

논문 2009-46CI-2-9

신경망의 노드 가지치기를 위한 유전 알고리즘

(Genetic Algorithm for Node Pruning of Neural Networks)

허기수*, 오일석**

(Gi-Su Heo and Il-Seok Oh)

요약

신경망의 구조를 최적화하기 위해서는 노드 또는 연결을 잘라내는 가지치기 방법과 노드를 추가해 나가는 구조 증가 방법이 있다. 이 논문은 신경망의 구조 최적화를 위해 가지치기 방법을 사용하며, 최적의 노드 가지치기를 찾기 위해 유전 알고리즘을 사용한다. 기존 연구에서는 입력층과 은닉층의 노드를 따로 최적화 대상으로 삼았다. 우리는 두 층의 노드를 하나의 염색체에 표현하여 동시 최적화를 꾀하였다. 자식은 부모의 가중치를 상속받는다. 학습을 위해서는 기존의 오류 역전파 알고리즘을 사용한다. 실험은 UCI Machine Learning Repository에서 제공한 다양한 데이터를 사용하였다. 실험 결과 신경망 노드 가지치기 비율이 평균 8~25%에서 좋은 성능을 얻을 수 있었다. 또한 다른 가지치기 및 구조 증가 알고리즘과의 교차검증에 대한 t -검정 결과 그들에 비해 우수한 성능을 보였다.

Abstract

In optimizing the neural network structure, there are two methods of the pruning scheme and the constructive scheme. In this paper we use the pruning scheme to optimize neural network structure, and the genetic algorithm to find out its optimum node pruning. In the conventional researches, the input and hidden layers were optimized separately. On the contrary we attempted to optimize the two layers simultaneously by encoding two layers in a chromosome. The offspring networks inherit the weights from the parent. For learning, we used the existing error back-propagation algorithm. In our experiment with various databases from UCI Machine Learning Repository, we could get the optimal performance when the network size was reduced by about 8~25%. As a result of t -test the proposed method was shown better performance, compared with other pruning and construction methods through the cross-validation.

Keywords : 신경망 최적화, 유전알고리즘, 데이터정규화, 노드 가지치기

(Neural Networks Node Optimization, Genetic Algorithm, Data Normalization, Node Pruning)

I. 서론

최근까지 신경망은 분류기를 사용하는 다양한 응용에서 우수한 성능을 제공하고 있다. 신경망의 성능은

신경망 구조 (입력 및 은닉 노드의 수)와 다양한 변수 (가중치 초기 값, 훈련 계수, 모멘텀 등)에 의해 좌우된다^[1].

최적화된 구조와 가중치를 갖는 신경망은 일반화 (generalization) 성능이 뛰어나다. 일반화 성능이 뛰어난 신경망 구조는 일반적으로 경험에 의해 또는 여러 번의 실험에 의해 결정된다. 하지만 이 방법은 많은 시간과 노력을 요구한다. 이런 단점을 극복하기 위해 작은 네트워크 구조에서 출발하여 점점 크기를 증가시켜 가는 구조 증가 (constructive) 알고리즘과 필요한 것보다 큰 네트워크 구조에서 시작하여 점점 크기를 줄여

* 정회원, 전북대학교 컴퓨터정보학과
(Department of Computer and Information Science,
Chonbuk National University)

** 정회원, 전북대학교 전자정보공학부 컴퓨터공학
(Department of Computer Engineering,
Chonbuk National University)

※ 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.
접수일자: 2008년9월5일, 수정완료일: 2009년2월26일

가는 연결소거 또는 노드 가지치기 (pruning) 알고리즘이 사용되고 있다^[2].

본 논문은 신경망의 노드 가지치기를 위한 유전 알고리즘을 제안한다. 신경망의 입력층과 은닉층은 서로 신호를 주고받는 직접적인 관계를 갖는다. 따라서 상호 밀접한 연관 관계를 가질 수밖에 없다. 하지만 기존의 방법들은 입력층과 은닉층의 노드를 따로 취급하여 최적화를 시도하였다^[3-6]. 본 논문에서는 입력층과 은닉층의 노드를 하나의 염색체로 표현하여 동시 최적화를 시도하였다. 입력층에 대한 가지치기는 특징 선택 문제이며, 은닉층에 대한 가지치기는 최적의 은닉 노드 개수를 결정하는 문제와 동일하다. 이 두 가지 문제를 동시에 해결하여 높은 성능의 신경망 구조를 찾는 것과 동시에 유전 알고리즘의 교차와 돌연변이 연산과정에서 생성되는 신경망 가중치는 부모의 가중치를 그대로 상속받는다. Yao는 여러 가지 문헌의 관찰을 통해 신경망 구조와 가중치를 동시에 최적화해야 가장 좋은 성능을 보인다고 하였다^[2].

II장에서는 기존의 신경망 최적화방법과 유전 알고리즘을 이용한 최적화 방법에 대해 살펴본다. III장은 본 논문에서 제안하는 NNPGA(Neural networks Node Pruning with Genetic Algorithm)알고리즘의 전체적인 구조와 더불어 신경망의 염색체 표현, 진화를 위한 연산을 설명한다. IV장은 제안한 알고리즘 평가를 위해 실험에 사용한 데이터 특징을 설명하고 실험 결과를 분석한다. V장에서 결론을 기술한다.

II. 관련 연구

1. 신경망 최적화

신경망 최적화란 일반화 능력이 뛰어난 구조와 가중치를 가진 신경망을 찾는 작업이다. 일반화 성능은 학습 데이터가 고정되어 있을 경우 신경망의 구조에 가장 큰 영향을 받는 것으로 알려져 있다^[2]. 하지만 불행하게도 적합한 네트워크의 구조를 사전에 알 수가 없다.

일반화 성능이 좋은 신경망을 찾기 위해, 보통 신경망의 구조를 여러 가지로 변경하며 성능 실험을 하여 그중 가장 좋은 것을 답으로 취하는 휴리스틱한 방법을 사용한다. 하지만 이 방법은 많은 시간과 노력이 필요하며, 작은 네트워크의 경우 초기 가중치 값에 민감하여 학습이 제대로 이루어 지지 않는 단점이 있다^[7].

이런 문제를 해결하기 위해 구조 증가 (constructive)

알고리즘과 연결 소거 또는 가지치기 (pruning) 알고리즘이 있다^[8]. 구조증가 알고리즘은 작은 네트워크 구조에서 시작하여 점점 크기를 증가 시켜가는 반면, 가지치기 알고리즘은 문제에 필요한 것보다 큰 네트워크 구조에서 시작하여 점점 크기를 줄여가며 최적화된 구조를 찾는다.

