

신뢰도 추정을 위한 분산 학습 신경 회로망

조 영 빈* , 권 대 갑**

A Variance Learning Neural Network for Confidence Estimation

Young B. Cho*, D.G. Gweon**

ABSTRACT

Multilayer feedforward networks may be applied to identify the deterministic relationship between input and output data. When the results from the network require a high level of assurance, consideration of the stochastic relationship between the input and output data may be very important. Variance is one of the effective parameters to deal with the stochastic relationship. This paper presents a new algorithm for a multilayer feedforward network to learn the variance of dispersed data without preliminary calculation of variance. In this paper, the network with this learning algorithm is named as a variance learning neural network(VALEAN). Computer simulation examples are utilized for the demonstration and the evaluation of VALEAN.

Key Words: Stochastic Process(확률과정), Gaussian Distribution(가우시안 분포), Variance Learning(분산학습), Confidence Estimation(신뢰도 추정), Multilayer Perceptron(다층 신경회로망, 다층 퍼셉트론)

1. 서 론

신경회로망은 병렬 분산 처리 기능을 가진 비선형 연산 소자인 뉴런을 기본단위로 하여 구성되며 미지의 입출력 관계를 학습할 수 있다. 신경 회로망에는 연상기억 메모리⁽¹⁾ (Associative Memory), 경쟁 학습 신경회로망 (Competitive Network), 다층신경망(Multilayer Feedforward Network) 등과 같은 여러 가지 형태가 있으며 각각 특정한 학습 규칙을 가지고 있다. Hetch-Nielsen⁽²⁾은 임의의 연속 함수가 무한개의 뉴런과 2개의 은닉층으로 구성된 다층신경망으로 구현될 수 있음을 증

명하였고 Hornik⁽³⁻⁵⁾등은 충분히 많은 뉴런과 3개의 은닉층을 가진 다층신경망으로 임의의 함수가 근사화 될 수 있음을 보였다.

다층신경망의 함수 근사화 기능은 오류역전파 알고리즘⁽⁶⁾과 같은 학습 이론의 발전으로 패턴 인식, 매개변수 추정, 인공지능등의 분야에 널리 적용 되어왔다. 이러한 다층신경망의 학습은 입출력 변수 사이에 존재하는 비확률적 관계를 근사화하는 것이 주 목적이다. 그러나, 신뢰도를 고려해야 하는 검사 작업과 같은 응용 분야의 경우 입출력 변수 사이의 확률적 관계를 무시하기 어려울 때가 있다. 이런 경우, 확률적 영향을 고려하기 위하여 인식 결

* 한국과학기술원 기계공학과 대학원

** 한국과학기술원 기계공학과

과의 오차에 대한 분산을 추정하는 것이 유용하다. 가우시안 분포를 가지는 데이터의 평균과 분산을 학습하기 위한 신경회로망에 대한 연구는 Weigend⁽⁷⁾ 와 William⁽⁸⁾ 등에 의해 수행되었다. 이들의 연구는 기본적으로, 데이터의 확률분포와 신경회로망의 입출력 관계사이의 유사성(likelihood)을 최대화하는 기법을 사용한 것이다.

본 연구에서는 다층신경회로망을 분산 추정에 사용하기 위하여 기존의 유사성 최대화(Likelihood maximization) 기법을 사용하는 대신 새로운 학습알고리즘을 개발하고 이 알고리즘의 전개과정 및 학습성능에 대하여 다루었다. 이하로 이 알고리즘을 이용한 다층신경회로망을 분산 학습 신경회로망 (VALEAN : VAriance LEarning Neural Network)으로 부르기로 한다. 이 알고리즘은 데이터가 가우시안분포임을 가정으로 데이터의 분산값을 추정하는 것을 목표로 한다. 본 논문에서는 확률적 관계를 가지는 데이터에 대한 내용을 2장, 학습알고리즘 개발 과정을 3장, 그리고 모의실험결과를 4장에 기술하였다.

