

# 유비쿼터스 환경에서 퍼지-신경망을 이용한 장치구성 방안

## An Automatic Device Configuration Using Fuzzy-Neuro on Ubiquitous Environment

노은영, 김두완, 정환목  
대구가톨릭대학교 컴퓨터정보통신공학부

Eun-Young Roh, Doo-wan kim, Hwan-Mook Chung

Faculty of Computer and Information Communication Engineering  
Catholic University of Daegu

E-mail : eyroh@cu.ac.kr, dooywan@cu.ac.kr, hmchung@cu.ac.kr

### 요 약

최근 컴퓨터의 소형화와 인터넷의 대중화로 유비쿼터스 컴퓨팅이 많은 주목을 받고 있다. 유비쿼터스 환경에서 사용자는 언제 어디서나 원하는 정보 서비스를 제한 없이 받을 수 있다. 인터넷의 대중화로 유비쿼터스 시대가 도래 하면서 유비쿼터스 환경을 어떻게 제시할 것인지와 어떤 서비스와 이용 방법을 사용자에게 제공할 것인지가 중요해 지고 있다.

본 논문에서는 유비쿼터스 환경에서 사용자의 목적에 가장 적합한 장치가 자동으로 선택되는 방안을 제안한다. 일반 사용자들가 작업에 필요한 장치를 간단한 언어항으로 입력하면 퍼지 신경망을 이용하여 최적의 장치를 선택하고 IP 어드레스를 이용하여 장치를 구성한다. 간단한 환경을 구성하여 실험해 보았다.

### 1. 서론

유비쿼터스란 컴퓨터를 사용하고 있다는 의식을 사용자에게 주지 않으면서 실세계에 컴퓨터를 대량으로 배치하고 네트워크화 하여 언제 어디서나 정보서비스를 받을 수 있게 하는 환경이다[1,2].

장치의 소형화와 인터넷의 대중화를 이용하여 유비쿼터스 환경을 어떻게 제시할 것인지와 어떤 서비스와 이용 방법을 사용자에게 제공할 것인지가 중요해 지고 있다. 자택이나 사무실 또는 외출한 곳 등에서의 정보 시스템 이용에 대한 연구가 진행되고 있다[3,4]. 또한 유비쿼터스 환경에서는 장치의 구성이 준비되지 않은 상황에서 사용자의 목적에 적합하도록 근처에 있는 장치들이 자율적으로 구성될 수 있어야 한다. 사용자가 늘 사용하는 장치가 갖춰진 환경에서만 아니라 어떤 장치가 있는지 잘 모르는 장소에서도 장치를 검색하여 사용자가 가지고 있는 장치와 주변에 있는 장치를 자동으로 구성 한다면 사용자의 장치 사용에 대한 부담을 줄일 수 있다[3].

본 논문에서는 장치의 자율적인 구성을 위하여 퍼지 신경망을 이용하여 사용 목적에 가장 적합한 장치를 선택하고, IP어드레스를 이용하여 장치를 구성하였다. 유비쿼터스 환경에서 사용자가 필요한 장치들을 언어 변수로 입력하면 자율적으로 주위의 있는 장치들을 선별하여 구성하도록 만들었다. 먼저 주변 환경으

로부터 입력받은 값들을 기반으로 사용자가 필요한 사항들을 입력하여 규칙의 적합정도를 전처리로 계산하고, 사용자가 입력한 언어항으로 역전파 알고리즘을 이용하여 가장 적합한 장치들을 선별하였다.

### 2. 관련연구

#### 2.1 유비쿼터스 컴퓨팅

유비쿼터스화 된 제품으로는 AV장치의 홈 네트워크 시스템이 대표적이다. 장소의 제약을 받는 단일 오브젝트 장치가 네트워크에 접속됨으로서 장치의 연계를 이루어 시간소비 문제와 장소 이동의 문제를 개선한 것이다. 유비쿼터스 환경을 제시하기 위해서는 사용자의 의사나 행동을 저해하지 않고 시간과 장소에 제한 없이 서비스가 제공되어야 한다.

