

뉴로-퍼지 모델의 신뢰도 계산 : 비교 연구

Reliability Computation of Neuro-Fuzzy Models: A Comparative Study

심현정 · 박래정* · 왕보현*

Hyun-Jeong Shim, Lae-Jeong Park and Bo-Hyeun Wang

(주)보이스웨어

* 강릉대학교 전자공학부

요 약

본 논문은 신경회로망과 같은 경험적 모델에서 출력별 신뢰 구간을 추정하는 세 가지 대표적인 방법을 검토하고, 검토한 방법을 뉴로-퍼지 모델에 적용하여 장단점을 비교 분석한다. 본 논문에서 고려한 출력별 신뢰 구간 계산 방법은 cross-validation을 이용한 stacked generalization, 회귀 모델에서 유도된 predictive error bar, 지역 표현하는 신경회로망의 특성에 기반한 local reliability measure이다. 간단한 함수 근사화 문제와 혼돈 시계열 예측 문제를 이용하여 모의 실험을 수행하고, 세 가지 신뢰도 추정 방법의 성능을 정량적, 정성적으로 비교 분석한다. 분석 결과를 기초로 각 방법의 장단점 및 특성을 고찰하고, 모델링 문제에서 모델의 출력별 신뢰 구간 계산 방법의 실제 적용 가능성을 탐색한다.

Abstract

This paper reviews three methods to compute a pointwise confidence interval of neuro-fuzzy models and compares their estimation performance through simulations. The computation methods under consideration include stacked generalization using cross-validation, predictive error bar in regressive models, and local reliability measure for the networks employing a local representation scheme. These methods implemented on the neuro-fuzzy models are applied to the problems of simple function approximation and chaotic time series prediction. The results of reliability estimation are compared both quantitatively and qualitatively.

Key Words : pointwise confidence interval, stacked generalization, predictive error bar, local reliability measure, neuro-fuzzy models.

1. 서 론

신경회로망의 함수 근사화 능력을 이용하여 비선형 모델을 개발하기 위한 연구가 지난 10여 년간 지속적으로 진행되어왔다. 비선형 시스템의 동정 및 제어 [1], 비선형 예측 모델 [2], 인식 및 분류 문제 [3] 등이 그 대표적인 예이다. 그러나 적용 가능성 측면에서 신경회로망의 이러한 성장에도 불구하고, 진력 수요 예측, 원자력 발전 등과 같이 안정성 및 신뢰성을 우선적으로 확보해야 하는 문제에서 신경회로망 모델과 같은 경험적 모델을 실제 활용하기 위해서는 모델의 신뢰도에 관한 고려가 필수적인 것으로 인식되고 있다.

이러한 인식에서 최근 신경회로망 모델의 출력별 신뢰도를 계산하는 연구가 활발하게 진행되고 있다. 1992년에 Wolpert는 통계학의 cross-validation 개념을 구현한 보조 신경회로망으로 신경회로망 모델의 출력에 대한 오차를 추정하는 stacked generalization 방법을 제안하였다 [4]. 1995년에 Kim 등은 Wolpert의 stacked generalization 방법에서

보조 신경회로망의 입력 변수의 수를 줄이는 방법을 제안하였다 [5]. Satchwell은 회귀 모델을 구현하는 신경회로망에서 출력의 오차의 분산을 추정하는 predictive error bar 방법을 제안하였고 [6], Lowe등은 레이디얼 베이스 함수 네트워크 (RBFN)을 이용하여 이를 실현하는 간단한 방법을 제안하였다 [7]. 한편 1992년 Leonard 등은 RBFN과 같이 지역 표현에 기반한 신경회로망에서 베이스 함수에 대한 지역적 오차 특성을 이용하여 신뢰 구간을 추정하는 통계적인 방법을 제안하였다 [8]. 이후 bootstrap 방법에 기초한 maximum likelihood로 분산을 추정하는 델타 방법 [9], Bayesian error bars [10] 등의 다양한 신뢰도 계산 방법이 제안되었다.

그러나, 신경회로망의 신뢰도를 계산하기 위해서 제안된 다양한 신뢰도 계산 방법 중에서 아직 일반적으로 받아들여지는 표준적인 방법이 존재하지 않는다. 본 논문에서는 이러한 문제를 해결하기 위해서 신경회로망의 신뢰도를 계산하는 3가지 대표적인 방법 (Wolpert, Lowe, Leonard)의 성능을 비교 분석하는 실험을 수행하여 각 방법의 장단점에 관하여 고찰한다. 이를 위해서 뉴로-퍼지 모델을 이용한 함수 근사화 문제와 비선형 예측 모델에 위에서 언급한 3가지 신뢰도 계산 방법을 적용하고, 추정된 모델의 신뢰도에 관한 결과를 정량적, 정성적으로 분석하고, 규칙 레벨에서 해석한다.

접수일자 : 2001년 5월 3일

완료일자 : 2001년 7월 21일

이 논문은 2000년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2000-042-E00032)

본 논문은 다음과 같이 구성되어 있다. 2장에서 신뢰도의 기본 개념과 본 논문에서 비교 실험한 3가지 신뢰도 계산 방법에 대해 간단하게 설명한다. 3장에서는 2장에서 설명한 신뢰도 계산 방법을 간단한 함수 근사화 문제와 혼돈 시계열 예측 문제에 적용하여 각 방법의 장단점 및 신뢰도 계산의 특성을 관찰한다. 4장에서는 결론을 제시한다.

2. 뉴로-퍼지 모델의 출력별 신뢰도 계산 방법

2.1 뉴로-퍼지 모델

본 논문에서는 뉴로-퍼지 모델에 기존의 신뢰도 계산방법을 적용하고자 하므로, 본 절에서는 먼저 뉴로-퍼지 모델의 구조와 학습 방법을 간략하게 소개한다. 뉴로-퍼지 모델은 다음과 같은 형태의 퍼지 규칙을 학습하고 추론한다:

- 규칙 1 : $(x_1 \text{ is } A_1^1) \text{ and } (x_2 \text{ is } A_2^1) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^1), \text{ then } y \text{ is } w^1.$
- 규칙 2 : $(x_1 \text{ is } A_1^2) \text{ and } (x_2 \text{ is } A_2^2) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^2), \text{ then } y \text{ is } w^2.$
- ...
- 규칙 p : $(x_1 \text{ is } A_1^p) \text{ and } (x_2 \text{ is } A_2^p) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^p), \text{ then } y \text{ is } w^p$

여기서 $x_j (1 \leq j \leq n)$ 는 입력 변수이고, y 는 출력 변수이다. A_j^i 와 $w^i (1 \leq i \leq p)$ 는 소속 함수에 의해 정의되는 퍼지 집합이며, 본 논문에서는 식 (1)에서 후건부 퍼지 집합이 상수의 형태를 갖는 간단한 구조의 퍼지 규칙을 고려한다.