2. 확률과정과 분산학습신경망

n 차원 입력변수 x 와 출력변수 y 사이의 관계 $y(x)$ 는 다음과 같이 비확률적 관계식 $y_d(x)$ 와 확률적 관계식 $y_s(x)$ 로 표현될 수 있다.

$$y(x) = y_d(x) + y_s(x) \tag{1}$$

여기서 비확률적 관계는 기존의 다층 신경망에 의해 구해지지만 확률적 관계 $y_s(x)$ 는 기술할 수 없으므로 학습에러로 취급된다. 만일 확률적 관계가 평균이 0이고 임의의 분산을 가진 가우시안 분포라면 다음과 같이 표현할 수 있다.

$$E[y_s|x] = 0 \tag{2}$$

$$p(y_s|x) = \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{y_s^2(x)}{2\sigma^2(x)}\right) \tag{3}$$

여기서 $\sigma^2(x)$ 은 확률적 관계식 $y_s(x)$ 의 분산이며 입력변수 x 의 함수이다. 위식에서 보는 바와 같이 확률적 관계식은 분산 $\sigma^2(x)$ 만으로 기술할 수 있다.

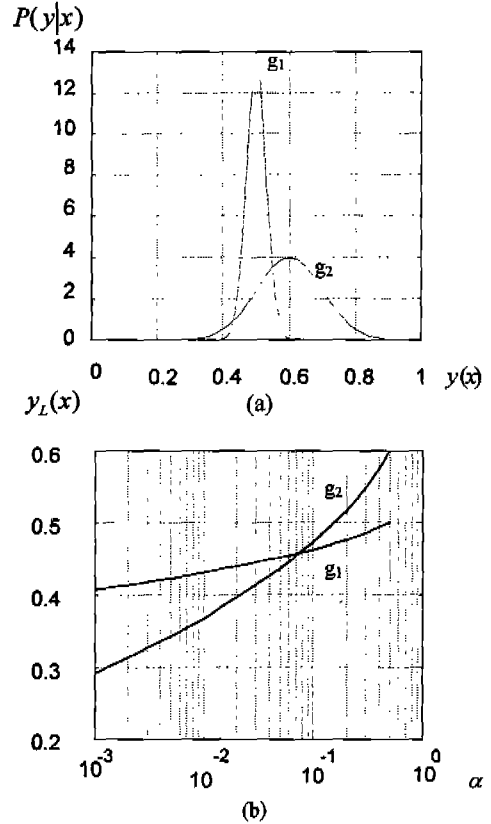


Fig. 1 Stochastic Process and Confidence Estimation

주어진 입력 x 에 대한 출력값 y 를 구할때 신뢰도를 고려하여야 한다면 확률적 관계식 $y_s(x)$, 즉 분산 $\sigma^2(x)$ 를 고려해야 한다. 만일 $(1-\alpha)$ 의 신뢰도로 보장할 수 있는 출력값의 하한값 $y_L(x)$ 을 구해야 한다면 다음과 같이 계산할 수 있다.

$$y_L(x) \leq y_d(x) - z_\alpha \sigma(x) \tag{4}$$

여기서 Z_α 는 신뢰계수 $(1-\alpha)$ 에 의해 구해지는 상수이다. Fig 1에서 보는 바와 같이 비확률적 관계(평균값)는 g_2 의 경우가 더 크지만 높은 신뢰도가 요구될 경우(여기서는 α 가 0.07보다 작을 경우) 하한값 $y_L(x)$ 은 g_1 의 경우가 더 크게 된다.

본 연구에서는 이와 같은 응용을 위하여 분산을 학습하는 신경 회로망을 개발하고자 하였으며 그 구조적 형태는 Fig 2과 같다. 기존의 다층 신경망NN1은 비확률적 합

