

# 사전 정보를 이용한 소프트웨어 개발노력 추정 신경망 구조 결정 (Decision of Neural Network Architecture for Software Development Effort Estimation using Prior Information)

박 석 규\*      유 창 열\*\*      박 영 목\*\*\*  
(Seok-Kyu Park) (Chang-Ryul Ryu) (Young-Mok Park)

## 요 약

소프트웨어 개발에서 점점 더 중요시되는 사항은 개발 생명주기의 초기에 개발과 관련된 노력과 비용을 추정하는 능력이다. 제안된 모델 대부분은 경험 데이터의 직관, 전문가 판단과 회귀분석의 조합에 기반을 두고 있으나 다양한 환경에 적용될 수 있는 하나의 모델을 개발하는 것이 불가능하였다. 본 논문은 기능 구성요소 형태들로 측정된 소프트웨어 규모로 소프트웨어 개발노력을 추정하는 신경망 모델을 제안한다. 신경망의 은닉 뉴런 수는 입-출력 관계로부터 휴리스틱하게 얻는 방법을 제안한다. 24개 소프트웨어 개발 프로젝트 사례연구를 통해 적합한 신경망 모델을 제시하였다. 또한, 회귀분석 모델과 신경망 모델을 비교하여 신경망 모델의 정확성이 보다 좋음을 보였다.

## ABSTRACT

An increasingly important facet of software development is the ability to estimate the associate cost and effort of development early in the development life cycle. Most of the proposed models are based upon a combination of intuition, expert judgement, and regression analysis of empirical data. Overall, the work has failed to produce any single model that can be applied with a reasonable degree of success to a variety of environments. This paper presents a neural network (NN) model that related software development effort to software size measured in function element types. The heuristic approach is applied to decide the number of hidden neurons in NN from the relationship between input-output pairs. The research describes appropriate NN modeling in the context of a case study for 24 software development projects. Also, this paper compared the NN model with a regression analysis model and found the NN model has better accuracy.

키워드(Keywords) : 신경망(Neural Network), 은닉 뉴런(Number of Hidden Neuron),  
소프트웨어 개발노력 (Software Development Effort), 사전 정보(Prior Information),  
신경망 구조(Neural Network Architecture)

---

\* 정회원 : 강원전문대학 교양전산 전임강사  
\*\* 정회원 : 남해전문대학 컴퓨터응용정보과 조교수  
\*\*\* 정회원 : 경상대학교 컴퓨터과학과 박사과정

논문접수 : 2001. 9. 8.  
심사완료 : 2001. 9. 19.

## 1. 서론

소프트웨어 개발 시 중요하게 제기되는 문제점으로 개발 생명주기의 초기단계에서 개발과 관련된 비용을 추정하는 능력이다. 소프트웨어 측정분야는 30년 이상 수많은 연구가 있어왔으나 소프트웨어 개발 비용에 영향을 미치는 다양한 속성들과 이들간의 관계 불명확으로 아직까지 구체적인 소프트웨어 비용 측정 모델이 없는 실정이다. 만약 소프트웨어 비용-개발노력을 추정하려면 소프트웨어의 규모를 알고 있어야만 한다. 소프트웨어의 규모를 측정하기 위한 소프트웨어 척도로 가장 많이 이용되는 방법이 LOC (Line Of Code)와 FPA (Function Point Analysis)이다. [1,2] LOC 척도를 기초로 한 모델로는 Boehm [3,4]의 COCOMO (COConstructive COst MOdel) 모델이 널리 사용되고 있으며, 소프트웨어 규모 측정 단위로 LOC를 사용할 경우 LOC에 대한 일반적으로 받아들일 수 있는 정확한 정의의 부족, 언어에 종속, 요구분석 또는 설계단계에서 정확한 LOC의 추정이 어렵고 코딩이 종료된 후 정확히 측정될 수 있으며, 규모에 대한 특정한 하나의 관점인 길이로서만 기능성 또는 복잡도 (Complexity)를 결정하는 문제점이 있다. 이의 대안으로 시스템의 기능성으로 소프트웨어 규모를 측정하는 척도로 FPA [5,6]가 제안되었다.

