

진화연산을 이용한 동적 귀환 신경망의 구조 저차원화

Structure Pruning of Dynamic Recurrent Neural Networks Based on Evolutionary Computations

김대준 · 심귀보

Dae Joon Kim and Kwee Bo Sim

중앙대학교 제어계측공학과

요 약

본 논문에서는 진화연산을 이용하여 동적 귀환 신경망의 구조를 저차원화하는 방법을 제안한다. 일반적으로 진화연산은 개체군을 이용한 탐색 방법으로서 신경회로망의 여러 가지 다른 성질을 동시에 최적화할 필요가 있을 때 유용한 방법이다. 본 연구에서는 동적 귀환 신경망의 구조를 저차원화하기 위하여 진화 프로그래밍으로 신경망의 구조를 탐색하고, 진화전략으로 신경망의 연결강도를 학습시킴으로서 전체적인 구조를 저차원화하였다. 신경망의 중간층 노드의 추가/삭제는 돌연변이 확률에 의하여 결정한다. 노드를 삭제할 경우에는 입력 연결강도의 총합이 가장 작은 노드를 삭제하고, 노드를 추가할 경우에는 미리 지정한 확률함수에 따라서 노드를 추가한다. 그리고 추가된 노드와 다른 노드와의 연결방법은, 서로 영향을 미칠 수 있는 모든 연결강도 중에서 확률적으로 선택하여 연결하였다. 마지막으로 제안한 저차원화 동적 귀환 신경망이 완전 연결된 신경망보다 더 좋은 성능을 얻을 수 있음을 예제로서 본 논문에서는 도립진자의 안정화 및 위치 제어와 로봇 매니퓰레이터의 비주얼 서보잉에 적용하여 컴퓨터 시뮬레이션을 통하여 그 유효성을 확인한다.

ABSTRACT

This paper proposes a new method of the structure pruning of dynamic recurrent neural networks (DRNN) using evolutionary computations. In general, evolutionary computations are population-based search methods, therefore it is very useful when several different properties of neural networks need to be optimized. In order to prune the structure of the DRNN in this paper, we used the evolutionary programming that searches the structure and weight of the DRNN and evolution strategies which train the weight of neuron and pruned the net structure. An addition or elimination of the hidden-layer's node of the DRNN is decided by mutation probability. Its strategy is as follows, the node which has minimum sum of input weights is eliminated and a node is added by predesignated probability function. In this case, the weight is connected to the other nodes according to the probability in all cases which can interact to the other nodes. The proposed pruning scheme is exemplified on the stabilization and position control of the inverted-pendulum system and visual servoing of a robot manipulator and the effectiveness of the proposed method is demonstrated by numerical simulations.

1. 서 론

뉴럴 네트워크를 이용하여 어떤 문제를 해결하고자 할 때 뉴럴 네트워크의 중간층의 개수와 연결강도의 수를 적절히 결정하는 것은 매우 중요한 문제이다. 만일 중간층과 연결강도의 수가 너무 많으면 자유 파라메타의 수가 많아져서 응답시간 및 학습시간이 많이 걸리며, 학습시 초기값에 따라 가중치가 전역 최소값에 도달하지 않고 계산량이 증가하는 등의 문제가 있다. 반대로 중간층과 연결강도의 수가 너무 적으면 입력력 패턴을 기억하지 못하고 학습이 최적해에 수

렴하지 않는다. 따라서 일반적으로 허용오차를 만족하는 범위 내에서 최소의 네트워크를 구하는 것이 가장 이상적이다. 즉, 최적 구조의 신경망이란 주어진 시스템에 대하여 최소의 노드수와 연결강도, 그리고 원하는 출력과 신경망의 출력과의 오차를 최소로 하는 연결강도로서 구성되는 것을 의미한다. 앞에서 최적구조의 신경망이란 단어를 사용하였는데 엄밀히 말해서 최적이라는 단어는 신경망의 어떤 면에 초점을 맞추느냐에 따라서 의미가 조금 바뀔 수 있다. 즉 신경망의 구조에 초점을 맞추느냐 혹은 성능에 초점을 맞추느냐에 따라서 약간 복잡한 구조에서 최고의

성능을 발휘할 수도 있다. 본 연구에서는 간단한 구조로서 좋은 성능을 발휘하기 위하여 신경망의 구조 저차원화에 초점을 맞추었다.

