteration과 비교 검토하였다. 허용에러를 판단하는 수식(17)에서의 β 값은 1.0으로 하였으며, 초기 학습 iteration이 200회가 지난후에 조건식을 적용하였다. 본 논문에서 PCA 신경망에서 사용된 학습상수인 α , η 그리고 EBP신경망에서의 모멘텀 α 학습률 η 를 구별하기 위해서 이하에서는 EBP신경망에서의 α 를 ALPHA로, η 를 ETA로 표기한다. EBP신경망과 제안한 MCNN신경망에서의 에러역전파 방법은 매 패턴의 입력시 가중치를 조정하는 on-line처리방법을 선택하였다.

1. 패리티 문제

기존 신경망에서 패리티 문제의 해결은 매우 유사한 입력에 대하여 서로다른 출력을 내야하므로 단일의 신경망 구조인 EBP에서는 학습이 상당히 어려우며 또한 중간 뉴런의 개수는 적어도 입력 뉴런의 2배 이상이 필요하다. 그러나 본 연구에서 제안한 MCNN신경망에서는 전체 입력패턴의 차원을 보다 적은 차원으로 감소시키고 이들 패턴들을 다시 분할하여 각 컴포넌트 신경망인 CBP에서 병렬적으로 학습시킴으로써 매우 빠른 학습을 수행시킬 수 있다. 본 실험에서는 패리티 2문제 (Exclusive-OR문제), 패리티 4문제 그리고 패리티 6문제를 대상으로 비슷한 신경망의 크기를 갖는 기존 EBP신경망과 그 실험결과를 비교 평가하였다. 첫째, 패리티 2문제에 대하여 MCNN 신경망을 구성하는 PCA신경망에서의 각 학습파라미터의 변화에 따른 학습속도에 대한 결과는 그림5, 6 과 같다. 그림 5.에서는 PCA에서의 가중치를 갱신하는데 있어서 학습률인 η 의 변화에 따른 학습iteration을 나타낸다.

실험결과 전체 학습iteration은 학습률 η 의 값이

0.35일 때 74.4회로 가장 빠름을 알 수 있다. 또한 전체 학습시간은 $10^{-2}(\text{sec})$ 이하로 매우 빠르다. η 의 값이 0.55이상을 넘은 경우에는 최대 허용학습 iteration인 3000번을 넘어서 학습이 안됨을 알 수 있었다.

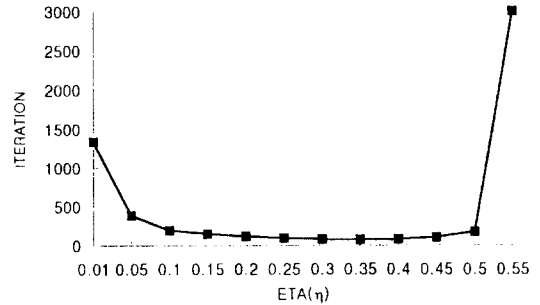


그림 5. η 값의 변화에 따른 PCA신경망 학습iteration

Fig. 5. Learning iterations with various η values on PCA Neural Network.

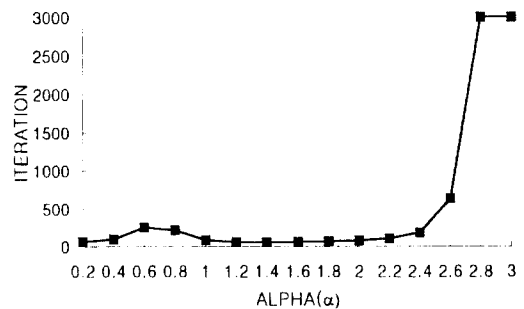


그림 6. α 값의 변화에 따른 PCA신경망 학습 iteration

Fig. 6. Learning iterations with various α values on PCA Neural Network.

그림 6.에서는 PCA에서의 다중 성분의 값, 즉 PCA 신경망의 출력값간의 차이를 변화시키는 역할을 하는 α 의 변화에 따른 학습iteration을 비교하였다. 식(8)에서 알 수 있듯이 α 의 값이 0.0일 때는 각 출력뉴런에서의 출력값은 동일하다. 가장 빠른 학습은 α 의 값이 1.2일때로써 60.8회의 학습iteration이 소요되었으며 이때의 전체 학습시간은 $10^{-2}(\text{sec})$ 이하이다. 따라서 PCA신경망에서의 α 값은 1에서 2사이가 적절함을 알 수 있다. α 의 값이 2.8이상인 경우, 학습시 가중치의 변화가 급격히 변화함으로써 oscillation현상이 발생하였으며 허용학습 iteration인 3000번을 초과하였다.

