

論文 99-36C-12-9

온닉노드의 생성·소멸에 의한 웨이블릿 신경망 구조의 자기 조직화

(A Self Organization of Wavelet Network Structure by Generation and Extinction of Hidden Nodes)

林成吉*, 李顯洙*

(Sung-kil Lim and Hyon-soo Lee)

요약

기존의 웨이블릿 신경망들의 구조는 주로 주파수·시간 공간으로 변환된 훈련 패턴의 분포와 웨이블릿 원도우와의 관계를 고려하여 결정한다. 또한 신경망 구조 결정 알고리즘과 네트워크 파라미터 학습 알고리즘을 분리하여, 우선 신경망 구조를 결정한 후, 출력 에러를 최소화하기 위한 학습을 수행한다. 그러나 이러한 방법은 학습을 시작하기 전에 훈련 패턴을 변환해야 하는 부가적인 전처리 과정이 필요하고, 초기에 구성된 신경망 구조는 변경되지 않는다는 단점을 가지고 있다.

본 논문에서는 별도의 처리 과정 없이 신경망의 출력과 교사 신호의 차이를 이용하여 웨이블릿 신경망 구조를 결정하는 방법을 제안한다. 제안하는 알고리즘은 네트워크 구조의 결정과 에러 최소화 학습을 동시에 수행하기 때문에 문제의 복잡도에 따라 적응적으로 온닉 노드의 수를 결정한다. 또한 학습에 의해 가장 큰 에러가 발생하는 영역에 온닉 노드를 추가하고 출력에 영향을 미치지 않는 노드를 제거하는 방법을 사용하여 네트워크의 구조를 최적화한다. 본 알고리즘은 훈련 패턴에 대한 전처리 과정을 없음으로써 학습하기 전에 모든 훈련 패턴을 알고 있어야 한다는 제약 조건을 없애고 시간의 변화에 따라 출력이 바뀌는 시스템에도 효과적인 적용이 가능하다.

Abstract

Previous wavelet network structures are determined by considering the relationship between wavelet windows and distribution of training patterns that are transformed into time-frequency space. Because it is separated two algorithms that determines wavelet network structure and that modifies parameters of network, learning process that minimizes output error of network is executed after the network structure is determined. But this method has some weaknesses that training patterns must be transformed into time-frequency space by additional preprocessing and the network structure should be fixed during learning process.

In this paper, we propose a new constructing method for wavelet network structure by using differences between the output and the desired response without preprocessing. Because the algorithm perform network construction and error minimizing process simultaneously, it can determine the number of hidden nodes adaptively as with the complexity of problems. In addition, the network structure is optimized by inserting new hidden nodes in the area that has maximum error and extracting hidden nodes that has no effect to the output of network. This algorithm has no constraint condition that all training patterns must be known, because it removes preprocessing procedure for training patterns and it can be applied effectively to systems that has time varying outputs.

* 正會員, 慶熙大學校 電子計算工學科

(Department of Computer Engineering, Kyung-Hee University)

※ 이 논문은 (1988)년 한국학술진흥재단의 학술연구비

에 의하여 지원되었음.

接受日字: 1999年3月23日, 수정완료일: 1999年11月9日

I. 서 론

웨이블릿 신경망(Wavelet Network)은 1990년대 초반에 에러 역전파 학습 알고리즘을 사용하는 다층 퍼셉트론 (Multi-Layer Perceptrons : MLP)의 단점을 보완하기 위해 함수 해석 도구 및 신호 처리 도구로 활발히 연구되고 있는 웨이블릿 이론을 신경망 구조에 적용한 모델이다^[1-3]. MLP는 패턴 인식 및 분류, 비신형 시스템의 제어, 음성 인식 등의 응용 분야에서 기존의 일반적인 알고리즘으로는 처리하기 어려운 문제들을 효과적으로 해결하고 있다. MLP는 실용적인 측면 뿐만 아니라 이론적으로도 $L^2(\mathbb{R})$ 공간상의 임의의 함수를 근사(approximation)할 수 있으며, 전역 근사 성질(universal approximation property)을 가지고 있음이 밝혀졌다^[4]. 그러나, MLP는 지역해에 수렴할 가능성이 많고, 4네트워크 파라메터의 초기화가 어렵다. 또한 네트워크의 근사 능력을 좌우하는 은닉 노드의 수를 결정하기 어렵다는 단점이 있다.

웨이블릿 신경망은 전방향 다층 신경망 구조를 가지고 있고 최급강하법(gradient descent method)에 의한 교사 학습을 수행하며, 은닉 노드의 활성화 함수로 웨이블릿 함수를 사용한다. 웨이블릿 신경망은 활성화 함수가 지역 함수의 성격을 가지고 있으므로, 학습에서 필요로 하는 계산량이 적고, 학습의 수렴이 빠르다. 또한 웨이블릿 변환을 이용하면 초기 연결 가중치를 결정할 수 있고, 프레임 성질(frame property)을 이용하여 은닉 노드의 수를 결정할 수 있다는 장점을 가지고 있다^[5-6].

웨이블릿 신경망에서 은닉 노드의 집합이 프레임이 되도록 은닉층을 구성하면 $L^2(\mathbb{R})$ 에 포함되는 모든 함수를 근사할 수 있지만, 신경망의 크기가 지나치게 커진다는 단점을 가지고 있다. 필요 이상으로 은닉 노드가 많은 신경망은 조정되어야 할 파라메터의 수가 많아지므로 학습에 많은 계산량이 필요하고 일반화 성능이 저하된다. 따라서 이와 같은 문제점을 해결하기 위해서는 웨이블릿 신경망 구조의 최적화가 필요하다.

웨이블릿 신경망 구조의 최적화에 대한 기존의 연구는 시간-주파수 공간으로 변환된 패턴의 분포를 신경망의 구조 결정에 사용하는 방법[8-9]이 주로 사용되고 있다. 이 방법들의 단점은 혼련 패턴의 변환을 위해 별

도의 전처리 과정이 필요하고 신경망의 구조를 결정하기 전에 모든 혼련 패턴이 제시되어야 한다는 것이다.

본 연구에서는 문제의 복잡도에 따라 적응적으로 은닉 노드의 수를 결정하며, 혼련 패턴에 대한 전처리 과정이 필요 없는 웨이블릿 신경망의 구조 결정 방법 및 학습 방법을 제안한다. 본 방법에서는 신경망의 출력과 교사 신호의 차이를 이용하여 새로 추가될 은닉 노드의 위치를 결정하며, 최급강하 학습을 통하여 웨이블릿 함수의 크기와 연결 가중치를 결정한다. 은닉 노드의 추가는 에러가 충분히 작아질 때까지 이루어지며, 학습 도중 출력에 미치는 영향이 작은 노드들은 삭제한다. 본 방법에 의해 생성되는 신경망은 학습이 진행됨에 따라 은닉 노드의 추가와 삭제가 반복되면서 점차적으로 신경망의 크기가 증가하므로 자기 성장 웨이블릿 신경망(Self-Growing Wavelet Neural Network : SGWNN)이라 부른다.