기존의 전방향(feedforward) 네트워크를 최적화 하기 위한 시도는 여러 방면에서 많이 시도가 되었다. 예를 들면 신경망을 이진인 구조행렬과 가중치 행렬로 표현한 유전 알고리즘을 이용하여 최적화하는 방법[1], 유전 알고리즘을 이용하여 전방향 구조를 최적화하고 온라인 학습 알고리즘에 의하여 최적의 가중치를 찾아내는 방법[2] 등이 있다. 이러한 전방향 네트워크는 구조상 정적인 사상만을 학습할 수 있으므로 시변 신호나 시계열 동적 사상을 학습하기 위해서는 제어기에 시스템의 입력을 tapped delay line으로 구성하거나, 시스템의 출력을 신경망의 입력으로 피드백하는 등의 변형이 필요하다. 반면 동적 귀환 신경망은 내부적으로 상태 피드백이 있기 때문에 입·출력 사이의 동적 사상을 학습할 수 있다. 따라서 정적 전방향 신경망과 같은 변형은 필요하지 않고 비선형 동적 시스템을 제어하는데 정적 귀환 신경망보다 내부적으로 상태 피드백이 있어 동특성을 가지는 동적 귀환 신경망이 적합하다.

동적 귀환 신경망의 구조를 최적화하는 연구는 아직까지는 많이 이루어져 있지 않다. Giles와 Omlin은 입력 연결강도의 총합이 작은 뉴런을 소거한 후에 재학습하는 방식[3]을 제안하였고, Yamamoto 등은 비선형 시스템을 선형시스템으로 근사한 후 Moore의 수법을 적용하여 중간층의 개수를 감소시킨 후에 재학습하는 방식[4]을 제안하였다. 하지만 단순히 입력 연결강도의 총합이 작은 노드만을 소거한다면 지역적 최소값에 빠질 우려가 있으며, Moore의 수법 또한 근사화에 의한 방법이므로 중요한 정보를 상실할 위험성이 있다.

따라서 본 논문에서는 동적 귀환 신경망의 구조를 저차원화하기 위하여 진화전략(ES)과 진화 프로그래밍(EP)을 이용하였다. 진화전략은 자연계의 진화현상에 기반한 계산 모델로서 실수치 탐색에 적합하고, 진화 프로그래밍은 예측문제의 해를 구하기 위하여 유한 상태 기계(FSM: finite state machine)를 진화시키는 방법에서 시작된 것으로서 최적화 문제의 해를 구하는 데에도 많이 사용된다. 본 연구에서는 진화 프로그래밍의 확률적 선택방식과 입력 연결강도의 총합이 작은 뉴런을 소거하는 방식을 융합하여 노드 및 연결강도를 삭제하였으며, 노드를 추가할 경우에는 미리 지정한 확률함수에 따라서 노드를 추가하였다. 그리고 추가된 노드와 다른 노드와의 연결방법은, 서로 영향을 미칠 수 있는 모든 연결강도 중에서 확률

적으로 선택하여 연결하였다. 또한 진화전략을 사용하여 탐색된 구조의 연결강도를 학습시킴으로써 지역적 최적해를 구하였다.

본 논문의 구성은 다음과 같다. 2장에서는 동적 귀환 신경망의 개요와 구조, 3장에서는 본 연구에서 제안하는 진화연산에 의한 구조의 저차원화 및 학습, 4장에서는 제안한 저차원화 동적 귀환 신경망을 도립 진자의 안정화 및 위치 제어와 로봇 매니퓰레이터의 비주얼 서보잉에 적용하여 컴퓨터 시뮬레이션을 통하여 제안한 수법의 유효성을 확인하고, 마지막으로 5장에서 결론을 맺는다.

