

## 가중치감소 신경망의 자동학습에 관한 연구<sup>1</sup>

황창하<sup>2</sup> · 나은영<sup>3</sup> · 석경하<sup>4</sup>

### 요약

신경망은 점차 분류 및 함수추정을 위한 현대 통계적 방법론으로 부각되고 있다. 신경망은 특히 선형 회귀함수를 일반화시키는 유연한(flexible) 방법을 제공하며 일반적 비선형 함수를 모수화하는 방법으로 간주된다. 본 논문에서는 함수추정을 위한 신경망을 생각한다. 신경망이 훈련자료를 과대적합하는 것을 피할 수 있도록 하는 간단한 방법은 정칙화(regularization)이다. 신경망에서는 정칙화를 위해 주로 가중치감소법(weight decay method)을 사용한다. 함수추정을 위해 가중치감소 신경망을 사용할 때 은닉노드 수, 가중치모수, 학습률 및 학습반복회수가 중요한 모수이다. 본 논문에서는 유전자 알고리즘을 사용하여 가중치감소 신경망의 중요한 모수들을 자동으로 최적화하는 방법을 제안하고 결과적으로 가중치감소 신경망을 자동학습하는 방법을 설명한다. 그리고 다른 함수추정방법들과 자동학습된 가중치감소 신경망을 비교분석한다.

주제어 : 가중치감소법, 신경망, 유전자 알고리즘, 최적화, 자동학습.

### 1. 서론

신경망은 점차 분류 및 함수추정을 위한 현대 통계적 방법론으로 부각되고 있다. 신경망은 특히 선형 회귀함수를 일반화시키는 유연한(flexible) 방법을 제공하며 일반적 비선형 함수를 모수화하는 방법으로 간주된다. 본 논문에서는 함수추정을 위한 신경망을 생각한다. 신경망의 학습을 위해 주로 사용되는 방법은 훈련자료에 대한 오차제곱합  $E(\mathbf{w})$ 에 최급강하법(steepest descent)을 적용하여  $E(\mathbf{w})$ 이 최소화 되도록 신경망의 가중치 벡터  $\mathbf{w}$ 를 수정하는 역전파(backpropagation) 알고리즘이다. 많은 은닉노드들을 갖는 신경망을 사용하면 그 신경망은 훈련자료를 더 잘 적합한다. 그리고 은닉노드의 수가 증가함에 따라 새로운 관측치들에 대한 성능이 처음에는 향상되다가 나중에는 점점 나빠지는 것을 알 수 있다. 따라서 신경망에서 은닉노드의 수를 결정하는 것은 매우 중요한 문제이다. 따라서 역전파 알고리즘을 사용할 때 학습률 및 학습반복회수는 은닉노드수와 더불어 신경망의 중요한 모수이다.

신경망이 훈련자료(training data)를 과대적합(overfitting)하는 것을 피할 수 있도록 하는 간단한 방법은 정칙화(regularization)이다. 신경망의 정칙화에 대한 자세한 내용을 위해 Werbos(1988)와 Bishop(1995)을 참고하라. 신경망에서는 정칙화를 위해 주로 가중치감소법(weight decay method)을 사용한다. 이 방법은 굴곡이 심한 함수에 벌칙을 가하기 위해

<sup>1</sup>본 연구는 2000학년도 대구가톨릭대학교 연구비 지원에 의한 것임.