본 논문의 II 장에서는 웨이블릿 신경망의 일반적인 구조와 학습에 대해 기술하고 기존의 웨이블릿 구조 최적화 방법에 대하여 고찰한다. III 장에서는 SGWNN의 구조와 학습 방법을 제안한다. IV 장에서는 함수 근사 문제를 대상으로 한 시뮬레이션을 통하여 SGWNN의 성능을 평가하고, 마지막으로 V 장에서는 본 연구의 결론을 내리고 향후 연구 방향을 제시한다.

II. 웨이블릿 신경망

1. 웨이블릿 신경망 구조

1990년대 초에 제안된 웨이블릿 신경망은 웨이블릿 이론을 신경망의 구조에 적용한 모델로 그림1과 같이 3층의 전방향 구조를 가지고 있다. 각 층간의 연결은 완전 연결이며 활성화 함수로는 입력과 출력층에서는 선형 함수를, 은닉층에서는 웨이블릿 함수를 사용한다. 웨이블릿 신경망의 출력은 다음 식(1)과 같이 정의된다.

$$\hat{y}(t) = \sum_{k=1}^K w_k \psi_{a_k, b_k}(t) \quad (1)$$

\hat{y} : 신경망의 출력 (근사 신호)

K : 은닉 노드의 수

k : k 번째 노드

w_k : 중간층과 출력층의 연결 가중치

ψ_{a_k, b_k} : 활성화 함수

식(1)에서 ϕ_{a_k, b_k} 는 허용(admissible) 조건을 만족하는 모 웨이블릿(mother wavelet) 함수의 변형으로써 다음과 같은 함수를 나타낸다.

$$\phi_{a_k, b_k} = \frac{1}{\sqrt{a_k}} \psi\left(\frac{t - b_k}{a_k}\right) \quad (2)$$

ϕ : 모 웨이블릿 함수

a_k : 웨이블릿 함수의 확장(dilation) 계수

b_k : 웨이블릿 함수의 이동(translation) 계수

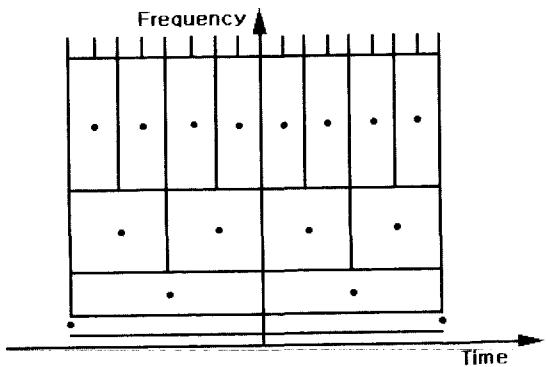


그림 2. 시간-주파수 공간에서의 웨이블릿 원도우
Fig. 2. Wavelet windows in time-frequency space.

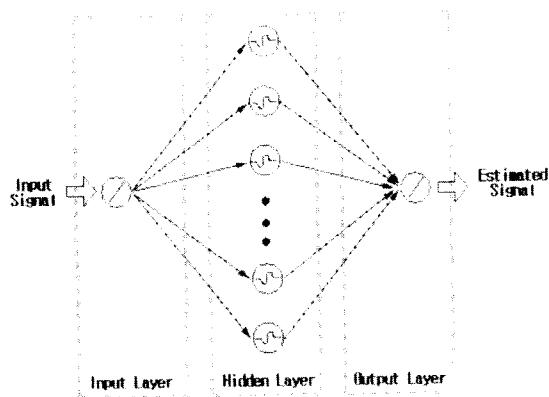


그림 1. 일반적인 웨이블릿 신경망의 구조

Fig. 1. Wavelet network structure in general case.

웨이블릿 함수는 시간-주파수 공간에서 직사각형의 원도우를 갖는다. 웨이블릿 함수 $\phi_{a,b}$ 의 원도우는 다음과 같이 정의된다.

$$[b - ax_{\min}, b + ax_{\max}] \times [\lambda_{\min}/a, \lambda_{\max}/a] \quad (3)$$

이때 훈련 패턴 집합의 원소 x_{\min}, x_{\max} 와 변환된 훈련 패턴의 주파수 성분 $\lambda_{\min}, \lambda_{\max}$ 은 $\forall \varepsilon > 0$ 에 대하여 다음 식(4)의 조건을 만족해야 한다. 식(4)의 조건은 웨이블릿 함수의 에너지가 대부분 원도우 안에 집중되어 있음을 의미한다.

$$1 - \varepsilon < \frac{\int_{x_{\min}}^{x_{\max}} |\psi(x)|^2 dx}{\int_R |\psi(x)|^2 dx} \quad (4)$$

and $1 - \varepsilon < \frac{\int_{\lambda_{\min}}^{\lambda_{\max}} |\Psi(\lambda)|^2 d\lambda}{\int_R |\Psi(\lambda)|^2 d\lambda}$

Ψ : ψ 의 푸리에 변환

그림 2는 $a = 2^i, b = j$ ($i, j \in \mathbb{Z}$)일 때, 웨이블릿 함수의 원도우를 나타낸 것이다.

2. 웨이블릿 신경망 학습

웨이블릿 신경망의 학습은 LMS 알고리즘을 사용한다. 네트워크의 구성에서 초기화된 파라미터들은 최급강하법에 의해 출력에러를 최소화하도록 학습된다^[7]. 웨이블릿 신경망의 온너 노드는 해당하는 웨이블릿 함수의 원도우 안에 포함되어 있는 패턴에 대해서만 학습을 하는데 이러한 학습을 지역적인 학습이라 한다. 최소화되는 출력 에러 함수의 수식은 다음과 같다.

$$E = \frac{1}{2} \sum_{i=1}^p [y(t_i) - \hat{y}(t_i)]^2 \quad (5)$$

$y(t_i)$: 교사 신호

$\hat{y}(t_i)$: 신경망의 출력

p : 훈련 패턴의 수

최급강하법을 사용하여 E 를 최소화하기 위해서는 식(5)를 각각의 파라미터로 편미분하여 변화량을 구한다. 각각의 파라미터에 대한 E 의 편미분은 식(6)과 같다.

$$\frac{\partial E}{\partial w_k} = - \sum_{i=1}^p (y(t_i) - \hat{y}(t_i)) \psi_{a_k, b_k}(t_i) \quad (6)$$

$$\frac{\partial E}{\partial b_k} = - \sum_{i=1}^p (y(t_i) - \hat{y}(t_i)) w_k \frac{\partial \psi_{a_k, b_k}(t_i)}{\partial b_k}$$

$$\frac{\partial E}{\partial a_k} = - \sum_{i=1}^p (y(t_i) - \hat{y}(t_i)) w_k \frac{\partial \psi_{a_k, b_k}(t_i)}{\partial a_k}$$

각각의 파라미터의 변화량은 식(7)과 같다.

$$\Delta w_k = - \frac{\partial E}{\partial w_k}, \Delta b_k = - \frac{\partial E}{\partial b_k}, \Delta a_k = - \frac{\partial E}{\partial a_k} \quad (7)$$

최급강하법에 의해 생성된 파라미터들은 식(8)과 같다.

$$w_k^{n+1} = w_k^n + \alpha_w \Delta w_k \quad (8)$$

$$b_k^{n+1} = b_k^n + \alpha_b \Delta b_k$$

$$d_k^{n+1} = d_k^n + \alpha_d \Delta d_k$$

이때 α_w , α_b , α_d 는 각 파라메터에 대한 학습률이고 각 파라메터의 위 첨자는 학습의 반복 회수를 나타낸다.