FP는 개발과정 초기에 프로그램의 기능적인 측면에서 소프트웨어 생산성을 측정하기 위한 척도로 IBM에 근무하는 Albrecht가 1979년에 제안하였다. FPA는 사용자에게 양도될 시스템의 기능으로 소프트웨어 시스템의 규모와 복잡도를 정량화하는 방법으로 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적, 개발 생명주기의 초기 단계인 요구분석 단계에서 측정 가능 등 장점이 있으며, LOC를 사용할 때의 주요 문제점을 극복할 수 있는 접근법이다. [2]

기존의 소프트웨어 개발노력 (Development Effort, Man-Month or Man-Year로 측정)을 추정하기 위한 모델들 [2,7-10]은 소프트웨어의 정보영역과 복잡도에 대한 주관적인 평가의 계수적 측정을 통한 실험적 관계를 통해 얻어지는 알고리즘적 기법으로 선형회귀 또는 다중회귀분석을 이용하여, 개발노력이 LOC, FP 또는 기능 구성요소 형태 (Function Element Type)들에 대해 선형이나 비선형 형태를 취함을 연구하였다. 이

모델들은 알고리즘적 기법을 사용함으로써 주관적으로 복잡도 가중치들이 설정되어 계산되고 복잡한 비선형 형태를 적절히 표현하지 못하였다.

본 논문은 자체의 학습 능력을 이용해 주어진 문제의 복잡한 비선형 관계를 표현할 수 있는 능력을 가진 신경망 모델을 이용해 소프트웨어 개발노력 추정분야에 적용하는 연구를 수행하고자 한다. 신경망 모델을 이용시 입력과 출력의 수는 해결하고자 하는 문제에 따라 결정되나 모델의 복잡도 즉, 은닉 뉴런의 적절한 수를 결정하는 문제는 제시된 해법이 없는 실정이다. 따라서, 신경망을 설계하기 위해 주어진 입력-출력 관계로부터 휴리스틱한 방법으로 은닉 뉴런 수를 사전에 결정하는 기법을 제시하고자 한다.

## 2. 신경망의 구조 결정

회귀분석 문제는 모수적 (Parametric)과 비모수적 (Nonparametric)방법으로 구분된다. 기존의 회귀분석 모델은 모수적 모델로 독립변수인 입력과 종속변수인 출력 간 기능적 관계형태를 사전에 알고 있어 결정할 수 있으나 미지의 모수값은 데이터를 통해 추정하는 방법이다. 이에 반해, 신경망 모델은 비모수적 방법으로 추정될 함수의 형태에 대해 사전 지식이 없는 것으로 자유 모수 (Free Parameter)들을 포함한 공식을 사용해 모델링 된다.

신경망을 이용해 주어진 문제를 풀기 위해서는 첫 번째로 이 문제를 해결할 수 있는 모델을 선택하는 것이다. 사실 많은 다른 신경망 형태와 모델이 적용될 수 있지만, 그 중에서도 가장 많이 알려져 있고, 가장 일반적으로 사용되는 것이 FFN (Feed-Forward Network)으로 본 논문은 FFN을 이용하고자 한다.

두 번째로, 신경망의 입력과 출력 뉴런의 개수를 결정하는 문제는 어떤 데이터를 입력으로 하여 어떤 데이터를 얻을 것인가에 따라 결정된다. 소프트웨어 개발노력을 추정 (출력)하기 위해 본 논문은 <표 1>의 개발노력 데이터에서 기능 구성요소 형태인 IN, OUT, INQ, FILE을 입력으로 하여 소프트웨어의 개발 노력 (EFFORT)를 추정하고자 한다.