언제나 어디서나 컴퓨터를 사용하기 위해서는 낮은 장소에서 주변에 있는 장치들을 이용하여 컴퓨터 시스템을 구성하고 이용하는 것이 필요하다. 즉, 미리 준비하지 않고 근처에 있는 장치들만을 이용해서 컴퓨터 시스템을 자율적으로 구성하여 사용자가 원하는 환경을 제공하는 시스템의 연구 개발이 필요하다.

#### 2.2 퍼지 신경망 시스템의 구성

퍼지-신경망은 인간의 지식과 생체의 구조를 모방한 새로운 정보처리 방법으로 퍼지 및 신경망을 이용하여

비선형인 입출력 관계를 용이하게 처리할 수 있다. 퍼지 추론과 신경망 추론은 각각 독립적으로 연구되어 왔으나 1980년대 중반부터 신경망과 퍼지이론의 융합에 대한 연구가 활성화됨에 따라 양 기술의 장점을 접목시킨 새로운 기술의 연구가 활발하게 이루어져 왔으며 이들의 신경망의 학습기능과 비선형성의 특성에 대한 연구가 퍼지 시스템에 반영되어지고 있다.

신경망은 학습 기능과 대량 병렬성을 보유하고 자기 조정(Auto Tuning) 기능을 갖는 유연성이 뛰어난 계산 모델이지만 지식 표현이 구조적으로 분명하지 않고 전문가의 애매한 지식을 반영하기가 힘들며 학습하는데 오랜 시간이 걸리는 단점이 있다. 반면 퍼지 시스템은 지식의 논리적, 구조적 표현을 명확하게 할 수 있고 퍼지 추론 등의 퍼지 연산을 통해 퍼지 정보처리를 할 수 있으며, 복잡한 시스템에 대한 개략적 기술이 가능한 시스템이다. 그러나 학습 능력이 없고 소속 함수와 규칙에 대한 조정 능력이 없어 변화하는 환경에 대한 적응성이 떨어지고 계산 시간이 많이 걸리는 단점이 있다. 퍼지-신경망 시스템은 퍼지 시스템 또는 신경망만을 이용하는 것보다 유용한 시스템이다.

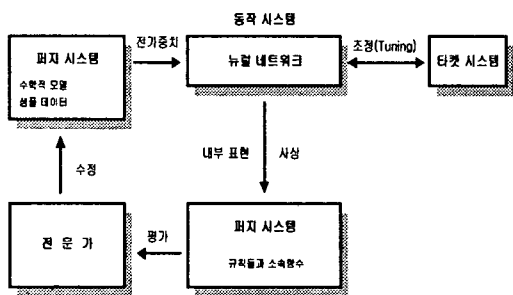


그림 1. 퍼지-신경망 시스템

그림 1은 퍼지-신경망 시스템의 구성으로 퍼지 규칙은 전문가로부터 획득되고, 신경망에서는 퍼지 시스템에서 선택된 학습 데이터를 사용하여 이 데이터를 학습한다[5].

퍼지-신경망 시스템은 각 단계가 은닉층 뉴런들에 의해 실행되는 것을 제외하고는 전통적인 퍼지 시스템의 여러 가지 성분으로 구성되며, 신경망 학습 능력은 지식 시스템을 강화하도록 제공된다.

퍼지 시스템은 인간의 애매한 사고를 컴퓨터로 처리할 수 있도록 하고 컴퓨터 내의 처리를 인간이 이해하기 쉽게 한다.

신경망은 학습기능을 가지며 임의의 비선형 함수의 근사가 가능하다. 신경망에 의해 퍼지 시스템을 실현하는 최대의 장점은 퍼지 규칙을 자동 획득할 수 있다는 것이다.