3. 웨이블릿 신경망 구조 결정에 관한 기준 연구
웨이블릿 신경망의 구조는 은닉 노드의 수와 각 노드의 확장(dilation), 이동(translation)의 분포로 결정된다. 웨이블릿 신경망의 구조를 결정하는 기준 방법으로는 프레임을 이용한 방법^[8], pruning 방법^[3], 시간-주파수 공간에서 훈련 패턴을 covering하는 방법^[8], 시간-주파수 공간에서 SOFM(Self Organizing Feature Map)을 사용하는 방법^[9] 등이 제안되었다.

프레임을 이용한 방법은 웨이블릿 이론의 프레임 성질을 이용하여 은닉 노드의 집합이 프레임을 이루도록 하는 방법이다. 이는 웨이블릿을 이용한 신경망 구성에 있어서 가장 단순하며 기본이 되는 방법이고, 거의 모든 신호 패턴에 대하여 우수한 성능을 보이는 신경망을 구성한다. 그러나 훈련 패턴의 수가 n 일 때, 프레임을 구성하는 은닉 노드의 수는 약 2^n 이기 때문에 신경망의 크기가 매우 커지는 단점이 있다.

pruning 방법은 웨이블릿 이론의 다중해상도를 이용하여 해상도(resolution) 단위로 은닉 노드를 추가한다. 또한 은닉 노드가 추가 될 때마다 여러 최소화 학습을 하고 연결 가중치가 작은 은닉 노드를 삭제한다. 이 방법은 알고리즘이 간단하고 학습과 구성을 반복하여 수행함으로써 주어진 문제를 해결 할 수 있다는 특징을 가지고 있다. 그러나 웨이블릿의 확장과 이동이 이산적인 값으로 고정되어 있어야 하므로 이후의 두 방법보다 은닉 노드의 수가 많아진다는 단점을 가지고 있다.

시간-주파수 공간에서 모든 변환된 훈련 패턴을 covering 하는 방법은 웨이블릿 신경망이 지역 학습을 한다는 특징을 이용한 것으로써 먼저 프레임을 이루도록 신경망을 구성한 후, 훈련 패턴을 가지지 않은 은닉 노드는 삭제하는 방법으로 구조를 최적화한다.

SOFM을 이용한 구조 결정 방법은 변환된 훈련 패턴을 입력으로, 은닉 노드의 윈도우를 클래스로 생각하여 모든 입력이 클래스에 포함 될 수 있도록 클래스를 증가시키는 변형된 SOFM을 적용한 방법이다.

위의 두 가지 방법은 은닉 노드의 수를 효과적으로 결정 할 수 있지만, 훈련 패턴을 시간-주파수 공간으로 변환해야 하므로 전처리에 많은 계산을 필요로 하고 신경망의 구조를 결정하기 전에 모든 훈련 패턴을 알고 있어야 한다는 제약 조건이 있다. 또한 신경망 구조 결정 알고리즘과 여러 최소화 학습 단계가 한번씩만 수행되므로 이미 구성된 신경망이 주어진 문제를 해결 할 수 없을 때는 신경망을 다시 구성하고 학습을 해야 한다는 단점을 가지고 있다.

III. 자기 성장 웨이블릿 신경망

1. 자기성장 웨이블릿 신경망 구조

본 논문에서 제안하는 자기 성장 웨이블릿 신경망(SGWNN)은 그림3과 같이 3층의 전방향 신경망 구조를 가진다.

입력층과 출력층은 완전 연결되어 있고 연결 가중치는 1로 고정된다. 은닉층과 출력층은 완전 연결되어 있고 연결 가중치는 학습을 통하여 웨이블릿 계수가 된다.

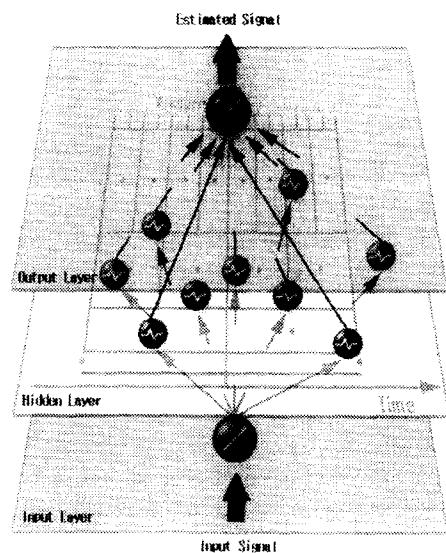


그림 3. SGWNN의 네트워크 구조

Fig. 3. SGWNN network structure.

네트워크의 출력 수식은 식(1)과 동일하며, 입력, 출력 노드에서는 선형 활성화 함수를, 은닉 노드에서는 식(9)과 같은 시그모이드 함수로부터 유도된 웨이블릿 함수를 사용한다.

$$\phi_{a,b} = -\frac{1}{\sqrt{a}} \left(\frac{t-b}{a} \right) \cdot \exp \left(-\left(\frac{t-b}{a} \right)^2 / 2 \right) \quad (9)$$

2 자기성장 웨이블릿 신경망의 학습

제안하는 SGWNN의 학습 과정은 네트워크의 구조 초기화, 온닉 노드 추가 및 삭제, 최소 제곱 에러(Least Mean Square : LMS) 학습의 세 부분으로 구성된다. SGWNN의 구조 초기화는 초기 온닉 노드의 수를 결정하고 각 노드의 파라메터를 초기화 한다. 초기화 과정이 끝나면 온닉 노드를 추가하고 출력 에러를 최소화하는 LMS 학습을 수행한다. LMS 학습 후 에러가 원하는 크기로 줄어들면 학습을 종료하고 그렇지 않으면 새로운 온닉 노드를 추가한 후 LMS 학습을 한다. 새로운 온닉 노드를 추가 할 때 네트워크에서 역할이 작은 노드, 즉 출력에 영향을 거의 미치지 않는 노드는 제거 한다. 온닉 노드의 추가 및 삭제 과정과 LMS 학습을 반복하여 에러가 허용 에러 이하가 되면 전체 학습을 종료한다.