그림 1은 두 개의 입력 변수와 하나의 출력 변수를 갖는 뉴로-퍼지 시스템의 구조이다. 두 개의 퍼지 규칙을 표현한 그림 1의 구조에서 입력 층의 입력 노드는 퍼지 규칙의 전건부 소속 함수 A_j^i 를 표현하며, 출력 층의 출력 노드는 후건부 소속 함수 w^i 를 표현한다. 임의의 입력 x 가 주어지면 뉴로-퍼지 모델의 출력은 다음 식으로 계산된다:

$$F(x, w) = \frac{\sum_{i=1}^p w^i R_i(x)}{\sum_{i=1}^p R_i(x)} \quad (2.1)$$

여기서 $R_i(x) = \prod_{j=1}^n \exp \left[- \left(\frac{x_j - C_j^i}{\sigma_j^i} \right)^2 \right]$ (2.2)

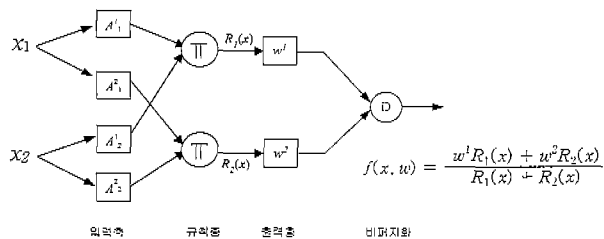


그림 1. 뉴로-퍼지 시스템의 구조.

뉴로-퍼지 시스템의 학습은 학습 데이터를 이용하여 입력 공간을 분할하는 구조 학습과 전건부 소속 함수 A_j^i 와 출력 노드의 후건부 인자 w^i 를 설정된 목적 함수를 최소화하도록

정하는 인자 학습으로 구성된다. 입력 공간 분할 문제를 해결하기 위한 다양한 방법이 제안되었는데 [11, 12, 13], 본 논문은 결정 트리를 이용하여 입력 공간을 분할하는 방법을 이용한다. 함수 근사화 문제에서 결정 트리에 기반한 입력 공간 분할 작업은 주어진 학습 데이터의 출력을 근접화하고, 근접화 결과를 결정 트리를 이용하여 퍼지 규칙의 형태로 변환하고, 변환된 퍼지 규칙을 활용하여 인자 학습의 초기 인자 값을 구하는 것으로 구성된다. 이에 관한 구체적인 내용은 [14]를 참조하기 바란다. 뉴로-퍼지 모델의 인자 학습을 위해서 본 논문에서는 오차 역전파 학습을 이용한다 [15].

2.2 신뢰도의 기본 개념

신경회로망 모델의 신뢰도를 이해하기 위해서 함수 근사화 문제를 고려하자. 함수 근사화 문제는 근사 함수 $F(x, w), x \in R^n$ 를 이용하여 미지의 연속 함수 $f(x)$ 를 근사화하는 것이다. 이를 해결하기 위해서 근사 함수 $F(x, w)$ 가 함수 $f(x)$ 를 가능한 잘 표현하도록 신경망은 신경망을 이용하여 근사 함수의 인자 w 를 학습하는 문제가 된다. 근사 함수를 학습하기 위하여 다음과 같이 m 개의 데이터로 구성된 학습 집합을 고려하자:

$$L = \{(x_i, f(x_i)) | i = 1, 2, \dots, m\} \quad (3)$$

소위 연역 학습 가설이라고 불리는 가정에서 보면 성공적인 학습은 학습 데이터에 포함되지 않은 입력 $x \in R^n - L$ 에 대하여 학습된 근사 함수 $F(x, w^* \leftarrow L)$ 가 미지의 함수 $f(x)$ 의 근사치가 됨을 의미하며, 이를 근사 함수의 일반화라 한다. 함수 근사화 문제에서 근사 함수의 신뢰도를 추정하는 다양한 방법이 존재한다. 가장 기본적이며 간단한 방법은 근사 함수의 일반화 성능을 추정하는 것이다. 이를 위해서 학습 집합 L 에 포함되지 않은 데이터로 구성된 테스트 집합 T 를 수집한다:

$$T = \{(x_i \in R^n - L, f(x_i)) | i = 1, 2, \dots, l\} \quad (4)$$

수집된 테스트 데이터에 대하여 식 (5)와 같이 일반화 오차를 이용한 근사 함수의 정확도를 계산한다:

$$MSE_T = \frac{1}{l} \sum_{i=1}^l \| f(x_i) - F(x_i, w^* \leftarrow L) \|^2 \quad (5)$$

테스트 집합에 대한 일반화 제곱 오차의 평균과 함께 일반화 제곱 오차의 분산을 계산하는 방법 또한 근사 함수의 신뢰도를 추정하기 위해서 일반적으로 사용하는 방법이다. 그러나 모델별 신뢰도를 계산하는 이와 같은 방법은 특정한 입력에 대한 모델의 출력별 신뢰도를 계산할 수 없다는 단점을 갖는다.

모델의 출력별 신뢰도를 계산하기 위해서 통계학의 구간 추정 이론에 기반한 신뢰 구간을 이용한다 [16]. 신뢰 구간은 특정한 입력에 대한 모델 출력의 오차 범위를 추정한다. 학습된 근사 함수에 입력 $x \in R^n - L$ 를 가하고, 연속 함수 $f(x)$ 가 다음의 오차 구간에 존재할 확률을 r 이라 가정하자:

$$\Pr \{ F(x, w^* \leftarrow L) - CL \leq f(x) \leq F(x, w^* \leftarrow L) + CL \} = r \quad (6)$$

여기서 확률 r 을 신뢰 상수라 하고, 오차 구간 $[F(x, w^* \leftarrow L) - CL, F(x, w^* \leftarrow L) + CL]$ 를 $(r \times 100) \%$ 신뢰 구간이라 한다. 주어진 r 에 대하여 CL 을 구하는 것이 신뢰 구간을

추정하는 작업이 된다.

그림 2에서 알 수 있듯이 신뢰 구간이 정확하게 추정되었다고 가정하면, 단일 모델에서 입력에 따른 $(r \times 100)\%$ 신뢰 구간의 넓이는 지역적으로 변하며, 이는 모델의 지역적 신뢰도를 반영한다. 또한 다수의 모델에 대하여 $(r \times 100)\%$ 신뢰 구간을 계산했을 때 신뢰 구간이 좁을수록 그렇지 않은 모델보다 신뢰할 수 있다. 본 논문의 모든 예제에서는 편의상 95% 신뢰 구간을 계산한다. 본 논문에서 비교하고자 하는 신경회로망 모델의 출력별 신뢰 구간 추정 방법 3가지를 다음 절에서 간략하게 소개한다.