신경망 가지치기 방법에는 민감도 방법 (sensitivity method), 정규화 방법 (regularization method) 또는 페널티 방법 (penalty terms), 유전 알고리즘의 진화연산을 이용하는 방법이 있다^[9].

첫째, 민감도 방법은 가중치의 중요도 기준을 정의한 후 훈련 과정에서 민감도가 가장 적은 가중치를 삭제해 나간다. 이 방법으로는 가중치 크기 기반 연결소거, 최적 두뇌 손실 (Optimal Brain Damage)^[10], 최적 두뇌 의사 (Optimal Brain Surgeon)^[11], 영향력 계수 (Impact Factor) 연결 소거법^[7]이 있다.

둘째, 정규화 또는 페널티 방법은 기존의 오차 함수에 벌칙 항을 더한 새로운 오차 함수를 정의하는 것이다^[12-13]. 이렇게 함으로써 신경망이 훈련하는 동안 불필요한 가중치를 0에 가깝게 학습 시킨다. 이것은 전체적인 출력 값의 변화를 완만하게 만들어 일반화 성능을 높이거나 나중에 연결소거를 제거하는데 도움이 된다.

마지막으로 유전 알고리즘은 가중치 및 노드를 염색체로 표현하고 교차 및 변이 연산 과정을 통하여 부모 세대보다 좋은 성능을 가진 자식세대로 진화하는 방식이다.

신경망 최적화를 위한 가지치기 방법은 다양하다. 이 가운데 민감도 방법이나 페널티 방법은 적절한 매개변수 조절과 정규화 매개 변수의 결정에 성능이 좌우되며, 전역적 최적 점을 찾지 못하고 지역적 최적 점에 쉽게 빠질 가능성이 있는 단점이 있다.

2 유전 알고리즘을 이용한 최적화

유전 알고리즘을 이용한 신경망 최적화는 가중치, 구조, 가중치와 구조 최적화로 분류할 수 있다^[2].

가중치 최적화란 미리 정의된 신경망 구조에 대해 가장 좋은 성능을 가진 가중치를 찾는 것이다. 다층 퍼셉트론 신경망을 훈련시키는 대표적인 오류 역전파 (error back-propagation) 알고리즘은 미분강하법 (gradient descent) 을 이용한다. 미분강하법은 지역 최적 점에 빠질 수 있으며 큰 규모의 신경망일 경우 학습시간이 오래 걸리는 단점이 있다. 이런 단점을 극복하기 위하여

전역 최적 점을 찾을 가능성이 높은 유전 알고리즘을 이용하는 방법이 연구되고 있다^[14~15].

구조 최적화란 유전 알고리즘을 이용하여 신경망의 구조 (즉, 은닉 층 개수와 노드의 개수)와 연결 형태 등을 선택하는 작업이다. Miller 등은 소규모 회로망의 최적화에 유전 알고리즘을 제시 하였다^[15]. Arena는 최적 노드의 수를 결정하는 유전 알고리즘을 제안하였으며^[16], 고정된 수의 노드를 최적으로 배치하는 연구도 발표하였다^[17].

가중치 및 구조 최적화는 앞의 두 방법을 결합한 형태로 신경망의 가중치와 구조를 동시에 최적화한다. Pujol의 실험은 가중치와 구조를 동시에 최적화할 때 좋은 성능을 보여 주었다^[18].

신경망과 유전 알고리즘의 결합에 있어 가장 중요한 문제는 신경망의 구조를 어떻게 염색체로 표현하는가 하는 문제이다. 가장 대표적인 방법으로 직접표현 (direct encoding) 과 간접 표현 (indirect encoding) 방법이 있다^[19].

직접 표현 방법은 신경망 노드 및 가중치를 이진스트링 또는 실수를 갖는 염색체로 표현한다. 직접 표현 방식에는 연결 기반 인코딩, 실수 인코딩, 노드 기반 인코딩, 층 기반 인코딩 등 다양한 방법이 있다^[20]. 이 방식은 신경망 구조를 쉽게 표현할 수 있는 장점을 가지고 있다. 하지만 신경망 구조가 복잡해지고 커질수록 염색체 길이가 길어져 연산 시간이 오래 걸리는 단점을 가지고 있다.

간접 표현 방법은 신경망 노드 및 가중치를 직접 표현하지 않고 그 값이 만들어지는 규칙을 정의하고 염색체로 표현한다. 가장 대표적인 방법으로 Kitano의 문법 인코딩 (grammar encoding)^[21]방법과 Gruau의 셀룰러 인코딩 (cellular encoding)^[22]방법이 있다. 간접 표현 방법은 신경망 구조가 복잡하더라도 염색체 길이를 짧게 표현할 수 있는 장점이 있는 반면, 연산과정에서 염색체 유전자를 인코딩 및 디코딩해야 하는 단점이 있다.

III. 신경망 최적화를 위한 유전 알고리즘

그림 1은 본 논문에서 제안하는 NNPGA (Neural networks Node Pruning with Genetic Algorithm)전체적인 알고리즘 이다.

```

P개의 초기 해집단을 생성한다.;
P의 해들의 품질을 평가한다.;
repeat {
    P에서 두 개의 해 p1, p2 선택한다.;
    offspring = crossover(p1, p2);
    // 부모 해로부터 교차와 변이를 통해
    mutation(offspring); // 자식 해를 만든다.
    부모 염색체의 구조로부터 가중치를 상속한다.;
    자식 염색체로 표현된 신경망 구조 훈련.;
    자식 해의 품질을 평가한다.;
    P내의 해중 자식 해와 비교 품질이 좋지 않을 해를
    대체한다.;
} until (종료 조건);
P내에서 가장 우수한 해 C를 선택한다. ;
선택된 C 염색체로 표현된 신경망 구조를 평가한다.;
    
```

그림 1. 제안한 NNPGA알고리즘
Fig. 1. Proposed NNPGA Algorithm.

1. 표현

신경망의 입력층과 은닉층에 있는 노드의 제거 여부를 '0'과 '1'로 표현한다. 그림 2는 가지치기 된 신경망과 해당 염색체를 보여준다. 입력층의 노드 개수를 i , 은닉층의 노드 개수를 h 라 할 때, 염색체는 $g = i + h$ 개의 비트를 갖는 비트열이다.