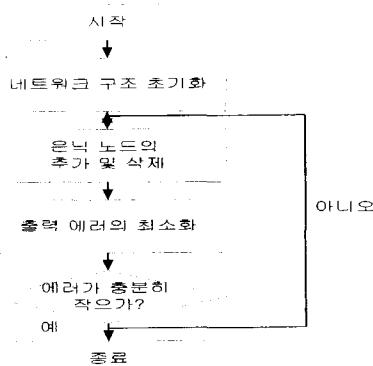


그림 4. SGWNN의 학습 과정
Fig. 4. SGWNN learning flow.

1) 초기화

초기에, SGWNN은 적은 수의 온닉 노드로 구성된다. LMS 학습을 하기 전에는 훈련 패턴에 대한 정보가 없으므로 초기의 온닉 노드 위치는 전체 패턴 공간에서 평등 분포(Uniform distribution)를 따르도록 결정한다. 온닉 노드의 위치를 나타내는 이동(translation) b_k 는 다음 식(10)과 같이 결정된다.

$$b_k = t_{\min} + k \times (t_{\max} - t_{\min}) / (n-1) \quad (10)$$

$[t_{\min}, t_{\max}]$: 훈련 패턴 공간

n : 초기 온닉 노드의 수

k : k 번째 온닉 노드 ($\in \{0, 1, \dots, n-1\}$)

온닉 노드의 크기를 나타내는 확장(dilation) a_k 의 초기값은 임의의 상수 c 로 결정한다. 이때 상수 c 는 온닉 노드에 의해 결정되는 시간상의 원도우에 5-6개의 훈련 패턴이 포함되도록 결정한다. 각 온닉 노드의 연결 가중치 w_k 는 0으로 초기화한다.

초기의 온닉 노드 수는 학습의 속도에 영향을 주지만 네트워크의 근사 능력에는 거의 영향을 미치지 않는다. 따라서 최적의 온닉 노드의 구성을 위해서는 초기 온닉 노드의 수를 0으로 하는 것이 좋다.

2) LMS 학습

네트워크의 초기화 과정이 끝나거나 온닉 노드가 추가 또는 삭제 과정이 끝난 후에는 출력 에러를 최소화하기 위한 학습을 수행한다. SGWNN에서는 최급강하법을 이용하여 네트워크의 파라메터인 웨이블릿의 확장, 이동, 그리고 연결 가중치를 조정한다. 최급 강하학습을 하는 동안은 네트워크에 온닉 노드의 추가나 삭제가 이루어지지 않기 때문에 이 학습 과정은 일반적인 전방향 신경망의 학습과 동일하다. 이 과정을 나타내면 다음과 같다.

Step 1) 훈련 패턴 t_i 를 선택

Step 2) 네트워크 출력 계산 (식(1))

Step 3) 출력 에러 계산 (식(11))

$$E(t_i) = \frac{1}{2} [y(t_i) - \hat{y}(t_i)]^2 \quad (11)$$

$y(t_i)$: 교사 신호

$\hat{y}(t_i)$: 신경망의 출력

Step 4) 네트워크 파라메터 갱신 (식(12))

$$w_k^{n+1} = w_k^n + \alpha_w \frac{dE(t_i)}{dw_k} \quad (12)$$

$$a_k^{n+1} = a_k^n + \alpha_a \frac{dE(t_i)}{da_k}$$

$$b_k^{n+1} = b_k^n + \alpha_b \frac{dE(t_i)}{db_k}$$

Step 5) 출력 에러가 최소화 되도록 1-4를 반복

3) 온닉 노드의 추가 및 삭제

최급강하 학습이 수행되는 동안에 네트워크는 현재의 온닉 노드수로 표현 할 수 있는, 목표 출력에 가장 가까운 값을 출력하도록 학습한다. 최급 강하 학습에서 학습이 충분히 이루어지면 출력 에러의 변화율이 거의

0에 가까워진다. 이 때의 출력은 그 네트워크가 표현할 수 있는 최적의 해에 가까우므로 보다 정밀한 해를 구하기 위해서는 은닉 노드의 수를 증가 시켜야 한다.

그러나 주어진 신경망의 구조에서 최적의 해가 구해졌는가에 대한 기준을 결정하는 것은 매우 어려운 문제이다. 본 논문에서는 신경망에 포함된 은닉 노드의 수가 적을수록 지역해의 존재가 적어지며, 학습이 충분히 이루어진 최급 강하 학습의 후반부에 에러의 변화가 매우 작아진다는 특성을 이용하여 다음과 같은 수식으로 은닉 노드의 수를 증가 시킨다.

$$\frac{\text{Previous Err.} - \text{Current Err.}}{\text{Previous Err.}} < \text{Cri}_1 \quad (13)$$

식(13)에서 전체 훈련 패턴이 입력되어 학습된 기간을 1 epoch로 정의 할 때, 현재 에러(Current Err.)는 현재 epoch에 대한 에러 제곱의 평균을, 이전 에러(Previous Err.)는 이전 epoch에 대한 에러 제곱의 평균을 나타낸다. Cri_1 은 에러의 변화율이 얼마나 작을 때 은닉 노드를 추가 할 것인지를 결정하는 상수로써 Cri_1 가 크면 추가되는 은닉 노드가 많아지고 작으면 적은 수의 은닉 노드가 추가된다.

식(13)을 이용하여 최급강하 학습의 종결 시점을 결정 할 때는 다음의 두 가지에 대하여 고려 해야한다. 첫 번째로는 최급 강하 학습을 시작하여 에러가 가장 급격히 감소하는 방향을 찾을 때까지 에러의 변화가 매우 불안정하다는 것이다. 이 기간에는 에러의 변화가 거의 없거나 심지어 에러가 증가하기도 한다. 따라서 이 기간 동안에는 식(13)의 조건을 만족하더라도 네트워크가 충분히 학습되었다고 결정할 수 없다. 따라서 최급강하 방향을 찾아 학습이 안정적으로 이루어 질 때, 즉 에러가 지속적으로 감소하기 시작 할 때까지는 식(13)의 조건에 대한 판단 없이 학습을 수행해야 한다. 본 논문에서는 이 기간의 학습을 학습의 하한(Lower bound of learning)이라고 정의하고 일정한 횟수의 반복 학습 후에 조건을 판단한다.

두 번째는 학습률이 커지면 에러의 진동 현상이 자주 발생한다는 것이다. 진동 현상이 발생하면 학습이 충분히 이루어져 에러의 변화율이 작아져도 식(13)으로는 은닉 노드의 추가 시기를 결정 할 수 없게 된다. 따라서 학습률은 너무 크지 않은 범위에서 결정 해야한다.