그림 7에서는 패리티2 문제에 대하여 각 CBP신경망

과 EBP신경망의 학습률 η 와 모멘텀 α 의 변화에 따른 학습속도를 비교하였다. 이때 MCNN에서의 PCA신경망 학습파라미터인 α 는 1.2로, η 는 0.4로 하였으며, 학습안정 상태를 평가하는 SSV(Stable State Value)는 10^{-3} 으로 하였다. 또한 패리티2 문제의 경우 매우 적은 4개의 패턴만이 존재하므로 전체 CBP신경망은 하나로 제한하였다.

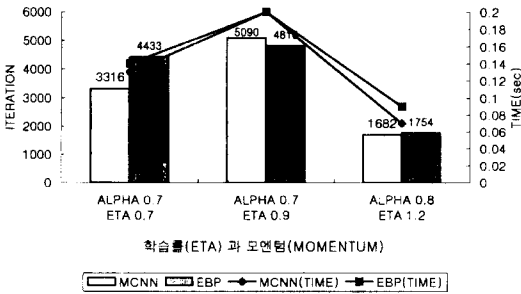


그림 7. 학습상수의 변화에 따른 MCNN과 EBP의 학습속도 비교

Fig. 7. Comparison results for MCNN and EBP with various learning parameters.

실험결과 MCNN신경망과 기존 EBP신경망이 비슷한 학습속도를 나타냄을 알 수 있었다. 그러나 각각 5회씩의 실험결과 MCNN의 경우에는 3회의 학습성공이 있는데 반하여 EBP의 경우 4회의 성공을 나타내어 학습은 EBP의 경우가 잘됨을 알 수 있었다. 그 이유로는 패리티2 문제의 경우 기존 EBP신경망이 이진 입력벡터를 사용하는 반면에 제안한 신경망의 경우에는 PCA신경망의 출력이 아날로그값으로 계산되어지기 때문에 CBP의 입력시 01패턴과 10패턴이 EBP의 패턴보다 유사한 아날로그 패턴으로 변환되어지기 때문이다. 이는 PCA신경망의 출력뉴런의 수를 증가시킴으로써 간단히 해결할 수 있다.

전체 신경망의 크기에 있어서는 EBP의 경우 최소 3개의 가중치연결이 필요한데 비하여, MCNN의 경우 전체 7개의 최소 가중치 연결 (PCA : 4개, CBP: 4개)이 필요로 하여진다. 학습속도는 MCNN의 경우 PCA 학습시간이 거의 무시해도 좋은 정도의 매우 빠른 학습을 나타내기 때문에 전체 학습시간은 EBP보다 약간 빠름을 보여준다. 결과적으로 α 가 0.8이고 η 가 1.2일 때 제안한 MCNN신경망이 전체 학습 iteration 1682회와 0.07sec의 학습속도로 가장 좋은 학습성능을 나타냄을 알 수 있다.

다음으로는, 패리티 4문제에 대하여 본 MCNN신경망과, EBP신경망을 비교 검토하였다. 표 1에서는 제안한 신경망과 이를 구성하는 각 PCA신경망, CBP신경망 그리고 기존 EBP신경망의 크기를 나타낸다.

표 1. 패리티 4문제 신경망 크기 비교
Table 1. Comparison results of network size of parity 4 problem.

Nets.	MCNN network				EBP network			
	PCA	CBP1	CBP2	CBP3	CBP4	EBP1	EBP2	EBP3
Size (con.)	8	12	12	12	12	40	45	50
Total Size (con.)	56					40	45	50

(I*J*K) : I : input neuron, J: hidden neuron, K: output neuron

이때 각 CBP신경망의 입력뉴런수는 2개로 중간 뉴런수는 동일하게 결정하였으며 전체 CBP신경망의 수는 4개로 고정하였다. MCNN에서의 PCA신경망 학습 파라미터인 α 는 2.0으로, η 는 0.01로 하였으며, 학습안정 상태를 평가하는 SSV는 10^{-3} 으로 하였다.