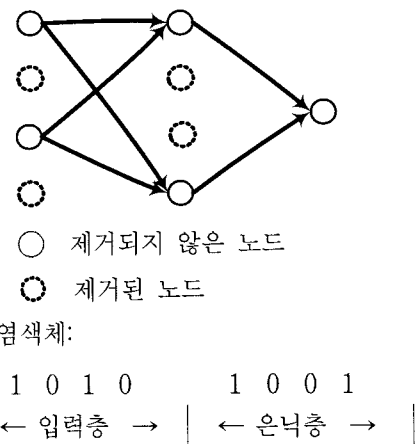


그림 2. 노드 가지치기 된 신경망의 염색체 표현
Fig. 2. Representation of chromosome.

2. 초기 해집단

이들 해집단은 N 개의 염색체를 갖는다. 염색체는 각 비트에 대해 0 또는 1의 임의수로 설정한다. 초기 해집단을 생성할 때 입력노드와 은닉 노드 각각 가지치기 비율을 유지하기 위해 모든 염색체를 1로 표현한 다음 가지치기 개수만큼 0 값을 임의의 위치에서 대체한다.

가지치기 비율을 $\alpha\%$ 라 하면, 각 염색체의 입력 및 은닉노드에 대해 $pi = \alpha \times i/100$, $ph = \alpha \times h/100$ 비트가 0으로 대체된다. P_{ij} 는 i 번째 염색체의 j 번째 비트를 표시한다. $random(x,y)$ 는 $[x,y]$ 사이의 임의의 수를 발생시킨다.

초기 해집단 생성:

```
for(a = 1 to N) {
  for(b = 1 to g) Pab = 1;
  repeat { // 입력 노드 가지치기 비율 유지
    k = random(1,i);
    if(Pak != 0) { Pak = 0; c++;}
  } until(c == pi);
  repeat { // 은닉 노드 가지치기 비율 유지
    k = random(i+1,h);
    if(Pak != 0) { Pak = 0; d++;}
  } until(d == ph);
}
```

3. 적합도 평가 및 선택

해집단의 i 번째 염색체의 품질 p_i 는 (1)에 의하여 평가한다.

$$p_i = \frac{1}{1 + \varepsilon}, i = 1, \dots, N \quad (1)$$

ε 은 i 번째 염색체에 해당하는 신경망 구조에 대해 검증 집합에 대해 범하는 RMSE (Root Mean Square Error) 이다.

$$\varepsilon = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{1}{m} \sum_{j=1}^m (y_j^i - t_j^i)^2} \quad (2)$$

- y_j^i : i 번째 샘플에 대한 j 번째 출력 노드의 출력 값
- t_j^i : i 번째 샘플에 대한 j 번째 출력 노드의 목표 값
- m : 출력층의 노드 개수
- n : 샘플 개수

선택연산은 해집단 내에서 부모 해를 고르기 위한 연산이다. 품질 비례 룰렛 휠, 토너먼트, 순위 기반 등 다양한 선택 연산이 있으나 공통된 원칙은 우수한 해가 선택될 확률이 높아야 한다는 것이다^[23].

본 논문에서는 순위 기반 룰렛 휠 방법을 사용하였다. 해집단 내의 해들을 적합도 평가의 품질에 따라 순위를 매긴 후 (2)의 비선형 적합도 배정 함수에 의해

선택 확률을 배정하였다^[23].

$$f(i) = q(1-q)^{i-1}, i = 1, \dots, N \quad (3)$$

q 값에 따라 선택압을 조절할 수 있으며 값이 클수록 강한 선택압이 부여된다. 아래 예는 룰렛 휠 방법의 가상 코드를 보여 준다. 이 코드를 두 번 수행하여 부모 해 두 개를 선택한다. 이들을 p_1 과 p_2 라 표기하자.

룰렛휠 선택 방법:

```
point = random(0,1);
sum = 0;
for(i = 1 to n) {
  sum = sum + f(i);
  if (point < sum) return i;
}
```

4. 교차와 돌연변이

선택 연산에 의해 선택된 두 개의 부모 염색체는 교차(crossover)와 돌연변이(mutation) 연산을 통해 보다 넓은 탐색 공간을 탐색할 수 있도록 한다^[23].

그림 3은 두 부모해 p_1 과 p_2 로부터 자식해를 얻는 3점 교차 연산의 예를 보여준다.

돌연변이 연산은 교차에 의해 만들어진 자식 염색체의 유전자 각각에 대해 $[0,1]$ 사이의 임의의 수 r 을 생성하고 이것이 주어진 돌연변이 확률(p_m)보다 작으면 0은 1로 1은 0으로 바꾼다. 그림 4는 돌연변이 연산에 의해 1이 0으로 유전자가 변이되는 예이다.

교차와 돌연변이 연산 이후 자식 염색체에는 가지치기 비율에 변화가 생긴다. 가지치기 비율을 유지하기 위하여 돌연변이 연산 이후 0의 개수가 가지치기 비율보다 높은 경우 0을 갖는 유전자 중에 임의로 선택하여

$$\begin{array}{cccccccc} p_1 : & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_2 : & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} \Rightarrow 11001010$$

그림 3. 3점 교차 연산

Fig. 3. An example of 3-point crossover.

변이 유전자

⇓

$$11001010 \Rightarrow 11011010$$

그림 4. 돌연변이

Fig. 4. Mutation.

1로 변경하고, 적은 경우에는 1인 유전자 중에 선택하여 0으로 변경한다.

5. 가중치 상속과 신경망 재훈련

이제 자식 염색체의 구조는 확정되었으므로, 두 부모로부터 가중치를 상속받아야 한다. 자식 염색체의 가중치는 다음 네 가지 경우 중의 하나에 속하게 된다.

- 1) p_1 가중치만 존재하는 경우, p_1 에서 상속 받음.
- 2) p_2 가중치만 존재하는 경우, p_2 에서 상속 받음.
- 3) p_1 과 p_2 의 가중치가 모두 존재하는 경우, p_1 과 p_2 의 가중치의 평균을 상속 받음.
- 4) p_1 과 p_2 가중치 모두 존재하지 않는 경우, $[-1,1]$ 사이의 임의의 수를 상속 받음.

신경망 학습에 사용되는 오류 역전파 알고리즘은 미분강화법의 이용으로 전역해를 찾지 못하고 지역 최적해에 빠지게 되는 단점이 있다. 이를 해결하고자 하는 다양한 방법 중 신경망의 가중치 교란(perturbation)을 이용하는 방법이 있다. 우리는 유전 연산에 의해 세대가 진화되는 과정에서 위의 4가지 방법으로 가중치를 상속함으로써 지역 최적해를 벗어나는 기회를 가지게 된다.