식(13)에 의해 은닉 노드의 추가가 필요하다고 판단되면 네트워크에 새로운 은닉 노드를 추가시킨다. 먼저

식(14)와 같이 새로운 은닉 노드가 추가될 시간상의 위치를 결정한다.

$$\text{insert} = \arg \{ \max \{ E(t_i) \}, t_i \in T \} \quad (14)$$

$E(t_i)$: 식(11)에서 성의한 출력 에러

T : 훈련 패턴 집합

새로운 은닉 노드의 각 파라메터들은 다음과 같이 초기화한다. 새로 추가된 은닉 노드가 최급 강하 학습을 수행하기 전에 네트워크의 출력에 영향을 주지 않도록 하기 위하여 연결 가중치를 0으로 초기화한다.

$$\begin{array}{ll} \text{확장} & a_{\text{new}} = c \\ \text{이동} & b_{\text{new}} = t_{\text{insert}} \\ \text{연결 가중치} & w_{\text{new}} = 0 \\ c & : \text{네트워크 초기화에 사용한 상수} \end{array} \quad (15)$$

최급강하 학습을 수행하면, 네트워크를 구성하는 은닉 노드 중에서 네트워크의 출력에 영향을 거의 미치지 않는 은닉 노드가 발생하게 된다. 이러한 노드들은 거의 사용되지 않으며 신경망의 학습을 느리게 만든다. 웨이블릿 신경망에서 은닉 노드가 출력에 미치는 영향 Out_k 는 식(16)과 같이 결정 할 수 있다.

$$\begin{array}{l} Out_k = \max \{ \psi_{a_k b_k}(t_i) \} \times w_k \\ t_i : \text{훈련 패턴}, t_i \in T \\ a_k : k\text{번째 노드의 확장} \\ b_k : k\text{번째 노드의 이동} \\ w_k : k\text{번째 노드와 출력 노드의 연결강도} \end{array} \quad (16)$$

$\max \{ \psi_{a_k b_k}(t_i) \}$ 는 식(9)의 미분 값이 0이 되는 t_i 를 구함으로써 계산 할 수 있다. $\max \{ \psi_{a_k b_k}(t_i) \}$ 는 웨이블릿 함수의 확장(dilation)에만 관계되므로 $b_k = 0$ 이라 하고 t_i 를 구하면 $t_i = \pm a_k$ 가 된다. 따라서 출력에 영향을 미치지 않으므로 삭제될 k 번째 노드는 다음과 같은 조건으로 결정한다.

$$|\max \{ \psi_{a_k 0}(t_i) \} \times w_k| \leq |\psi_{a_k 0}(a_k) \times w_k| < \text{Cri}_2 \quad (17)$$

Cri_2 는 출력에 어느 정도까지 영향을 미치는 노드를 삭제 할 것인가를 결정하는 상수이다. Cri_2 가 작다면 삭제되는 노드의 수는 적어져 네트워크를 구성하는 노드의 수는 증가하고, Cri_2 가 너무 크다면 출력 에러를

원하는 만큼 감소시킬 수가 없다. 따라서 Cri_2 는 원하는 출력 에러를 고려하여 결정해야 한다.

온닉 노드의 추가 및 삭제 과정과 에러 최소화 학습을 반복하여 다음의 식(18)을 만족하면 네트워크의 구성 및 학습을 종료한다.

$$E(t_i) < Cri_3, \quad \forall t_i \in T \quad (18)$$

T : 훈련 패턴의 집합

IV. 시뮬레이션 결과 및 고찰

본 논문에서는 SGWNN을 함수 근사 문제에 적용하여 시뮬레이션을 수행함으로써 제안한 모델의 성능을 평가한다. 함수 근사에는 $1 \times n \times 1$ 다층 전방향 신경망이 사용되며, 신경망의 입력으로는 시간상의 좌표, 교사 신호로는 함수 값이 사용된다.

네트워크의 성능을 평가하기 위하여 반복 학습 횟수, 학습 수행시간, 네트워크의 크기, 그리고 일반화 에러를 비교한다. 학습의 횟수는 모든 훈련 패턴이 한번씩 학습되었을 때를 1회 학습으로 하였다. 네트워크의 크기는 조정되어야 할 파라미터 및 온닉 노드의 수를 사용하였으며, 일반화 에러는 테스트 패턴 집합이 학습된 패턴과 학습되지 않은 패턴으로 구성되었을 때, 각 패턴에 대한 에러 제곱의 평균으로 정의하였다.

훈련 집합은 함수의 정의 구역 $[-1,1]$ 에서 동일한 간격으로 100개의 패턴을 샘플링하여 구성하며, 테스트 집합은 마찬가지 방법으로 200개의 패턴을 샘플링하여 학습된 패턴과 학습되지 않은 패턴이 동일한 수를 가지고도록 하였다.

근사의 대상이 되는 함수는 다음과 같다.

$$F_1(t) = \sin(3t) \cos(5(t-0.5)) \quad (19)$$

$$F_2(t) = \sin(50(t+1)^2/3)$$

$$F_3(t) = \sin^2 10t (e^{(10t+1)\sin 30}) * (\sin 10t + \cos 20t) + 10)/60$$

시뮬레이션에 사용한 신경망 모델은 표1과 같은 상수를 정의하여 사용한다.

제안한 모델의 성능을 평가하기 위하여 에러 역전파 학습 알고리즘을 사용하는 MLP, [10]에서 제안한 CWNN과 비교한다. 함수 근사에 사용하는 MLP는 온닉 노드에서 시그모이드 함수를 활성화 함수로 사용하

며, 출력 노드에는 선형 함수를 사용한다. MLP의 온닉 노드는 SGWNN과 동일한 수를 가지고록 하였다. 학습률은 0.1, 0.01, 0.001로 실험하여 평균적으로 성능이 가장 좋은 0.001에 대한 결과를 나타내었다. 최대 반복 학습 횟수는 500,000번으로 하였으며, 연결 가중치는 $[-1,1]$ 에서 임의의 수로 초기화하였다.

표 1. 시뮬레이션에서 사용한 SGWNN의 상수

Table 1. Constants for SGWNN in Simulation.

초기 온닉 노드의 수	0
c (확장을 위한 상수)	0.02
각 파라미터에 대한 학습률 $\alpha_a, \alpha_b, \alpha_w$	0.5, 0.5, 0.1
학습의 하한	50회
Cri_1 (노드 추가 조건)	0.0001
Cri_2 (노드 삭제 조건)	$Cri_1/2$
Cri_3 (허용 에러)	0.001
최대 반복 횟수	20,000회

CWNN은 에러가 발생하는 시간상의 영역에 온닉노드를 추가하여 네트워크를 구성하며, 온닉 노드의 삭제는 이루어지지 않는다. 하나의 온닉 노드가 추가된 후 임의의 상수만큼 LMS 학습을 수행한 다음 새로운 온닉 노드가 추가된다. 표2의 결과는 온닉 노드의 추가 사이의 학습을 100번에서 1000번까지 변화시키며 학습한 평균값이다.

표 2. 모델간의 시뮬레이션 결과 비교

Table 2. The compared simulation results of each model.