위의 표 1에서 제안한 MCNN의 경우 전체 신경망의 크기는 56개로 비교된 EBP신경망의 크기와 비슷함을 알 수 있다. 그러나 학습속도는 제안한 신경망이 보다 매우 빠르며, EBP1과 EBP2 신경망은 허용학습 iteration인 50,000을 초과함으로써 학습이 제대로 이루어 지지 않았다. 패리티4 문제에 대한 각 신경망의 학습속도를 나타내면 그림 8,9와 같다.

위의 그림 8에서 MCNN 신경망의 학습iteration은 CBP4 신경망의 학습iteration과 동일하며, 따라서 CBP2 신경망이 컴포넌트 신경망 중 가장 늦게 학습이 되었음을 알 수 있다. 기존 EBP신경망과의 비교에 있어서 EBP3 신경망의 경우에 학습iteration이 41903회로 제안한 신경망이 약 15.3배 빠름을 알 수 있다.

다음 그림 9에서는 각 신경망에 대한 학습시간을 비교하였다. 제안한 MCNN신경망의 경우 전체 학습시간은 0.222 초로 EBP3신경망에 비하여 31.4배의 속도향상이 나타났다. 학습 iteration과 학습속도의 차이는 약 2배로 이는 제안한 신경망을 구성하는 CBP신경망의 병렬성으로 인하여 한 번 iteration에 소요되는 시간이 EBP신경망보다 매우 빠르기 때문이다.

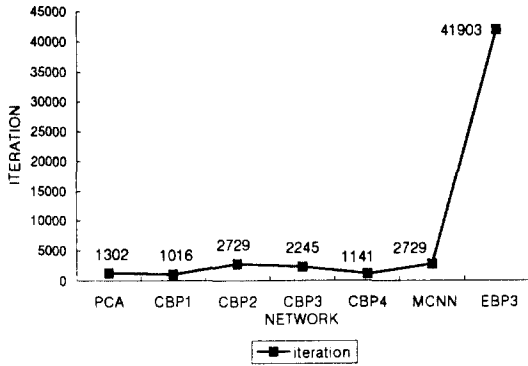


그림 8. 패리티4 문제 각 신경망의 학습iteration 비교

Fig. 8. Comparison results of learning iterations for each neural networks.

마지막으로 64개의 패턴을 갖는 패리티 6문제에 대한 학습성능을 평가하였다. 표 2에서는 제안한 신경망과 이를 구성하는 각 PCA신경망, CBP신경망 그리고 기존 EBP신경망의 크기를 나타낸다. 각 CBP신경망의 중간뉴런수는 입력뉴런수의 2배로 중간 뉴런수를 동일하게 결정하였으며 전체 CBP신경망의 수는 EBP신경망과의 비교를 위하여 8개로 고정하였다.

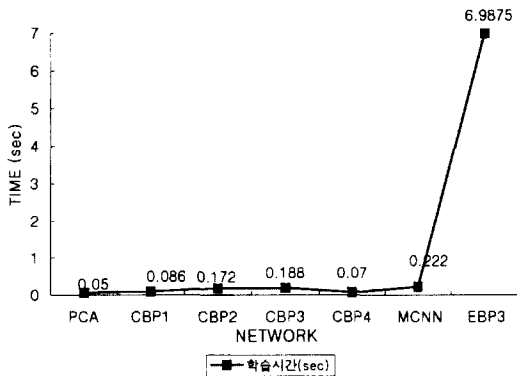


그림 9. 패리티4 문제 각 신경망의 학습시간 비교

Fig. 9. Comparison results of learning time for each neural networks.

제안한 MCNN에서의 PCA신경망 학습파라미터인 α 는 2.0으로, η 는 0.001로 하였으며, 학습안정 상태를 평가하는 SSV는 10^{-3} 으로 하였다. η 의 값이 패리티4 문제 보다 낮은 것은 보다 작은 값으로 가중치를 변화시켜 안정된 상태를 유지하기 위해서이다. η 의 값을 0.01로 한 경우에는 학습이 되지 않았다.

표 2. 패리티 6문제 신경망 크기 비교

Table 2. Comparison results of network size of parity 6 problem.