<sup>2</sup>경북 경산시 하양읍 대구가톨릭대학교 정보통계학과 교수

<sup>3</sup>경북 경산시 하양읍 대구가톨릭대학교 전산통계학과 박사과정

<sup>4</sup>경남 김해시 어방동 인제대학교 데이터정보학과 교수

오차제곱합에 가중치제곱합을 더한  $E(\mathbf{w}) + \lambda \sum_{ij} w_{ij}^2$ 에 최급강하법(steepest descent)을 적용하여 가중치 벡터  $\mathbf{w}$ 를 수정하는 방법이다. 이 방법은 가중치  $w_{ij}$ 의 크기를 축소시키고자 하는 의도를 가지고 있다. 학습을 위해 가중치감소법을 사용하는 신경망을 본 논문에서는 가중치감소 신경망이라 부른다. 일반적인 역전파 알고리즘의 성공여부는 훈련표본의 질에 전적으로 의존한다. 일반적인 역전파 알고리즘의 문제점들은 지역극소에 빠질 수 있다는 점과 긴 훈련 시간이 필요하고 이상치에 민감하다는 것 등의 단점을 가지고 있다. 그러나 가중치감소 신경망은 이런 문제들을 어느정도 극복하기 때문에 분류 및 함수추정을 위해 Ripley와 Ripley(1998)는 가중치감소 신경망을 사용할 것을 제안한다.

가중치감소 신경망을 사용할 때의 중요한 문제점은 모수  $\lambda$ 를 어떻게 선택해야 하는가이다. 만약 가중치들의 사전분포가  $p(\mathbf{w}) \propto \exp(-\lambda \sum_{ij} w_{ij}^2)$ 이면  $E(\mathbf{w}) + \lambda \sum_{ij} w_{ij}^2$ 을 최소화하는 문제는 가중치들의 사후분포를 최대화하는 문제로 귀착된다. 사전 분포는 평균이 0이고, 분산이  $1/2\lambda$ 인 정규분포를 따른다고 하면 입력값들의 로지스틱함수 적합은  $\pm 3$ 주변이며, 전체 입력값의 표준오차는 2정도가 된다. 만약  $[0,1]$ 사이에 입력값들의 수가 적다면 가중치들의 표준오차는 5정도를 제안하고 있으며 이 때  $\lambda = 1/50$ 이다. 이런 논의가 다소 보수적이라는 견해도 있지만 기본적으로  $\lambda \approx 10^{-3} \sim 10^{-1}$ 를 제안하고 있다. 한편, Ripley(1996)는  $\lambda \approx 10^{-4} \sim 10^{-2}$ 를 제안하고 있다.

가중치감소 신경망에서 은닉노드수, 가중치감소항의 계수  $\lambda$ , 학습률, 학습반복회수 등의 모수를 최적화하는 문제는 매우 중요하다. 그러나 본 논문에서는 학습률을 최적화 대상에서 제외한다. 왜냐하면 학습률은 신경망의 학습이론에 따라 반복회수  $t$ 의 함수  $\frac{1}{t}$ 에 비례하므로 실제로 프로그램 작성시 고려하면 된다. 가중치감소 신경망의 모수들을 최적화하기 위해 본 논문에서는 유전자 알고리즘(genetic algorithms, GA)의 사용을 제안한다. 유전자 알고리즘은 특히 최적화해야 할 모수의 수가 많을 때 강한 장점을 가지고 있다. 이론적으로는 충분한 은닉노드수를 가진 3층 전방향 신경망이 어떤 연속함수라도 근사할 수 있다고 말하지만, 몇 개의 은닉노드가 최적의 수인지는 명백하지가 않다. 신경망의 복잡도(complexity)는 은닉노드수와 가중치감소 신경망의 계수  $\lambda$ 에 의해 통제된다. 사실 가중치감소항을 사용하면 은닉노드의 수는 덜 중요하게 되는 경향이 있다. 자주 간과되는 한가지 어려운 문제는 신경망 적합의 알고리즘들이 적합기준의 최소점이 아닌 극소점만을 찾는다는 것이다. 가중치감소법이 사용될 때에도 일반적으로 여러개의 극소점들이 존재할 수 있다. 비록 많은 극소점들을 찾을 수 있다고 해도 이들 중 어떤 것을 선택해야 하는지 분명치 않다. 왜냐하면 훈련자료들을 잘 적합한다고 해도 검정자료들을 잘 추정하는 것은 아니기 때문이다.

이런 여러 가지 문제점을 고려하여 모수들을 최적화하고 결과적으로 가중치감소 신경망의 구조를 최적화하기 위해, 적합도를 최대화하는 여러 변수를 동시에 찾을 수 있는 최적화 알고리즘인 유전자 알고리즘을 사용한다. 이때 적합도로는 훈련자료들을 잘 적합하고 나아가 검정자료들을 잘 추정하기 위해 일반화교차타당성(generalized cross validation, GCV)의 함수를 사용한다.

