

# 진화 스트레티지를 이용한 CMAC 망 최적 설계

## Optimal Design of CMAC network Using Evolution Strategies

○이 선우\*, 김상권\*, 김중환\*\*

\*삼성전자 생활시스템연구소  
Tel:0331-8200-3278, Fax:0331-200-3298  
E-mail: senu@secns.sec.samsung.co.kr

\*\*한국과학기술원 전기 및 전자공학과  
Tel:042-869-3448  
johkim@vivaldi.kaist.ac.kr

**Abstracts** : This paper presents the optimization technique for design of a CMAC network by using an evolution strategies(ES). The proposed technique is designed to find the optimal parameters of a CMAC network, which can minimize the learning error between the desired output and the CMAC network's as well as the number of memory used in the CMAC network. Computer simulations demonstrate the effectiveness of the proposed design method.

**Keywords** : CMAC network, Optimization, Evolution Strategies(ES)

### 1 서론

1970년대 미국의 Albus에 의해 포유류의 소뇌를 모델링하여 “Cerebellar Model Articulation Controller (CMAC)”이라는 인공망과 이에 사용되는 학습 알고리즘이 제안되었다 [1]. 이 후 여러 연구자들에 의해 CMAC 알고리즘의 수렴성이 이론적으로 증명되었으며, 개선된 학습 알고리즘과 함께, Albus의 CMAC 망 맵핑의 개선 방법 [2]등도 제안되었다. 이러한 이론적 연구 뿐 아니라 응용에 있어서도 80년대 이후 많은 분야에서 CMAC 망을 이용한 여러 연구가 수행되고 있다. Miller [3]는 로봇의 실시간 제어를 위해 CMAC 망을 사용한 광범위한 실험과 연구를 수행하였고, 사구간 특성을 가지는 시스템에 대하여 CMAC 망을 이용하는 제어 기법도 소개되었다 [4].

CMAC 망은 다층 신경망(multi-layer artificial network)이 가지는 특징들을 대부분 포함하고 있을 뿐 아니라, 학습이 빠르고, 추가 학습(Incremental learning)이 용이하다는 점과 일반화(generalization)의 정도가 파라미터에 의해 임의로 조절된다는 점등의 여러 장점을 가진다. 그러나 구조와 알고리즘의 단순성에 비해 설계 파라미터인 일반화(generalization) 파라미터와 각 입력 변수에 대한 양자화 함수의 분해능(resolution)을 어떤 값으로 결정해야 최소의 메모리 수와 학습 오차를 가지는 지에 대해서는 제안된 방법이 없는 실정이다.

이와 같은 최적화 문제에 대한 해법들로는, 최근 소위 진화 알고리즘(evolution algorithm)이라는 카테고리 안에 유전자 알고리즘(GAs), 진화 스트레티지(ESs), 진화

프로그래밍(EPs)등과 같은 알고리즘에 대한 많은 연구가 이루어지고 있다. 이러한 알고리즘들은 자연계의 적자생존 원리를 모사하여 전역 최적해를 찾도록 제안된 것으로써, 연산 능력의 발전에 힘입어 보다 많은 응용이 이루어지고 있다 [5].

본 논문에서는 상기한 문제에 대하여 진화 스트레티지(이하 ES로 약함)를 이용하여 CMAC 망의 학습에 지대한 영향을 주는 설계 파라미터인 일반화 파라미터와 입력 변수에 대한 양자화 분해능을 최소의 메모리 수를 가지면서도 원하는 정도의 학습 오차를 나타내주는 최적 파라미터 값을 결정하는 방법을 제안하고자 한다. CMAC 망의 경우에 학습 오차와 메모리 수는 서로 상반되는 성질을 가지고 있다. 즉, 학습 오차를 줄이기 위해서는 보다 많은 메모리가 요구된다. 제안된 최적화 설계 방법은 이들 사이의 가중치를 적절히 조절함으로써 적절한 최적의 값을 찾아준다.