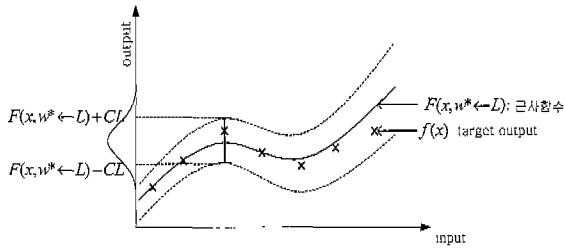


그림 2. 비선형 모델의 신뢰 구간

2.3 모델의 출력별 신뢰 구간 계산 방법

2.3.1 Wolpert의 stacked generalization [4]

Stacked generalization (SG)은 그림 3과 같이 근사 함수를 표현하는 신경회로망 F에 보조 신경회로망 P를 연결하여 F의 신뢰 구간을 추정하는 절차이다. SG 절차의 핵심은 보조 신경회로망 P의 학습 데이터를 어떻게 구성하느냐는 것인데, 이를 위해서 cross-validation 개념을 적용한다 [17].

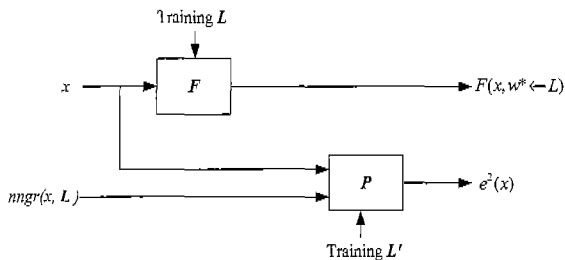


그림 3. Wolpert의 stacked generalization.

학습 시스템의 일반화 성능을 향상하기 위해서 고안된 cross-validation은 그림 4와 같이 주어진 학습 집합 L을 아래의 조건이 만족하도록 K개의 부분 집합 L_i 로 분할한다:

$$\bigcup_{i=1}^K L_i = L \quad (7.1)$$

$$L_i \cap L_j = \emptyset, \text{ for } i \neq j \quad (7.2)$$

여기서 L_i 를 GMDH (group method for data handling) 분할 집합이라 하고, K는 m/K 이 정수가 되도록 선택한다. m/K 개의 데이터로 구성된 i 번째 분할 집합 L_i 를 검증 집합 (validation set) L_2 라하고, F를 테스트하는 단계에서 이를 사용한다. L_2 를 제외한 나머지 데이터 (즉, $L - L_2$)를 L_1 이라 하고, 이를 F를 학습할 때 사용하기 위한 학습 집합 (learning set)으로 준비한다.

학습 집합 L_1 을 이용하여 학습된 F를 L_2 를 이용하여 검증하는 단계에서 검증 데이터의 입력에 대한 F의 출력 오차를 확보할 수 있다:

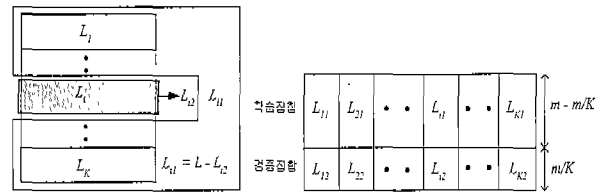


그림 4. GMDH 분할 집합.

$$e(x_k) = F(x_k, w^* \leftarrow L_1) - f(x_k), \text{ for } x_k \in L_2, k=1, 2, \dots, m/K \quad (8)$$

출력별 신뢰 구간을 추정하기 위하여 F의 검증 단계에서 확보한 출력 오차의 제곱 $e^2(x_k)$ 을 P의 학습 데이터의 desired 출력으로 하고, 학습 데이터의 입력 벡터 x_{aug} 를 x_k 와 학습 집합 L_1 에 속하는 데이터 중에서 x_k 에 가장 가까운 벡터를 augment하여 구성한다:

$$x_{aug}(x_k) = [x_k, mng(x_k, L_1)] \quad (9)$$

그러므로 P의 학습 데이터에서 입력 벡터의 차원은 F의 학습 데이터의 입력 벡터 차원의 두 배가 된다. 검증 집합 L_2 에 속한 모든 데이터에 대하여 위의 과정을 수행하면 m/K 개의 학습 데이터로 구성된 L'_i 을 얻을 수 있다:

$$L'_i = \{(x_{aug}(x_k), e^2(x_k)) \mid k=1, 2, \dots, m/K\} \quad (10)$$

i 를 1에서부터 증가시키면서 F의 학습을 분할 집합의 수 K번 반복하면, P의 학습 집합 L' 이 구성된다:

$$L' = \bigcup_{i=1}^K L'_i \quad (11)$$

구성된 L' 을 이용하여 보조 신경회로망 P를 학습하고, 전체 학습 집합 L을 이용하여 신경회로망 F를 재학습하여 SG 절차를 완료한다. 만약 임의의 입력 벡터 x 가 주어지면, 신경회로망 F에서 함수 근사화의 일반화 오차가 다음과 같이 계산된다:

$$e(x) = F(x, w^* \leftarrow L) - f(x), \text{ for } x \in T \quad (12)$$

입력 벡터 x 를 이용하여 보조 신경회로망 P의 입력 벡터 x_{aug} 를 구성하고, 이를 P에 가하면 x 에 대한 F의 일반화 오차의 제곱이 추정된다:

$$e^2(x) = P(x_{aug}, w^* \leftarrow L') \quad (13)$$

여기서 $x_{aug} = [x, mng(x, L)]$ 이다. 보조 신경회로망 P에서 계산된 $e^2(x)$ 를 이용하여 $F(x, w^* \leftarrow L)$ 의 신뢰 구간을 추정하는 작업은 보조 신경회로망 P의 학습을 2.3.2절에서 소개할 통계적 환경의 회귀 모델로 해석하면 가능하다.

SG 절차에서 F와 P의 선택은 자유롭다. 예를 들어, 오차 역전과 학습을 사용하는 다층 퍼셉트론, RBFN, 뉴로-퍼지 모델, 결정 트리 등과 같이 학습에 의해 모델을 구성하는 시스템은 모두 F와 P로 사용할 수 있다. 또한 F를 구현하기 위하여 다수의 모듈을 사용하는 경우 SG는 함수 근사화 문

제에서 학습 데이터에 대한 출력의 바이어스를 추정하여 일반화 성능을 향상하는 방법으로 사용된다. 이에 관한 자세한 사항은 [4]를 참조하기 바란다.