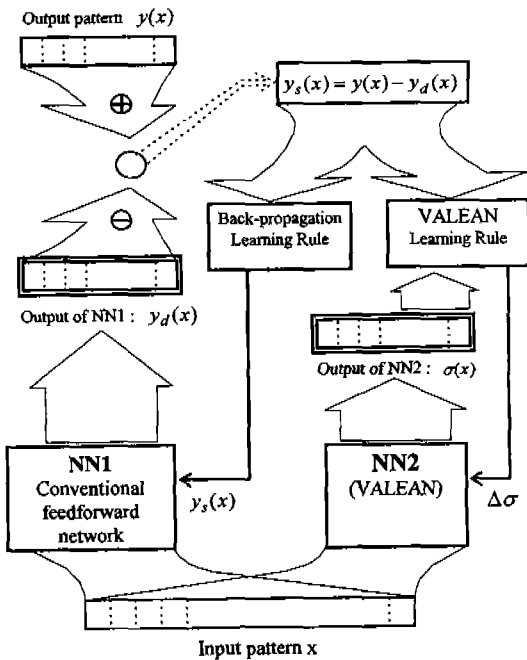


Fig. 2 Schematic Diagram of Variance Learning

수 $y_d(x)$ 를 학습하기 위한 것이며 나란히 사용된 분산 학습 신경망 NN2는 확률적 함수관계 $y_s(x)$ 를 학습하기 위한 것이다. 신경회로망 NN1은 위의 그림에서 보는 바와 같이 일반적인 오류역전파와 신경회로망이다. 주어진 입력패턴 x 와 그에 따른 출력 패턴 y 가 결정되면 우선 NN1의 학습이 진행된다. 이때 NN1은 비확률적인 관계, 즉 $y_d(x)$ 를 학습하게 된다. 만일 신경 회로망 NN1의 학습이 성공적이라면 비 확률적 함수관계 $y_d(x)$ 는 NN1에 의해 충분히 기술될 수 있다. 이 경우 확률적 관계를 추정하기 위해 분산 학습 신경 회로망 NN2가 학습을 시작한다.

기존의 오류 역전파 알고리즘은 입력과 그에 대응되는 출력을 이용하는데 분산을 학습하고자 하는 경우에는 주어진 입력 패턴 x 에 대하여 출력값은 분산 $\sigma^2(x)$ 가 되어 학습하기 전에 미리 구해야 하는 까다로운 점이 있다. 만일 다차원 입력 패턴일 경우에는 일일이 분산값 $\sigma^2(x)$ 을 구하는 작업은 매우 힘든 일이 될 수 있다. 여기서는 입력 패턴 x 와만 $y_s(x)$ 을 이용하여 자동으로 분산 $\sigma^2(x)$ 를 구하는 학습 알고리즘을 개발하고자 한다.

3. 분산 추정을 위한 학습 법칙

가우시안 분포를 가지는 데이터의 분산을 학습하는 알고리즘에 설명을 집중하기 위하여 본 논문에서는 Fig 2의 NN2의 학습에 대해서만 서술하기로 한다. 학습 시간 t 에서 분산 학습 신경 회로망의 출력은 분산값의 추정치 $\sigma_N^{(t)}(x)$ 로 표현되며 이 값은 주어진 입력 패턴 x 의 함수이다. 신경 회로망의 학습 과정을 학습 시간 t 와 $t+1$ 에서 아래의 식 (5)와 같이 표현한다면 $\Delta\sigma_N^{(t)}(x)$ 은 학습 시간 t 에서의 학습 오차로 생각할 수 있다.

$$\sigma_N^{(t+1)}(x) = \sigma_N^{(t)}(x) + \Delta\sigma_N^{(t)}(x) \quad (5)$$

신경 회로망의 웨이트들은 임의의 값으로 초기화되므로 분산 학습 신경망의 초기 출력값 $\sigma_N^{(0)}(x)$ 는 임의의 값을 가지게 된다. 그러나 학습이 반복될 수록 학습 오차 $\Delta\sigma_N^{(t)}(x)$ 의 피드백으로 신경 회로망의 출력은 참값에 접근하게 된다. $\Delta\sigma_N^{(t)}(x)$ 의 크기는 충분히 작은 일정한 값이며 부호만 바뀌는 매우 단순한 상황일때 아래와 같은 임의의 확률 함수로 표현될 수 있다.

$$|\Delta\sigma^{(t)}| \ll \sigma_N^{(t)} \quad (6)$$

$$\text{sign}(\Delta\sigma^{(t)}) = \begin{cases} +1 & p[\Delta\sigma > 0] \\ -1 & p[\Delta\sigma < 0] \end{cases}$$

여기서 확률 함수는 다음과 같은 3가지 제한 조건을 만족하여야 분산 학습 신경 회로망이 참 값에 수렴한다.

Constraint 1 : (7.a)

$$E[\Delta\sigma | \sigma_N = \sigma_T] = 0$$

Constraint 2 :

$$\begin{aligned} P[\Delta\sigma < 0 | \sigma_N > \sigma_A] &> P[\Delta\sigma < 0 | \sigma_N < \sigma_A] \\ P[\Delta\sigma > 0 | \sigma_N > \sigma_A] &< P[\Delta\sigma > 0 | \sigma_N < \sigma_A] \end{aligned} \quad (7.b)$$