### 2 CMAC 망

CMAC 망은 그림 1에 나타냈듯이, 입력과 출력의 관계 정보를 분산하여 저장하는 일종의 look-up 표이며, 또한 학습 알고리즘에 의해 표의 내용을 적응적으로 고쳐나가는 학습 능력이다. 그림 CMAC 망의 구조와 학습 알고리즘에 대해 간략히 살펴보자.

다음과 같은 다입력, 단출력 비선형 함수  $f(x)$ 를 고려하자.

$$f(x) : X \rightarrow \mathbb{R}, \quad X \subset \mathbb{R}^n \quad (1)$$

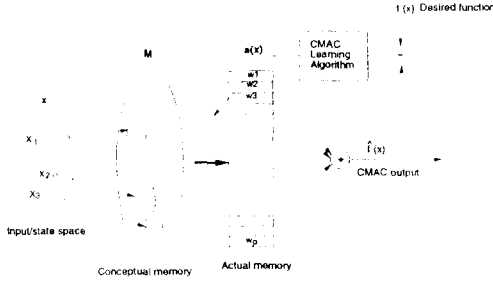


그림 1: CMAC 망의 기본 구조

여기서  $\mathbf{x}$  는  $n$  차원의 벡터로 다음과 같이 정의된다.  $\mathbf{x} \triangleq [x_1, x_2, \dots, x_n]^T \in X$ .

이 같은 비선형 함수를 CMAC 망을 이용하여 모사하기 위해 그림 1에 나타난 것처럼, CMAC 망은 주어진 입력 변수를 미리 결정된 분해능( $R_i, i = 1, \dots, n$ )을 이용하여 상대 변수로 양자화(quantization)하고,  $\mathbf{x} \rightarrow \mathbf{a}(\mathbf{x})$  맵핑을 통해  $\rho$ 개의 가중치를 선택하고, 이 선택된 가중치들을 합하여 출력을 만들어 낸다. 이를 다음 식과 같이 벡터 형태로 간단히 표현할 수 있다.  $\mathbf{x} \rightarrow \mathbf{a}(\mathbf{x})$  맵핑에 대한 자세한 설명은 생략하겠다.

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \sum_{i=1}^{\rho} \omega_{(i)} \\ &= \mathbf{a}^T(\mathbf{x})\mathbf{w} \end{aligned} \quad (2)$$

여기서  $\mathbf{x}$ 는 입력 벡터,  $\omega_{(i)}$ 는 주어진 입력  $\mathbf{x}$ 에 따라 전체 가중치 집합  $\{\omega_1, \omega_2, \dots, \omega_p\}$  중에서, 선택되어진  $\rho$ 개의 가중치를 나타낸다. 또한 상수  $\rho$ 는 일반화 파라미터(generalization parameter)라고 불리는 설계 파라미터로써, 인접한 입력과 공유하는 가중치의 수를 결정함으로써 일반화 특성을 결정하는 매우 중요한 파라미터다.

이제 이와 같은 가중치들의 학습 알고리즘을 살펴보자.

1.  $k$ 번째 스텝에서  $\mathbf{x} \rightarrow \mathbf{a}$  맵핑에 의해 유도된 연상 벡터에 의해  $\rho$ 개의 가중치가 선택된다.
2. 다음의 식과 같이 선택된 가중치들의 합으로 CMAC 망의 출력을 계산한다.

$$\hat{f}(\mathbf{x}_k) = \mathbf{a}^T(\mathbf{x}_k)\mathbf{w}_k$$

3. 모사하려는 실제 시스템의 출력과의 오차를 이용하여 다음과 같이 선택된 가중치들에 대해 값을 수정한다.

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \eta(\hat{f}(\mathbf{x}_k) - f(\mathbf{x}_k))\mathbf{a}(\mathbf{x}_k)$$

여기서  $\eta$ 는 학습 계수다.

4. 요구된 정밀도를 얻지 못했다면 위의 과정을 반복한다.

### 3 ES를 이용한 최적 설계

ES는 실험적인 최적화 문제를 풀기위하여 1960년대에 독일의 베를린 기술 대학(Technical University of Berlin)에서 개발된 이후, 여러 어려운 최적화문제에 성공적으로 적용되어왔다. 개체로써 실제 값을 가지는 벡터를 사용하고, Gaussian 돌연변이, 재조합(recombination) 그리고 상대적으로 매우 강한 선택 압력 (strong selective pressure)을 가지는 것 외에도 ES는 돌연변이 파라미터(mutation parameter)인 표준편차 값이 탐색되는 과정에서 자기-적응(self-adapt)한다는 것이 가장 주된 특징이다 [6].