퍼지-신경망의 연구로서는 주식데이터나 상업 데이터 해석의 응용, 음성인식, 화상인식, 센서로의 응용, 로봇, 화상이해로의 응용 등이 있다.

### 3. 퍼지-신경망을 이용한 장치 구성 과정

#### 3.1 유비쿼터스 환경에서의 장치 구성

먼저 사용자가 작업하려고 하는 상황에 가장 적합한 장치들을 설정하기 위하여 IP 어드레스를 통해 주

변 장치들을 인식한다. 사용자의 환경에 적합한 장치들을 구성하기 위하여 필요한 요구사항들에 대한 응답을 받는다. 응답된 항목들을 퍼지화하여 각 항목에 대한 적합정도를 전처리 단계에서 계산하고 이 적합정도에 따라 결론부를 합성하여 가장 적합한 시스템을 결정한다.

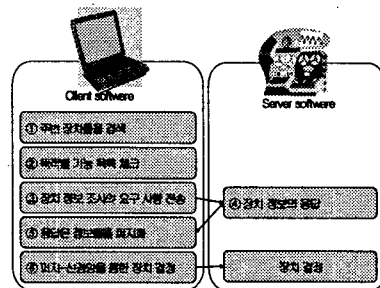


그림 2. 전체 시스템 구성

#### 3.2 퍼지-신경망 시스템의 장치 구성

퍼지-신경망 시스템은 전처리 단계와 신경망으로 구성한다. 전처리 단계에서는 퍼지 시스템을 처리하기 위해 입력과 조건부의 적합정도를 계산한다. 신경망은 전처리 단계에서의 적합정도에 따라 결론부를 합성하여 비퍼지화하고 학습을 한 후 사용목적에 가장 적합한 주변 시스템을 결정한다.

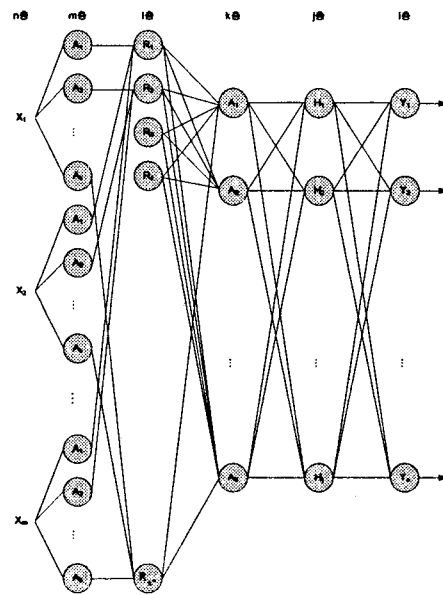


그림 3. 퍼지-신경망 구조

$n$ 층에서 각 노드는 하나의 입력변수에 해당하고, 이 입력변수에 대하여 정의된 언어항을 나타내는  $m$ 층의 노드들과 연결된다.  $l$ 층의 노드는 각 퍼지 규칙의 조건부에 해당하고,  $m$ 층에서 계산된 적합정도를  $\min$  연산을 통해 합성하는 역할을 한다.  $k$ 층에는 출력변수에 정의된 언어항의 개수만큼 노드가 있고, 노드는 각 언어항에 대응하고 이 언어항을 결론부에 가지고 있는 퍼지 규칙들의 적합정도를  $\max$  연산을 통해 합성한다. 이것을 신경망의 입력 데이터로 사용하고, 신경망의 학습 방법으로 오차역전파 알고리즘을 사용한다.

### 3.3 소속함수 생성

전체집합  $U$  에 퍼지집합을 정의하기 위하여 임의의 언어적 변수를 결정한다.