Constraint 3 :

$$\begin{aligned} P[\Delta\sigma < 0 | y_s] &\leq 1, P[\Delta\sigma > 0 | y_s] \leq 1 \\ P[\Delta\sigma < 0 | y_s] + P[\Delta\sigma > 0 | y_s] &= 1 \end{aligned}$$

where σ_T^2 : the true value of variance (7.c)

σ_N^2 : the estimated variance by VALEAN

σ_A^2 : an arbitrary value of variance

y_s : a stochastic relationship between input and output

첫번째 제한 조건은 학습 결과가 바이어스되지 않음을 요구하는 것이며 두번째 제한 조건은 참값에 수렴함을 요구하는 것이고 마지막 제한 조건은 확률 함수의 일반적인 요구 조건이 된다. 여기서 두번째 제한 조건은 다음과 같이 서술될 수 있다.

“추정치 $\sigma_N^{(l)}(x)$ 가 임의의 σ_A 값보다 크면 작은 경우보다 감소할 확률이 커진다”

“추정치 $\sigma_N^{(l)}(x)$ 가 임의의 σ_A 값보다 작으면 큰 경우보다 증가할 확률이 커진다”

따라서 충분한 학습 후에는 반드시 임의의 값에 수렴하게 된다.

위의 제한 조건을 모두 만족하는 학습 규칙을 찾기 위해 식 (7.a)를 다음과 같이 정리한다.

$$\begin{aligned} E[\Delta\sigma | \sigma_N = \sigma_T] &= \int P_{\sigma_T}(y_s) \cdot [P[\Delta\sigma > 0] - P[\Delta\sigma < 0]] \Delta\sigma dy_s \quad (8) \\ &= \int P_{\sigma_T}(y_s) \cdot [1 - 2P[\Delta\sigma < 0]] \Delta\sigma dy_s \\ &= 0 \end{aligned}$$

$|\Delta\sigma|$ 가 일정하다고 가정하였으므로 위 식 (8)은 다음과 같이 간단히 정리된다.

$$\int P_{\sigma_T}(y_s) \cdot P[\Delta\sigma < 0] dy_s = \frac{1}{2} \quad (9)$$

위 식 (9)는 첫 번째 제한 조건의 다른 표현으로 생각할 수 있다. 따라서 식 (9)를 만족하는 $P[\Delta\sigma < 0]$ 를 학습에 사용할 경우 반드시 임의의 값에 수렴하게 된다.

임의의 서로 다른 2개의 데이터 y_{s1}, y_{s2} 에 대하여 가우시안 함수는 아래의 식을 항상 만족한다.

$$\begin{aligned} \frac{P_{\sigma_N}(y_{s2})}{P_{\sigma_N}(y_{s1})} - \frac{P_{\sigma_A}(y_{s2})}{P_{\sigma_A}(y_{s1})} &= \begin{cases} > 0 & \text{if } \sigma_N > \sigma_A \\ < 0 & \text{if } \sigma_N < \sigma_A \end{cases} \quad (10) \\ &\text{for any } |y_{s1}|, |y_{s2}| \end{aligned}$$

위 식에서 알 수 있는 바와 같이 학습 법칙 $P[\Delta\sigma < 0]$

를 $\frac{P_{\sigma_N}(y_{s2})}{P_{\sigma_N}(y_{s1})}$ 와 비례하는 임의의 함수로 표현하면 두 번째

제한 조건을 만족시킬 수 있다. 식 (10)은 식(11)의 조건에서도 역시 만족되고 정규화된 가우시안 함수를 식

(12)와 같이 정의하면 식 (10)은 (13)과 같이 간단히 표현될 수 있다.

$$y_{s1} = 0 \quad (11)$$

$$\bar{P}_{\sigma}(y_s) = \exp\left(-\frac{y_s^2}{2\sigma^2}\right) \quad (12)$$

$$\bar{P}_{\sigma_N}(y_s) - \bar{P}_{\sigma_A}(y_s) = \begin{cases} > 0 & \text{if } \sigma_N - \sigma_A > 0 \\ < 0 & \text{if } \sigma_N - \sigma_A < 0 \end{cases} \quad (13)$$

정규화된 가우시안 분포함수와 가우시안 분포함수 사이에는 다음과 같은 관계가 성립한다.

$$\int P_{\sigma}(y_s) \bar{P}_{\sigma}^n(y_s) dy_s = \frac{1}{\sqrt{n+1}} \quad (14)$$

만일 학습을 위한 확률 함수 $P[\Delta\sigma < 0]$ 를 아래와 같이 선택한다면 앞에서 설명한 3가지 제한 조건을 모두 만족하는 매우 간단한 함수가 된다.

$$P[\Delta\sigma < 0] = \bar{P}_{\sigma_N}^{-3}(y_s) \quad (15)$$

$$P[\Delta\sigma > 0] = 1 - \bar{P}_{\sigma_N}^{-3}(y_s)$$

분산 학습 알고리즘의 동작은 식 (15)와 Fig 3의 모식도에서 보는바와 같이 정리될 수 있다.