다른 이점으로는 기존 유전 알고리즘을 이용한 신경망 최적화에서는 먼저 유전 알고리즘을 이용하여 최적의 신경망 구조를 결정한다. 이렇게 결정된 구조에 대해 다시 유전 알고리즘을 적용 가중치를 최적화 하는

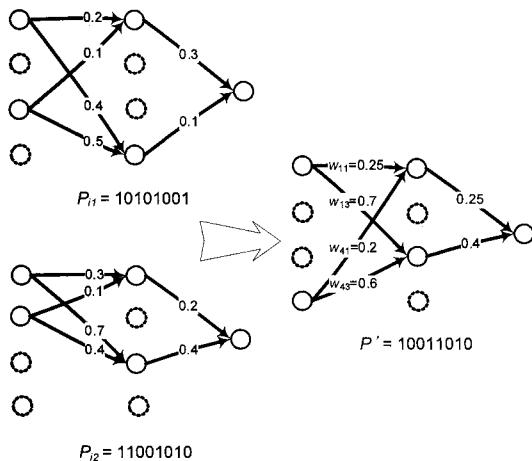


그림 5. 두 부모 염색체로부터 가중치를 상속 받는 자식 염색체

Fig. 5. Weight inheritance.

방식으로 신경망 구조와 가중치 최적화를 시도하였다. 이로 인하여 신경망 구조와 가중치를 결정하는데 많은 시간이 필요하다. 하지만 우리는 신경망 구조 결정과 동시에 가중치 최적화를 피함으로써 이런 단점을 최소화 하였다.

이제 새로운 구조의 자식 신경망이 만들어졌고 가중치도 상속받아 설정되어 있다. 이 신경망은 설정된 가중치를 초기 값으로 간주하고 오류 역전파 알고리즘을 사용하여 재학습을 수행한다.

6. 대치 및 종료

부모 세대로부터 가중치를 상속받아 재훈련된 자식 해는 해집단 내의 해 하나와 대치를 함으로써 좋은 품질을 유지하는 해집단으로 진화 하게 된다.

우리는 안전 상태 유전 알고리즘 사용으로 하나의 해가 생기면 즉시 부모해와 비교를 통해 대치한다. 이 때 자식해가 부모해 중의 하나보다 품질이 좋은 경우에는 부모해와 대치하고 그렇지 않을 경우 해집단 내의 가장 나쁜 해를 대치한다.

유전 알고리즘의 종료 조건은 진화 세대가 주어진 조건(T)에 도달하면 종료한다.

IV. 실험결과 및 토론

1. 실험 환경

(1) 실험 데이터 특징 및 정규화

우리는 제안한 방법의 성능을 비교 평가하기 위해 실세계의 다양한 분류 문제를 제공하는 UCI Machine Learning Repository 데이터를 이용하였다^[24].

표 1은 실험에 사용한 데이터 종류와 각 데이터에 대한 샘플 개수 · 특징 개수 · 부류 개수 등 데이터의 특징을 정리한 표이다. 이 데이터들은 특징에 손실이 없으며 다양한 샘플 개수를 가지고 있다. 특징 개수 및 부류 개수 또한 다양하여 신경망 노드 가지치기 비율에 따른 가지치기된 입력 및 은닉 노드 개수의 다양성을 확보 할 수 있는 특징이 있다.

입력 데이터는 신경망의 입력 노드에 적용하기 전 일정한 범위 값으로 정규화(normalization)한다. 하나의 샘플을 표현하기 위해서는 다양한 특징 결합이 필요하며 이 특징들은 다양한 범위의 값으로 표현된다. 이런 특징 값들은 신경망의 훈련 과정에서 큰 값의 특징 값이 작은 값의 특징 값에 우위를 차지하여 분류기의 성

표 1. 실험 데이터 및 MLP 구조

Table 1. Data sets used for experiment and MLP structure.

Data set	샘플 개수	특징 개수	부류 개수	MLP구조
balance-scale(1)	625	4	3	4-12-3
german(2)	1000	24	2	24-8-2
glass(3)	214	9	7	9-28-7
heart-statlog(4)	214	13	2	13-8-2
ionosphere(5)	351	34	2	34-8-2
iris(6)	150	4	3	4-12-3
pima-indians(7)	768	8	2	8-8-2
segmentation(8)	2310	19	7	19-24-7
sonar(9)	208	60	2	60-8-2
vehicle(10)	846	18	4	18-16-4
Vowel(11)	990	10	11	10-22-11
waveform-noise(12)	5000	40	3	10-12-3
zoo(13)	101	17	7	17-28-7

능을 떨어뜨리는 결과를 초래한다. 이와 같은 특성을 방지하기 위해 특징 값의 분포는 유지하면서 일정한 범위 안으로 특징 값의 크기를 조정하는 정규화 과정이 필요하다. 실험에서는 우리는 입력 데이터를 선형 함수를 이용하여 정규화 하였다.^[25]

(2) 유전 알고리즘에 사용한 매개 변수

유전 알고리즘에 사용한 매개 변수는 다음과 같다.

- 안정 상태 유전 알고리즘 (steady-state GA)
- 해집단 크기 (N): 20
- 교차 확률(p_c): 1.0 (매번 교차가 발생)
- 교배 확률(p_m): 0.01
- 순위기반 선택압(q): 0.25
- 종료조건 (T): 150세대

(3) 신경망 구성·학습 및 성능 평가

실험에 사용한 신경망은 은닉층이 한 개인 다층 퍼셉트론 (multi-layer perceptron)이며, 학습은 오류 역전파 (error back-propagation) 알고리즘을 사용하였다.

신경망 초기 구조는 입력 노드와 출력 노드의 수에 따라 결정된다. 입력 노드는 데이터의 특징 개수로 결정되며, 은닉 노드는 실험에 의해 보통 출력 노드의 3배 이상이 되는 시점부터 평균적으로 성능이 저하됨을 확인할 수 있었다. 이런 이유로 우리는 충분히 큰 신경

망 구조를 위해 은닉 노드의 개수를 출력 노드의 4배로 초기 신경망을 구축하였다. 신경망 학습 종료 조건은 학습회수(epoch) 10회 마다 검증 데이터를 적용, 에러 값이 연속적으로 5회 증가하는 경우 학습을 중단 한다.^[26]

신경망의 훈련 과정에서의 과잉 적응(over-fitting) 현상을 방지하고 제안한 방법을 명확하게 평가하기 위해 실험 데이터를 훈련·검증·테스트 데이터로 나눌 필요가 있다. 이를 위해 10-겹 교차검증 구조(ten-fold Cross-validation schema)를 적용하였다. 교차검증을 위해 실험 데이터를 임의의 동일한 크기 10개의 부분 집합으로 분할한다. 이 중 8개의 부분 집합은 훈련 데이터로 나머지 두 개는 각각 검증 및 테스트 데이터로 분류한다. 신경망 훈련과정에서 훈련 데이터와 검증 데이터를 사용하고 최종적인 노드 가지치기된 신경망 성능은 테스트 데이터를 이용하여 정확률을 평가 한다. 최종적인 성능은 10번의 교차검증 정확률에 대한 평균과 표준편차에 의하여 평가 한다.