<표 1> IBM 개발노력 데이터  
 <Table 1> IBM Development Effort Data

Case No.	기능 구성요소 형태				FP	개발 노력 (EFFORT)
	입력 (IN)	출력 (OUT)	조현 (INQ)	화일 (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

세 번째로, 주어진 문제를 표현하기 위한 모델의 적절한 구조 (복잡도)를 선택하는 것이다. 너무 복잡한 모델은 잡음에까지도 적합되어 과적합이 발생할 수 있으며, 너무 단순한 모델은 주어진 문제를 학습할 능력이 없게 된다. 따라서, 주어진 문제를 적절히 표현할 수 있는 모델의 복잡도를 결정하는 것이 중요하다. 본 논문은 1개의 은닉층을 이용하고 은닉층과 출력층 뉴런의 작동함수로 시그모이드 함수와 선형함수를 각각 사용하고자 한다. 이와 같은 모델 구조와 작동함수를 선택한 이유는 다음과 같다.

- (1) Cybenko [11]는 비선형 함수인 시그모이드 작동함수를 가진 1개의 은닉층 뉴런의 수를 충분히 갖고 있으면 모든 함수를 표현할 수 있다는 보편적 근사 이론 (Universal Approximation

Theorem)을 증명해보였으며, Barron [12]은 은닉층에 시그모이드를 출력층에 선형 작동함수를 사용해, 1개의 은닉층을 가진 FFN이 함수 근사 능력이 있음을 밝혔다.

- (2) 출력층 뉴런의 작동함수로 시그모이드 함수를 사용하면 이 함수가 표현할 수 있는 한정된 데이터 범위인 [0, 1]로 주어진 데이터를 정규화시켜 훈련을 수행하고 훈련결과 얻어진 데이터를 원래의 데이터 범위로 역변환 시켜야하는 불편이 따른다. 이에 반해, 출력층 뉴런의 작동함수를 선형으로 사용하면, 이러한 불편을 해소 가능하다.

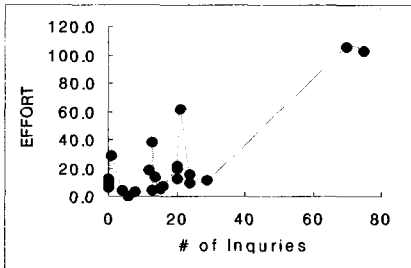
마지막으로, 주어진 문제에 적합한 신경망의 은닉층 뉴런의 수를 결정하는 문제이다. 이는 주어진 문제를 적절히 표현할 수 있는 능력으로 문제에 따라 달라질 수 있어 명확한 법칙이 존재하지 않으며, 단지 시행착오법 (Trial-and-Error)으로 구할 수 있다. 만약 주어진 문제에 대한 사전 지식을 갖고 있다면 사용자가 신경망을 훈련시키기 전에 신경망의 구조를 결정할 수 있을 것이며, 시행착오법을 이용해 적절한 구조를 결정하는데 소요되는 시간을 절약하면서, 주어진 문제에 보다 적합한 망을 추정할 수 있는 장점이 있다. 그러나 현재까지 이 분야에 대한 연구가 이루어지지 않은 실정이다. 따라서, 3장에서는 휴리스틱한 방법으로 신경망을 훈련시키기 전에 은닉 뉴런 수를 결정하는 기법을 제안한다.

### 3. 휴리스틱 은닉 뉴런 수 결정 기법

주어진 함수를 근사 시키는데 필요한 은닉 뉴런 수를 사전정보 (Prior Information)을 이용해 결정하는 방법을 고려해 보자. 본 연구에서는 <표 1>의 다변량 입력, 하나의 변수를 출력하는 문제를 해결하기 위해 FFN (FeedForward Network)에 대해  $i$  개의 입력 ( $i=1,2,3,4$ ),  $j$  개의 은닉 뉴런 ( $j=1,2,\dots$ ), 1 개의 출력 뉴런을 사용한다.

획득된 데이터로부터 FFN의 초기 은닉 뉴런 수를 결정하는 방법 (이를 사전 정보를 활용해 FFN을 구성하기 때문에 piFFN 알고리즘이라 칭한다.)은 다음과 같이 결정된다.