2.3.2 Lowe의 predictive error bar [7]

Predictive error bar (PEB)는 SG와 유사한 전략을 취하고 있으나, 함수 근사화 문제를 통계적 환경으로 확장하여 얻어진다. 통계적 환경에서 함수 근사화 문제는 회귀 모델(regressive model)로 표현된다 [18]:

$$D = f(X) + \epsilon \quad (14)$$

여기서 X 는 독립 변수로 구성된 랜덤 벡터이고, D 는 랜덤 변수이며, 회귀 함수 $f(X)$ 는 결정론적 함수라 가정한다. ϵ 은 X 와 D 의 상관 관계에서 불확실성을 표현하는 랜덤 오차이고, 평균 $E[\epsilon]$ 은 0이다. 회귀 모델에서 x 가 주어졌을 때 $f(x)$ 는 회귀 모델의 출력 D 의 조건 평균이 된다:

$$f(x) = E[D|X=x] \quad (15)$$

신경회로망 $F(x, w)$ 를 이용하여 식 (14)로 주어진 회귀 모델을 구현하기 위하여 랜덤 벡터 X 로부터 학습 집합을 구성한다고 하자:

$$L = \{(x_i, d_i) | i = 1, 2, \dots, m\} \quad (16)$$

신경회로망 F 의 연결 벡터 w 는 아래 주어진 비용 함수를 최소화하여 구해진다:

$$C(w) = \frac{1}{m} \sum_{i=1}^m (d_i - F(x_i, w))^2 \quad (17)$$

식 (17)로 주어진 비용 함수는 회귀 모델의 직교 특성을 이용하여 다음과 같이 표현된다:

$$C(w) = \frac{1}{m} \sum_{i=1}^m \epsilon_i^2 + \frac{1}{m} \sum_{i=1}^m (f(x_i) - F(x_i, w))^2 \quad (18)$$

식 (18)의 우변 첫 항은 학습 집합 L 에 대한 회귀 모델의 랜덤 오차의 분산이 된다. 오차의 분산이 연결 벡터 w 와 무관한 값으로 주어지기 때문에 비용 함수의 최소화는 식 (18)의 두 번째 항을 최소화하는 문제가 된다. 최소제곱오차로 주어진 비용 함수를 최적화 하여 얻어진 신경회로망 F 의 출력은 임의의 입력 벡터 x 에 대한 회귀 모델의 출력 D 의 조건 평균을 근사화 한다:

$$F(x, w^* \leftarrow L) \approx f(x) = E[D|X=x] \quad (19)$$

SG에서와 같이 신뢰 구간을 추정하기 위한 보조 신경회로망 P 를 학습하는데 필요한 학습 집합 L' 을 미리 학습한 신경회로망 $F(x, w^* \leftarrow L)$ 를 이용하여 구성한다:

$$L' = \{(x_i, e^2(x_i)) | i = 1, 2, \dots, m\} \quad (20.1)$$

여기서 L' 의 desired 출력은 다음과 같이 계산된다:

$$e^2(x_i) = (d_i - F(x_i, w^* \leftarrow L))^2 \quad (20.2)$$

구성된 학습 집합 L' 을 이용하여 보조 신경회로망 $P(x, w)$ 를 다음의 비용 함수를 최적화 하도록 학습한다:

$$C'(w) = \frac{1}{m} \sum_{i=1}^m (e^2(x_i) - P(x_i, w))^2 \quad (21)$$

학습이 완료된 보조 신경회로망 P 에 임의의 입력 x 를 가

하여 얻은 출력은 신경회로망 F 의 출력과 회귀 함수 f 의 출력의 제곱 오차의 조건 평균이 된다:

$$P(x, w^* \leftarrow L) \approx E[(F(x, w^* \leftarrow L) - E[D|X=x])^2 | X=x] \quad (22)$$

만약 $e(x)$ 가 평균이 0인 정상 분포를 따른다고 가정하면, $P(x, w^* \leftarrow L)$ 는 $e(x)$ 의 분산이 되고, 이를 이용하여 $F(x, w^* \leftarrow L)$ 의 95% 신뢰 구간을 다음과 같이 추정할 수 있다:

$$CI = [F(x, w^* \leftarrow L) - n_{0.95} P(x, w^* \leftarrow L) \leq f(x) \leq F(x, w^* \leftarrow L) + n_{0.95} P(x, w^* \leftarrow L)] \quad (23)$$

여기서 $n_{0.95}$ 는 정상 분포에서 95% 신뢰도의 결정 값이다.

1998년에 Lowe 등은 그림 5에서와 같이 PEB의 F 와 P 를 구현할 때 RBFN의 입력층과 은닉층을 공유하는 간단한 방법을 제안하고, 제안된 방법이 기존의 다른 신뢰 구간 계산 방법과 비교하여 구현이 용이하며 정확한 신뢰도 계산이 가능함을 입증하였다 [7]. PEB 방법에서 보조 신경회로망의 학습 집합 L' 을 얻기 위해서 SG 절차에서 사용한 cross-validation 개념을 활용할 수도 있으나, 본 논문은 Lowe의 방법을 재현하기 위해서 학습 집합 전체를 이용하여 한번만 신경회로망 F 를 학습하여 L' 을 구성한다. 이 경우 PEB는 통계적인 환경에서 유도된 방법이지만 기본적으로 SG 절차의 특별한 경우로 생각할 수 있다.

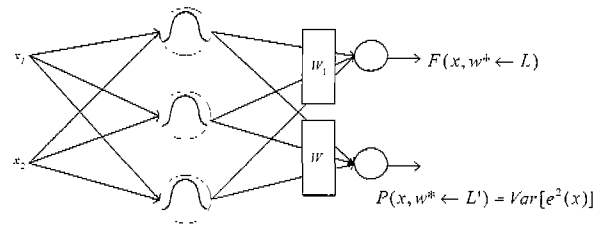


그림 5. Lowe의 predictive error bar의 RBFN 구현.

2.3.3 Leonard의 local reliability measure [8]

위에서 언급한 SG와 PEB는 연역 학습을 하는 어떠한 모델에도 적용 가능한 방법이다. 1992년 Leonard는 RBFN, 뉴로-퍼지 모델과 같이 지역 표현에 기반한 신경회로망에 적용할 수 있는 출력별 신뢰 구간 계산 방법을 제안하였다 [8]. 제안된 신뢰 구간 계산 방법은 각 지역에 해당하는 베이스 함수에 대한 지역 신뢰 구간을 계산하고, 임의의 테스트 입력에 대해서 해당하는 베이스 함수의 신뢰 구간의 가중 평균을 모델 출력의 신뢰 구간으로 출력한다.