본 논문에서는 유전자 알고리즘을 사용하여 가중치감소 신경망의 중요한 모수들인 은닉노드수, 가중치감소항 계수를 자동으로 최적화하는 방법을 제안하고 결과적으로 가중치감소 신경망을 자동학습하는 방법을 설명한다. 그런데 사전실험에 의해 학습반복회수는 1,000번이면 충분하여 계산시간을 줄이기 위해 학습반복회수를 그냥 1,000번으로 고정시키고 최적화 대상에서 제외시켰다. 그리고 본 논문에서는 Hwang 등(1994)이 사용한 다섯 개의 함수에 대해 자동학습된 가중치감소 신경망을 일반적 신경망, PPR(projection pursuit regression) 및 SVM(support vector machine)과 비교 분석한다. PPR은 고차원 비선형 함수 추정에 사용되는 대표적인 통계적 방법이며 SVM은 최근 기계학습분야에서 함수추정을 위해 가장 널리 사용되는 효율성이 매우 높은 방법이다.

## 2. 유전자 알고리즘 적용방법

본절에서는 유전자 알고리즘을 사용하여 가중치감소 신경망의 중요한 모수인 은닉 노드수와 가중치감소항 계수를 자동으로 최적화하는 방법을 설명한다. 1970년대 미국의 Holland에 의해 제안된 유전자 알고리즘은 자연도태의 유전적인 메카니즘에 기초한 탐색알고리즘이며, 그 목적은 세대(generation)가 지날 수록 주어진 환경에 잘 적응한 개체(individual)가 발생한다는 가설에 입각하여 최적 개체를 찾는 것이다. 유전자 알고리즘의 원리를 살펴보면 그림1과 같다.

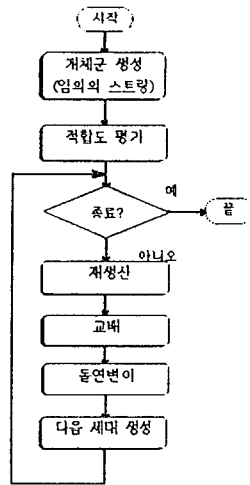


그림 1: 유전자 알고리즘 흐름도

전형적인 유전자 알고리즘은 프로그램에서 임의의 값으로 초기화된 개체들의 집합으로 시작한다. 즉 탐색공간에서 임의의  $N$ 개의 점들을 선택하여 개체군(population)을 형성한다. 여기서 개체군이란 어떤 세대를 형성하는 개체들의 집합을 의미하며 각 개체는 유전자의 집합으로 구성된 염색체를 가지고 있다. 개체군에서 주어진 문제에 적합한 개체를 확률적으로 재생산(reproduction)하여, 교배(crossover)와 돌연변이(mutation)를 통해 다음세대(offsprings)를 만들고 그 개체군에서 원하는 수준의 해가 존재하거나, 미리 정해 놓은 최대 세대수를 넘거나 또는 다른 종료 조건이 만족될 때까지 반복된다. 본 논문에서는 최대 생성 세대수를 100으로 설정하고 반복하였다.

유전자 알고리즘의 수행을 위해서 초기값의 설정이 필요하다. 초기값으로 개체군의 크기 30, 교배확률  $p_c = 0.9$ , 돌연변이확률  $p_m = 0.03$ , 은닉노드수 4, 가중치감소항 계수의 값 0.01을 사용하였다. 유전자 알고리즘에서 일반적으로  $p_c$ 의 값은 0.6 ~ 0.95이고  $p_m$ 의 값은 0.05 이하이다. 사실 최대 생성 세대수, 개체군의 크기, 교배확률, 돌연변이확률 등의 초기값은 유전자 알고리즘 수행시 주로 사용되는 값으로 설정되었다. 그리고 적합도 함수로는 훈련자료들을 잘 적합하고 나아가 검증자료들을 잘 추정하기 위해 사용되는 GCV의 함수 형태인  $1/(1 + GCV)$ 를 사용하였다. Orr(1998)의 GCV 계산방법을 사용하여 다음과 같이 GCV를 구한다. 입력  $\mathbf{x}$ 의 신경망 출력은  $f(\mathbf{x}) = \sum_{j=1}^m w_j h_j(\mathbf{x})$ 로 표현가능하다. 여기서 행렬  $\mathbf{H}$ 를  $\mathbf{H} = (H_{ij}) = h_j(\mathbf{x}_i)$ 로 두면