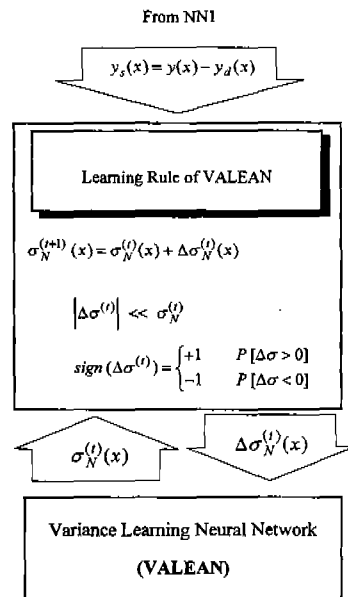


Fig. 3 Learning Rule of VALEAN

참고로 $P[\Delta\sigma < 0]$ 와 데이터 y_i 사이의 다음과 같은 결과는 매우 주목할 만하다. $P[\Delta\sigma < 0]$ 와 $P[\Delta\sigma > 0]$ 가 똑같이 0.5를 가지는 데이터 y_i 는 식(12)와 (15)로부터 아래와 같이 계산될 수 있다.

$$y_i = \pm \sqrt{\ln 2^{\frac{2}{\sigma}}} \cdot \sigma \quad (16)$$

y_i 로부터 y_i 까지 가우시안 분포의 적분값은 정규 분포 표에서 확인할 수 있는 바와 같이 0.5이다. 이것은 분산 학습 신경 회로망의 학습이 제대로 진행되어 참 값에 도달했을 때는 학습된 분산값을 증가시키거나 감소시키는 확률이 동일하다는 의미로서 학습 알고리즘이 분산 학습 신경 회로망의 출력을 참 값에 수렴시키는 것을 보장한다는 것을 나타내는 증거가 된다.

4. 모의 실험

지금까지 설명한 분산 학습 알고리즘의 모의 실험을 위하여 임의의 가우시안 분포를 생성하였다. 여기서 $N(m(x), \sigma(x))$ 와 는 주어진 입력 패턴 x 에 대한 평균과 분산을 의미한다. 입력 패턴 x_1, x_2, x_3 는 0.5 0.6 그리고 0.7이며 각각의 입력 패턴에 대한 데이터의 평균과 분산은 각각 아래와 같다. 여기서는 각각의 패턴에 대하여 25개, 모두 75개의 데이터를 모의 실험에 사용하였다.

$$\begin{aligned} m(x_1) &= m(x_2) = m(x_3) = 0 \\ \sigma(x_1) &= 0.027304 \\ \sigma(x_2) &= 0.051190 \\ \sigma(x_3) &= 0.085624 \end{aligned} \quad (17)$$

평균값은 비확률적인 관계로서 Fig 2의 NN1에 의해 학습될 수 있다. 학습 방법은 Weigend 등에 의해 실험된 바와 같으며 NN1의 학습이 완료된 후 분산 학습 신경 회로망 NN2가 확률적 관계인 분산을 학습하기 위해 동작하기 시작한다. 여기서는 1개의 입력 노드, 3개의 은닉 노드, 1개의 출력 노드를 가진 다층 신경 회로망으로 구성하였다. 3장에서 설명한 바와 같이 수정항은 전체 학습 과정 동안 일정한 작은 값을 가지도록 하였으며 현재의 추정 값에 비례한 값을 가지도록 하는 방법도 실험하였

다. 추정치가 참 값에 충분히 가까워지고 학습이 충분히 천천히 진행된다면 수정항의 크기가 일정하다는 가정을 그대로 적용할 수 있다.