ONets.	MCNN network		EBP network		
	PCA	CBP1 ~ CBP8	EBP1	EBP2	EBP3
Size (con.)	18	24*8	70	140	210
Total Size (con.)	210		70	140	210

(I*J*K) : I : input neuron, J: hidden neuron, K : output neuron

위의 표 2에서 MCNN의 경우 전체 신경망의 크기는 210개로 비교된 EBP3신경망의 크기와 같다. 그러나 EBP1-3까지의 대상 신경망은 모두 허용iteration 50,000번을 초과 함으로써 학습이 되지 않았다. 그러나 제안한 MCNN신경망의 경우 전체 학습iteration 5716.8회 그리고 전체 학습시간 1.6초의 빠른 속도로 학습을 하였음을 알 수 있다. 이를 그래프로 나타내면 그림 10.과 같다.

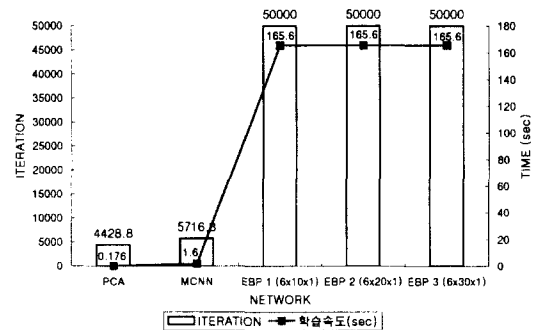


그림 10. 패리티 6 문제 학습속도 비교

Fig. 10. Comparison results of learning speed for each neural networks.

결과적으로 문제의 크기가 증가할수록, 또는 입력패턴들이 서로 유사할수록 전체 패턴에 대하여 전체 가중치 값이 적용되는 단일 구조의 신경망에서는 해결하기가 상당히 어려움을 알 수 있다. 그러나 제안한 신경망에서는 입력패턴의 차원크기를 줄이고, 이들 패턴들을 각각의 컴포넌트 신경망에 나누어 학습시킴으로써 매우 빠른속도로 학습할 수 있으며, 또한 전체 신경망의 학습효율을 증가시킬 수 있다.

2. Seven Segment 인식 및 영문자 인식문제

위의 패티터 문제에서는 비슷한 신경망의 크기를 갖는, 제한한 신경망과 기존 EBP신경망의 학습속도에 대하여 평가하였다. seven segment 인식 문제와 영문자 인식문제에서는 본 연구에서 제안한 MCNN신경망의 동적 컴포넌트 신경망의 부가와 이에 따른 전체 신경망의 크기를 EBP신경망과 비교 평가하였다. 표 3에서는 seven segment 인식문제에서 대상으로하는 전체 신경망의 크기를 비교하였다. 이때 MCNN신경망을 구성하는 각각의 PCA신경망의 크기는 입력뉴런 10개 출력뉴런 2개로 전체 20개의 가중치 연결을 가진다.

표 3. Seven segment문제 신경망 크기 비교
Table 3. Comparison results of network size of Seven segment problem.

Nets.	MCNN network				EBP network					
	MCN N1	MCN N2	MCN N3	MCN N4	EBP 1	EBP 2	EBP 3	EBP 4	EBP5	EBP6
Total Size (con.)	68	84	100	118	85	102	119	136	153	170

그림 11은 식 (10)-(12)를 이용하여 원 입력패턴들이 PCA신경망의 학습을 통하여 전체 3개의 컴포넌트 신경망인 CBP1,2,3로 분할되는 과정을 나타낸다. 이때 제3영역과 제4영역의 패턴은 하나의 신경망상에서 학습되었다. 왜냐하면 제4영역의 패턴의 수가 1개이므로 단일의 컴포넌트신경망 상에서의 학습이 불가능하기 때문이다. 또한 각 영역을 결정하는 threshold θ 는 각 컴포넌트 신경망으로 입력되는 패턴값을 식(10)과 식(11)에 의하여 분할하여 Y_1 에 대한 θ_1 과 Y_2 에 대한 θ_2 로 결정하였다. 따라서 CBP1에서는 4개의 패턴을 학습하게 되며 CBP2,3에서는 각각 3개의 패턴이 학습하게 된다.