$$GCV = \frac{\sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2}{n - tr(\mathbf{P})}$$

여기서  $\mathbf{P} = \mathbf{I}_p - \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T$ 이고  $\mathbf{A} = \mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_m$ 이다.

유전자 알고리즘의 실험은 586PC에서 MATLAB으로 구현되었으며 실험의 과정은 다음과 같다.

[1단계] 위에서 제시한 초기값에 의하여 한 세대 개체군에 대하여 30개의 개체가 만들어지고 각 개체에 대하여 적합도를 계산한다. 선택메커니즘에 의해 그 세대의 일부 개체가 선택(재생산)되어진다. 즉, 적합도를 정규화(적합도/전체적합도)하여 개체군의 개체수 만큼 균등난수(uniform random)을 구한다. 그 다음 정규화한 적합도를 개체수 만큼 반복해서 누적시킨 값을 균등난수와 비교하여 누적값이 큰 인덱스를 찾는다. 그 인덱스의 개체가 재생성된다.

[2단계] 한 개체군의 30개의 개체 즉 30개의 염색체를 2로 나누어 15개의 쌍으로 짝지운다. 교배 위치가 선택되어져 교배확률보다 작으면 그 위치에서 교배가 일어나게 된다. 본 논문에서 사용한 MATLAB 툴박스의 유전자 알고리즘에서는 (개체수/2 만큼의 균등난수) × (염색체의 길이-1)를 정수화한 값이 교배위치가 된다. 이것을 염색체의 길이 만큼 반복한다. [3단계] 한 개체군에서 그 개체군의 각 개체에 대하여 돌연변이 확률보다 작은 염색체의 위치에서 돌연변이가 발생한다. 개체수 30만큼 균등난수를 생성하여 그 값이 돌연변이 확률보다 작은 값의 위치를 찾는다. 그리고 그 위치에 해당하는 염색체의 비트에서 값을 0은 1로, 1은 0으로 반전시킨다.

[4단계] 이런 과정을 한번 마치면 새로운 개체군의 한 세대가 만들어진다. 이것은 다시 적합도를 평가하여 종결조건이 만족될 때까지 반복수행한다.

### 3. 모의실험

본 논문의 모의실험의 첫번째 목적은 함수추정문제에서 유전자 알고리즘을 사용하여 어떻게 가중치감소 신경망의 모수들을 자동으로 최적화하고 결과적으로 가중치감소 신경망을 어떻게 자동학습하는지를 설명하는 것이다. 그리고 두번째 목적은 수식표현에서 너무 비슷하고 과대적합을 피하고 예측력을 높이기 위해 정칙화(regularization)를 사용하는 면에서 너무 비슷한 SVM과 역전파 알고리즘보다 우수하다고 알려진 PPR과 비교 분석하는 것이다.

Wahba 와 Wold(1975)가 제안한 French 곡선  $y_i = f_{true}(x_i) + \epsilon_i$ 에 대해 가중치감소 신경망이 어떻게 일변량 비선형 함수를 추정하는지 모의실험을 통하여 비교분석하였다. 여기서  $f_{true}(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$ 이며  $\epsilon \sim N(0, 0.2^2)$ 이다. 입력값  $x$ 는 구간  $[0, 3]$ 에서 랜덤으로 발생된 100개 값이다. 한편, 오염된 자료의 생성을 위해  $N(0, 9 \times 0.2^2)$ 에서 10개의 오차항을 생성하고  $N(0, 0.2^2)$ 에서 90개의 오차항을 생성하였다. 오염된 자료와 오염되지 않은 자료에 대해 커널, 스플라인, PPR, SVM 및 가중치감소 신경망을 사용해서 실험을 하였는데 대체로 SVM 및 가중치감소 신경망이 보다 더 좋은 결과를 보여주었다. 그러나 지면 관계상 결과를 생략하고 다음의 이변량 비선형함수 추정과 관련된 결과만을 언급하는데 이 결과만으로도 가중치감소 신경망의 성능을 충분히 평가할 수 있다. 모의실험은 586PC에서 MATLAB으로 구현되었다.