$A_1 = \text{Very Low}, A_2 = \text{Low}, \dots, A_k = \text{Very High}$  를 가능 변수라 하면 이 때 정의된 퍼지집합의 수에는 제한이 없다. 전체 퍼지집합은 가능한 언어변수에 의해 레벨화 되고  $A_1, A_2, \dots, A_k$  를 언어 변수의 언어값인 퍼지 집합으로 둔다. 전체 집합  $U$  에서 퍼지집합  $A_1, A_2, \dots, A_k$  를 다음과 같이 나타낸다.

$$\begin{aligned} A_1 &= \{u_1/1.0, u_2/0.5, u_3/0.0, u_4/0.0, \dots, u_k/0.0\} \\ A_2 &= \{u_1/0.5, u_2/1.0, u_3/0.5, u_4/0.0, \dots, u_k/0.0\} \\ A_3 &= \{u_1/0.0, u_2/0.5, u_3/1.0, u_4/0.5, \dots, u_k/0.0\} \\ &\vdots \\ A_k &= \{u_1/0.0, u_2/0.0, u_3/0.0, \dots, u_{k-1}/0.5, u_k/1.0\} \end{aligned} \quad (1)$$

여기서,  $u_i (i = 1, 2, 3, \dots, k-1, k)$  는 원소이고,  $'/'$ 의 아래의 수는  $u_i$  의 소속도이다.

### 3.4 전처리 과정

퍼지 신경망 시스템은 입력 변수  $X_i$ 가  $m$ 개이고, 출력 변수  $Y_j$ 가  $n$ 개인  $l$ 개의 퍼지 규칙  $R_k$ 로 구성된 다음과 같은 퍼지 시스템에 대한 퍼지 신경망이다. 전처리 단계에서 계산된 규칙의 조건부와 입력과의 적합 정도를 계산한다.

- $R_1 : \text{if } X_1 \text{ is } A_i \text{ and } \dots \text{ and } X_m \text{ is } A_i \text{ then } Y_1 \text{ is } C_i \text{ and } \dots \text{ and } Y_n \text{ is } C_i$
- $R_2 : \text{if } X_1 \text{ is } A_i \text{ and } \dots \text{ and } X_m \text{ is } A_i \text{ then } Y_1 \text{ is } C_i \text{ and } \dots \text{ and } Y_n \text{ is } C_i$
- $\vdots$
- $R_l : \text{if } X_1 \text{ is } A_i \text{ and } \dots \text{ and } X_m \text{ is } A_i \text{ then } Y_1 \text{ is } C_i \text{ and } \dots \text{ and } Y_n \text{ is } C_i$

퍼지 규칙의 조건부에 해당하는 정보는 전처리 단계에서 계산하고 결론부에 대한 정보는 신경망 단계에서 구현한다.

$l$ 층의 뉴런들은 각 퍼지 규칙의 조건부를 나타내고, 이 뉴런들은 중간층에서 계산한 입력과 규칙의 언어항간의 적합 정도를 min 연산에 의해 결합한다. 그리고 이들 뉴런의 출력값은 신경망의 입력값으로 전달된다.  $k$ 층에는 출력변수에 정의된 언어항의 개수만큼 노드가 있고, 노드는 각 언어항에 대응하고 이 언어항을 결론부에 가지고 있는 퍼지 규칙들의 적합 정도를 max 연산을 통해 합성한다.

### 3.5 신경망 시스템

퍼지 규칙의 학습에서는 규칙의 구조 자체를 학습하는 것(전진부의 추가 또는 삭제, 언어항의 대체, 퍼지 규칙 개수 변경)과 소속함수를 학습함으로써 성능을 향상시키는 방법을 채택하고 있다. 각 퍼지 규칙은 후진부에 대해서는 언어항의 형태에 제한이 없지만 전진부에 대해서는 미분 가능한 형태를 갖는 파라미터에 의해 표현될 수 있는 언어항만 오는 형태이다.