2. 동적 귀환 신경망의 개요와 구조

80년대 후반부터 뉴럴 네트워크의 연구는 BP학습과 대칭결합의 네트워크에 의한 에너지 최소화라고 하는 2개의 원리를 축으로 해서 전개되어 왔는데 이들 기본 모델은 구조상 정적인 사상만을 학습할 수 있고, 시변 신호나 시계열 동적 사상을 학습하기 위해서는 구조상의 변형을 해 주어야 한다. 이러한 정적인 신경망의 능력과 한계가 명확하게 되자 전방향 혹은 대칭이라고 하는 결합형태의 제한이 없는 임의의 동적 귀환 신경망이 등장하였고 생물학적, 공학적 응용에 초점을 맞춘 연구가 진행되고 있다[5,6].

본 연구에서 사용하는 동적 귀환 신경망은 그림 1에서 보이는 바와 같이 내부적으로 상태 feedback과 self feedback이 존재하고 뉴런이 서로 비대칭 결합하고 있는 상호 결합형 뉴럴 네트워크이기 때문에 동적인 특성을 나타냄은 물론 시계열 데이터를 용이하게 처리할 수 있다[10].

동적 귀환 신경망에 있는 하나의 동적 뉴런은 위의 그림 2와 같은 구조를 가진다. 즉, 각각의 뉴런은

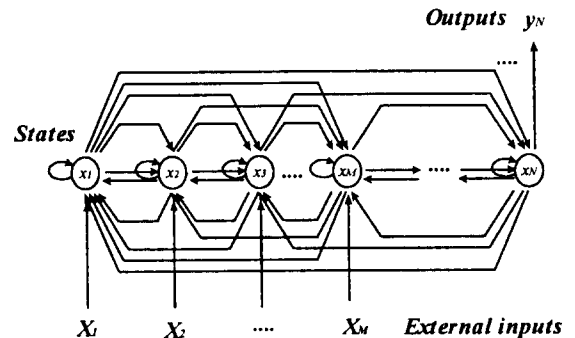


그림 1. 동적 귀환 신경망의 구조
Fig. 1. Structure of DRNN

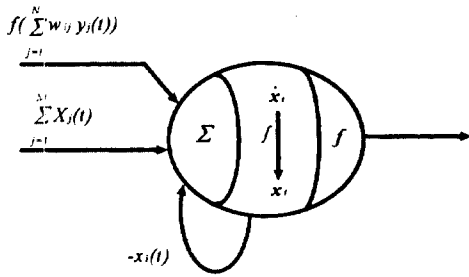


그림 2. 동적 뉴런의 구조
Fig. 2. Structure of dynamic neuron

self feedback에 의한 입력, 외부 입력, 다른 뉴런으로부터 들어오는 입력의 합을 전체 입력으로 하여 비선형 시그모이드 함수를 사용함으로써 -1~1로 정규화된 출력값을 얻는다.

그림 1에서 보인 동적 귀환 신경망의 상태 방정식과 출력방정식은 다음의 식 (1)과 (2)와 같이 된다.

$$\tau_i \frac{dx_i}{dt} = -x_i(t) + f\left(\sum_{j=1}^N w_{ij} y_j(t)\right) + \sum_{j=1}^M X_j \quad (1)$$

$$y_k(t) = f(x_k(t)) \quad (k=1, \dots, K: K\text{는 출력개수}) \quad (2)$$

여기서,

- w_{ij} : 뉴런 j 에서 뉴런 i 로의 결합하중
- τ : relaxation time scale
- x_i : 시간 t 에서의 뉴런 i 의 내부상태
- y_i : 시간 t 에서의 뉴런 i 의 출력
- X_i : 시간 t 에서의 뉴런 i 의 외부입력이 된다.