그림 1의 뉴로-퍼지 모델을 식 (3)으로 주어진 학습 집합을 이용하여 학습하고, 모델의 오차를 구한다:

$$e(x_i) = F(x_i, w^* \leftarrow L) - f(x_i), \text{ for } x_i \in L, i = 1, 2, \dots, m \quad (24)$$

여기서 모델의 오차 $e(x_i)$ 는 Gaussian 랜덤 변수라 가정한다. 구해진 m 개의 샘플을 이용하여 h 번째 퍼지 규칙에 대한 모델 오차의 지역 평균과 지역 분산을 추정한다:

$$\hat{\mu}_h = \frac{1}{n_h} \sum_{i=1}^m R_h(x_i) e(x_i) \quad (25)$$

$$s_h^2 = \frac{1}{(n_h - 1)} \left[\sum_{i=1}^m R_h(x_i) e^2(x_i) \right] \quad (26)$$

3. 모 의 실 험

여기서 n_h 는 h 번째 규칙과 연관된 입력 데이터의 수를 의미하며 다음과 같이 계산된다:

$$n_h = \sum_{i=1}^m R_h(x_i) \quad (27)$$

h 번째 규칙에 대한 모델 오차의 지역 평균을 0이라 가정하면, 식 (28)과 같이 정의된 랜덤 변수 t 는 자유도가 $(n_h - 1)$ 인 Student t -분포를 따른다:

$$t(n_h - 1) \sim \frac{\widehat{\mu}_h - \mu_h}{s_h/\sqrt{n_h}} = \frac{-\mu_h}{s_h/\sqrt{n_h}} \quad (28)$$

여기서 μ_h 는 h 번째 규칙에 대한 모델 오차의 실제 지역 평균값이다. h 번째 규칙에서 모델 오차의 실제 지역 평균값이 존재할 95% 신뢰 구간은 h 번째 규칙과 연관된 모델 오차의 샘플 평균 $\widehat{\mu}_h$ 의 분산 s_h^2/n_h 을 이용하여 다음과 같이 구할 수 있다:

$$CI_h = [\widehat{\mu}_h - CL_h \leq \mu_h \leq \widehat{\mu}_h + CL_h] \quad (29)$$

여기서 $CL_h = t_{0.95} \frac{s_h}{\sqrt{n_h}}$ 이고, $t_{0.95}$ 는 Student t -분포에서 $(n_h - 1)$ 자유도를 갖는 95% 신뢰도의 결정 값이다.

일반적으로 모델 오차의 평균에 대한 신뢰 구간보다 우리는 임의의 입력에 대한 모델의 출력 오차에 대한 신뢰 구간이 필요하다. 이를 위해서 평균의 분산과 모델 오차의 분산의 합을 사용하여 h 번째 규칙에 대한 오차 한계를 계산한다 [19]:

$$CL_h' = t_{0.95} s_h \sqrt{(1 + 1/n_h)} \quad (30)$$

임의의 테스트 입력 x 에 대한 모델의 오차 한계는 각 규칙에 대하여 식 (30)에서 계산한 규칙별 오차 한계의 가중 평균으로 계산된다:

$$CL(x) = \frac{\sum_{h=1}^H CL_h' R_h(x)}{\sum_{h=1}^H R_h(x)} \quad (31)$$

모델의 오차 한계를 이용하여 실제 출력 $f(x)$ 가 존재하는 95% 신뢰 구간을 다음과 같이 추정한다:

$$CI = [F(x, w^* - L) - CL(x) \leq f(x) \leq F(x, w^* + L) + CL(x)] \quad (32)$$

식 (31)과 (32)에서 알 수 있듯이 모델의 출력 오차에 대한 신뢰 구간은 각 퍼지 규칙의 신뢰 구간을 가중 평균하여 구할 수 있으며, 뉴로-퍼지 모델에 은닉층을 공유하는 별도의 출력층을 추가하여 그림 6과 같이 간단하게 구현할 수 있다.

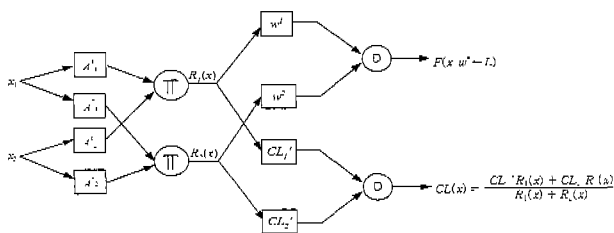


그림 6. Leonard의 local reliability measure.

본 장에서는 2.3절에서 언급한 3가지 신뢰 구간 계산 방법의 성능을 비교 분석하고 각 방법의 장단점을 파악하기 위해서 간단한 함수 근사화 문제와 비선형 예측 문제에 신뢰도 계산 방법을 적용하는 모의 실험을 수행한다. 각 신뢰도 계산 방법에 대한 성능 평가는 1) 정량적 분석, 2) 정성적 분석, 3) 규칙 단위 분석으로 이루어진다. 정량적 성능 평가는 테스트 집합을 이용하여 신뢰 구간의 통계적 특성을 분석한다. 예를 들어, 95% 신뢰 구간을 계산한 경우 테스트 데이터의 출력이 계산된 신뢰 구간에 몇 %나 포함되어 있는가를 정량적으로 계산한다. 신뢰 구간의 정성적 분석 기법은 학습 데이터에서 입력 데이터의 밀도와 출력이 포함된 잡음에 대한 신뢰 구간의 영향을 평가한다. 학습 집합에서 입력의 밀도가 높은 영역에서는 신뢰 구간이 좁을 것이고, 출력에 포함된 잡음이 큰 구간에서는 신뢰 구간이 넓게 나타날 것이다. 마지막으로 신뢰 구간의 규칙 단위 해석은 뉴로-퍼지 모델에서 학습한 각 규칙의 신뢰도를 평가한다. 뉴로-퍼지 모델의 규칙은 입력 공간에서 특정한 영역을 표현하기 때문에 규칙 단위에서의 신뢰도 분석은 뉴로-퍼지 모델의 성능 향상을 위한 가이드를 제공할 수 있다.

3.1 간단한 함수 근사화 문제

본 절에서는 단일 입력 단일 출력을 갖는 간단한 함수 근사화 문제에 2.3절에서 설명한 신뢰 구간 계산 방법을 적용하고, 각 방법의 성능 평가를 수행한다. 함수 근사화에 사용된 함수는 다음과 같다:

$$y_1(x) = 0.5 \sin(1.5\pi x + \pi/2) + 2.0 + v_1 \quad (33)$$

$$y_2(x) = 0.5 \sin(1.5\pi x + \pi/2) + 2.0 + v_2 \quad (34)$$

여기서 입력 변수 x 는 $[-1, 1]$ 에서 uniform 분포를 갖고, v_1 과 v_2 는 Gaussian 잡음이다. v_1 의 표준 편차는 0.03이고, v_2 의 표준 편차는 $(0.045 + 0.04x)$ 이다. 식 (34)로 주어진 함수에서 x 가 1에 접근하면 출력에 포함된 잡음이 커짐을 알 수 있다. 본 절에서 사용하는 마지막 함수 $y_3(x)$ 는 식 (33)의 형태를 그대로 가지나, 입력 변수 x 가 다음의 확률 밀도 함수로 표현되는 경우를 고려한다:

$$f(x) = \begin{cases} 0.5 + 0.45x, & -1 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

식 (35)에서 알 수 있듯이 $y_3(x)$ 에 대한 학습 집합은 $x = -1$ 근처에서 적은 수의 데이터를 갖고, $x = 1$ 근처에서 많은 수의 데이터를 갖는다.