그림 3의 퍼지-신경망 시스템에서 입력 데이터를  $X_i (i = 1, 2, \dots)$ 라고 하면, 입력층  $k$ 의 뉴런은 조건부와 입력의 적합 정도가 입력으로 사용되고, 출력층  $i$ 의 뉴런에서는 적합 정도에 따라 결론부가 합성되어 비퍼지화 된 값이 출력된다. 은닉층  $j$ 의 뉴런에서는 은닉층을 나타낸다. 네트워크의 학습에는 오차 역전파 방법을 사용한다. 오차에 대한 성능평가 척도  $E$ 로는 기대되는 값  $t_i$ 와 실제 출력값  $o_i$ 에 차의 제곱합을 사용한다.

$$E = \frac{1}{2} \sum (t_i - o_i)^2 \quad (2)$$

이 때 오차역전파 알고리즘에 따른  $i$ 층과  $j$ 층의 가중치 조정 값  $\Delta W_{ij}$ 는 다음과 같이 계산한다.

$$\begin{aligned} \Delta W_{ij} &= -\eta \frac{\partial E}{\partial W_{ij}} = -\eta \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial W_{ij}} \\ &= \eta (t_i - o_i) f'(\text{net}_i) o_j \end{aligned} \quad (3)$$

$$\frac{\partial E}{\partial o_i} = -(t_i - o_i) \quad (4)$$

$$\frac{\partial o_i}{\partial \text{net}_i} = f'(\text{net}_i) = f(1 - f) \quad (5)$$

$$\frac{\partial \text{net}_i}{\partial W_{ij}} = \frac{\partial \sum W_{ij} o_j}{\partial W_{ij}} = o_j \quad (6)$$

$$\text{net}_i = \sum W_{ij} o_j \quad (7)$$

$$o_i = f(\text{net}_i) \quad (8)$$

$$\epsilon_i = -\frac{\partial E}{\partial \text{net}_i} = -\frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial \text{net}_i} = (t_i - o_i) f'(\text{net}_i) \quad (9)$$

여기서,  $t_i$  는 목표값,  $o_i$  는 실제 출력값,  $\Delta W_{ij}$  는 출력층  $i$ 와 은닉층  $j$ 간의 가중치 조정 값을 나타낸다. 그리고,  $j$ 층과  $k$ 층간의 가중치 조정 값  $\Delta W_{jk}$ 는 다음과 같이 계산한다.

$$\begin{aligned} \Delta W_{jk} &= -\eta \frac{\partial E}{\partial W_{jk}} = -\eta \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial W_{jk}} \\ &= -\eta \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} o_k \\ &= \eta \left( -\frac{\partial E}{\partial o_j} f'(\text{net}_j) o_k \right) = \eta \epsilon_j o_k \end{aligned} \quad (10)$$

$$\frac{\partial E}{\partial o_j} = \sum \frac{\partial E}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial o_j} \quad (11)$$

$$= \sum -\frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial o_j} = \sum_i \epsilon_i W_{ij}$$

$$\epsilon_j = -\frac{\partial E}{\partial \text{net}_j} = -\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \quad (12)$$

$$= \sum_i \epsilon_i W_{ij} f'(\text{net}_j)$$

그림 4에서 이미 학습된  $i, j, k$  층을 통해 평가 결과에서의 오차를 역전파 시켜 입력과 각 규칙의 부합 정도에 대한 오차값을 간접적으로 계산하는 방법을 사용한다. 오차 역전파 알고리즘에 의해  $i, j, k$  층에서 역전파 되는 오차는 다음과 같다.

