

# RBF 망 이용 소프트웨어 개발노력 추정 성능향상

이 상 운<sup>†</sup> · 박 영 목<sup>\*\*</sup> · 박 재 흥<sup>\*\*\*</sup>

## 요 약

소프트웨어 개발에서 점점 더 중요시되는 사항은 개발 생명주기의 초기에 개발과 관련된 노력과 비용을 추정하는 능력이다. 소프트웨어 개발노력과 비용추정을 위한 대부분의 모델이 선형회귀분석 절차를 사용하였다. 그러나 소프트웨어의 복잡성, 개발환경의 다양성으로 인해 소프트웨어 개발노력과 비용 추정은 점점 더 부정확해지고 있다. 이 목적을 달성하기 위해서는 비선형 방법을 사용해야만 한다. 따라서, 본 논문은 소프트웨어 개발노력을 추정하는데 비선형 관계를 표현 가능한 RBF망 모델을 제안한다. 24개 소프트웨어 개발 프로젝트 사례연구를 통해 적합한 RBF 망 모델을 제시하였다. 또한, 회귀분석 모델과 RBF 망 모델을 비교하여 RBF 망 모델의 정확성이 가장 좋음을 보였다.

## Improving Estimative Capability of Software Development Effort using Radial Basis Function Network

Sang-Un Lee<sup>†</sup> · Young-Mok Park<sup>\*\*</sup> · Jae-Heung Park<sup>\*\*\*</sup>

## ABSTRACT

An increasingly important facet of software development is the ability to estimate the associated cost and effort of development early in the development life cycle. In spite of the most generally used procedures for estimation of the software development effort and cost were linear regression analysis. As a result of the software complexity and various development environments, the software effort and cost estimates that are grossly inaccurate. The application of nonlinear methods hold the greatest promise for achieving this objective. Therefore, this paper presents an RBF (radial basis function) network model that is able to represent the nonlinear relation for software development effort. The research describes appropriate RBF network modeling in the context of a case study for 24 software development projects. Also, this paper compared the RBF network model with a regression analysis model. The RBF network model is the most accuracy of all.

키워드 : RBF 망(Radial Basis Function Network), 기저함수(Basis Function), 개발노력(Development Effort), 선형회귀분석(Linear Regression Analysis)

### 1. 서 론

소프트웨어 개발시 중요하게 제기되는 문제점으로 개발 생명주기의 초기단계에서 개발과 관련된 노력과 비용을 추정하는 능력이다. 소프트웨어 개발노력 추정 분야는 과거 30년 동안 활발한 연구가 수행되었으며, 새롭고 보다 나은 모델 개발이 되었다. 그럼에도 불구하고, 대부분의 모델이 공통적으로 문제점은 소프트웨어 개발 생명주기의 초기단계에서 모델의 입력으로 사용되는 프로젝트의 속성들이 추정 시점에서 타당성 있는 확실성을 갖고 있지 못하기 때문에 발생하는 제한된 유용성이다. 제안된 모델 대부분은 개발요원의 노력, 사업기간과 개발비용을 추정하기 위해 시스템 속성들과 개발과정 환경요소를 고려하였다. 가장 일반적으로 사용된 모델은 독립변수의 실질적인 개수를 사용한

다중 선형 회귀분석 (Multiple Linear Regression Analysis) 이다. Morin[1]은 소프트웨어 비용추정을 위한 기존의 많은 모델들을 연구하였으나 개발노력을 예측하는데 정확하고 신뢰할 수 있는 방법을 발견하는데 실패하였다. 또한, 개발 노력 추정은 선형 인과관계 (Cause-Effect)의 단순한 문제가 아니며 다변량 간에 관계를 갖고 있어 복잡한 함수로 데이터 보간법의 비선형 모델을 적용해야 함을 추천하였다. Putnam[2]도 추정분야의 가장 중요한 목적은 정확한 추정식을 구하는 것이며, 이 목적을 생성하는데 비선형 방법 적용이 가장 타당해 보임을 제안하였다.