함수 근사화 문제를 해결하기 위하여 위의 세 함수로부터 각각 200개의 데이터를 수집한다. 수집된 200개의 데이터에서 처음 50개는 학습 데이터로, 나머지 150개는 테스트 데이터로 활용한다. 뉴로-퍼지 모델을 학습하기 위해서 2.1절에서 언급한 것과 같이 먼저 뉴로-퍼지 모델의 구조 학습을 수행한다. 구조 학습을 위해서 k -means 알고리즘을 이용하여 학습 데이터의 출력을 군집화 하고, 출력 군집을 입력 공간에 투영한다. 여기서 군집의 수는 최초에 2에서 시작하여 1씩 증가시켜 가며 군집의 수가 4가 될 때까지 반복 수행한다. 입력 공간에 투영하는 작업을 완료하면 결정 트리를 사용하여 입력 공간을 분할한다. 입력 공간에 대한 분할 정보를 이용하여 뉴로-퍼지 모델을 초기화하고, 오차 역전파 학습을 이용하여 뉴로-퍼지 모델의 인자 학습을 수행한다.

표 1은 인자 학습을 완료하고 테스트 데이터를 이용하여 뉴로-퍼지 모델의 일반화 성능을 평가한 결과를 요약한다. 표 1에서 알 수 있듯이 세 함수 모두에서 규칙의 수가 증가하면 학습 오차는 감소하나, 테스트 데이터에 대한 오차는 오히려 증가하는 현상을 관찰할 수 있다. 이는 단순한 구조를 갖는 모델의 일반화 성능이 그렇지 않은 모델의 일반화 성능보다 우수하다는 Occam's razor를 시연하는 것으로 해석될 수 있으며, 모델의 구조를 결정하는 과정이 모델의 일반화 성능에 필수적임을 입증하는 결과이다.

표 1. 뉴로-퍼지 모델의 함수 근사화 성능.

함수	$y_1(x)$			$y_2(x)$			$y_3(x)$		
균집의 수	2	3	4	2	3	4	2	3	4
규칙의 수	3	7	9	3	7	10	3	6	8
학습오차 (RMSE)	0.113	0.026	0.020	0.115	0.051	0.047	0.108	0.034	0.030
테스트오차 (RMSE)	0.110	0.051	0.052	0.100	0.067	0.087	0.110	0.045	0.048

표 2. 함수 근사화에서 신뢰 구간의 정량적 분석 결과.

Models	$y_1(x)$	$y_2(x)$	$y_3(x)$
Stack generalization	76.00%	88.00%	91.33%
Predictive error bar	72.00%	79.33%	84.67%
Local Reliability measure	84.00%	88.00%	91.33%

세 함수 각 각에서 일반화 성능이 가장 우수한 뉴로-퍼지 모델에 대하여 2.3절에서 소개한 방법으로 테스트 집합에 대한 95% 신뢰 구간을 계산한다. 신뢰 구간 내에 실제 출력이 존재하는 확률을 정리한 표 2로부터 다음과 같은 정량적 분석 결과를 얻을 수 있다. 첫째, Wolpert의 방법과 Leonard의 방법이 거의 유사한 성능을 보이지만, Leonard 방법에서 95%에 가장 가까운 신뢰 구간이 계산되었다. 둘째, 주어진 함수에 따라 추정된 신뢰 구간의 정확성이 변하였다. 즉, 주어진 문제의 통계적 특성에 따라 동일한 신뢰도 계산 방법이라도 추정된 신뢰 구간의 정확성이 달라질 수 있다고 판단된다. 셋째, 세 방법 모두에서 실제 신뢰 구간보다 더 tight하게 신뢰 구간을 추정하였다.

계산된 신뢰 구간의 정성적 분석을 위한 그림 7은 세 함수 각 각에 대하여 추정된 신뢰 구간과 학습된 퍼지 규칙의 소속 함수를 나타낸다. 신뢰 구간내의 실선은 모델의 출력이고, x 마크는 함수의 실제 출력 값이다. 그림 7로부터 다음의 결과를 유도 할 수 있다. 첫째, 입력 데이터의 밀도가 작은 영역의 신뢰 구간이 큰 영역의 신뢰 구간보다 크게 계산되었다. (그림 7의 (a-3), (b-3), (c-3) 참조.) 이는 함수 근사화에서 입력 데이터의 밀도 때문에 모델의 일반화 성능이 떨어지는 것을 신뢰 구간의 계산에서 세 방법 모두 잘 반영할 수 있음을 의미한다. 둘째, Lowe와 Leonard의 방법에서는 학습 데이터에 포함된 잡음이 큰 영역에서 계산된 신뢰 구간이 그렇지 않은 영역의 신뢰 구간보다 넓게 계산되었으나, Wolpert의 방법에서는 이와 같은 신뢰 구간의 지역적 특성을

을 관찰 할 수 없었다. (그림 7의 (a-2), (b-2), (c-2) 참조.) 이와 같은 현상은 학습된 퍼지 규칙의 소속 함수의 두께를 신뢰 구간의 크기에 비례하도록 표시한 그림 7에서 명확하게 관찰할 수 있다. 이에 대한 이유로는 Wolpert의 SG에서 학습이 정확하게 수행된 F에서 구해진 보조 신경회로망 P의 학습 데이터가 신뢰 구간의 지역적 특성을 잘 표현하고 있지 못하기 때문으로 판단된다.

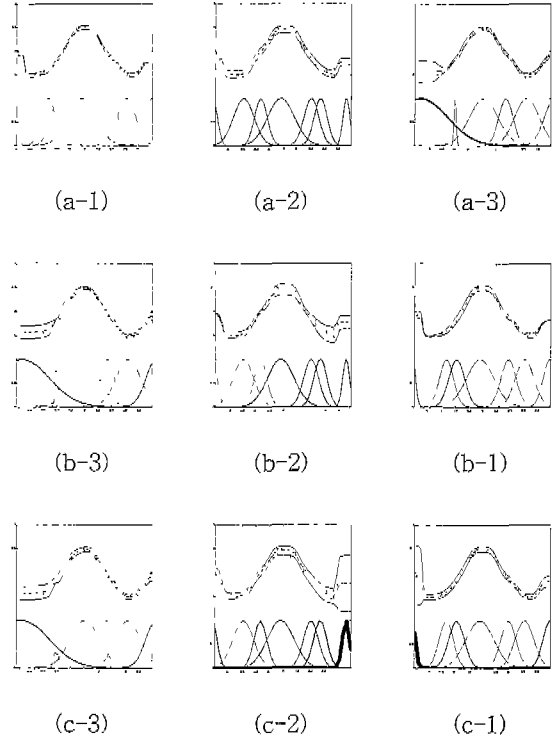


그림 7. 뉴로-퍼지 모델에서 추정된 신뢰 구간 (x: 실제 출력), (a) SG by Wolpert, (b) PEB by Lowe, (c) LRM by Leonard, (1) $y_1(x)$, (2) $y_2(x)$, (3) $y_3(x)$.