$$|\Delta\sigma| = k \quad (18)$$

$$|\Delta\sigma| = \gamma \times (\sigma + \epsilon \cdot \gamma^2) \quad (19)$$

Fig. 4는 k 값이 각각 0.0001, 0.001 그리고 0.01일 때의 학습 과정을 도시한 것이다. 여기서 보는 바와 같이 학습 과정 전반에 걸쳐 일정한 수정 값을 유지할 경우는 수정 값의 크기에 민감하게 변하는 것을 알 수 있다. 첫 번째의 경우는 분산의 크기에 비해 수정 값이 너무 작아서

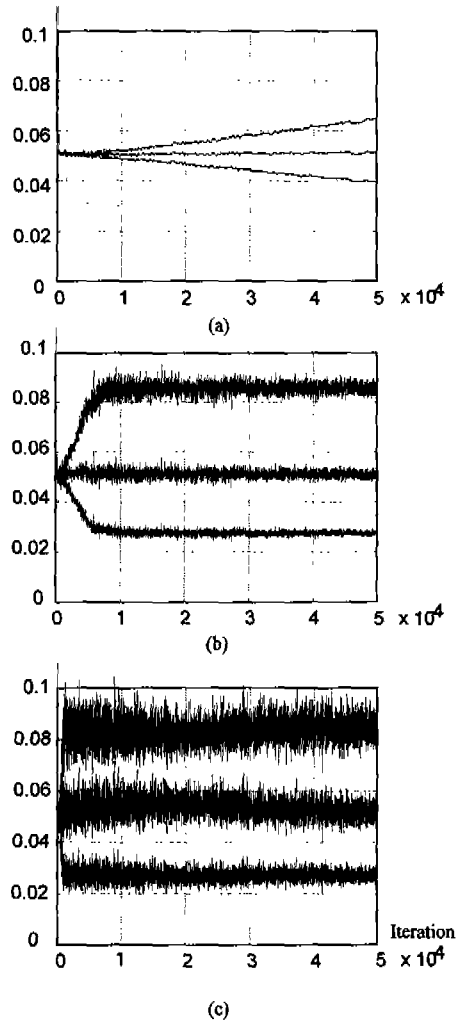


Fig. 4 Learning history of VALEAN for constant correction

50,000번의 학습에도 수렴하지 못하고 느린 학습을 보여 주었다. 이에 비해 세 번째의 경우는 너무 큰 진동 모습을 보여 일정한 값에 수렴 되지 않았다. 일반적으로 주어진 문제에 따라 k 값을 일일이 지정하기는 어려우므로 식 (18)의 경우는 실제적용에 무리가 따를 수 있다.

Fig 5에는 식 (19)를 이용한 학습 결과를 보여준다. 여기서 γ 는 일정 초기값에서 서서히 0으로 수렴하는 값으로서 충분한 학습후는 진동하는 출력을 일으키지 못하도록 하기 위한 것이다. 또한 ε 은 초기에 추정값이 매우 작아 학습을 하지 못하는 현상을 피하기 위한 값으로서 학습 초기에만 필요한 값이므로 빨리 소멸되도록 하였다. 이렇게 수정치의 크기를 정할 경우 그림에서 보는 바와 같이 빠르고도 안정적인 학습 결과를 얻을 수 있었으며 학습 성능이 민감하게 변하지 않는 바람직한 결과를 얻을 수 있었다.

학습에 사용된 신경망의 은닉노드의 갯수가 작았음에도 불구하고 학습 에러의 크기는 Table 1에서 보는 바와 같이 1%내외로 매우 작아 학습이 잘되는 것을 알 수 있다.

용접 강도를 추정하거나 결함을 검사하는 응용 분야의 경우 신경 회로망등을 이용하여 품질 상태를 추정하려는

시도는 많이 있어왔다. 그러나 이 경우 신경망의 출력값은 비확률적 관계만을 추정하는 것이므로 추정치에 대하여 높은 신뢰도가 요구될 경우 불확실성 정도, 즉 확률적 관계를 추정하는 것은 매우 중요하다. 검사 작업을 통해 불량품이 양품으로 오인되어 문제를 일으킬 확률을 ppm 단위로 조정하려는 경우, 품질 추정 알고리즘의 추정 오차, 즉 분산의 크기는 매우 중요한 변수가 된다. 용접품의 강도예측과 같은 응용분야의 경우 검사품의 품질은 식 (4)에서 보는 바와 같이 분산의 크기와 요구되는 신뢰도 수준에 따라 추정 알고리즘이 제시하는 추정치 $y_{d(x)}$ 보다 과소 평가되는 것이 바람직하다.