비선형 모델 사용은 개발노력 추정을 단순하고 쉽게 적용할 수 있는 공식을 생성하지는 못하지만 주어진 함수의 복잡한 비선형적 특징을 잘 표현할 수 있는 장점이 있다. 이상운[3]은 비선형 함수를 근사시키는 능력을 가진 가장 일반적으로 사용되고 있는 신경망의 일종인 FFN(FeedForward Network)을 소프트웨어 개발노력 추정 모델로 적용할 수 있음을 연구하였다. 그러나 FFN의 경우, 모델의 구조를 결

† 정 회 원 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

\*\* 준 회 원 : 경상대학교 대학원 컴퓨터학과

\*\*\* 정 회 원 : 경상대학교 컴퓨터학과 교수

논문접수 : 2001년 4월 16일, 심사완료 : 2001년 7월 23일

정하는 것으로 최적의 은닉 뉴런 수를 찾는 문제가 제기된다. FFN의 경우 1개의 은닉층에 시그모이드 작동함수를 가진 뉴런을 무한히 갖고 있으면 어떠한 함수도 표현할 수 있다는 보편적 근사 이론이 증명되었다[4, 5]. 그러나, 최적의 은닉 뉴런 수는 주어진 문제의 복잡도에 따라 결정되는 단점이 있다. 이에 반해, RBF(Radial Basis Function) 망의 최대 은닉 뉴런 수 (기저함수의 개수)는 최대 입력 데이터 개수가 되며, FFN에 비해 수렴 속도가 빠르고, 작은 오차와 높은 신뢰성을 가지는 장점이 있다[6]. 따라서, 본 논문은 주어진 문제의 복잡한 비선형 관계를 표현할 수 있는 능력을 가진 RBF 망을 이용해 소프트웨어 개발노력 추정 분야에 적용하는 연구를 수행하고자 한다.

2장에서는 소프트웨어 개발노력과 비용추정에 관한 관련 연구와 문제점을 기술하고, 3장에서는 비선형 관계를 표현할 수 있는 RBF 망을 살펴본 후 소프트웨어 개발노력 추정 분야에 적용될 수 있는 RBF 망을 설계한다. 4장에서는 설계된 RBF 망을 이용해 소프트웨어 개발노력을 추정할 결과를 기존 모델들과 비교 분석하여 제안된 RBF 망 모델의 성능이 보다 좋음을 제시한다.

2. 관련 연구 및 문제점

기존의 소프트웨어 개발노력과 비용을 추정하기 위한 모델들은 소프트웨어의 정보영역과 복잡도에 대한 주관적인 평가의 계수적 측정을 거치는 실험적 관계를 통해 얻어지는 알고리즘적 기법으로 선형회귀 또는 다중회귀분석을 이용하였다. 이들 모델들은 개발노력이 LOC(Line of Code)[7, 8], FP(Function Point)[9, 10] 또는 기능구성요소 형태 (Function Element Type)들에 대해 선형이나 비선형 형태를 취함을 연구하였다[11]. FP 척도는 소프트웨어 프로젝트 추정, 생산성과 품질 평가에서 일관성과 객관성을 제공하는 구체적으로 공인되고 표준화된 척도로 프로젝트 계획 중에 소요 비용과 일정을 추정하는데 유용하게 사용되며, LOC를 이용한 COCOMO 모델[7, 8]에 비해 소프트웨어의 개발 초기 단계인 요구분석 단계에서 측정이 가능한 장점을 갖고 있다.

FP를 이용한 개발노력 추정에 관한 연구로서, Albrecht et al.[10]은 <표 1>의 데이터에 대해 개발노력 E가 FP에 대해 식 (1)의 선형형태를 취함을 연구하였다. Kemerer[12]는 ABC 회사에서 개발된 15개의 프로젝트에 대해 식 (1)과 유사한 선형모델과 3차항 까지를 고려한 곡선회귀모델을 제안하였다. Albrecht et al.[10]의 데이터에 대해, Matson et al.[11]은 개발노력 E가 FP에 대해 식 (2)의 비선형 형태를 취하는 것이 보다 정확한 추정을 할 수 있음을 보였다. Matson et al.[11]은 다양한 개발언어와 개발업체로부터 수집된 104개의 프로젝트에 대해 식 (1)과 유사한 모델과 식 (3)의 모델을 제안하였다. 또한, FP 보다는 기능 구성요소 형태들을 이용해 직접 개발노력을 추정하는 식 (4)

의 회귀모델을 제시하였다.

$$E = -13.39 + 0.0545 \cdot FP \tag{1}$$

$$\sqrt{E} = 1.000 + 0.00468 \cdot FP \tag{2}$$

$$\ln(E) = 2.51 + 1.00 \ln(FP) \tag{3}$$

$$E = 3.80 + 0.00119 \widehat{IN}^2 + 0.00210 \widehat{OUT}^2 + 0.00608 \widehat{INQ}^2 + 0.0045 \widehat{FILE}^2 \tag{4}$$