또, 본 연구에서는 출력으로서 -1~1까지의 값을 얻기 위해서 다음의 식 (3)과 같은 시그모이드 함수를 사용한다.

$$f(x) = \frac{2}{1 + e^{-\beta x}} - 1 \quad (3)$$

단, x 는 뉴런의 net input 이고, β 는 시그모이드 함수의 기울기이다.

3. 진화연산에 의한 구조의 저차원화 및 학습

본 논문에서는 동적 귀환 신경망의 구조에 초점을 맞추어서 주어진 시스템에 대하여 초기에 최대 노드와 최대 연결강도로 구성된 구조로부터 진화를 시작

하여, 원하는 출력과 신경망의 출력과의 오차를 최소로 하는 신경망의 최소의 노드수와 연결강도를 갖는 구조를 저차원화된 신경망으로 정의한다.

신경망의 구조를 탐색하기 위하여 진화 프로그래밍의 확률적 탐색방법과 입력하중의 총합이 작은 노드를 삭제하는 방법을 융합하여 사용하였으며, 진화 전략을 이용하여 탐색되어진 구조의 연결강도를 학습하였다.

3.1 진화 전략[7,8]

진화전략은 자연계의 진화현상에 기반한 모델로서 국소해가 많은 최적화 문제의 해를 구하는 것을 목적으로 1963년 독일의 I. Rechenberg에 의해 제안되었다. 진화전략의 특징은 주 연산자로는 돌연변이를 사용하고 주로 실수치 탐색에 쓰인다.

진화전략에서의 개체는 $(\vec{x}, \vec{\sigma})$ 와 같이 실수벡터로 나타낸다. 여기서 \vec{x} 는 상태공간상의 위치벡터이며, $\vec{\sigma}$ 는 표준편차 벡터가 된다. 이 때 돌연변이에 의한 다음 세대의 개체는 다음과 같이 표현된다.

$$\vec{x}^{t+1} = \vec{x}^t + N(0, \vec{\sigma}) \quad (4)$$

단, $N(0, \vec{\sigma})$ 는 평균이 0 이고, 표준편차가 $\vec{\sigma}$ 인 임의의 수를 나타낸다.

진화전략의 선택은 확률적이 아니고 결정적으로 행해진다. 즉, 부모벡터의 선택방법에 의해 나누어진 (μ, λ) -ES와 $(\mu+\lambda)$ -ES 중에서 적합도가 높은 순서대로 μ 개의 개체를 선택한다. 또한 진화전략에서는 최

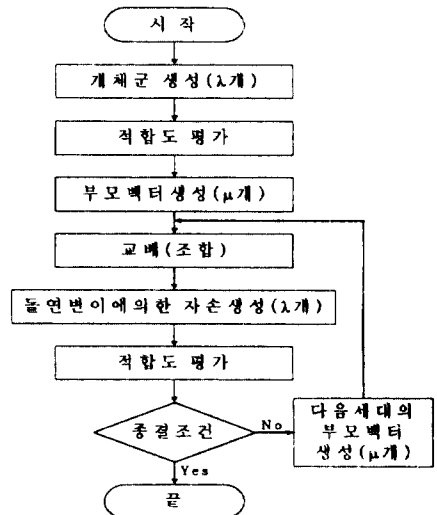


그림 3. 진화전략의 흐름도
Fig. 3. Flowchart of evolution strategies

적해를 찾기 위해서 수렴율을 최적화하는 1/5 규칙을 적용한다. 1/5 규칙이란 "성공하는 돌연변이의 비율을 1/5 로 하라는 것이다. 만약 성공하는 돌연변이의 비율이 1/5 보다 커지면 σ 를 크게 하고, 1/5 보다 작아지면 σ 의 값을 작게 한다." 이 규칙의 직관적인 의미는 "탐색의 효율을 높이기 위하여 돌연변이에 의해 성공하는 비율이 커지면 탐색을 보다 큰 보폭으로 계속하고, 그렇지 않으면 탐색의 보폭을 줄여라"라고 하는 것이다.