Char.	Network	F_1	F_2	F_3
온닉 노드 수	MLP	8 (24)*	21 (63)	22 (66)
	CWNN	12.7	25.9	25.2
	SGWNN	8 (24)	21 (63)	22 (66)
학습 횟수	MLP	456,345	500,000	500,000
	CWNN	6,060	13,817	12,932
	SGWNN	10,796	6,194	13,240
학습 시간 (ms)	MLP	10시간이상	10시간이상	10시간이상
	CWNN	59,746	277,813	246,686
	SGWNN	118,640	151,260	348,950
일반화 에러	MLP	0.000366	0.074425	0.014309
	CWNN	0.000097	0.001091	0.007470
	SGWNN	0.000044	0.000455	0.008210

* 온닉 노드의 수에서 괄호 안의 숫자는 신경망에서 조정될 파라미터의 수

표 2는 각 모델에 대한 시뮬레이션 결과를 나타낸 것이다. 먼저 네트워크의 크기를 결정하는 은닉 노드의 수를 비교해 보면, SGWNN이 기존의 CWNN보다 효과적으로 은닉 노드의 수를 결정하고 있음을 알 수 있다. 1회 학습에 소요되는 시간은 은닉 노드의 수가 많을수록 길어지므로 학습의 반복 횟수보다는 전체 학습을 수행한 시간이 정확한 성능을 측정할 수 있는 기준이 된다. MLP의 경우는 CWNN이나 SGWNN과는 비교가 안될 정도로 오랜 시간의 학습이 필요하며, F_2 나 F_3 와 같은 복잡한 과정에 대해서는 학습이 수렴되지 않았다. 학습의 수행 속도를 봤을 때, CWNN이나 SGWNN보다 좋은 성능을 보이고 있다.

CWNN의 경우, LMS 학습이 수행되어야 하는 기간에 대한 판단 기준이 없다. 따라서 상수 횟수만큼 학습을 행한 후에 은닉 노드를 추가해야 한다. 그러나 LMS 학습을 어느 정도 수행해야 하는 가는 학습률, 입력 신호의 종류, 샘플링 비율 등에 영향을 받기 때문에 네트워크의 크기를 줄이고 학습의 속도를 빠르게 하는 상수를 찾아내는 것은 어려운 작업이다. 그러나 SGWNN은 LMS 학습의 종료를 적응적으로 결정함으로써 입력 신호의 종류에 관계없이 은닉 노드의 수를 적절히 결정 할 수 있다.

따라서 적응성이 없는 CWNN의 학습 수행 속도가 다소 빠른 경향을 보이며, 은닉 노드의 최적화에서는 SGWNN이 좋은 성능을 보이고 있다.

그림 5-7은 각각 F_1 , F_2 , F_3 신호에 대한 각 모델의 출력과 목표 신호를 나타낸다.

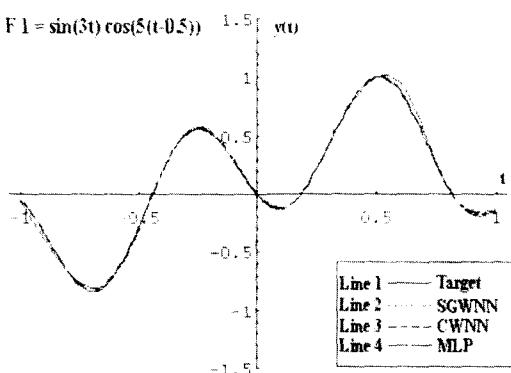


그림 5. F1에 대한 네트워크 출력 비교
Fig. 5. Outputs of each model for F1.

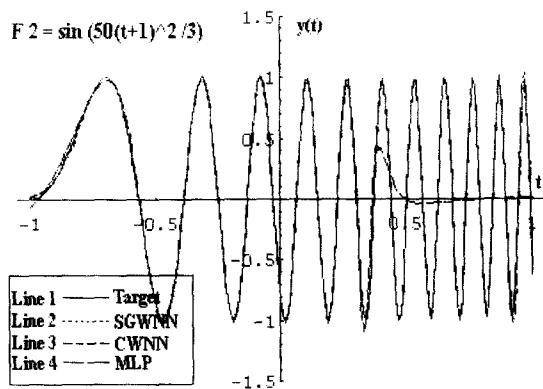


그림 6. F2에 대한 네트워크의 출력 비교
Fig. 6. Outputs of each model for F2.

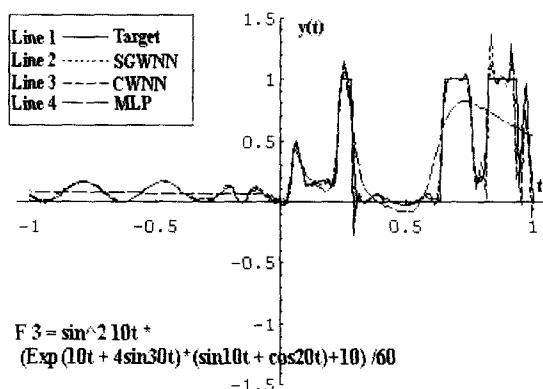


그림 7. F3에 대한 네트워크의 출력 비교
Fig. 7. Outputs of each model for F3.

F_1 에 대한 출력을 살펴보면 표 2에서와 같이, SGWNN과 CWNN의 출력은 일반화 에러가 매우 작아 목표 출력과 거의 구분되지 않는다. 그러나 MLP의 경우는 신호의 첫 부분과 끝 부분에 약간의 차이가 있음을 알 수 있다.

F_2 와 F_3 에 대하여, MLP는 수렴을 하지 못 했기 때문에 목표 출력과 많은 차이를 보이고 있다. CWNN과 SGWNN은 F_2 의 경우는 잘 근사하고 있지만 F_3 에 대해서는 두 모델 모두 비교적 큰 에러를 가지고 있다. F_3 에 대한 SGWNN의 출력은 사인파와 유사한 부분은 적절히 근사를 하고 있지만, 구형파 부분은 많은 에러가 발생하고 있다. 이것은 구형파 부분에서 일반화 성능이 떨어진다는 것을 의미하며, 이러한 현상은 은닉층에서 사용하는 웨이블릿 함수가 사인파에 가까운 보양이기 때문이다.

그림 8은 F_2 에 대한 온닉 노드의 위치와 웨이트의 크기를 시간-주파수 공간에 나타낸 것이다. 가로축은 훈련 패턴의 시간상의 위치를 나타내며, 세로축은 웨이블릿 함수의 확장(dilation)을 나타낸다. 확장(dilation)은 주파수와 역의 관계에 있으므로 세로축의 값이 0에 가까운 쪽이 고주파이고 값이 커질수록 저주파 영역이다. 굵은 실선으로 나타낸 것은 양의 웨이트를 가지는 온닉 노드이며, 가는 실선은 음의 웨이트를 가지는 온닉 노드이다. 원의 지름은 상대적인 웨이트의 크기를 나타낸다. 그림 9는 같은 공간에서 온닉 노드의 출력 크기를 나타낸 것이다.