#### 3.1 실험함수와 실험자료

실험연구를 위해 유전자 알고리즘을 사용하여 가중치감소함을 갖는 신경망의 모형을 결정하고 결정된 신경망 모형으로 Hwang 등(1994)이 사용한 5개의 이변량 비선형함수를 추정하는 문제를 생각한다. 5개의 함수는 다음과 같다.

• Simple Interaction Function

$$g^{(1)}(x_1, x_2) = 10.391\{(x_1 - 0.4) \cdot (x_2 - 0.5) + 0.36\}$$

• Radial Function

$$g^{(2)}(x_1, x_2) = 24.234\{r^2(0.75 - r^2)\}, r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$

• Harmonic Function

$$g^{(3)}(x_1, x_2) = 42.659\{(2 + x_1)/20 + Re(z^5)\},$$

여기서  $z = x_1 + ix_2 - 0.5(1 + i)$  혹은

$$g^{(3)}(x_1, x_2) = 42.659\{0.1 + \tilde{x}_1(0.05 + \tilde{x}_1^4 - 10\tilde{x}_1^2\tilde{x}_2^2 + 5\tilde{x}_2^4)\},$$

여기서  $\tilde{x}_1 = x_1 - 0.5, \tilde{x}_2 = x_2 - 0.5$ .

• Additive Function

$$g^{(4)}(x_1, x_2) = 1.3356 [1.5(1 - x_1) + e^{2x_1 - 1} \sin\{3\pi(x_1 - 0.6)^2\} + e^{3(x_2 - 0.5)} \sin\{4\pi(x_2 - 0.9)^2\}]$$

• Complicated Interaction Function

$$g^{(5)}(x_1, x_2) = 1.9 [1.35 + e^{x_1} \sin\{13(x_1 - 0.6)^2\}e^{-x_2} \sin(7x_2)]$$

모의실험을 위한 225개의 훈련자료(training data)의 가로축 좌표값  $\{(x_{\ell 1}, x_{\ell 2})\}$ 의  $x_{\ell 1}, x_{\ell 2}$ 는 서로 독립이며, 균등분포로  $U[0, 1]$ 로 부터 생성되었다. 5개의 함수에 대한 모의실험에서 항상 같은 225개의 쌍 $\{(x_{\ell 1}, x_{\ell 2})\}$ 이 사용된다. 그리고 이들 가로축 좌표값에 대해 오차가 없는 훈련자료  $y_{\ell}^{(j)} = g^{(j)}(x_{\ell 1}, x_{\ell 2}), \ell = 1, 2, \dots, 225, j = 1, 2, \dots, 5$ 를 만들었다. 아울러 iid 정규오차를 더하여 훈련자료  $y_{\ell}^{(j)} = g^{(j)}(x_{\ell 1}, x_{\ell 2}) + 0.25\epsilon_i, \ell = 1, 2, \dots, 225, j = 1, 2, \dots, 5$ 를 만들었다. 여기서,  $\epsilon \sim N(0, 1)$ 이다. 그리고 적합된 모형의 성능평가를 위해서 검정자료(test data)를 사용하는데 이 검정자료는  $[0, 1]^2$ 상의 10,000개의 등간격 격자점  $x_{\ell i} = (2\ell - 1)/200, \ell = 1, \dots, 100, i = 1, 2$ 과 이들 격자점에 대한 10,000개의 함수값  $\{g^{(j)}(x_{\ell 1}, x_{\ell 2})\}$ 로 구성된다.