앞의 그림 3은 기본적인 진화전략의 흐름도를 나타낸 것이다.

3.2 진화 프로그래밍[8]

진화 프로그래밍은 1960년대에 L.J. Fogel에 의해서 제안되었으며, 예측문제의 해를 구하기 위하여 유한상태 기계를 진화시키는 방법으로, 최근에는 돌연변이 연산자를 이용한 최적화 문제의 해를 구하는 방법으로 많이 사용되고 있다.

3.2.1 동적 귀환 신경망의 구조 표현

본 논문에서는 N개의 노드를 가진 동적 귀환 신경망의 연결강도를 그림 4와 같이 행렬형태로 나타내었다. 그림에서 행과 열은 각각 노드로 들어오는 입력과 다른 노드로 나가는 출력값을 나타낸다.

구조의 탐색시에는 그림의 $N \times N$ 행렬을 1(연결을 의미함)과 0(연결되어 있지 않음을 의미함)으로 표현하여 신경망의 구조정보를 나타내었고, 학습시에는 연결강도의 값을 -1~1의 실수값으로 하였다. 이 때 주 연산자로 돌연변이를 사용함으로써 동적 귀환 신경망의 구조 탐색과 학습을 동시에 행하였다.

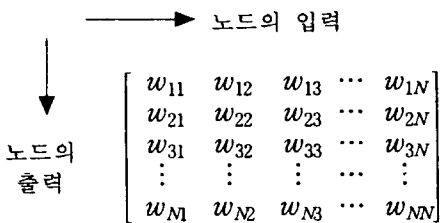


그림 4. 동적 귀환 신경망의 연결강도 행렬
Fig. 4. Weight matrix of DRNN

3.2.2 적합도 함수

신경망의 구조를 저차원화하고 적절한 연결강도를 구하기 위해서는 구조와 연결강도에 대한 평가기준이 있어야 한다. 본 논문에서는 그림 4의 $N \times N$ 행렬을 저차원화하기 위하여 진화 프로그래밍의 적합도 함수를 진화전략에서 구한 적합도 함수와 노드 및 연

결강도 함수의 곱의 형태로 식 (5)와 같이 정하였다.

$$fit_{EP} = fit_{ES} \cdot f(n) \cdot f(w) \quad (5)$$

$$f(n) = \frac{a}{n_{max} - n_{min}} (n - n_{min}) \quad (0 < a \leq 1) \quad (6)$$

$$f(w) = \frac{b}{w_{max} - w_{min}} (w - w_{min}) \quad (0 < b \leq 1) \quad (7)$$

식 (5)에서 fit_{ES} 는 진화전략에서 구한 적합도 함수, $f(n)$ 은 노드의 함수, $f(w)$ 는 연결강도의 함수를 각각 나타내고, 식 (6)과 식 (7)의 a 와 b 는 노드와 연결강도의 함수가 전체 적합도에 미치는 비율을 나타낸다.

만일 적합도 함수를 합인 형태로 설정하면 어떤 특정한 함수의 영향을 많이 받고 다른 함수의 영향이 전체 적합도에 적절히 고려되지 않을 수 있다. 따라서 본 연구에서는 곱의 형태로 적합도를 표현하여 각 함수의 비율로서 전체 적합도를 표현하였다.

노드와 연결강도의 함수는 노드와 연결강도의 개수가 증가함에 따라 감소하는 형태를 취하기 위하여 식 (6) 및 식 (7)과 같이 정하였다.