<표 1> IBM 개발노력 데이터

Case No.	기능 구성요소 형태				FP	개발 노력 (EFFORT)
	입력 (IN)	출력 (OUT)	조회 (INQ)	파일 (FILE)		
1	15	15	6	3	199	0.5
2	33	17	8	5	224	2.9
3	25	28	4	22	500	3.6
4	7	12	13	8	209	4.1
5	17	17	15	5	289	4.9
6	12	15	0	15	260	6.1
7	28	41	16	11	417	7.5
8	34	14	0	5	205	8.0
9	27	20	24	6	400	8.9
10	13	19	0	23	283	10.0
11	41	27	29	5	512	10.8
12	70	27	0	12	428	11.1
13	61	68	0	11	694	11.8
14	43	40	20	35	682	12.0
15	45	64	14	16	680	12.9
16	28	38	24	9	512	15.8
17	42	57	12	5	606	18.3
18	40	60	20	15	794	19.0
19	40	60	20	12	759	21.1
20	10	69	1	9	431	28.8
21	48	66	13	50	1235	38.1
22	69	112	21	39	1572	61.2
23	25	150	75	60	1750	102.4
24	193	98	70	36	1902	105.2

개발노력 추정 분야의 대부분의 연구는 알고리즘적 노력추정 모델링에 초점을 두고 있다. 이 과정에서 노력은 측정된 출력을 얻기 위한 척도를 가진 입력 또는 노력과 연관된 수학적 공식을 사용해 분석된다. 이 공식은 과거 이력자료 분석으로부터 제시된 형식적인 모델(Formal Model)을 사용하며, 모델의 정확도는 특정 개발환경을 반영하여 가중치 조절을 포함한 모델의 조정(Calibration)으로 향상될 수 있으나, 일반적으로 이 방법은 추정에 매우 큰 불일치성을 갖고 있다.

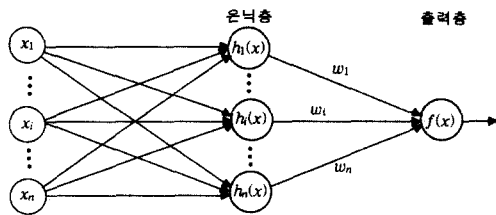
Cybenko[4]는 다층 전향망이 비선형 함수인 시그모이드 작동함수를 가진 1개의 은닉층으로 구성되어 있으면 주어진 모든 함수를 표현할 수 있다는 보편적 근사 이론(Universal Approximation Theorem)을 증명해 보였으며, Barron[5]은 은닉층에 시그모이드를 출력층에 선형 작동함수를 사용해, 1개의 은닉층을 가진 FFN이 함수 근사 능력이 있음을 밝혔다. 이 이론에 근거하여, 이상운[3]은 개발노력 추정에 FFN을 적용할 수 있음을 연구하였다. FFN에 대해 최적의 은닉층 뉴런의 개수를 결정하기 위해 1~30개까지 은닉 뉴런 수를 변경시켜 가면서 훈련 결과 가장 적은 오차를 나타내는 은닉 뉴런수를 가진 FFN 모델 구조를 선택

하고 이 구조가 갖고 있는 은닉 뉴런 수를 최적의 은닉 뉴런수로 결정하는 시행착오법(Trial-and-Error)을 이용하여 좋은 추정 결과를 얻었다. 그러나 최적의 은닉 뉴런 수는 주어진 문제의 복잡도에 따라 달라질 수 있는 문제점이 있다. 따라서, 본 논문은 FFN과 같이 정적으로 은닉 뉴런 수를 결정하는 방법 대신, 주어진 문제에 따라 동적으로 최적의 은닉 뉴런 수를 결정하는 RBF 망[13,14]을 적용해 개발노력을 추정하고자 한다.