PPR은 차원문제(curse of dimensionality)를 극복한 고차원 비선형 함수추정에 사용되는 대표적인 통계적 방법이며 SVM은 최근 기계학습분야에서 함수추정을 위해 가장 널리 사용되는 효율성이 매우 높은 방법이다. 따라서 일반적 신경망(NN), 가중치감소 신경망(NNWD), SVM 및 PPR을 비교분석하는 것은 나름대로 의미가 있다. 비교분석을 위한 측도로 우리는 Hwang 등(1994), Roosen 과 Hastie(1994) 등이 사용한 FVU(fraction of variance unexplained)를 사용한다. 실험조건 또한 그들의 것과 동일하다. 한편 FVU는 다음과 같이 정의된다.

$$FVU = \frac{\sum_{\ell=1}^n (\hat{g}(x_{\ell}) - g(x_{\ell}))^2}{\sum_{\ell=1}^n (g(x_{\ell}) - \bar{g})^2}$$

여기서,  $g = g^{(j)}, j = 1, \dots, 5$ 이다.

### 3.2 자동학습 알고리즘의 구조

유전자 알고리즘을 사용하여 가중치감소 신경망의 중요한 모수들을 자동으로 최적화하는 알고리즘의 구조는 다음과 같다.

1. 초기화 단계

① 유전자 알고리즘:

최대생성 세대수(maximum generations) = 100

- 개체군의 크기(population size)= 30, 단, no generational overlap  
 교배확률(Crossover probability)= $p_c=0.9$   
 돌연변이 확률(mutate probability)= $p_m=0.03$
- ② 가중치감소 신경망: (유전자 알고리즘을 이용하여 찾는 모수들)  
 은닉노드수 = 4  
 가중치감소량 계수값 =  $\lambda=0.01$
- ③ 가중치감소 신경망 종결조건:  
 학습반복수 : 1000

## 2. 유전자 알고리즘 시작

초기값 : 2진화

- ① 은닉노드수 : 1개 ~ 15개, 4 bit 사용, 초기값 = 0100  
 ② 가중치 감소값 : 0.1 ~ 0.000001, 4 bit + 3 bit 사용  
 앞의 4 bit는 1 ~ 15를 표현하기 위해 사용  
 (단, 10 ~ 15인 경우는 9로 바꿔줌)  
 뒤의 3 bit는 소숫점의 자리수가 7이 되게 사용  
 초기값 : 0001 과 010 => 1 과 2 =>  $1 / 10^2 = 0.01$

## 3. While (최대적합도값) Repeat{

가중치감소 신경망 실행 (2진화 -> 10진화) 반복 1,000회  
 : 세대수 만큼 적합값 구함  
 적합도 계산 ( $1/(1+GCV)$ )

개체군 => 2진화

%재생산(Reproduce)

- ① 정규화 : 적합값(fitness)/전체적합값(sum(fitness));  
 ② 균등난수 생성 : size(fitness)적합값의 행렬의 크기만큼  
 균등난수를 생성한다.

for

- ③ 적합값 개체 길이 만큼 반복  
 ④ 정규화값 누적  
 ②의 균등난수와 ④의 값을 비교하여 큰 인덱스들  
 찾아 바꾼다

end

%교배(Crossover) < 교배확률 : 15쌍 개체군

for 검색체의 길이만큼 반복

crossover point : 교배확률보다 작은 위치에서 교배(1점 교배)

end

%돌연변이(Mutate) < 돌연변이 확률 : 30개 개체군

mutation point : 돌연변이확률보다 작은 위치에서 변이  
 (0은 1로 1은 0으로 반전)

개체군의 한 세대 생성

}

4. 세트 생성된 한 세대 => 10진와  
 은닉노드수와 가중치감소량  $\lambda$  값 출력

### 4. 모의실험 결과

유전자 알고리즘은 근본적으로 탐색을 위한 알고리즘으로 탐색공간에 대한 제약, 예를 들어 공간의 연속성과 같은 제약 조건을 갖지 않고 적합기준을 최적화시키는 여러개의 변수들을 동시에 탐색할 수 있기 때문에 여러 가지 문제에 적용이 가능하다. 그리고 주어진 상황에 적응적으로 대처해서 탐색을 한다는 특징을 가지고 있다. 본 논문에서는 유전자 알고리즘을 가중치감소 신경망의 모수들을 최적화하는데 적용하였으며 유전자 알고리즘의 수행결과로 결정된 가중치감소 신경망(NNWD)을 이변량 비선형 함수추정문제 적용하여 일반적 신경망(NN), PPR 및 SVM과 비교분석하였다.