Step 1. 주어진 모든 입력들 각각에 대해 입력 변수 값을 오름차순으로 정렬시켜 출력과의 관계를 그래프를 그린다. [그림 1]



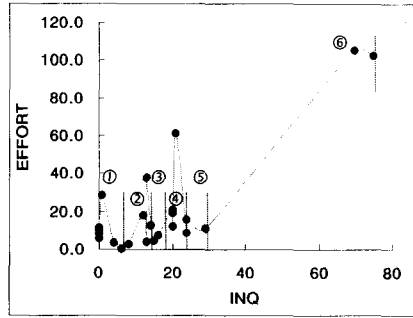
[그림 1] 조희 수와 개발노력 관계  
[Fig. 1] Relationship Between Inquiry and Development Effort

신경망에서 은닉층은 주어진 입력의 복잡한 관계를 표현하는 기능을 수행하며, 다수의 은닉 뉴런의 출력이 선형으로 결합되어 출력층에 전달되어 원하는 출력을 얻는다. 따라서, 은닉층의 뉴런 수는 주어진 문제를 표현할 수 있는 최소한의 개수로 결정될 수 있을 것이다.

Step 2. Step 1에서 각 입력 변수별로 시그모이드 함수 또는 Gaussian 함수로 출력을 표현할 수 있는 수를 휴리스틱하게 결정한다. [그림 2]에서 6개)

Step 3. 각 입-출력변수 관계로부터 결정된 은닉 뉴런 수의 최대 값으로 신경망의 은닉 뉴런수를 선택한다.

Step 4. Step 2에서 결정된 은닉 뉴런 수를 이용해 FFN을 구성한다.



[그림 2] 은닉 뉴런 수 결정  
[Fig. 2] Decision for Number of Hidden Neuron

Step 5. FFN을 구성한 후 신경망을 훈련시켜 주어진 문제에 적합하도록 신경망의 연결강도 값을 결정한다.

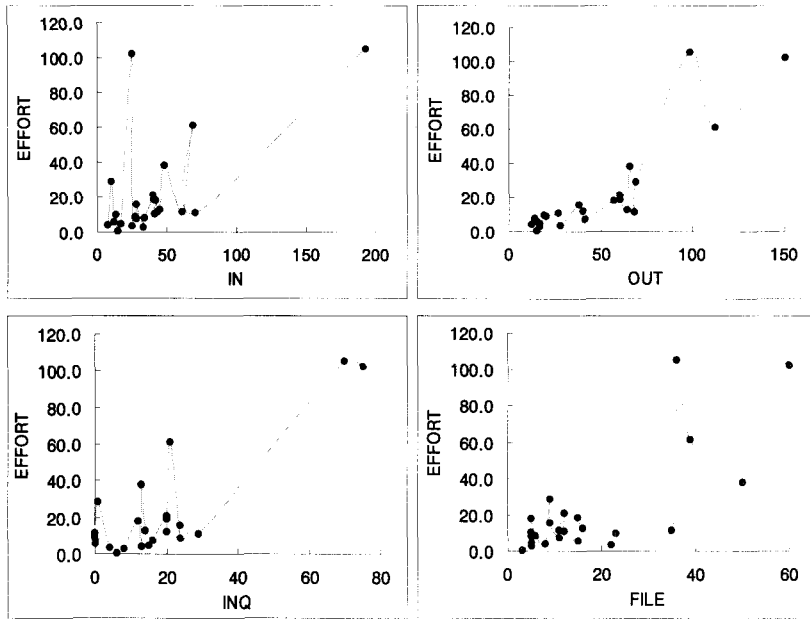
#### 4. 실험 및 결과 분석

Albrecht et al. [6]는 <표 1>의 IBM Data Processing Services에서 개발된 24개 소프트웨어 수집 데이터에 대해, 외부 사용자 입력, 조희, 출력과 화일의 수에 의해 프로젝트의 FP (Function Point) 값을 계산하였으며, FP를 이용하여 개발노력을 추정하였다. 또한, Matson et al. [2]는 기능 구성요소 형태를 이용해 개발노력을 추정하기 위해 <표 1>의 데이터에 대해서  $c_4IN = \widehat{IN}$  과 같이 치환하여 식 (1)의 회귀모형을 제안하고 이에 대해 식 (2)의 모델을 제시하였다.