최적화된 신경망 구조를 찾기 위한 시간 복잡도는 유전 알고리즘의 세대 진화 과정에서 신경망 훈련 시간과 밀접한 관련이 있다. 우리는 안전 상태 유전 알고리즘의 사용으로 최초의 해집단 성능 평가를 위한 신경망 훈련 횟수(N)과 종료조건 세대 동안 적합도 평가를 위한 신경망 훈련 횟수(T)의 합인 ($N+T$)의 시간 복잡도를 갖는다.

2. 실험결과

본 실험에서 우리는 제안한 방법을 비교 평가하기 위해 입력층, 은닉층, 입력층과 은닉층 각각에 대해 노드 가지치기 비율을 동일하게 유지하며 실험 하였다. 신경망 노드 가지치기를 어느 정도 했을 때 가장 좋은 성능을 보이는지 알아보기 위해 표 2에서와 같이 노드 가지치기 비율을 일정하게 증가시켜가며 성능을 측정 하였다.

예를 들어 Ionosphere dataset의 경우 완전결합 신경망에 대해 노드 가지치기 비율 50%는 34개 입력노드에 대해 17개, 8개 은닉 노드에 대해 4개 가지치기된 신경망 구조의 성능이다. 표 2는 10-겹 교차 검증을 이용하여 완전결합 신경망 구조의 입력 및 은닉 노드(I+H), 입력 노드(I), 은닉 노드(H) 세 가지 각각 4~50% 까지 일정한 범위에서 가지치기된 신경망 구조 성능의 평균과 표준편차를 계산한 결과이다. Ionosphere dataset경

표 2. 노드 가지치기 비율에 따른 성능 비교
Table 2. Accuracy according to the node pruning rate.

(단위 %)