유전자 알고리즘을 사용한 모수 최적화의 결과는 표1과 같으며 이 결과를 가중치감소 신경망에 적용한 함수추정의 결과는 표2에 주어져 있다.

Table 1. 유전자 알고리즘의 모수 최적화 결과

함수	오차가 없는 자료		오차가 있는 자료	
	$\lambda$	노드수	$\lambda$	노드수
$g^{(1)}$	0.0001	9	0.00006	10
$g^{(2)}$	0.0001	10	0.00008	10
$g^{(3)}$	0.000001	9	0.00001	10
$g^{(4)}$	0.00001	8	0.00009	8
$g^{(5)}$	0.00005	9	0.00009	10

표2에 의하면 오차를 포함하지 않는 훈련자료 및 검정자료에 대해서는 함수  $g^{(1)}$ 의 경우에 NNWD가 훨씬 좋은 결과를 보여주며, 함수  $g^{(1)}$ 을 제외한 나머지 4개의 함수들의 경우에 SVM이 가장 좋은 결과를 보여준다. 오차를 포함하는 훈련자료 및 검정자료에 대해서는 함수  $g^{(5)}$ 를 제외한 나머지 4개의 함수들의 경우에 NNWD가 가장 좋은 결과를 보여주며, 함수  $g^{(5)}$ 의 경우에 SVM이 다소 좋은 결과를 보여준다.

그러므로 수식표현에서 너무 비슷하며, 과대적합을 피하고 예측력을 높이기 위해 다같이 정칙화를 사용하는 가중치감소 신경망과 SVM이 다른 두 방법에 비해 다소 좋은 결과를 보여준다. 다른 비선형 함수를 추정할 때도 훈련자료 및 검정자료에 대해서 대체로 정칙화(regularization)를 사용하는 두 방법 즉, 가중치감소 신경망과 SVM이 좋은 결과를 보아 줄 것으로 예상된다.

Table 2. FVU에 의해 결정되는 정확도

함수	방법	오차가 없는 자료			오차가 있는 자료		
		노드수	훈련표본	검정표본	노드수	훈련표본	검정표본
$g^{(1)}$	NN	5	0.000534	0.001471	5	0.064836	0.070192
		10	0.000227	0.001338	10	0.064651	0.072336
	PPR	3	0.00075	0.001077	3	0.053925	0.080629
		5	0.000048	0.001095	5	0.050652	0.080352
	NNWD SVM	9	0.000012	0.000070	10	0.000642	0.001032
			0.000297	0.001632		0.005243	0.008412
$g^{(2)}$	NN	5	0.005934	0.007648	5	0.068849	0.083526
		10	0.003537	0.005400	10	0.058644	0.073024
	PPR	3	0.025906	0.034620	3	0.057967	0.082920
		5	0.001163	0.006069	5	0.058581	0.084407
	NNWD SVM	10	0.001453	0.004513	10	0.003649	0.005901
			0.000356	0.004492		0.006249	0.013994
$g^{(3)}$	NN	5	0.524021	0.424022	5	0.399383	0.566565
		10	0.142020	0.151487	10	0.146998	0.231935
	PPR	3	0.360373	0.577145	3	0.185445	0.321773
		5	0.112422	0.242643	5	0.138678	0.341155
	NNWD SVM	9	0.089426	0.225482	10	0.027876	0.141155
			0.012061	0.130577		0.028167	0.104874
$g^{(4)}$	NN	5	0.045945	0.019688	5	0.073419	0.086255
		10	0.004273	0.006913	10	0.059601	0.075581
	PPR	3	0.000389	0.000692	3	0.041929	0.091219
		5	0.000427	0.001915	5	0.038907	0.089882
	NNWD SVM	8	0.009025	0.036994	8	0.007918	0.035539
			0.000589	0.008519		0.017026	0.055434
$g^{(5)}$	NN	5	0.214445	0.236293	5	0.249337	0.286426
		10	0.025470	0.070035	10	0.096857	0.138735
	PPR	3	0.140191	0.227764	3	0.294997	0.543458
		5	0.016889	0.038313	5	0.060511	0.192520
	NNWD SVM	9	0.039365	0.180712	10	0.023723	0.122997
			0.001209	0.030118		0.019193	0.024875