3.2 혼돈 시계열 예측 문제

본 절에서는 혼돈 시계열 예측 문제에 신뢰 구간 계산 방법을 적용하고, 그 결과를 비교 분석한다. 예측시스템을 개발하기 위해서 필요한 데이터를 식 (36)과 같이 주어진 Mackey-Glass 동역학 시스템을 이용하여 수집한다 [20]:

$$x(k+1) = (1-a)x(k) + \frac{bx(k-\tau)}{1+x^{10}(k-\tau)} \quad (36)$$

여기서 $a = 0.1$, $b = 0.2$, $\tau = 17$ 이다. 예측 모델의 구조는 식 (37)로 주어진다 가정하자:

$$x(k+\Delta k) = F(x(k), x(k-6), x(k-12), x(k-18)) \quad (37)$$

여기서 Δk 는 예측 리드 타임이고, 본 예제에서는 Δk 를 6, 24, 85로 정하고 세 개의 예측 모델을 구현하였다.

예측 모델을 학습하기 위해서 식 (36)를 이용하여 2000개의 데이터를 생성한다. 생성된 2000개의 데이터 중에서 $k=125$ 에서부터 $k=1124$ 까지 1000개의 데이터를 수집해 처음 500개는 학습 데이터로 나머지 500개는 테스트 데이터로 사용한다. 수집된 학습 데이터를 이용하여 뉴로-퍼지 모델의

구조 학습과 인자 학습을 수행하고, 테스트 데이터를 이용하여 예측 모델의 일반화 성능을 평가한다. 표 3에서 각 예측 리드 타입에 대한 예측 모델의 학습 성능과 일반화 성능을 요약한다. 표 3에서 모델 1은 표 1과 같이 규칙의 수가 증가하면서 학습 오차가 작아지지만 일반화 성능은 감소함을 알 수 있다. 그러나 모델 2와 모델 3은 규칙의 수가 증가하면서 학습 성능과 일반화 성능이 모두 개선되고 있다. 이 경우 정확한 예측시스템을 개발하는 입장에서는 군집의 수를 늘려 더 많은 퍼지 규칙을 갖는 예측 모델을 만들어야 하지만, 본 논문의 목표가 신뢰 구간 계산에 있기 때문에 모델 2에서는 28개, 모델 3에서는 60개의 규칙으로 예측 시스템의 정확도 사양이 만족되었다고 가정한다.

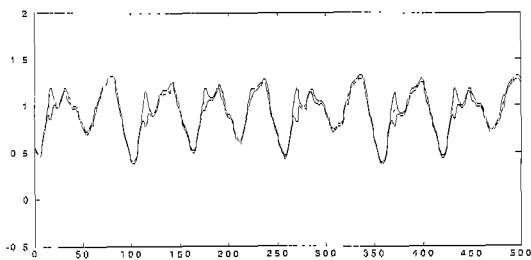
학습된 예측 모델 중에서 일반화 성능이 가장 우수한 모델을 선택하여 신뢰 구간을 추정하고, 테스트 데이터에 대하여 예측 값이 신뢰 구간 내에 존재할 확률을 표 4에 요약하였다. 표 4로부터 3.1절의 함수 근사화 문제와는 다르게 Leonard의 신뢰 구간 추정 방법이 다른 두 방법보다 월등히 우수함을 알 수 있다. 또한, 예측 리드 타임이 길어지더라도 (즉, 정확한 예측이 어렵게 되어도), Leonard 방법으로 추정된 신뢰 구간은 이를 잘 반영하고 있으나, Wolpert와 Lowe의 방법은 신뢰 구간을 더욱 tight하게 계산하고 있음을 알 수 있다. 그림 8은 예측 모델의 출력과 Leonard의 방법에 의해 추정된 신뢰 구간을 보인다.

표 3. 혼돈 시계열 예측시스템의 학습 및 일반화 성능.

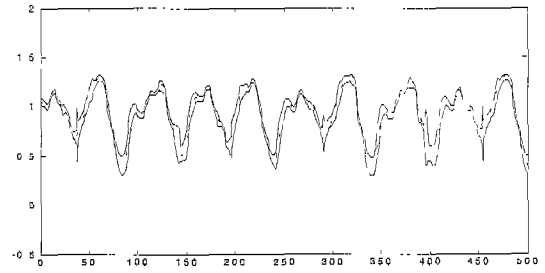
	모델 1 ($\Delta k=6$)			모델 2 ($\Delta k=24$)			모델 3 ($\Delta k=85$)		
	2	3	4	2	3	4	2	3	4
군집의 수	2	3	4	2	3	4	2	3	4
규칙의 수	10	25	26	8	24	28	21	50	60
학습성능 (RMSE)	0.011	0.008	0.007	0.018	0.012	0.010	0.054	0.027	0.018
일반화성능 (RMSE)	0.014	0.0120	0.0124	0.023	0.023	0.017	0.089	0.059	0.056

표 4. 시계열 예측 모델의 정량적 신뢰 구간 분석.

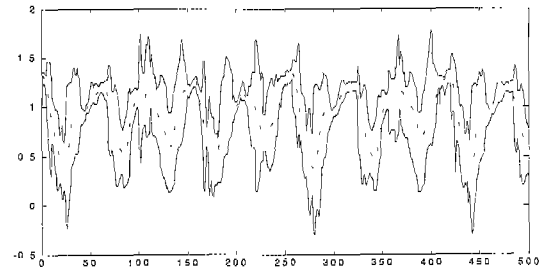
Models	모델 1 (규칙의 수=25)	모델 2 (규칙의 수=28)	모델 3 (규칙의 수=60)
Stack generalization	76.6 %	79.4 %	69.8 %
Predictive error bar	79.0 %	78.8 %	61.0 %
Local reliability measure	84.4 %	92.9 %	98.4 %



(a)



(b)



(c)

그림 8. Mackey-Glass 혼돈 시계열 예측에서 Leonard의 신뢰 구간 추정 결과 (실선: 신뢰 구간, 점선: 실제 출력) (a) 모델 1, (b) 모델 2, (c) 모델 3.