ES 알고리즘은 다른 진화 알고리즘들과 같이 무작위로 만들어진 초기 개체 집단 (initial population)을 기반으로 무작위 연산 방법의 일종인 돌연변이(mutation), 재조합(recombination)등의 연산자들을 이용하여 탐색 공간내에서 최적의 해를 찾도록 구성된 알고리즘이다. 각 개체의 적합도(fitness)는 각 응용에 따라 정의되는 비용 함수를 이용하여 구해지며 이러한 비용 함수의 최대화 혹은 최소화하는 최적의 해를 찾는 것이 알고리즘의 목적이다. ES는 진화 알고리즘의 하나로써, 탐색 공간에서 더 나은 적합도를 가지는 개체들을 만들기 위해 EP가 돌연변이(mutation) 연산자만을 이용하고, GA가 재조합(recombination) 연산을 주로 이용하는 것과 달리 이 두가지를 모두 고루 이용한다.

본 논문에서는 (5+25)-ES를 사용하여 위에서 정의된 CMAC 망의 설계 파라미터, 즉, 일반화 파라미터,  $\rho$ 와 각 입력의 양자화 분해능,  $R_i, i = 1, \dots, n$ 를 구하였다. 이를 정리하면 아래와 같다.

1. (초기화) 최초로 무작위 수를 이용하여 5개의 개체를 생성한다. 하나의 개체는 다음 식의 벡터로 정의된다.

$$\mathbf{I}^i = (z^i, \sigma^i)^T, \forall i \in \{1, \dots, 5\} \quad (3)$$

여기서 벡터  $\sigma^i$ 는 알고리즘의 내부 변수이며, 초기값은 1이다. 시스템 파라미터  $z_j^i, \forall j \in \{1, \dots, n\}$ 는 각각  $\rho$ 와 각 입력에 대한 분해능,  $R_i, i = 1, \dots, n$ 이며, 주어진 최대-최소값의 범위내에서 균등 분포를 가지는 무작위 수로 초기화한다.

2. (자손 생성)

- 재결합 : 전역 중간 재결합 (global intermediate recombination) 연산 방법을 이용하였으며, 5개의 부모 개체로부터 25개의 자손 개체를 다음 식과 같이 생성한다.

$$\begin{aligned} z_j^i &= z_j^S + r \cdot (z_j^T - z_j^S), \forall i \in \{1, \dots, 100\}, \\ &\quad \forall j \in \{1, \dots, 15\} \end{aligned} \quad (4)$$

여기서 인덱스  $S$ 와  $T$ 는 10개 부모 개체로부터 무작위로 선택된 개체를 나타내며,  $r$ 은  $[0,1]$ 사이에서 균등 분포를 가지는 무작위 수를 나타낸다.

- 돌연변이: 각 파라미터별로 다음 식과 같이 돌연변이 연산을 행한다.

$$\begin{aligned}\sigma_j^{i'} &= \sigma_j^i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_j(0, 1)), \\ z_j^{i'} &= z_j^i + \sigma_j^{i'} \cdot N_j(0, 1),\end{aligned}\quad (5)$$

여기서  $N(0, 1)$ 와  $N_j(0, 1)$ 는 평균이 0이고 편차가 1인 정규 분포를 가지는 무작위 수를 나타내고, 상수  $\tau' = 1/\sqrt{2n}$ ,  $\tau = 1/\sqrt{2\sqrt{n}}$ 는 실험적으로 구해진 값들로 정하였다. 위와 같은 돌연변이 연산에 의해 만들어진 개체가 미리 결정된 범위를 넘으면, 올바른 범위 안에 있을때 까지 반복하여 연산을 행한다.