### 3. 개발노력 추정 RBF 망 설계

#### 3.1 RBF 망

벡터공간  $V$ 의 벡터집합을  $S$ 라하고,  $S$ 에 있는 모든 벡터가 서로 1차 독립이며,  $V$ 의 모든 벡터는  $S$ 에 있는 모든 벡터의 1차 결합(선형결합)으로 표현될 수 있을 때  $S$ 를  $V$ 의 기저함수(Basis Function)라 한다. 따라서, RBF 망은 주어진 입력들을 분류할 수 있는 군집(Clustering)을 결정하는 기저함수를 찾는 것으로, 기저함수만 있으면 입력 벡터 공간의 모든 점들을 기저함수의 선형결합으로 표현할 수 있다는 개념에서 유래되었다. RBF 망은 (그림 1)의 2개 층(은닉층과 출력층)으로 구성되어 있다. 은닉층 뉴런은 전형적인 다층 신경망의 구조인 시그모이드 작동함수를 갖고 있지만, 입력의 가중합을 취하지 않으며, 은닉 뉴런의 작동함수는 가우시안(Gaussian) 함수 ( $e^{-x^2}$ )와 같은 방사대칭 기저함수(Radially Symmetric Basis Function)를 사용하여 FFN과 다른 특징을 갖고 있다.

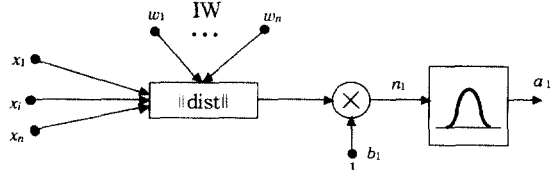


(그림 1) RBF 망

각 은닉층 뉴런의 출력(그림 2)의  $a_1$ 은 기저함수의 중심과 망의 입력간 거리에 의해 결정되는 기저함수로 표현된다. 출력층(그림 3)은 선형 작동함수를 가지며 선형결합으로 은닉층 출력의 가중합을 생성한다. RBF 망에 있는 은닉층 뉴런은 기저함수에 근접된 입력에만 응답하여(그림 4)와 같이 지역적으로 응답하는 영역을 가진다. 이것은 전역 응답을 생성하는 시그모이드 작동함수를 가진 표준 다층신경망인 FFN과 상반되는 역할을 수행한다.

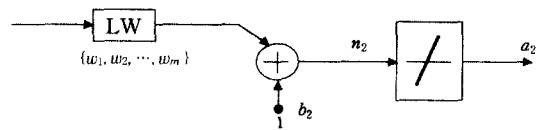
RBF 망은 FFN에 비해 최적의 은닉 뉴런 수(기저함수 개수)를 결정함에 있어 주어진 문제에 적합하도록 은닉 뉴런 수를 0개로부터 시작하여 점진적으로 추가시키면서 주어진 문제에 적합한 뉴런 수를 찾는 방법을 사용할 수 있다. 또한, RBF 망은 전형적인 다층 FFN 보다 빠른 수렴속도, 보

다 적은 외삽법 오차와 높은 신뢰성으로 인해 공학분야에서 점점 더 관심의 대상이 되고 있다.



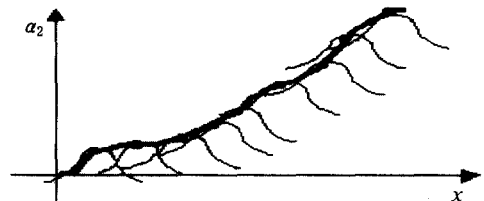
$$a_1 = \text{radbas}(\|IW - X\|b_1) = e^{-n^2} = e^{-(IW - X1b_1)^2}$$

(그림 2) 은닉층 출력



$$a_2 = \text{purelin}(LWa_1 + b_2)$$

(그림 3) 출력층 출력



(그림 4) 지역 응답 RBF 망의 출력

#### 3.2 개발노력 추정 RBF 망 설계

##### 3.2.1 망의 구조 결정

RBF 망은 FFN 설계시 제기되는 몇 개의 은닉층과 은닉층에 몇 개의 은닉 뉴런을 사용할 것인가의 문제는 발생되지 않는다. 단지 1개의 은닉층을 사용하고 은닉층의 뉴런 개수는 최악의 경우 입력 데이터의 최대 개수까지만 되며, 최적의 개수는 망의 훈련과정에서 자동적으로 결정된다[15]. 따라서, (그림 1)의 RBF 망을 사용한다.

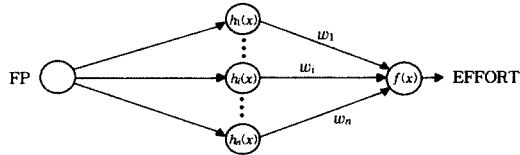
##### 3.2.2 입-출력 선택

RBF 망 설계시 중요한 사항은 망의 입력과 출력을 결정하는 것이다. 이는 어떤 데이터를 입력으로 하여 어떤 데이터를 얻을 것인가에 의존하며, 훈련을 시작하기 이전에 결정해야 한다. 소프트웨어 개발노력(출력)을 추정하기 위한 목적을 달성하기 위해, 소프트웨어 개발노력에 관련된 수집된 변량(Variable) 들을 입력으로 선택할 수 있다. 본 논문에서는 <표 1>의 수집된 데이터로부터, 입력으로 FP 만 이용하는 경우(그림 5)와 기능 구성요소 형태를 이용하는 경우(그림 6)로 신경망의 입-출력을 결정한다.