Data set	가지치기 대상	4%	8%	10%	15%	20%	25%	30%	35%	40%	45%	50%
(1)	I+H	91.29±3.55	<u>92.74±4.40</u>	90.97±5.87	91.77±2.55	90.97±3.32	90.48±4.70	90.97±4.74	90.65±3.37	87.42±3.37	90.00±5.34	90.81±3.54
	I	92.74±4.46	<u>92.90±3.69</u>	91.94±3.61	92.26±2.48	92.10±5.02	92.10±3.13	90.00±3.13	90.65±3.04	90.00±3.66	90.81±2.89	90.48±2.55
	H	<u>96.94±1.83</u>	95.65±2.05	96.77±1.25	95.97±2.31	94.52±2.72	95.48±2.26	94.19±3.32	92.74±3.77	93.39±4.70	93.23±4.88	92.26±3.80
(2)	I+H	73.30±4.67	73.60±3.64	72.70±3.23	73.10±4.44	74.40±2.24	<u>74.40±2.24</u>	73.00±3.66	74.10±3.11	73.50±4.36	73.00±3.07	74.00±5.10
	I	72.60±4.61	73.90±2.26	71.70±3.13	74.30±2.83	72.30±3.98	73.90±4.32	73.50±5.50	75.20±3.54	71.00±3.10	<u>74.90±3.67</u>	74.70±2.24
	H	73.20±5.36	73.70±3.74	72.30±2.79	<u>75.00±3.07</u>	73.70±5.53	73.50±4.25	73.60±4.98	74.50±4.88	74.40±5.71	73.80±6.67	72.70±4.52
(3)	I+H	65.24±11.48	<u>70.00±11.2</u>	64.76±14.0	62.38±13.5	64.29±8.58	62.86±12.38	64.29±10.05	60.00±9.57	64.76±11.9	62.38±15.42	59.52±16.39
	I	60.95±10.39	63.81±10.26	62.38±10.31	62.38±9.39	57.14±12.95	60.95±13.93	61.43±10.31	57.62±13.21	<u>64.29±12.09</u>	56.19±13.09	59.05±13.33
	H	62.38±13.38	63.33±11.28	59.52±7.45	63.33±9.05	<u>64.29±8.04</u>	62.38±10.95	60.48±8.26	58.57±13.97	59.05±13.33	61.90±10.65	59.52±12.64
(4)	I+H	81.48±5.74	82.96±6.02	81.85±4.52	82.96±3.39	80.37±6.21	<u>85.19±3.31</u>	80.74±2.77	82.22±6.15	83.33±5.80	84.07±5.51	81.11±5.84
	I	78.89±8.77	80.74±9.19	80.00±8.80	81.48±8.11	80.74±7.55	<u>82.22±7.55</u>	81.85±8.02	80.37±8.45	80.00±8.64	78.89±6.64	80.00±9.40
	H	80.37±5.51	80.37±5.25	81.48±6.41	78.52±4.63	80.00±6.46	81.11±6.07	81.48±5.97	83.33±6.03	82.96±4.44	<u>83.70±6.67</u>	82.96±6.87
(5)	I+H	89.43±4.44	88.57±6.13	89.14±6.98	90.57±4.79	90.00±4.65	89.71±4.08	<u>91.14±3.25</u>	90.57±3.84	90.29±4.46	90.29±4.98	90.57±3.39
	I	91.14±6.57	90.00±3.89	90.57±5.72	89.71±4.28	90.86±3.79	89.14±3.33	88.86±4.32	88.00±9.88	89.14±6.98	<u>90.71±4.13</u>	89.43±6.65
	H	88.86±5.64	87.43±5.14	88.00±5.08	<u>86.86±4.28</u>	87.14±5.61	87.71±4.79	87.14±7.69	87.71±6.77	86.86±6.29	86.86±6.41	87.71±5.27
(6)	I+H	96.67±4.47	94.67±7.18	96.00±4.42	96.67±4.47	<u>97.33±4.42</u>	91.33±9.91	94.67±10.24	95.33±4.27	89.33±9.52	92.00±10.24	91.33±9.33
	I	96.00±3.27	96.00±4.42	<u>97.33±4.27</u>	95.33±4.27	93.33±6.67	96.67±4.47	93.33±11.93	86.67±16.06	92.00±11.08	92.00±8.33	92.67±8.14
	H	97.33±4.42	97.33±4.42	97.33±4.22	96.67±4.47	92.67±11.72	96.67±4.47	96.67±4.47	92.00±13.60	96.67±4.47	<u>98.00±5.27</u>	96.00±4.42
(7)	I+H	<u>77.50±6.40</u>	77.24±7.18	76.84±7.21	77.11±6.42	75.53±6.04	77.11±6.82	76.58±6.93	75.00±7.04	75.39±5.68	73.68±7.58	75.92±5.20
	I	77.63±3.53	75.92±4.16	76.05±4.19	76.45±4.01	76.71±3.86	<u>77.76±2.85</u>	76.97±4.33	75.53±3.18	76.05±3.42	74.08±5.61	76.45±2.91
	H	76.45±4.22	76.58±4.43	76.97±4.09	76.97±4.37	76.58±4.07	76.45±4.26	<u>76.97±4.00</u>	75.79±5.10	76.58±4.59	76.71±4.48	76.45±4.30
(8)	I+H	99.44±0.34	<u>99.57±0.54</u>	99.48±0.47	99.44±0.51	99.44±0.51	99.39±0.401	99.39±0.44	99.48±0.32	99.31±0.48	99.39±0.35	99.35±0.48
	I	99.44±0.48	99.44±0.48	99.39±0.48	99.48±0.47	99.39±0.48	99.44±0.48	99.48±0.48	99.48±0.47	99.38±0.51	<u>99.44±0.48</u>	99.31±0.48
	H	99.39±0.44	99.39±0.44	99.39±0.44	99.44±0.48	99.39±0.44	<u>99.48±0.47</u>	99.38±0.32	99.35±0.40	99.38±0.32	99.38±0.47	99.32±0.47
(9)	I+H	78.50±7.43	81.00±5.83	82.00±6.78	78.50±7.09	80.00±9.49	<u>91.50±7.76</u>	81.00±8.00	79.50±11.06	78.00±9.00	80.00±5.48	81.00±4.90
	I	79.00±8.00	80.00±8.37	78.50±9.23	79.50±9.60	82.50±11.24	79.00±7.00	<u>84.50±9.07</u>	83.00±8.72	82.50±7.50	78.50±12.66	79.50±8.79
	H	83.50±5.94	86.00±7.35	85.50±7.57	84.50±6.50	<u>85.00±7.07</u>	84.00±7.35	81.00±9.43	82.00±10.30	82.00±9.80	83.50±8.08	83.00±6.78
(10)	I+H	81.31±4.16	82.62±5.93	81.00±4.84	<u>85.36±4.33</u>	82.38±4.96	82.38±3.98	83.45±3.30	80.71±4.29	79.88±4.72	79.57±5.63	78.69±3.71
	I	<u>86.07±3.61</u>	85.24±3.07	84.88±4.02	84.64±4.14	83.45±4.00	83.57±3.83	81.07±3.78	83.10±5.13	81.67±2.73	79.88±5.82	80.95±4.39
	H	84.17±3.20	85.83±3.34	85.95±2.31	84.52±2.87	<u>86.07±3.37</u>	85.71±2.82	84.64±2.94	84.40±2.84	85.12±3.38	84.52±2.06	84.42±3.06
(11)	I+H	<u>89.09±4.04</u>	87.88±4.38	87.58±4.56	86.77±1.72	85.05±2.92	81.72±3.45	81.73±4.77	76.67±4.81	78.79±3.10	70.10±4.59	77.27±3.73
	I	<u>90.20±2.12</u>	88.79±2.61	89.49±2.93	88.89±2.02	88.48±3.76	85.05±3.31	86.77±3.87	84.24±3.39	82.83±4.69	81.01±3.16	80.30±4.55
	H	<u>90.71±2.70</u>	89.29±3.10	89.39±3.48	89.29±3.00	90.00±3.76	87.78±3.18	87.47±3.17	84.85±4.24	82.93±6.43	85.05±3.93	80.30±3.07
(12)	I+H	83.88±1.20	83.82±1.87	84.40±1.50	83.34±1.74	84.66±1.76	84.78±2.17	84.32±1.65	<u>84.88±1.64</u>	84.54±1.58	83.96±1.31	83.20±1.34
	I	84.00±2.03	<u>84.92±1.84</u>	84.40±1.71	84.48±2.35	84.50±2.26	84.38±2.70	84.02±2.77	83.06±2.14	82.88±3.26	83.16±1.63	82.04±2.84
	H	84.24±1.35	84.00±1.07	84.02±1.82	84.62±1.53	83.82±0.79	84.14±1.17	83.96±1.63	84.02±1.47	84.74±1.53	<u>84.82±1.80</u>	83.02±1.63
(13)	I+H	95.00±6.71	96.00±6.63	94.00±8.00	<u>97.00±6.40</u>	95.00±6.71	94.00±8.00	96.00±6.63	94.00±9.17	92.00±8.72	94.00±10.20	92.00±8.72
	I	96.00±4.90	97.00±4.58	97.00±4.58	<u>97.00±4.58</u>	95.00±6.71	94.00±4.90	94.00±6.63	96.00±4.90	96.00±6.63	95.00±6.71	94.00±4.90
	H	96.00±6.63	96.00±6.63	96.00±6.63	97.00±4.58	<u>97.00±4.58</u>	96.00±6.63	96.00±6.63	96.00±6.63	96.00±6.63	94.00±8.00	95.00±6.71

* I(Input Layer): 입력 노드, H(Hidden Layer): 은닉 노드, I+H(Input+Hidden Layer): 입력+은닉 노드

우 입력 및 은닉노드를 동시에 가지치기 할 경우 30%, 즉 완전결합 신경망 구조에 대해 입력노드 11개, 은닉

노드 3개 가지치기 될 때 가장 좋은 성능을 보였다. 입력 노드에 대해서는 45%(16개), 은닉 노드에 대해서는

표 3. 제안한 방법과 다른 방법들과의 성능 비교

Table 3. Comparison of the proposed NNPGA and others four algorithms.