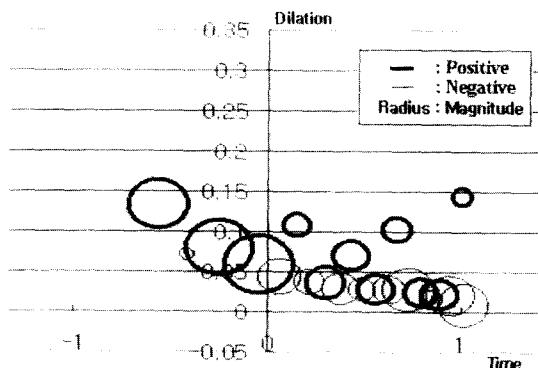


그림 8. F_2 에 대한 시간-주파수 공간에서의 온닉 노드 위치와 웨이트의 크기

Fig. 8. positions and weights of hidden nodes in time-frequency for F_2 .

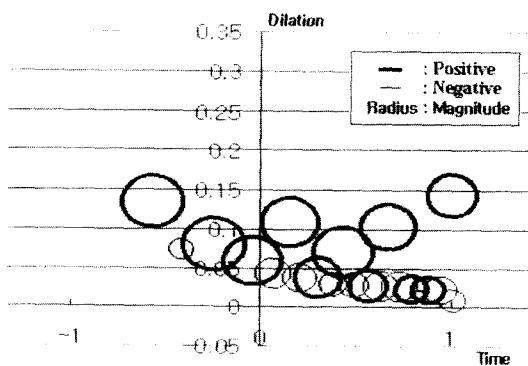


그림 9. F_2 에 대한 시간-주파수 공간에서의 온닉 노드 위치와 출력의 크기

Fig. 9. positions and outputs of hidden nodes in time-frequency for $F2$.

온닉 노드의 출력은 확장(dilation)의 제곱근에 반비례하고 웨이트에 비례하므로 확장(dilation)이 작은 온

닉 노드들은 웨이트가 상대적으로 적은 값을 가지고 있어도 출력은 큰 값을 가지고 있음을 알 수 있다. F_2 는 시간축에서 뒤로 갈수록 주기가 짧아지며 진폭은 일정한 신호이다. 그림 8에서는 확장(dilation)이 점점 작아지는 현상을 보이며 그림 9에서는 온닉 노드의 출력이 비슷한 크기를 가짐으로써 이러한 신호의 특성을 잘 반영하고 있다.

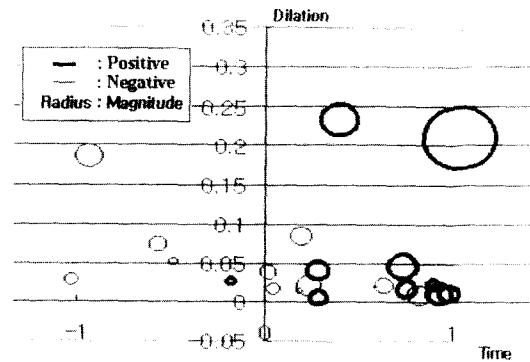


그림 10. F_3 에 대한 시간-주파수 공간에서의 온닉 노드 위치와 웨이트의 크기

Fig. 10. position and weight of hidden nodes in time-frequency for $F3$.

그림 10은 F_3 에 대한 웨이트를 시간-주파수 공간에서 나타낸 것인데, 구형파의 신호에 해당하는 부분에 온닉 노드가 모여 있으며 이 부분에서 신호의 근사를 위해 많은 온닉 노드가 필요함을 알 수 있다.

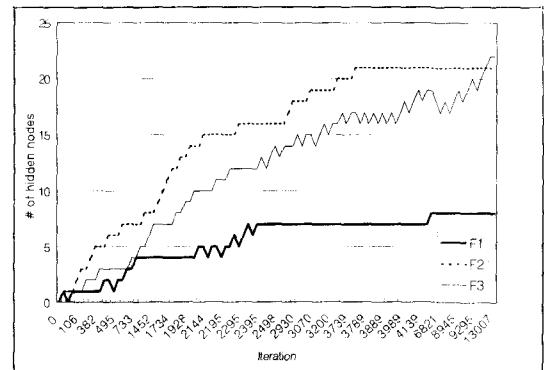


그림 11. 대상 신호에 따른 온닉 노드 수의 변화

Fig. 11. The number of hidden nodes for each signal.

그림 11은 각 신호에 대하여 학습이 진행됨에 따라 온닉 노드 수의 변화를 나타낸 것이다. 첫 번째 신호에

대해서는 8개의 은닉 노드로 학습이 종료되었으며 학습을 수행하는 동안 은닉 노드의 추가는 14번, 삭제는 6번 발생하였다. 두 번째 신호에서는 은닉 노드의 삭제 없이 계속 추가되어 21개로 학습이 종료되었으며 세 번째 신호는 은닉 노드의 추가 40번과 삭제 18번으로 은닉 노드의 추가 및 삭제가 비교적 빈번히 발생하며 학습이 진행되다가 22개의 은닉 노드로 학습이 종료되었다. 두 번째 신호에서 은닉 노드의 삭제가 이루어지지 않은 것은 본 논문에서 제안한 은닉 노드의 추가 방법을 사용했을 때 모든 은닉 노드가 적절한 위치에 추가되었음을 의미한다. 반면 다른 두 신호에서는 적절하지 않은 은닉 노드의 추가가 이루어 졌으며 신호가 복잡할수록 적절한 은닉 노드의 위치를 결정하는 것이 어려움을 보여주고 있다.

SGWNN이 학습 도중에 훈련 패턴이 변하는 문제에 대해서도 적용 가능함을 보이기 위해 F_1 신호의 정의 구역을 [-1, 2.8]로 확장하여 10개의 훈련 집합을 구성하여 학습을 수행하였다. 첫 번째 훈련 집합은 [-1, 1]에서 100개의 패턴을 샘플링하여 구성하고 두 번째 훈련 집합은 [-0.8, 1.2]에서 100개의 패턴을 샘플링하여 구성하였다. 같은 방법으로 신호의 시간축을 따라 0.2씩 진행시켜 10번째 훈련 집합은 [0.8, 2.8]에서 샘플링하여 구성하였다.

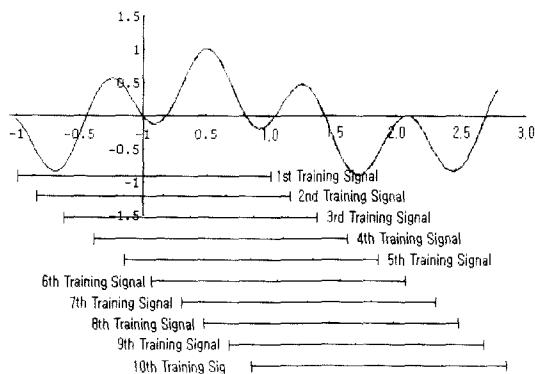


그림 12. 훈련 패턴이 변할 때의 훈련 패턴의 영역
Fig. 12. Domains of each set of training patterns.

그림 12는 시뮬레이션에서 사용한 신호와 각각의 훈련 집합에 대한 영역을 나타낸 것이다. 표 3은 훈련 패턴 집합이 변함에 따라 은닉 노드의 수와 학습 횟수의 변화를 나타낸 것이다. 시뮬레이션의 결과 학습 패턴이 변화할 때 신호가 복잡해지면 은닉 노드의 추가가 이루어지며 추가되는 학습은 최대 원래 신호를 학습하는

시간의 1/2정도가 필요함을 알 수 있다.