또한 진화 과정에서 재생산은 적합도 비례선택과 엘리트 선택법을 동시에 사용하였다. 적합도 비례선택은 적합도가 큰 개체에 대하여 선택될 확률이 높다. 반면 적합도가 낮은 개체도 확률이 낮기는 하지만 선택될 가능성을 가지고 있으므로 비록 현재세대에는 나쁜 개체라 할지라도 다음 세대에 좋아질 수 있는 가능성을 전혀 배제하지 않았으며, 또한 엘리트 선택법에 의해 가장 좋은 해를 보존하여 다음 세대에 남김으로써 돌연변이에 의하여 특별히 좋은 해가 소실되는 것을 방지하였다.

3.2.3 돌연변이 확률

노드 및 연결강도의 돌연변이 확률은 그림 5와 같이 정하였다. 그림에서 Pn_{max} , Pw_{max} 는 각각 노드와 연결강도 돌연변이 확률의 최대값을 나타내고, Pan/Pdn 은 노드의 추가/삭제 확률, Paw/Pdw 는 연결강도의 추가/삭제 확률을 각각 나타내고 식 (8) 및 식 (9)와 같이 표현한다. 즉, 탐색되어진 구조의 연결강도를 학습할 때 진화전략의 적합도에 따라서 노드 및 연결강도의 돌연변이 확률에 변화를 주어서 적합도 함수의 값이 좋으면 비례적으로 노드 및 연결강도의 삭제 확률을 크게 하여 신경망의 성능을 유지하면서 구조를 저차원화 하였다.

$$Pan, Paw = (1 - fit_{ES}) \cdot (Pn_{max} \text{ or } Pw_{max}) \quad (8)$$

$$Pdn, Pdw = fit_{ES} \cdot (Pn_{max} \text{ or } Pw_{max}) \quad (9)$$

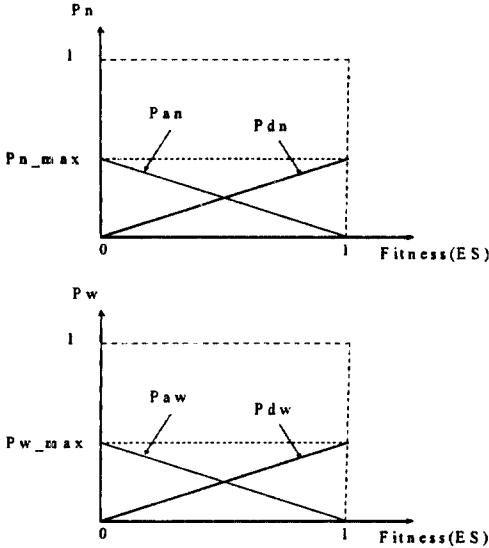


그림 5. ES 적합도 변화에 대한 노드 및 연결강도의 추가 및 삭제 확률
 Fig. 5. Probability of addition or elimination of node and weight according to ES fitness

일반적으로 큰 돌연변이 확률은 대역적 탐색(exploration)능력을 향상시키기 때문에 진화의 초기에 적합도가 높은 탐색공간을 찾는데 유리하지만, 동시에 국소적 탐색(exploitation)능력을 저하시킴으로써 어느 정도 좋은 해를 찾은 후에는 오히려 수렴속도를 저하시킬 수 있다. 따라서 본 연구에서는 진화 프로그래밍의 적합도 함수의 값이 좋을수록 돌연변이 확률의 크기를 줄임으로써 탐색된 좋은 개체를 보존하면서 진화를 계속할 수 있도록 식 (10)과 같이 돌연변이에 크기 변화를 주었다(그림 6). 이 식에서 S_{min} 은 돌연변이 확률의 최소 크기로 진화 프로그래밍에서의 적합도(fit_{EP})가 1인 경우에도 약간의 돌연변이가 일어날 수 있도록 미리 정해 준 값이다.