$$E = \beta_0 + \beta_1 \widehat{IN} + \beta_2 \widehat{OUT} + \beta_3 \widehat{INQ} + \beta_4 \widehat{FILE} + \varepsilon \quad (1)$$

$$E = 3.80 + 0.00119 \widehat{IN}^2 + 0.00210 \widehat{OUT}^2 + 0.00608 \widehat{INQ}^2 + 0.0045 \widehat{FILE}^2 \quad (2)$$

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 추정 정확도를 평가하는 것이 필요하다. 회귀분석의 경우 회귀직선에 의해 종속변수가 설명되어지는 정도를 나타내는 결정계수 (Coefficient of determination)  $R^2$ 이 있다. 종속변수의 값은 독립변수에 의해 결정되어지는 부분과 미지의 오차의 합으로 나타나며, 총 변동을 설명하는데



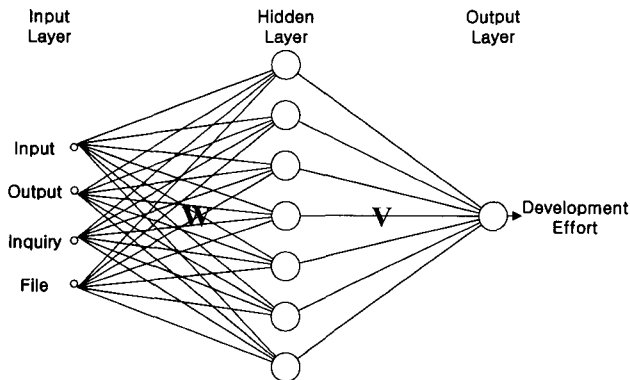
[그림 3] 입-출력 관계

[Fig. 3] Relationship Between Input and Output

있어서 회귀직선에 의해 설명되는 변동이 기여하는 비율이  $R^2$ 이다. 따라서,  $R^2$  ( $0 \leq R^2 \leq 1$ )이 0에 가까우면 추정된 회귀직선은 쓸모가 없으며, 값이 클수록 쓸모 있는 회귀직선이 된다. 본 논문에서는 제안된 모델의 성능을 기존 회귀분석 모델들과 비교 평가하기 위해  $R^2$ 와 상대오차 척도를 사용한다.

#### 4.1 piFFN을 이용한 개발노력 추정

제안된 방법을 적용하기 위해 첫 번째로, <표 1>의 데이터를 이용해 기능 구성요소형태 (IN, OUT, INQ, FILE) 각각의 입력에 대해 원하는 출력인 개발노력간 관계는 [그림 3]의 그래프로 표현된다.



[그림 4] 4-7-1 FFN

[Fig. 4] 4-7-1 FeedForward Network

입력과 개발노력 관계는 약 7개, 출력과 개발노력 관계는 약 7개, 조희와 개발노력 관계는 약 6개, 화일과 개발노력 관계는 약 7개의 은닉 뉴런 수가 선택되었으며, 이중 가장 큰 수 인 7로 은닉 뉴런 수를 선택하였다. 선택된 은닉 뉴런 수, 4개의 입력과 1개의 출력 (개발노력)을 이용해 [그림 4]의 FFN을 설계하였으며, 은닉 뉴런의 작동 함수로는 시그모이드 (Sigmoid), 출력 뉴런의 작동 함수는 선형 (Linear)을 사용하였다.

역전파 (Backpropagation) 알고리즘을 사용해 신경망을 훈련시켰다. 시행착오법을 사용해 여러번의 훈련 결과 은닉층의 입력 가중치  $w(w_{ij}, i = \text{은닉뉴런}, j = \text{입력변수})$ , 은닉층의 출력 가중치  $v(v_{ij}, i = \text{은닉뉴런}, j = 1)$ , 은닉층 뉴런의 바이어스 (Bias)  $b_1(b_{ij}, i = \text{은닉뉴런}, j = 1)$ 과 출력층 뉴런의 Bias  $b_2$ 는 좌적으로 다음과 같이 얻어졌다.