표 3. 훈련 패턴이 변할 때 학습 결과

Table 3. Results of learning when the set of training patterns is changed.

훈련집합	은닉노드수	학습횟수	에러
1	8	10796	0.00004320
2	8	461	0.00024150
3	8	738	0.00015886
4	8	979	0.00014867
5	8	963	0.00016827
6	9	3217	0.00013651
7	9	369	0.00026090
8	9	4506	0.00006951
9	10	2264	0.00009807
10	10	2731	0.00004142

표 3에서 은닉 노드의 수를 살펴보면, 첫 번째 훈련 집합에 대하여 8개의 은닉 노드로 학습이 수렴하였다. 이후로 여섯 번째와 아홉 번째 훈련 집합에서 은닉 노드가 하나씩 증가하였다. 신호가 진행됨에 따라 은닉 노드의 위치도 같이 변화되므로 새로 추가되는 은닉 노드는 신호가 진행되는 방향에서 추가된다. 신호가 진행되는 반대 방향에서는 은닉 노드가 삭제되어야 하는데, 시뮬레이션의 결과에서는 노드가 삭제되지 않고 있다.

그림 13은 6번째 훈련 집합과 9번째 훈련 집합에 대한 학습이 종료되었을 때, 은닉 노드의 위치를 나타낸 것이다. 은닉 노드의 삭제가 적절히 이루어 지지 않는 것은 그림 13에서 볼 수 있듯이 신호의 후반부에 은닉 노드가 모이는 현상이 발생하여 각 노드의 출력이 분산되는 효과를 나타내기 때문이다.

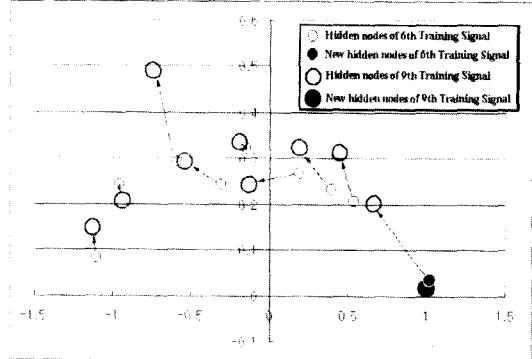


그림 13. 6번째와 9번째 집합에 대한 은닉 노드의 위치
Fig. 13. The position of hidden nodes for 6th and 9th training set.

V. 결 론

본 논문에서는 웨이블릿 신경망의 새로운 구성 방법과 학습 알고리즘을 제안하였다. 제안한 방법은 여러 최소화 학습을 수행하며 스스로 온닉 노드의 수를 증가, 감소시키며 온닉층의 구조를 결정한다. 제안한 방법에 의해 구성되는 SGWNN은 웨이블릿 이론의 함수 지역화 특성을 사용한다.

SGWNN은 웨이블릿 네트워크의 특성을 가지고 있으므로, 지역해에 수렴함 없이 해로 빠르게 수렴하며 신호의 일부가 변형되어도 신경망의 전체적인 재학습 없이 지역적인 재학습만으로 새로운 신호를 균사할 수 있다. 또한 훈련 패턴에 대하여 전처리 과정이 필요 없으므로 전체적인 학습의 속도를 빠르게 할 수 있으며 훈련 패턴을 모두 알고 있어야 한다는 제약이 없으므로 보다 넓은 범위의 응용 분야에 적용 할 수 있다.

시뮬레이션 결과, 제안한 SGWNN은 적용하는 문제에 따라 적응적으로 온닉 노드의 수를 결정하며, 신호의 특성을 적절히 반영하여 온닉 노드의 구조를 결정함을 보았다. 또한 학습 도중 훈련 패턴이 변하는 문제에 대해서도 SGWNN을 적용 할 수 있음을 보았다. 그러나 신호가 진행될 경우는 온닉 노드의 삭제가 적절히 이루어지지 않고 있어 온닉 노드의 출력이 분산되기 전에 효과적으로 온닉 노드를 삭제하는 방법이 요구된다.

본 논문에서 적용한 함수 균사 문제는 함수 해석은 물론, 비선형 시스템의 제어에 적용되어 사용 할 수 있으며, 패턴 인식 및 분류 문제에도 적용 가능하다. 또한 제안한 신경망 이용하여 모델을 음성에서의 특징 추출, EEG 신호 분석을 위한 특징 추출, 화상 이미지 처리 등의 실질적인 응용 분야에 적용하는 연구가 필요하다.

참 고 문 헌

- [1] H.Suz, B.Telfer, S.Kadambe, "Neural Network Adaptive Wavelets for Signal Representation and Classification", Optical Engineering, Vol.31, pp. 1907-1916, 1992.

- [2] Q. Zhang, A. Benveniste, "Wavelet Networks", IEEE Trans. Neural Networks, vol. NN-3, no 6, pp. 889-898, 1992.
- [3] B. Bakshi, G. Stephanopoulos, "Wavelets as Basis Functions for Localized Learning in a Multi-resolution Hierarchy", IJCNN 92, Vol 2, pp. 140-145, 1992.
- [4] K.Hornik, M. Stinchcombe, H.White, "Multilayer feedforward networks are universal approximators", Neural Networks, vol. 2, pp. 359-366, 1989.
- [5] B. Delyon, A. Juditsky, A. Benveniste, "Accuracy Analysis for Wavelet Approximations", IEEE Trans. Neural Networks, Vol. 6, No.2 pp. 332-348, 1995.
- [6] A. Ypma and R.P.W. Duin, "Using the Wavelet for function approximation", ASCI97, Netherlands, June, 1997.
- [7] Gaviphat Lekutai , "Adaptive Self-Tuning Neuro Wavelet Network Controllers", PhD thesis, March 31, 1997, Blacksburg, Virginia.
- [8] Y.C. Pati and P.S. Krishnaprasad, "Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformations", IEEE Trans. Neural Networks, 4, (1), pp. 73-85, 1993.
- [9] K.Kobayashi, T. Torioka, "A Wavelet Neural Network for Function Approximation and Network Optimization", Proc. of ANNIE 94, Vol 4, pp. 505-510, 1994.
- [10] S.K. Lim, C.H. Park, H.S. Lee, "Constructive Neural Network using Adaptive Wavelet for Function Approximation", Proc. of ICONIP '98, Vol 3, pp. 1224-1227, 1998.

저자소개

林 成 吉(正會員)

李 顯 淮(正會員) 第 33 卷 B編 第 10 號 參照

1997년 경희대학교 수학과 졸업(이
학사). 1999년 경희대학교 전자계
산공학과 대학원 석사과정 졸업(공
학석사). 1999년~현재 경희대학교
전자계산공학과 박사과정. 주관심
분야는 신경회로망, 패턴인식, 신호

처리 등임