그림 12와 그림 13에서는 표 1의 MCNN4과 EBP3 신경망을 대상으로 학습파라미터의 변화에 따른 학습 iteration과 학습속도를 비교평가 한다. 그림 12와 그림 13에서 전체 가장 좋은 성능을 보인 경우는 학습률 ETA 가 1.5이고, 모멘텀 $ALPHA$ 가 0.8인 경우에 MCNN4가 학습iteration 4620, 학습시간 1.7초로 EBP3보다 빠르게 학습되었음을 알 수 있다. 이때 전체 신경망의 크기는 MCNN4가 118개의 연결을 그리고 EBP6의 경우 170개의 연결을 가지므로 약 31%의

네트워크 크기의 감소를 가진다.

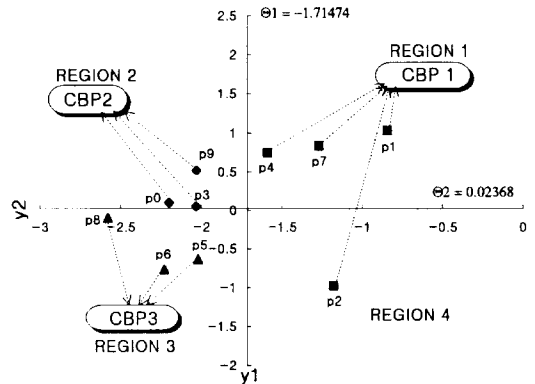


그림 11. PCA출력과 영역결정에 의한 각 CBP 신경망의 패턴분할
Fig. 11. Pattern distribution of each CBP neural networks by PCA output and region Determination.

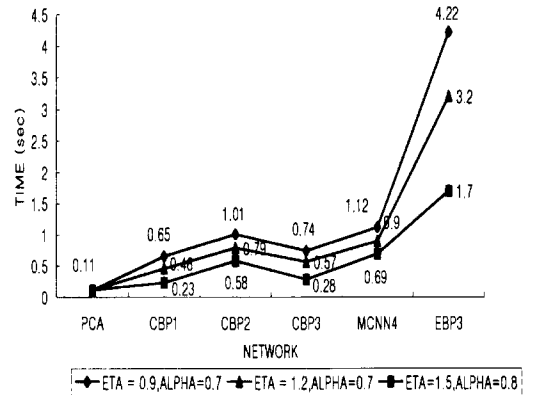


그림 12. 학습률 및 모멘텀의 변화에 따른 학습 iteration 비교
Fig. 12. Comparison results of learning iteration with learning rate & momentum.

중간 뉴런수의 증가에 따른 MCNN신경망과 EBP신경망의 비교평가는 그림 14와 같다. 이때 학습iteration은 두가지 모두 비슷하였지만, 제안한 MCNN신경망에서는 분할되어진 입력패턴에 대한 병렬 학습을 수행하므로 기존 EBP신경망과 비교하여 매우 빠른 학습시간을 나타냄을 알 수 있다.

그림 14에서 EBP4의 신경망이 EBP5, EBP6신경망에 비하여 느린 학습iteration을 나타냄에도 학습시간이 3.7초로 빠른 이유는 전체 신경망의 크기가 두 신경

망보다 작기 때문이다.

표 4에서는 영문자 인식을 대상으로 MCNN신경망과 EBP신경망의 전체 신경망 크기를 비교한다. MCNN신경망에서 초기 영역의 결정은 3개의 입력을 바탕으로 8개의 영역이 결정되어지나, 패턴을 포함하지 않은 영역과 하나의 패턴만이 존재함으로써 학습이 불가능한 영역을 인접영역에 포함하므로써 초기의 5개의 비교적 크기가 큰 CBP신경망에서 최종11개의 작은 크기의 CBP신경망으로 분할되어 진다. 따라서 초기 신경망의 크기는 530개의 연결을 갖지만, 최종 신경망의 크기는 362개로 크게 단축되어 진다.

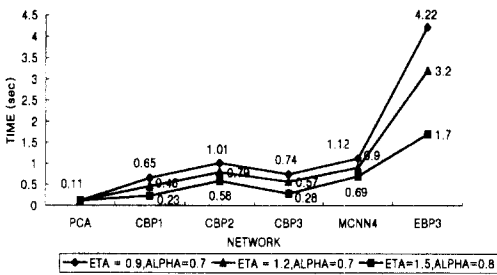


그림 13. 학습률 및 모멘텀의 변화에 따른 학습시간 비교

Fig. 13. Comparison results of learning time for MCNN and EBP with learning rate & momentum.