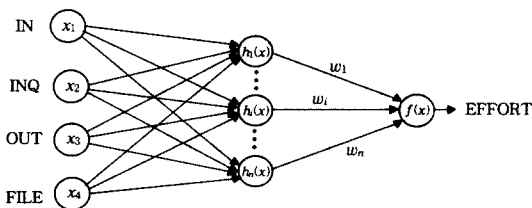
##### 3.2.3 기저함수의 폭 결정

RBF 망을 적용하는데 있어서 다음으로 해결해야 할 사항

은 기저함수의 폭을 결정하는 것이다. 기저함수의 폭은 입력 영역에서 각 기저함수가 표현할 수 있는 영역 (지역 정보 표현 범위)을 결정하는 것으로 입력 벡터간 거리보다 크게 기저함수의 폭을 선택하면 좋은 일반화 능력을 얻을 수 있다. 일반적으로 0.1과 2 사이의 값을 취하고 있으며[12], 본 논문에서는 일반적으로 많이 적용되고 있는 1.0을 취한다.



(그림 5) FP 이용 개발노력 추정 RBF 망



(그림 6) 기능 구성요소 형태 이용 개발노력 추정 RBF 망

3.2.4 기저함수 개수 결정

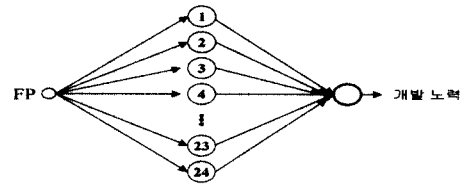
은닉 뉴런(기저함수)의 적절한 개수 선택 문제는 구성 알고리즘을 사용하여 기저함수가 없는 상태에서 망을 초기화시킨 후 훈련과정에서 망의 출력이 설정된 SSE(Sum Square Error) 이하의 결과를 얻거나 기저함수 개수가 입력 데이터의 최대 개수만큼 되는 기준 중 어느 하나를 만족할 때까지 기저함수를 하나씩 추가하면서 훈련을 수행하는 방법[13]을 이용한다.

4. 실험 및 결과 분석

RBF 망의 훈련 종료를 결정하는 기준으로 SSE가 사전에 설정된 값 이하가 되거나 최대 주어진 입력의 개수만큼 기저함수가 되는 경우이다. RBF 망을 훈련시킬 때, 사전에 설정된 SSE 값에 따라 모델의 성능과 기저함수의 개수도 다르게 나타난다. 본 실험에서는 SSE를 0.0, 1.0, 5.0, 10.0, 20.0, 50.0, 100.0과 200.0인 경우를 고려하였다. 본 실험에는 <표 1>의 데이터에 대해 기존의 개발노력 추정 회귀분석 모델인 Albrecht et al.[10], Matson et al.[11]과 이상운[3]의 FFN 모델과 제안된 RBF 망 모델의 성능을 비교, 분석한다.

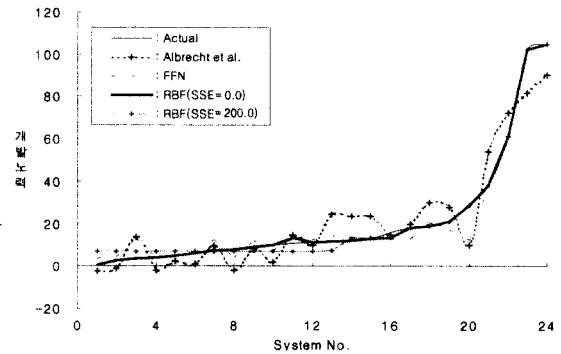
4.1 개발노력 추정 실험

<표 1>의 데이터에 대해 시스템별로 FP를 입력으로 개발노력을 추정하여 보자. FP를 이용해 개발노력을 추정하기 위해 (그림 5)의 RBF 망에 대해 실험하였다. FP를 이용한 개발노력 추정 RBF 망은 훈련 결과 (그림 7)과 같이 얻었다.