4. 결론

신경회로망과 같이 학습 데이터에 의존하는 경험적 모델에서 출력별 신뢰 구간을 계산하는 능력은 안정성과 신뢰성이 중요한 문제에서 필수적이다. 더욱이 지금까지 제안된 다양한 신뢰 구간 추정 방법 중에서 어느 것이 일반적으로 가장 우수한 성능을 보이는지 밝혀져 있지 않기 때문에, 본 논문에서는 대표적인 세 가지 출력별 신뢰 구간 계산 방법 (Wolpert, Lowe, Leonard)을 간단한 함수 근사화 문제와 혼돈 시계열 예측 문제에 적용하여 추정된 신뢰 구간의 성능을 비교하는 모의 실험을 수행하였다. 실험 결과를 정량적, 정성적으로 분석한 결과 지역 표현하는 신경회로망의 특성을 고려하여 신뢰 구간을 추정하는 Leonard의 방법이 일관되게 좋은 성능을 유지함을 알 수 있었다.

일반적으로 뉴로-퍼지 모델의 구조 학습은 적은 수의 퍼지 규칙으로도 보다 정확한 모델을 얻는데 기여하는 것으로 알려져 있다. 본 논문에 그 결과를 포함하지 않았지만, 결정 트리에 기반한 구조 학습을 선행한 뉴로-퍼지 모델의 신뢰 구간 계산 결과가 구조 학습을 수행하지 않은 경우보다 우수하였다 [21]. 이는 구조 학습이 정확한 모델을 얻는 것 뿐만 아니라 모델의 정확한 신뢰도 계산에도 중요함을 의미한다.

본 논문의 결과를 기초로 향후 진행되어야 할 연구의 방향은 크게 두 가지로 요약된다. 첫째, 검토한 신뢰 구간 계산 방법을 단기 전력 수요 예측과 같이 모델의 정확도뿐만 아니라 예측의 신뢰도가 중요한 실제 문제에 적용하여 그 성능을 입증하는 연구를 진행할 계획이다. 둘째, 정확한 신뢰도 계산을 바탕으로 비선형 모델의 신뢰도를 향상시키는 방법에 관한 연구를 수행할 예정이다. 지역적 특성을 효과적으로 반영

하여 신뢰 구간을 계산하는 방법에서 규칙 단위로 수행한 신뢰 구간 분석이 뉴로-퍼지 모델의 신뢰도를 향상하는데 효과적으로 활용될 수 있을 것이라 기대된다.

참 고 문 헌

- [1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp. 4-27, 1990.
- [2] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modeling," Los Alamos National Laboratory Tech. Rep. A-UR-87-2662, 1987.
- [3] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, 1989.
- [4] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241-259, 1992.
- [5] K. Kim. and E. B. Bartlett, "Error estimation by series association for neural network systems," *Neural Computation*, vol. 7, pp. 799-808, 1995.
- [6] C. Satchwell, "Finding error bars (the easy way)," *Neural Computing Applications Forum*, Edition 5, 1994.
- [7] D. Lowe and K. Zapart, "Point-wise confidence interval estimation by neural networks : A comparative study based on automotive engine calibration," *Technical Report NCRG/98/007* 10, 1998.
- [8] J. A. Leonard, M. A. Kramer and L. H. Ungar, "A neural network architecture that computes its own reliability," *J. of Computer Chem. Eng.*, vol. 16, no. 9, pp. 819-835, 1992.
- [9] R. Tibshirani, "A comparison of some error estimates for neural network models," *Neural Computation*, vol. 8, pp. 152-163, 1996.
- [10] C. Williams, C. Qazaz, C. M. Bishop and H. Zhu, "On the relationship between Bayesian error bars and the input data density," *IEE. Conference Publication* no. 409, pp. 160-165, 1995.
- [11] M. Sugeno and T. Yasukawa, "A fuzzy logic based approach to qualitative modeling," *IEEE Trans. on Fuzzy Systems*, vol. 1, no. 1, pp. 7-31, 1993.
- [12] M. Kubat, "Decision trees can initialize radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 9, no. 5, pp. 813-821, 1998.
- [13] B. H. Wang and H. J. Cho, "Structure identification of neuro-fuzzy models using genetic algorithms," *submitted to Fuzzy Sets and Systems*, 2000.
- [14] 박영진, 심현정, 왕보현, "뉴로-퍼지 모델을 이용한 단기 전력 수요 예측 시스템," *대한 전기 학회 논문지*, vol. 49A, no. 3, pp. 107-117, 2000.
- [15] P. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, 1993.
- [16] H. Stark and J. W. Wood, *Probability, Random Process and Estimation Theory for Engineers*, Prentice Hall, pp. 270-288, 1994.
- [17] M. Stone, "Cross-validatory choice and assessment of statistical prediction," *J. Roy. Stat. Sec. Ser. B.36*, pp. 111-147, 1974.
- [18] H. White, "Learning in artificial neural networks: A statistical perspective," *Neural Computation*, vol. 1, pp. 425-464, 1989.
- [19] N. R. Draper and H. Smith, *Applied Regression Analysis*, 2nd Eds., John Wiley & Sons, Inc., 1966.
- [20] R. M. Kil and J. Y. Choi, "Chaotic time series prediction based on a network with Gaussian kernel function," in: *Proc. JCEANF*, Seoul, Korea, pp. 95-99, 1993.
- [21] 심현정, "뉴로-퍼지 모델에 기반한 단기 전력 수요 예측시스템의 신뢰도 계산에 관한 연구," *강릉대학교 전자공학과 석사학위논문*, 2001.

저 자 소 개



심 현 정 (Hyun-Jeong Shim)
1999년 : 강릉대학교 전자공학과 공학사
2001년 : 강릉대학교 전자공학과 공학 석사
2001년~현재 : (주)보이스웨어 연구원
관심분야 : 데이터 마이닝, 신경회로망, 음성인식 및 합성

Phone : +82-2-2188-5761
Fax : +82-2-2188-5776
E-mail : hjshim@voiceware.co.kr



왕 보 현 (Bo-Hyeun Wang)
1987년 : 연세대학교 전기공학과 공학사
1990년 : Georgia Institute of Tech. 공학 석사
1991년 : Georgia Institute of Tech. 공학 박사
1991년~1998년 : LG 종합기술원 책임연구원

1998년~현재 : 강릉대학교 전자공학과 조교수
관심분야 : 기계 학습, 데이터 마이닝, 예측 시스템

Phone : +82-33-640-2384
Fax : +82-33-640-2244
E-mail : bhw@kangnung.ac.kr



박 래 정 (Lae-Jeong Park)
1991년 : 서울대학교 전기공학과 공학사
1993년 : 한국과학기술원 전기및전자공학과 공학석사
1997년 : 한국과학기술원 전기및전자공학과 공학박사
1997년~1999년 : LG종합기술원 정보기술 연구소 선임연구원

2000년~현재 : 강릉대학교 정보전자공학부 전임강사
관심분야 : 기계 학습, 진화 연산, 데이터 마이닝

Phone : +82-33-640-2389
Fax : +82-33-640-2244
E-mail : ljpark@kangnung.ac.kr