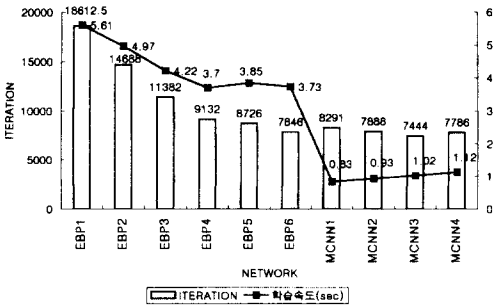


그림 14. 중간 뉴런수의 변화에 따른 학습속도 비교()안은 중간뉴런수

Fig. 14. Comparison results of learning speed for MCNN and EBP with hidden neurons increase.

그림 15에서는 영문자 인식에 대한 MCNN신경망과 EBP신경망 사이의 학습속도를 비교 평가한다. 그림 15에서 PCA신경망은 25개의 입력뉴런을 가지며 따라서 학습속도에 있어서 1.24초로 위의 다른 실험대

상문제 보다 다소 늦은 학습속도를 가진다. MCNN신경망은 전체 6652회의 학습 iteration 과 2.05초의 학습속도로 EBP신경망 중 가장 빠른 EBP1 신경망보다 속도면에서 약 8배의 속도향상을 보였다. 결과적으로 Seven segment 인식 문제 및 영문자 인식문제에서는 EBP보다 훨씬 적은 신경망의 가중치연결을 가지면서도 본 MCNN 신경망이 보다 빠른 속도로 학습을 수행함을 알 수 있었다.

표 4. 영문자 인식 문제 신경망 크기 비교
Table 4. Comparison results of network size of Alphabet character recognition problem.

Nets.	MCNN network		EBP network		
	Initial MCNN	last MCNN	EBP1	EBP2	EBP3
Total Size	530	362	765	663	510
(con.)					

제한한 MCNN신경망은 실험결과 기존 EBP신경망에 비하여 매우 빠른 학습속도를 얻을 수 있었으며, 전체 신경망의 크기가 또한 크게 감소되었다. 또한 영역별 학습에 따른 패턴의 분할학습을 독립적인 컴포넌트 신경망에서 병렬 수행함으로써 신경망의 구조가 학습의 난이도에 따라서 적응적으로 변화 할 수 있는 장점을 가진다.

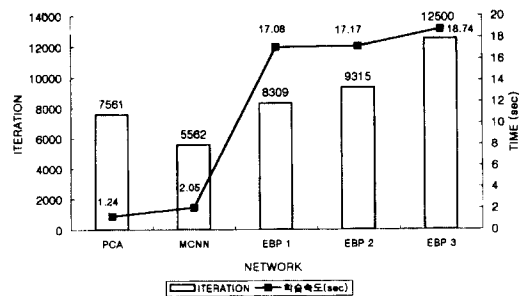


그림 15. 영문자 인식 학습속도 비교
Fig. 15. Comparison results of learning speed for each neural networks.

패리티 문제처럼 입력패턴간의 유사성이 많음으로 인하여 학습이 잘되지 않는 패턴들의 경우 이를 분할하여 지역적인 학습을 수행 할 수 있으며, 이를 보강하

기 위해서 PCA의 출력뉴런수를 증가시킴으로써 패턴 간의 유사도의 차이를 늘려서 문제를 좀더 쉽게 해결할 수 있다. 그러나 PCA출력인 주성분값이 반드시 원 입력패턴들 사이의 유사도를 보장하지는 않으므로, 유사한 패턴간의 재 분할 학습을 수행하기 위한 앞으로의 연구가 필요하며, 또한 비학습패턴에 대하여 올바른 출력을 내 주기 위해서는 좀 더 향상된 패턴 분할방법에 대한 연구가 진행되어야 한다.

V. 결 론

본 연구에서는 PCA (Principal Component Analysis)방법을 이용하여 입력벡터의 차원을 감소시켜 작은 차원을 갖는 주성분 패턴벡터로 표현한 후 다중의 컴포넌트 신경망인 CBP에서 패턴들을 영역별로 분할 학습하는 새로운 신경망을 제안하였다.

또한 제안한 신경망에서는 학습이 어려운 CBP신경망이 가지는 패턴을 재 분할하여 새로운 CBP 신경망을 생성하고 학습시킴으로써 동적인 신경망의 구성이 가능하도록 하였으며, 각 CBP 신경망의 학습은 독립적이고 병렬적으로 수행할 수 있도록 하였다.