(단위 %)

Data set	NNPGA	NNPCv	C5.0	M5'	N2C2S
balance-scale	92.74±4.40	95.41±0.90(○)	77.60±1.00(●)	86.40±0.70(●)	92.30±0.80(○)
german	74.40±2.24	73.10±1.20(○)	71.20±1.00(●)	72.90±0.70(○)	70.10±1.60(●)
glass	70.00±11.28	64.90±2.30(●)	67.50±2.60(○)	70.50±2.80(○)	65.50±3.00(●)
heart-statlog	85.19±3.31	78.15±2.80(●)	78.70±1.40(●)	82.20±1.00(●)	77.50±1.00(●)
ionosphere	91.14±3.25	90.58±1.40(○)	88.90±1.60(●)	89.70±1.20(○)	89.50±2.20(○)
iris	97.33±4.42	96.67±1.00(○)	94.50±0.70(●)	94.70±0.70(●)	96.60±0.60(○)
pima	77.50±3.40	75.49±1.10(●)	74.50±1.20(●)	76.20±0.80(○)	76.00±0.80(○)
segment	99.57±0.54	96.97±0.30(●)	96.80±0.20(●)	97.00±0.20(●)	95.10±0.40(●)
sonar	82.00±6.78	78.65±2.30(●)	74.70±2.80(●)	78.50±3.40(●)	76.50±1.50(●)
vehicle	85.36±4.33	77.78±1.20(●)	72.90±1.20(●)	76.50±1.30(●)	84.20±0.70(○)
vowel	89.09±4.04	86.07±1.10(●)	79.80±1.30(●)	81.70±1.10(●)	88.80±1.40(○)
waveform-noise	84.88±1.64	83.20±0.80(●)	75.40±0.50(●)	82.00±0.20(●)	85.60±0.20(○)
zoo	97.00±6.40	94.00±2.50(●)	91.80±1.10(●)	92.10±1.30(●)	94.30±1.40(●)

* NNPCv, C5.0, M5', N2C2S 데이터는 [27-30] 참고 자료

15%(2)가 가지치기 될 때 가장 좋은 성능을 보였다.

표 2는 UCI Repository의 13개 실험 데이터에 대해 입력과 은닉 노드를 동시에 가지치기 할 경우와 입력 및 은닉 노드 각각을 가지치기 할 경우의 성능을 비교하였다. 표에서 볼 수 있듯 동시에 가지치기 방법이 각각 가지치기 하는 방법과 비교하였을 경우 6개 데이터에 대해 우수한 성능을 보였다. 또한 나머지 데이터에 대해서도 제거된 노드 수에 비해 성능 면에서 많이 뒤떨어지지 않음을 알 수 있었다. 이는 기존 방법과 비교하여 우수하거나 또는 조금의 성능 저하를 초래한다 할지라도 분류기 훈련 및 인식에 소요되는 시간을 줄일 수 있는 이점이 있다.

제안한 방법의 실험 결과 모든 실험 데이터에 대해 적게는 4%에서 많게는 30%까지 가지치기되었을 때 최적의 성능을 보였다. 평균적으로 8~25% 가지치기 된 경우 가장 좋은 성능을 보임을 알 수 있다.

표 3은 동일한 UCI 데이터에 대해 제안한 NNPGA 방법의 성능을 비교 평가하기 위해 실험에 의해 최고

성능을 가진 값과 다른 신경망 가지치기 및 구조증가 방법을 이용하였다.

NNPCv^[27] 방법은 신경망 훈련과정에서 페널티 함수(penalty function)와 교차검증 방법을 통해 성능에 영향이 적은 신경망 가지를 제거함으로써 성능을 향상 시켰다. C5.0^[28], M5'^[29] 방법은 결정트리(decision tree)를 이용하여 신경망 가중치를 결정함으로써 성능을 향상 시켰다. N2C2S^[30] 방법은 구조증가(construction) 방법으로 은닉노드를 증가시켜가며 최적의 성능을 가지는 신경망 구조를 찾았다. 위의 다른 알고리즘과의 검증을 위해 최대한 이들 실험과 동일한 실험 환경에서 검증하였다.

다른 알고리즘과의 통계적 검증을 위해 우리는 t -검정을 하였다. $H_0: \mu_1 - \mu_2 = 0, H_a: \mu_1 - \mu_2 > 0$ 가설에서 유의수준 $\alpha = 0.01$ 인 양측검증(two-tailed test) 결과 귀무가설이 기각 될 때 제안하는 방법의 성능이 우수하다 판단한다. 표 3에서 제안하는 방법과 다른 방법들과의 t -검정결과 제안하는 방법의 성능이 우수한 경우 (●),

표 4. 제안한 방법과 다른 알고리즘의 비교 평가 요약

Table 4. Summarization of evaluation to compare proposed algorithm with other algorithms.

	NNPCv	C5.0	M5'	N2C2S
NNPGA우수(●)	9	12	9	6
NNPGA열등(○)	4	1	4	7

그렇지 않을 경우 (○) 으로 표시하였다.

표 4의 비교 평가 요약 에서 볼 수 있듯 실험 데이터에 대한 다른 알고리즘과 비교하여 평균적인 성능 면에서 우리가 제안하는 NNPGA 방법이 좋은 결과를 보였다. 다만 N2C2S방법에 비해 제안한 방법의 인식 성능 편차가 크게 발생하였다. 그 결과 waveform-noise 데이터를 제외한 모든 데이터에 대해 평균 성능은 뛰어나지만, 표준편차 범위가 크게 발생하여 t-검정 결과 성능이 열등하다. 하지만 N2C2S 구조증가 알고리즘은 은닉노드에 대해서만 최적화를 시도한 반면, 제안하는 알고리즘은 입력과 은닉노드를 동시에 최적화하였다. 따라서 어느 정도의 성능저하를 고려한다 하더라도 N2C3S 방법에 비해 간단한 신경망 구조를 얻게 되는 이점을 얻을 수 있다.

V. 결 론

본 논문은 신경망의 입력과 은닉층 노드 가지치기 및 가중치 상속을 동시에 하여 최적의 신경망 구조와 가중치를 찾는 유전 알고리즘을 제안하였다. 노드 가지치기를 입력과 은닉 노드 동시에 수행함으로써 기존 입력 또는 은닉 노드 별개로 가지치기 되었을 때 보다 우수한 성능의 신경망 구조를 찾을 수 있다. 이와 더불어 진화 과정에서 가중치 상속은 교란의 역할로써 지역 최적점을 극복하여 우수한 성능을 얻을 수 있다. 마지막으로 우리는 신경망 구조와 가중치 최적화를 동시에 포함으로써 유전 알고리즘의 진화 및 신경망의 학습에 필요한 시간 단축 효과를 얻을 수 있었다.