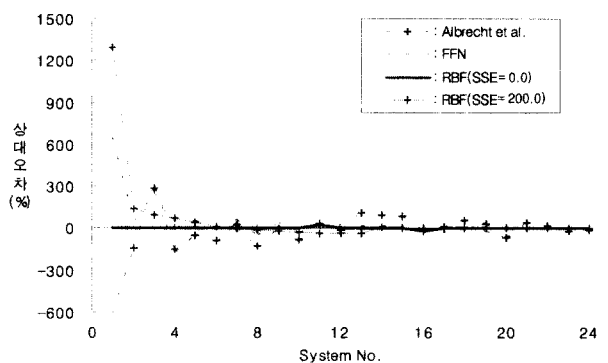
(그림 7) FP 이용 개발노력 추정 RBF 망 모델(SSE=200인 경우)

실험 결과 얻은 개발노력 추정치는 (그림 8)에, 상대오차는 (그림 9)에 제시하였다.



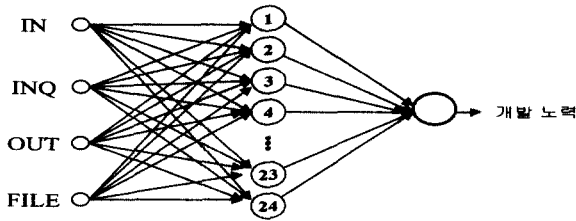
(그림 8) FP 이용 개발노력 추정 결과

(그림 8)에서 개발노력이 시스템별 FP에 대해 비선형 형태를 취한다. 이에 비해 기존의 선형회귀분석 모델들은 이들 관계를 선형적으로 표현하기 때문에 정확한 개발노력을 추정하지 못하였다. FFN은 선형회귀분석 모델들에 비해 비선형을 표현하는 능력을 갖고 있지만, 본 논문이 제안한 RBF 망 모델보다 성능이 저하된 결과를 나타내었다. (그림 8)과 (그림 9)에서 SSE를 0.0으로 설정한 RBF 망 모델이 실제 값에 거의 일치하는 결과를 나타내 최적의 모델로 선택될 수 있다. SSE가 200.0인 경우에도 기존의 FFN이나 회귀분석 모델보다 좋은 성능을 보였다.



(그림 9) FP 이용 개발노력 추정 상대오차

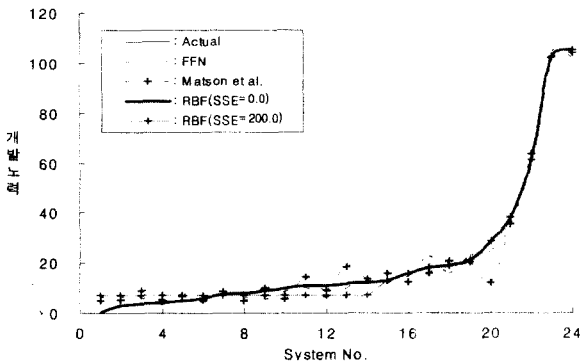
(그림 6)의 RBF 망으로 기능 구성요소 형태를 이용해 개발노력을 추정하기 위해 실험하였다. RBF 망 훈련 결과 얻은 모델은 (그림 10)과 같다. 실험 결과 개발노력 추정 결과는 (그림 11)에, 상대오차는 (그림 12)에 제시하였다.



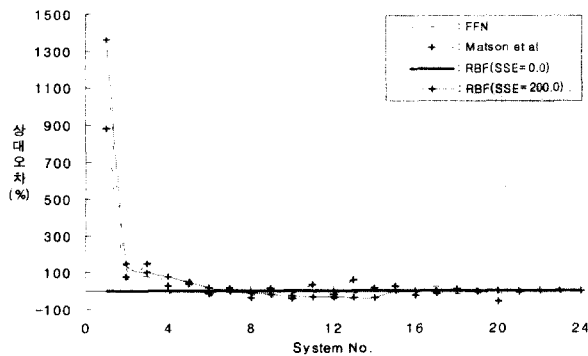
(그림 10) 기능 구성요소 형태 이용 개발노력 추정 RBF 망 모델 (SSE = 200인 경우)