실험결과 기존 다층 신경망상에서의 에러역전파 학습알고리즘과 비교하여 학습속도를 크게 향상 시켰으며, 전체 신경망의 크기 또한 크게 감소됨을 알 수 있었다. 제안한 신경망은 특히 대규모의 데이터와, 입력력벡터의 크기를 갖는 문제에 대해서 매우 효율적이며, 특히 데이터의 압축 및 복원 문제에 효과적으로 적용되어질 수 있다.

앞으로의 연구방향은 PCA신경망의 출력으로 나오는 패턴들의 출력값, 즉 주성분값에 대한 분할문제를 연구하여 좀 더 향상된 패턴 분할방법을 제시하고 이를 제안한 신경회로망에 적용시키는 것이다. 또한 제안한 MCNN 신경망에 대한 하드웨어 구현시의 타당성과 효율성에 대한 앞으로의 연구가 진행될 것이다.

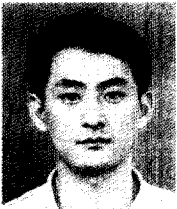
참 고 문 헌

- [1] S.Haykin, Neural Network A Comprehensive Foundation, Mcmillan Publishing Co., 1993.
- [2] J.M. Zurada, Introduction to Artificial Neural Systems, WEST Publishing Co., 1992.
- [3] K.A.Marko, J.James, J.Dosdall, and J.Murphy, "Automotive Control System Diagnostics Using Neural Nets for Rapid Pattern Classification of Large Data Sets," in Proc. 2nd Int. IEEE Joint conf. On Neural Networks, Washington D.C., pp 13-17, 1989.
- [4] A.Cichocki and R.Ubenhauen, Neural Networks for Optimization and Signal Processing, WILEY Publishing co., 1992.
- [5] T.Samad, "Backpropagation improvements based on heuristic arguments," Proc. of IJCNN, Washington D.C., pp 565-568, 1990.
- [6] T.Vogl and Mangis, "Accelerating the convergence of the backpropagation method", Biol., Cybernetics, pp 257-263, 1988.
- [7] C.H.kwan, C.H.Park and H.S.Lee, "Efficient strategies for Error Updating to Improve Performance Backpropagation Learning," Proc. of ICNN, Perth Australia, pp 2825-2830, 1995.
- [8] J.Park and I.W.Sandberg, "Universal Approximation using Radial Basis Function Networks", Neural Computation, vol 3, pp 246-257, 1991.
- [9] R.A.Jacobs and M.I.Jordan, "A Competitive modular connectionist Architecture," Advances Neural Information Processing Systems 3, pp 763-773, San Diego, CA: Morgan aufmann, 1991.
- [10] Y.H.Pao, Adaptive Pattern Recognition and Neural Networks, Addison Wesley Co., pp 197-222, 1989.
- [11] E.Oja, "Principal components, minor components and linear neural networks," Neural Networks, vol 5, pp 927-935, 1992.
- [12] T.D.Sanger, "An optimality principle for unsupervised learning," Advances in Neural Information Processing Systems, D.S., Touretzkey editor, Morgan Kaufman, pp 11-19, 1989.
- [13] J.hertz, A.Krogh and R.G.Palmer, Introduction to the Theory of Neural Computation, Addison Wesley Co., 1991.
- [14] E.,Oja, "A simplified neuron model as a principal component analyzer", J.Math. Biology, vol 15, pp 267-273, 1982.

[15] R.J.Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches,

John Wiley and Sons, Inc., 1991.

— 저 자 소 개 —



朴 贊 虎(正會員)

1991년 경희대학교 전자계산공학과 졸업(공학학사), 1993년 경희대학교 전자계산공학과 대학원 석사과정 졸업(공학석사), 1993년 ~ 현재 경희대학교 전자계산공학과 박사과정, 주관심분야는

신경회로망, 패턴인식, 컴퓨터 구조, 병렬처리 등임



李 顯 洙(正會員)

1979년 경희대학교 전자공학과 졸업(공학학사), 1982년 일본 경응의속대학 전기공학과 석사과정 졸업(공학석사), 1985년 일본 경응의속대학 전기공학과 박사과정 졸업(공학박사), 1985년 ~ 현재 경희대학교 전

자계산공학과 교수, 주관심분야는 Parallel Processing, VLSI architecture 설계 및 신경망 컴퓨터 등임