3. (평가) 각 개체는 CMAC 망의 설계 파라미터인 일반화 파라미터와 양자화 함수의 분해능을 나타내므로, 이같은 값들을 이용하여 하나의 CMAC 망을 설계하고 주어진 훈련 데이터를 이용하여 CMAC 망 학습을 수행한 후 학습 시의 학습 오차의 RMS 값과 총 메모리 수를 계산하여 이를 다음 식과 같이 가중치를 구해 각 개체의 비용함수를 산출한다.

$$e(k) = f(\mathbf{x}(k)) - \hat{f}(\mathbf{x}(k)) \quad (6)$$

$$e_{RMS} = \sqrt{\frac{\sum_{k=1}^{N_s} e(k)^2}{N_s}} \quad (7)$$

$$J_e(\hat{\mathbf{z}}) = \alpha e_{RMS} + \beta p \quad (8)$$

여기서  $N_s$ 는 훈련 데이터의 수를 나타내며,  $p$ 는 총 메모리의 수이고,  $\alpha$ ,  $\beta$ 는 학습 오차의 RMS 값과 메모리 수의 상대적 가중치를 결정하는 상수다.

4. (선택) 위에서 구해진 비용 함수값에 따라 순위를 매긴 후, 가장 우수한 개체군(가장 작은 비용함수)으로 5개를 선택하여, 새로운 세대를 형성한다.
5. (종결 조건 판단) 만약 지정된 시간이 되거나 적합한 해를 찾았으면 알고리즘을 종결하고, 그렇지 않으면 2의 과정으로 돌아가 알고리즘을 반복한다.

이와 같은 방법을 통해 최적의 파라미터 값을 결정할 수 있다.

#### 4 컴퓨터 시뮬레이션

제안된 방법의 유용성을 확인하기 위해 다음과 같은 2-입력, 1-출력의 2개의 비선형 함수를 사용하였다.

$$f_1(x_1, x_2) = \sin(x_1)\sin(x_2), \quad 0 \leq x_1 \leq 2\pi, \quad 0 \leq x_2 \leq \pi \quad (9)$$

$$f_2(x_1, x_2) = (x_1^2 - x_2^2)\sin(x_1), \quad -1 \leq x_1, x_2 \leq 1 \quad (10)$$

이와 같은 비선형 함수를 학습하기 위해 다음 식과 같은 범위를 가지는 일반화 파라미터  $\rho$ 와 입력 분해능  $R_1, R_2$  범위를 설정하였으며, 각 파라미터의 허용 범위 내에서

균등 분포의 무작위 입력 샘플 200개를 훈련 데이터로 사용하였다.  $\mathbf{x} \rightarrow \mathbf{a}(\mathbf{x})$  맵핑 방법은 Albus의 방법이 사용되었으며, CMAC 가중치의 학습은 Normalized Least Mean Square(NLMS) 방법을 ( $\eta=1$ ) 사용하였다. 식 (8)에 나타낸 바와 같이 비용 함수 계산에 사용되는 가중치들은 함수  $f_1(\cdot)$ 의 경우에는  $\alpha = 1$ ,  $\beta = 1e - 4$ 로, 함수  $f_2(\cdot)$ 의 경우에는  $\alpha = 1$ ,  $\beta = 1e - 5$ 로 설정하였다.

$$2 \leq \rho \leq 100$$

$$10 \leq R_1 \leq 200$$

$$10 \leq R_2 \leq 100$$

그림 2 (a)와 (b)에 각 함수에 대한 50 세대까지 ES 알고리즘 수행 결과를 도시하였으며, 각 세대내에서의 최적해의 비용함수 값을 표시하였다. 최종으로 구해진 설계 파라미터의 값은 다음과 같다.

$$f_1(x_1, x_2) : \rho = 26, R_1 = 92, R_2 = 49 \quad (11)$$

$$f_2(x_1, x_2) : \rho = 17, R_1 = 95, R_2 = 39 \quad (12)$$

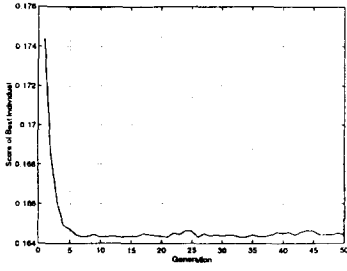
그림에서 알 수 있듯이 매우 빨리 적절한 파라미터 값으로 수렴함을 알 수 있다. 그림 3과 4에 함수  $f_1(\cdot)$ 과  $f_2(\cdot)$ 의 출력과 위의 최적해를 이용해 설계된 CMAC에 의해 학습된 출력을 도시하였다. 각 그림은 3차원의 그물망 형태의 출력과 등고선 결과도 동시에 나타내고 있으며, 두 함수 모두에서 최적의 메모리 수를 가지면서도 매우 근사한 출력 결과를 나타낸다.