기능 구성요소 형태 4가지를 이용하여 개발노력을 추정하는 경우, 기능 구성요소 형태 상호간에 복잡한 상관관계를 갖고 있어 개발노력 추정이 보다 어려워진다. 또한, 개발노력과 4개의 기능 구성요소 형태간 관계가 복잡한 비선형 형태를 취한다. 따라서, 기존의 다중 선형 회귀분석 모델을 이용할 경우 복잡한 비선형 형태를 구현하기 어려워 모델의 성능이 저하된다. 이에 반해 신경망인 FFN과 RBF 망은 복잡한 비선형 형태를 표현 가능하므로 성능 향상을 얻을 수 있다. (그림 11)에서 신경망 모델들이 보다 정확한 개발노력을 추정할 수 있음을 알 수 있으며, 그 중에서 RBF 망 모델의 성능이 실제 개발노력 데이터에 거의 일치되게 추정하고 있다. (그림 11)과 (그림 12)에서 FP를 이용한 개발노력 추정 결과인 (그림 8)과 동일한 결과로 SSE를 0.0에서 200.0 사이의 어떤 값으로 취하던 간에 기존의 FFN과 회귀분석 모델보다 좋은 성능을 나타내었다.

4.2 모델성능 분석



(그림 11) 기능 구성요소 형태 이용 개발노력 추정 결과



(그림 12) 기능 구성요소 형태 이용 개발노력 추정 상대오차

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 추정 정확도를 평가하는 것이 필요하다. 본 논문에서는 제안된 모델이 주어진 문제를 얼마나 설명할 수 있는가를 표현하는 결정계수(Coefficient of determination)  $R^2$ 와 모델의 출력 값이 원하는 목표 값에 상대적으로 얼마나 오차가 있는지를 알 수 있는 상대오차 척도를 사용하였다.

이들 비교척도를 사용해 모델의 성능을 비교한 결과를 <표 2>에 제시하였다. <표 2>에서 제안된 RBF 망이 FP를 이용한 경우와 기능 구성요소 형태를 이용한 경우 모두 기존 모델보다 성능 ( $R^2$ )이 향상되었으며, SSE를 0.0으로 설정할 경우 최적의 결과로 주어진 문제를 거의 100% 표현하였다. 단지 은닉 뉴런 (기저함수)의 개수는 입력의 개수와 동일하여 FFN에 비해 복잡한 모델임을 알 수 있다. SSE를 200으로 설정할 경우, 두 가지 모델 모두 기저함수의 개수는 10개가 필요하며, FFN 모델 보다 단순하면서도 99% 이상의 정확도를 갖는다.

<표 2> 개발노력 추정 모델 성능

입력	모델	은닉 뉴런수	모델 성능	
			$R^2$	MSE
FP	Albrecht et al.	-	87.40%	97.3219
	Matson et al.	-	89.90%	-
	FFN	6	93.77%	20.7701
	RBF (SSE = 0.0)	24	99.93%	0.5208
	RBF (SSE = 1.0)	24	99.93%	0.5208
	RBF (SSE = 5.0)	24	99.93%	0.5208
	RBF (SSE = 10.0)	24	99.93%	0.5208
	RBF (SSE = 20.0)	20	99.90%	0.7411
	RBF (SSE = 50.0)	16	99.77%	1.8014
	RBF (SSE = 100)	13	99.50%	3.8494
기능 구성요소 형태	RBF (SSE = 200)	10	99.08%	7.1446
	Matson et al.	-	97.50%	25.7000
	FFN	15	98.60%	7.6155
	RBF (SSE = 0.0)	24	100.00%	5.54E-28
	RBF (SSE = 1.0)	23	100.00%	0.0000
	RBF (SSE = 5.0)	22	99.98%	0.1200
	RBF (SSE = 10.0)	20	99.96%	0.3178
	RBF (SSE = 20.0)	18	99.90%	0.7603
	RBF (SSE = 50.0)	16	99.77%	1.7992
	RBF (SSE = 100)	14	99.57%	3.3285
RBF (SSE = 200)	10	99.05%	7.3392	

5. 결론 및 향후 과제

소프트웨어 비용추정 모델 분야에서 소프트웨어 개발노력을 추정하는 모델로 다층 FFN을 적용할 수 있으나 최적의 은닉 뉴런 수를 결정하는 문제점을 갖고 있다. 이 문제점을 해결하는 방안으로 본 논문은 RBF 망을 이용해 은닉 뉴런 수를 훈련을 통해 자동적으로 결정하는 장점과 더불어 개발노력 추정 능력을 최대한으로 향상시킬 수 있음을 보였다.

신규로 개발될 소프트웨어에 대한 개발노력을 추정하고

자 한다면, RBF 망을 이용함으로써, 신경망의 구조를 결정하기 위해 소요되는 노력을 줄이면서도 보다 정확한 개발 노력 데이터를 추정할 수 있을 것이다. 또한 추정된 개발 노력 데이터를 이용하여 개발에 소요되는 비용이나 개발기간 추정에도 제안된 신경망 모델을 활용할 수 있을 것이다.