## 참고 문헌

1. 공성곤, 김인택, 박대희, 박주영, 신요한 (1995). 유전자 알고리즘, 그린.
2. Bishop, C. M. (1995). Neural Networks for Pattern Recognition, Oxford.
3. Chapelle, O. and Vapnik, V. N. (2000). Model Selection for Support Vector Machines, *Advances in Neural Information Processing Systems*, 12, 230-236.
4. David, E. G. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley.



5. Gunn, S. (1998). Support Vector Machines for Classification and Regression, *ISIS Technical Report*, U. of Southampton.
6. Hwang, J-N, Lay, S-R, Maechler, M., Martin, D. and Schimert, J. (1994). Regression Modeling in Back-Propagation and Projection Pursuit Learning, *IEEE Transactions on Neural Networks*, **5**, 342-353.
7. Orr, M. J. L. (1998). Optimising the Widths of Radial Basis Functions. In *Fifth Brazilian Symposium on Neural Networks, Belo Horizonte, Brazil*.
8. Ripley, B. D. (1996). Pattern Recognition and Neural Networks, *Cambridge University Press*.
9. Ripley, B. D. and Ripley, R. M. (1998). Neural Networks as Statistical Methods in Survival Analysis. In *Artificial Neural Networks: Prospects for Medicine* eds R. Dybowski and V. Gant, Landes Biosciences Publishers.
10. Roosen, C. B. and Hastie, T. J. (1994). Automatic Smoothing Spline Projection Pursuit, *Journal of Computational and Graphical Statistics*, **3**, 235-248.
11. Smola, A. J. and Scholkopf, B. (1998). A Tutorial on Support Vector Regression, *NeuroCOLT2 Technical Report*, NeuroCOLT.
12. Wahba, G. and Wold, S. (1975). A Completely Automatic French Curve, *Communications in Statistics*, **4**, 1-17.
13. Werbos, P. (1988). Backpropagation: Past and future. In *Proceedings of the IEEE International Conference on Neural Networks*, *IEEE Press*. 343-353.

## A Study on Automatic Learning of Weight Decay Neural Network <sup>5</sup>

Changha Hwang <sup>6</sup> · Eunyoung Na <sup>7</sup> · Kyungha Seok <sup>8</sup>

### Abstract

Neural networks are increasingly being seen as an addition to the statistics toolkit which should be considered alongside both classical and modern statistical methods. Neural networks are usually useful for classification and function estimation. In this paper we concentrate on function estimation using neural networks with weight decay factor. The use of weight decay seems both to help the optimization process and to avoid overfitting. In this type of neural networks, the problem to decide the number of hidden nodes, weight decay parameter and iteration number of learning is very important. It is called the optimization of weight decay neural networks. In this paper we propose a automatic optimization based on genetic algorithms. Moreover, we compare the weight decay neural network automatically learned according to automatic optimization with ordinary neural network, projection pursuit regression and support vector machines.

*Key Words and Phrases:* weight decay, neural network, genetic algorithm, optimization, automatic learning

---

<sup>5</sup>This research was supported by the Catholic University of Daegu research grants in 2000.

<sup>6</sup>Associate Professor, Department of Statistical Information, Catholic University of Taegu

<sup>7</sup>Graduate School, Catholic University of Daegu.

<sup>8</sup>Associate Professor, Department of Data Science, Inje University