$i$  층의 뉴런에 역전파 되는 오차

$$\epsilon_i = -\frac{\partial E}{\partial net_i} = (t_i - o_i) f'(net_i) \quad (13)$$

$j$  층의 뉴런에 역전파 되는 오차

$$\epsilon_j = -\frac{\partial E}{\partial net_j} = \sum_i \epsilon_i W_{ij} f'(net_j) \quad (14)$$

#### 4. 퍼지-신경망 시스템에 대한 적용 예

유비쿼스트 환경하에서 인식된 장치들 중 사용자에게 가장 적합한 시스템을 선정하기 위해 각각의 기능 목록에 대한 응답을 받아서 퍼지-신경망 시스템에 적용하여 주변 장치들 중 필요한 장치들을 결정한다. 본 실험에서는 간단히 세 가지 기능 정보에 대해서만 체크하였다. 응답 값은 다섯 가지 언어항으로 설정하여 실험하였다. 입력  $n$  층의 노드 수는 3개로 두었고,  $m$  층을 각 다섯 가지 언어항으로 15개의 노드로 설정하였다. 각 규칙에 따라 1층의 노드 수를 125개로 설정하였고,  $k$  층은 퍼지 언어항에 따라 5개의 노드를 두었다.  $j$  층은 은닉층으로서 오차역전파 학습을 위해 5개의 노드로 설정하였고 출력층은 2개의 노드로 설정하였다. 여기서 학습율은 0.6, 모멘텀은 0.8, 그리고 평균 오차율은 0.1로 설정하였다.

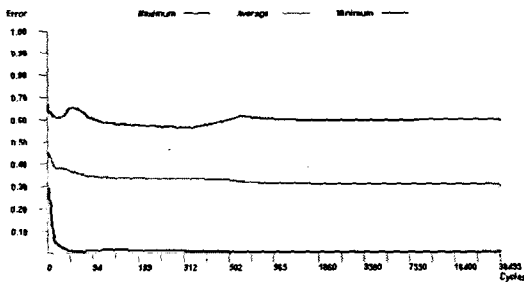


그림 4. 퍼지-신경망의 학습 반복회수

입력값으로 CPU와 그래픽 카드가 좋은 장치들을 사용하고자 하였을 경우 출력값으로 Y2의 장치가 Y1의 장치보다 우수하다는 결과가 나왔다.

Outputs	
Names	Values
Y1	0.443979
Y2	0.990588

그림 5. 결과 값

#### 5. 결론

인터넷의 대중화로 유비쿼스트 시대가 도래하면서 유비쿼터스 환경을 어떻게 제시할 것인지와 어떤 서비스와 이용 방법을 사용자에게 제공할 것인지가 중요해지고 있다.

본 논문에서는 주변의 환경을 사용자가 작업하고자 하는 최적의 환경을 지원하기 위해 현재 작업 환경에 가장 알맞은 장치들을 제공하는 방법을 제안하였다. 일반 사용자들은 필요한 장치의 정확한 값을 이해하기 어려울 수 있기 때문에 사용자가 필요한 사항들을 언어항으로 확인하여 최적의 환경을 설정하도록 하였다. 전처리 단계에서 규칙의 적합정도를 계산하고 이 값을 이용하여 신경망을 통해 학습하여 사용자가 지금 작업하고자 하는 최적의 장치들을 구성할 수 있도록 하였다.

#### 6. 참고문헌

- [1] M.Weiser, "The Computer for the 21th Century", Scientific American 265, No. 3, pp.94-104,1991.
- [2] M.Weiser, "Some Computer Science Issues in Ubiquitous Computing", Communications of the ACM, Vol.36, No.7, pp.74-83,1993.
- [3] M. Minami, K. Sugita, H. Morikawa, and T. Aoyama, "A Design of Internet Application Platform for Ubiquitous Computing environment", IEICE transactions on communications, Vol.J85-B, No.12, p.2313-2330, 2002.
- [4] Faiyaz Doctor, Hani Hagrass, Victor Callaghan, "An Intelligent Fuzzy Agent Approach for Realising Ambient intelligence in Intelligent Inhabited Environments", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2004.
- [5] R. Masuoka, N. Watanabe, A. Kawamura, Y. Owada, K. Asakawa "Neuro Fuzzy System-Fuzzy Inference Using a Structured Neural Network," Proceeding of the International Conference on Fuzzy Logic & Neural Networks, pp. 173~177, 1990.