본 연구에서, 주어진 모든 데이터를 이용한 소프트웨어 개발 노력 추정문제에 RBF 망을 이용시 최상의 성능을 얻었다. 그러나 신경망의 훈련에 사용되지 않는 미지의 데이터에 적합한지 여부를 판단하는 모델의 일반화 능력에 대한 RBF 망의 적합성 연구는 수행되지 못하였다.

또한, 실험에 사용된 프로젝트는 최근의 소프트웨어 개발 환경, 개발 특성 등을 반영하지 않고 있다. 즉, 최근 다양한 개발환경과 언어를 사용한 충분한 프로젝트 규모를 갖고 있는 데이터에 적합한지 여부는 데이터 수집의 어려움으로 검증되지 못하였다. 따라서, 추후 ISBSG Benchmark Release 6[15]의 데이터를 이용하여 이 분야에 대한 연구가 이루어 질 것이다.

**참 고 문 헌**

[1] L. H. Morin, "Estimation of Resources for Computer Programming Projects," M.S. Thesis, Univ. North Carolina, Chapel Hill, NC, 1973.  
 [2] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol.SE-4, No.4, pp.345-361, 1978.  
 [3] 이상운, "신경망을 이용한 소프트웨어 개발 노력 추정", 한국 정보처리학회논문지(D), 제8권 제3호, pp.241-246, 2001.  
 [4] G. Cybenko, "Approximation by Super-positions of A Sigmoidal Function," Mathematics of Control, Signals and Systems, Vol.2, pp.303-314, 1989.  
 [5] A. R. Barron, "Neural Net Approximation," In Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems. New Haven, CT. Yale University, pp.69-72. 1992.  
 [6] H. Demuth and M. Beale, "Matlab Neural Network Toolbox, User's Guide Version 3.0," Math Works, Inc. 1997.  
 [7] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.  
 [8] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol.10, No.1, pp.7-19, 1984.  
 [9] A. J. Albrecht, "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA, pp.83-92, 1979.  
 [10] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Line of Code and Development Effort Prediction : A Software Science Validation," IEEE Trans. on Software Eng., Vol.SE-9, No.6, pp.639-648, 1983.  
 [11] J. E. Matson, B. E. Barrett, and J. M. Mellichamp, "Software Development Cost Estimation Using Function Points,"

IEEE Trans. on Software Eng., Vol.20, No.4, pp.275-287.  
 [12] C. F. Kemerer, "Reliability of Function Point Measurement - A Field Experiment," Communications of ACM, Feb. 1993.  
 [13] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Square Learning for Radial Basis Function Networks," IEEE Trans. on Neural Networks, Vol.2, No.2, pp. 302-309, 1991.  
 [14] D. S. Broomhead and D. Lowe, "Multivariate Functional Interpolation and Adaptive Networks," Complex Systems, Vol.2, pp.321-355, 1988.  
 [15] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.



**이 상 운**

e-mail : sangun\_lee@hanmail.net

1983년~1987년 한국항공대학교 항공전자 공학과(공학사)

1995년~1997년 경상대학교 컴퓨터학과 (석사)

1998년~2001년 경상대학교 컴퓨터학과 (박사)

1992년~현재 국방품질관리소 항공전자 장비및소프트웨어 품질 보증 담당  
 관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성), 소프트웨어 매트릭스, 신경망, 뉴로-퍼지



**박 영 목**

e-mail : ympark@cjcc.chinju.ac.kr

1983년~1989년 경상대학교 전산통계학과 (학사)

1997년~1999년 경상대학교 산업정보공학과 (석사)

2000년~현재 경상대학교 컴퓨터과 학과 (박사과정)

관심분야 : 소프트웨어 공학(소프트웨어 시험 및 소프트웨어 신뢰성), 신경망, 멀티미디어



**박 재 흥**

e-mail : pjh@nongae.gsnu.ac.kr

1973년~1978년 충북대학교 수학과(학사)

1978년~1980년 중앙대학교 전산학과(석사)

1985년~1988년 중앙대학교 전산학과(박사)

1983년~현재 경상대학교 컴퓨터학과 (교수)

관심분야 : 소프트웨어 신뢰성, 시험도구 자동화, 시스템 분석 및 설계, 신경망