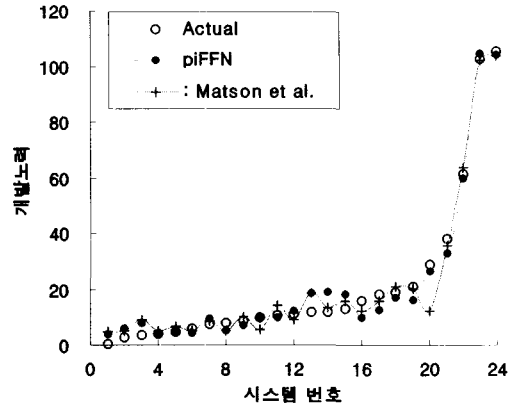
$$W = \begin{bmatrix} -2.0223 & 6.8386 & -0.3858 & 2.8411 \\ +2.4903 & 6.8386 & -0.3858 & 2.8411 \\ -5.6827 & -6.1171 & -4.3074 & -2.9536 \\ 10.0308 & 6.1421 & 10.2208 & 9.7119 \\ -1.6156 & -0.6198 & -1.1431 & -1.2300 \\ 14.1285 & -8.3745 & 5.4207 & -1.0976 \\ 0.0169 & 0.0168 & 0.0146 & 0.0206 \end{bmatrix}$$

$$V = \begin{bmatrix} 1.8385 \\ 8.5340 \\ 10.4537 \\ 1.4112 \\ 1.5346 \\ -17.7297 \\ 111.1601 \end{bmatrix} \quad b_1 = \begin{bmatrix} 11.9098 \\ 5.2398 \\ -10.1534 \\ 3.9037 \\ 6.8659 \\ 1.7813 \\ -4.0186 \end{bmatrix} \quad b_2 = 8.0768$$

### 4.2 실험결과 분석

실험결과 얻어진 개발노력 추정은 [그림 5]에 표기되어 있다. 그림에서 제안된 piFFN 모델이 Matson et al. [2]의 선형회귀 모델보다 복잡한 비선형 형태를 취해 주어진 데이터에 보다 적합함을 알 수 있다.

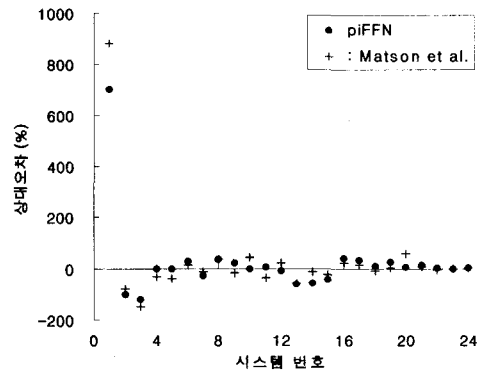
상대오차는 [그림 6]에 표기되어 있으며, 그림에서 전반적으로 제안된 piFFN 모델이 Matson et al. [2]의 회귀분석 모델보다 상대오차가 적음을 알 수 있다.



[그림 5] 개발노력 추정

[Fig. 5] Development Effort Estimation

Matson et al. [2] 모델은  $R^2$ 는 97.5%, MSE는 25.7을 얻은데 반해 사전 정보를 활용해 은닉 뉴런 수를 7개로 사용한 제안된 piFFN 모델은  $R^2$ 는 98.28%, MSE는 13.2774로 상대오차 분석 결과 Matson et al. [2] 모델 보다 좋은 결과를 나타내었다.