다양한 데이터에 대해 실험한 결과 8~25% 정도의 가지치기에서 최적의 성능을 보임을 알아내었다. 신경망의 계산량은 예지의 개수에 비례한다. 따라서 20% 가지치기하였다면, 계산량이 대략 20% 감소됨을 뜻한다. 대용량 콘텐츠의 처리 및 모바일 환경에 적합한 인

식기 개발에는 이 정도의 속도 개선은 의미가 있다.

향후 연구에서는 최적의 가지치기 비율을 자동으로 결정하는 방법을 개발할 것이다.

참 고 문 헌

- [1] L. Fausett, Fundamentals of Neural Networks, Prentice Hall, 1994.
- [2] X. Yao, "Evolving Artificial Neural Networks," Proc. of the IEEE, Vol. 87, No. 9, pp. 1423-1447, 1999.
- [3] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid Genetic Algorithms for Feature Selection," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26, No. 11, pp. 1424-1437, 2004.
- [4] G. Castellano and A. M. Fanelli, "An Iterative Pruning Algorithm for Feedforward Neural Networks," IEEE Trans. on Neural Networks, Vol. 8, No. 3, pp. 519-531, 1997.
- [5] W. Wang, W. Lu, A. Leung, S. M. Lo, Z. Xu, and X. Wang, "Optimal Feed-forward Neural Networks Based on the Combination of Constructing and Pruning by Genetic Algorithms," Proc. of the 2002 Intl. Joint Conf. on Neural Networks, Vol. 1, pp. 636-641, 2002.
- [6] Z. Guo and R. E. Uhring, "Using Genetic Algorithms to Select Inputs for Neural Networks," Intl. Workshop on Combinations of Genetic Algorithms and Neural Networks, pp. 223-234, 1992.
- [7] 이하준, "임팩트 팩터 정규화 기법을 이용한 신경 회로망 연결소거 알고리즘", 한국과학기술원 석사 학위 논문, 2001.
- [8] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, 2006.
- [9] R. Reed, "Pruning Algorithms - A Survey," IEEE Trans. on Neural Networks, Vol. 4, No. 5, pp. 740-747, 1993.
- [10] Y. Cun, J. S. Denker, and S. A. Solla, "Optimal Brain Damage," Advances in Neural Information Processing Systems 2, pp. 598 - 605, 1990.
- [11] B. Hassibi and D. G. Stork, "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon," Advances in Neural Information Processing Systems 5, pp. 164 - 171, 1992.
- [12] B. E. Segee and M. J. Carter, "Fault Tolerance of Pruning Multilayer Networks," in Proc. Intl. Joint Conf. Neural Networks, Vol. II(Seattle), pp. 447-452. 1991.
- [13] E. D. Karnin, "A Simple Procedure for Pruning

- Back-propagation trained Neural Networks," IEEE Trans. Neural Networks, Vol. 1, No. 2, pp. 239-242, 1990.
- [14] J. Branke, "Evolutionary Algorithms for Neural Network Design and Training," Proc. of the First Nordic Workshop on Genetic Algorithms and their Applications, 1995.
- [15] G. Miller, P. Todd, and S. Hedge, "Designing Neural Networks using Genetic Algorithms," Proc. 3rd Intl. Joint Conf. on Genetic Algorithms, pp. 379-384, 1989.
- [16] P. Arena, R. Caponetto, L. Fortuna, and M. G. Xibilia, "Genetic Algorithm to Select Optimal Neural Network Topology," Proc. 35th Midwest Symposium on Circuit and Systems, Washington, 1992.
- [17] P. Arena, R. Caponetto, L. Fortuna, and M. G. Xibilia, "MLP Optimal Selection Via Genetic Algorithms," Proc. Intl. Conf. Neural Networks and Genetic Algorithms, Innsbruck, Austria, 1993.
- [18] J. C. F. Pujol and R. Poli, "Evolving the Topology and the Weights of Neural Networks Using a Dual Representation," Applied Intelligence, Vol. 8, No. 1, pp. 73-84, 1998.
- [19] Y. LeCun, "Generalization and Network Design Strategies," Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, Canada, 1989.
- [20] P. Koehn, "Combining Genetic Algorithms and Neural Networks: The Encoding Problem," Master's thesis, University of Tennessee, Knoxville, 1994.
- [21] H. Kitano, "Designing Neural Networks Using Genetic Algorithms with Graph Generation System," Complex Systems, Vol. 4, pp. 461-467, 1990.
- [22] F. Gruau, "Neural Networks Synthesis Using Cellular Encoding and The Genetic Algorithm," PhD thesis, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, France, 1994.
- [23] 문병로, 유전 알고리즘, 두양사, 2003.
- [24] <http://mllearn.ics.uci.edu/MLRepository.html>
- [25] Neural Network FAQ, 2004 [online] Available: <ftp://ftp.sas.com/pub/neural/FAQ.html>, W. S. Sarles, periodic posting to the Usenet newsgroup comp.ai.neural-net, URL.
- [26] P. P. Palmes, T. Hayasaka, and S. Usui, "Mutation-Based Genetic Neural Network," IEEE Trans. on Neural Networks, Vol. 16, No. 3, pp. 587-600, 2005.
- [27] T. Q. Huynh and R. Setiono, "Effective Neural Network Pruning Using Cross-validation," Proc. Intl. Joint Conf. on Neural Networks, pp. 972-977, 2005.
- [28] R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufman, San Mateo, CA, 1993.
- [29] E. Frank, Y. Wang, S. Inglis, G. Holmes and I. H. Witten, "Using Model Trees for Classification," Machine Learning, Vol. 32, No. 1, pp. 63-76, 1997.
- [30] R. Setiono, "Feedforward Neural Network Construction Using Cross-validation," Neural Computation, Vol. 13, No. 12. pp. 2865-2877, 2001.

 저 자 소 개



허 기 수(정회원)
 2000년 전북대학교 기계공학부
 산업공학과 학사 졸업.
 2004년 전북대학교 정보과학
 컴퓨터 정보 석사 졸업.
 2004년 ~ 전북대학교 컴퓨터정보
 학과 박사과정.

<주관심분야 : 영상처리, 패턴인식, 컴퓨터비전>



오 일 석(정회원)
 1984년 서울대학교 컴퓨터공학과
 학사 졸업.
 1992년 KAIST 전산학과
 박사 졸업.
 1992년 ~ 전북대학교 전자정보
 공학부 컴퓨터공학
 전공 교수.

<주관심분야 : 문서영상처리, 패턴인식, 컴퓨터비
 전>