Table 1. Result of the modified method using Equation (19)

Input pattern x	σ_T	σ_N (% Error) ^(*) $\varepsilon=0.2$
0.5	0.027304	0.027080(0.82%)
0.6	0.051190	0.052818(1.23%)
0.7	0.085624	0.085237(0.45%)

* Averaged value obtained for the 500 time steps

$$\% \text{ Error} = \left| \frac{\sigma_T - \sigma_N}{\sigma_T} \right| \times 100\%$$

5. 결 론

높은 신뢰성이 요구되는 검사 작업과 같은 응용 분야에 있어서 입출력 변수 사이에 존재하는 불확실한 관계, 즉 확률적 관계의 규명은 매우 중요하다. 이런 경우 분산의 추정은 검사 결과의 신뢰성을 보증하기 위하여 유용하게 사용될 수 있다. 본 논문에서는 분산을 추정하기 위한 새로운 학습 알고리즘을 개발하였고 컴퓨터 모의 실험을 통하여 이를 검증하였다. 알고리즘의 유도를 단순화 하기 위하여 수정항의 크기가 학습 과정 전반에 걸쳐 항상 일정하다고 가정하였으나 2가지 추가적인 파라미터를 사용하여 학습 과정이 보다 안정적이고 빠른 수렴 성능을 가지도록 향상시킬 수 있었다. 1차원 입력에 대해서만 모의 실험 하였지만 학습 방법은 다차원 입력 공간에서도 역시

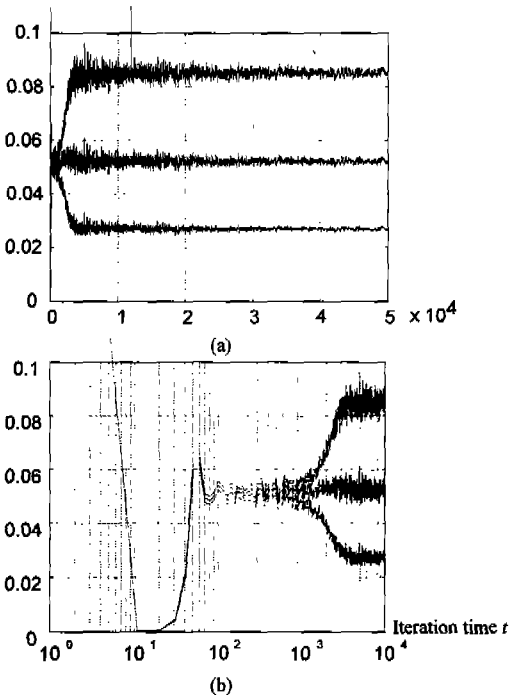


Fig. 5 Learning history of VALEAN for flexible correction

동일하므로 입력 변수가 다차원인 경우도 마찬가지로 적용할 수 있다. 이러한 분산학습 신경회로망을 이용하면 검사 작업등에서 필요한 "신뢰도를 고려한" 추정 작업등을 수행할 수 있는 토대를 제공한다.

참 고 문 헌

1. Hopfield, J. "Neurons with graded response have collective computational properties like those of two-state neurons." Proceedings of National Academy of Sciences (U.S.A), 81, pp3088-3092., 1984.
2. Hecht-Nielsen, R. "Kolmogorov mapping neural network existance theorem." IEEE International Conference on Neural Networks, 3, pp. 11-13, 1987.
3. Hornik, K., Stinchcombe, M., and White, H. "Multilayer feedforward networks are universal approximators." Neural Networks, 2, pp359-366., 1989.
4. Irie, B., and Miyake, S. "Capabilities of three-layered perceptrons." IEEE International Conference on Neural Networks, 1, pp641-648. , 1988.
5. Parzen, E. "On the estimation of a probability density function and mode." Annals of Mathematical Statistics, 33, pp1065-1076., 1962.
6. Rumelhart, D., Hinton, G., and Williams, R. "Learning internal representations by error propagation." In D. Rumelhart, & J. McClelland(Eds.). Parallel Distributed Processing: Exploration in the Microstructure of cognition. Cambridge, MA: MIT press. 1986.
7. Weigend, A. S., and Nix, D. A. "Predictions with confidence intervals (local error bars)." Proc. Int. Conf. Neural Inform. Process, Seoul, Korea, pp847-852, 1994.
8. Peter M. Williams, "Using Neural Networks to Model Conditional Multivariate Densities", Neural Computation, 8, pp843-854, 1996.