[그림 6] 상대오차 분석

[Fig. 6] Analysis of Relative Error

## 5. 결론 및 향후 과제

본 논문은 최근들어 소프트웨어 비용추정 모델 분야에서 소프트웨어 개발노력을 추정하는 모델로 신경망 모델을 적용시 제기되는 신경망의 구조를 사전에 결정할 수 있는 방법을 제시하여 개발노력 추정 능력을 향상시킬 수 있음을 보였다. 기존의 개발노력 추정 회귀분석 모델에 비해 제안된 신경망 모델이 적은 은닉 뉴런 수를 사용하여 종속변수인 개발노력과 독립변수인 기능 구성요소 형태 사이의 복잡한 비선형적 관계를 보다 잘 표현함으로써, 보다 정확한 추정 결과를 얻었다.

신규로 개발될 소프트웨어에 대한 개발노력을 추정하고자 한다면, 기능 구성요소 형태를 계산할 경우, 이들 입-출력 변수 간 관계에 대한 그래프를 그려 사전에 은닉 뉴런 수를 결정하면 신경망의 구조를 결정하기 위해 소요되는 노력을 줄이면서도 보다 정확한 개발노력 데이터를 추정할 수 있을 것이다. 또한 추정된 개발노력 데이터를 이용하여 개발에 소요되는 비용이나 개발기간 추정에도 제안된 신경망 모델을 활용할 수 있을 것이다.

본 연구에는 휴리스틱한 방법으로 신경망의 은닉 뉴런 수를 결정하는 방법을 제안하였으나, 주어진 문제의 복잡도에 따라 은닉 뉴런 수가 달라 질 수 있다. 이에 대한 보다 정확한 정보를 사전에 얻는 방법으로 기저함수를 사용하는 RBF (Radial Basis Function) 신경망을 적용하는 방법이 있을 수 있다. 따라서, 추후 이 분야에 대한 연구가 이루어 질 것이다.

### ※ 참고문헌

- [1] L. A. Laranjeira, "Software Size Estimation of Object-Oriented Systems," IEEE Trans. Software Eng., Vol. 16, pp. 64-71, Jan. 1990.
- [2] 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.
- [3] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.
- [4] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol. 10, No. 1, pp. 7-19, 1984.
- [5] A. J. Albrecht, "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA, pp. 83-92, 1979.
- [6] 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.
- [7] T. Demarco, "Controlling Software Projects : Management Measurement & Estimation," New York : Yourdon Press, 1982.
- [8] A. J. Albrecht, "Measuring Application Development Productivity," in Programming Productivity : Issues for the Eighties, C. Jones, ed. Washington, DC : IEEE Computer Society Press, 1981.
- [9] C. F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," Communication ACM, Vol. 30, No. 5, pp. 416-429, 1987.
- [10] C. F. Keremer, "Reliability of Functional Point Measurement - A Field Experiment," Communications of ACM, Feb. 1993.
- [11] G. Cybenko, "Approximation by Superpositions of A Sigmoidal Function," Mathematics of Control, Signals and Systems, Vol. 2, pp. 303-314, 1989.
- [12] 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.

박 석 규



1988년 4월 ~ 2001년 2월  
진주산업대학교 전산실장  
1990년 3월 ~ 1992년 8월  
경남대학교 컴퓨터공학과  
석사  
2001년 3월 ~ 현재  
강원전문대학 교양전산  
전임강사  
관심분야 :  
시스템 분석 및 설계,  
멀티미디어, 정보검색

유 창 열



1983년 ~ 1987년  
경상대학교 전자계산학과  
학사  
1987년 ~ 1994년 경상대학교  
대학원 전자계산학과 석사  
1994년 ~ 1999년 경상대학교  
대학원 전자계산학과 박사  
1996년 ~ 현재  
남해전문대학  
컴퓨터응용정보과 조교수  
관심분야 : 소프트웨어 공학,  
멀티미디어, 정보통신 등

박 영 목



1983년 ~ 1989년 경상대학교  
전산통계학과 학사  
1997년 ~ 1999년 8월  
경상대학교  
산업정보공학과 석사  
2000년 ~ 현재 경상대학교  
컴퓨터학과 박사과정  
관심분야 : 소프트웨어 공학  
(소프트웨어 시험 및  
소프트웨어 신뢰성), 신경망