#### 5 결론

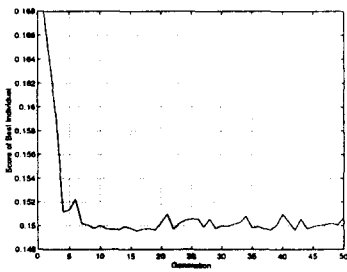
본 논문에서는 CMAC 망을 설계하는데 있어서 설계자가 결정해야 하는 일반화 파라미터와 각 입력에 대한 양자화 함수의 분해능 값을 최소의 메모리를 사용하면서도 적절한 학습이 가능한 최적의 값을 ES를 이용하여 결정하는 방법을 제안하였다. 일반적으로 일반화 파라미터가 CMAC 망에 끼치는 영향은 일반화 파라미터 값이 커질수록 메모리 수는 기하 급수적으로 줄어드나 학습 간섭이 심화하여 학습 오차가 증가하는 경향이 있다. 제안된 방법은 이와 같은 상반된 성질에 대해 적절한 가중치를 이용하여 복합적인 최적 설계가 가능하며 컴퓨터 시뮬레이션에서 입증한바와 같이 학습 오차가 크게 증가하지 않으면서도 가장 적은 메모리를 가지는 CMAC 망의 설계가 가능하다.

#### 참고문헌

- [1] J. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (cmac)," *ASME J. Dyn. Sys., Meas., and Contr.*, pp. 220-227, Sept. 1975.
- [2] M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control*. UK: Prentic Hall, 1994.

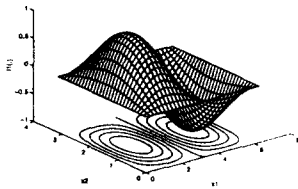


(a) 함수  $f_1(\cdot)$

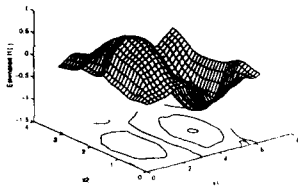


(b) 함수  $f_2(\cdot)$

그림 2: ES 최적해의 비용 함수 값

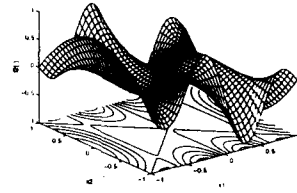


(a)  $f_1(\cdot)$

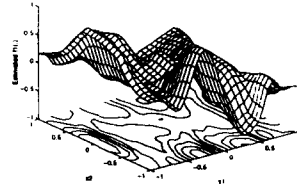


(b)  $\hat{f}_1(\cdot)$

그림 3: 함수  $f_1(\cdot)$ 과 CMAC 망 출력  $\hat{f}_1(\cdot)$



(a)  $f_2(\cdot)$



(b)  $\hat{f}_2(\cdot)$

그림 4: 함수  $f_2(\cdot)$ 과 CMAC 망 출력  $\hat{f}_2(\cdot)$

- [3] W. Miller, F. H. Glanz, and L. G. Kraft, "Cmac: An associative neural network alternative to backpropagation," *Proc. IEEE*, vol. 78, no. 10, pp. 1561–1567, 1990.
- [4] S.-W. Lee and J.-H. Kim, "Control of systems with deadzones using neural network based learning controller," in *IEEE Int. Conf. on Neural Networks*, vol. 4, (Orlando, USA), pp. 2535–2538, June 1994.
- [5] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Networks*, vol. 5, pp. 3–14, January 1994.
- [6] T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford: Oxford University Press, 1996.