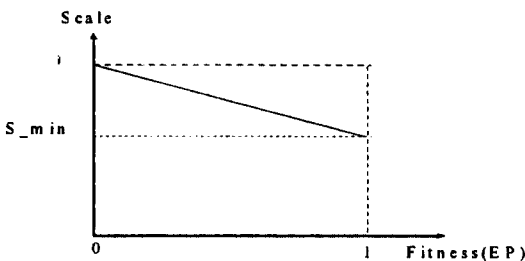


그림 6. EP적합도 변화에 대한 돌연변이 확률의 크기 변화
 Fig. 6. Scale of mutation probability according to EP fitness

$$Scale = (S_{min} - 1) \times fit_{EP} + 1 \quad (10)$$

그림 5와 그림 6에서 알 수 있는 바와 같이 진화전략에서 구한 적합도 함수의 값이 높으면 노드와 연결강도의 삭제 확률을 증가시켜 구조를 저차원화하고, 진화 프로그래밍의 적합도 함수값이 높으면 돌연변이 확률의 크기를 감소시켜 좋은 개체를 보존하면서 진화를 계속한다.

3.2.4 구조 저차원화

본 논문에서 제안한 노드 및 연결강도의 추가 및 삭제의 전략은 다음과 같다.

- 1) 돌연변이 확률에 의하여 노드의 추가 및 삭제를 결정한다.
- 2) 노드를 삭제할 경우 진화전략에서 구한 $N \times N$ 행렬에서 입력하중의 합이 가장 작은 것을 선택한다.
- 3) 노드를 추가할 경우 행과 열을 추가하고 연결강도는 다른 노드와의 관계에 의해서 다음과 같이 확률적으로 더해 준다.

• 임의의 노드 a 에서

$$\text{추가된 노드로 연결될 확률} : \frac{C_{out,a}}{T_N}$$

• 추가된 노드에서

$$\text{임의의 노드 } a \text{로 연결될 확률} : \frac{C_{in,a}}{T_N}$$

여기서

$C_{out,a}$: 노드 a 에서 나가는 연결강도의 수

$C_{in,a}$: 노드 a 로 들어오는 연결강도의 수

T_N : 전체 노드수

가 된다.

4. 시뮬레이션에 의한 검토

본 논문에서는 3장에서 구한 진화 프로그래밍의 적합도 함수와 돌연변이 확률에 의하여 동적 귀환 신경망의 구조를 저차원화 하였다. 저차원화 된 동적 귀환 신경망의 유효성을 확인하기 위하여 도립진자 시스템의 안정화 및 위치 제어(4.1절)[9]와 로봇 매니퓰레이터의 Image-based 비주얼 서보잉(4.2절)[10]에 각각 적용하였다. 위의 두 가지 예제에서 신경망의 최소 노드수와 최대 노드수를 미리 정해 주었다. 최소 노드수를 제한한 것은 중간층이 없이 입력과 출력만을 가지는 경우가 가장 저차원화 된 구조이기 때문이고, 최대 노드수를 미리 제한한 것은 신경망의 성능이 노드수의 증가에 단순히 비례하지 않으며 노드수가 증가하면 탐색영역이 증가하여 유전 탐색이 어려워지기 때

문이다.

각각의 예제에 대한 시뮬레이션을 위하여 연결강도를 학습시킬 때 진화전략은 (10+90)-ES, 돌연변이 확률은 0.2, 표준편차는 0.1로 설정하였으며 진화전략과 진화 프로그래밍의 개체군의 크기를 각각 100, 20개로 하였다. 또한 진화할 때 노드와 연결강도의 돌연변이 확률의 평균을 약 0.25, 0.2로 하기 위하여 독립진자 시뮬레이션에서는 $Pn_max=0.05$, $Pw_max=0.2$, 비주열 서보잉 시뮬레이션에서는 $Pn_max=0.025$, $Pw_max=0.2$, $S_min=2/3$ 로 설정하였고, 노드와 연결강도의 영향을 20%씩 반도록 a, b 값을 정하였으며 3장에서 구한 적합도 함수에 따라 동적 귀환 신경망의 구조를 탐색하였다.

4.1 독립 진자의 안정화 및 위치 제어

본 절에서는 3장에서 제안한 수법을 비선형성이 강한 독립진자 시스템의 안정화 및 위치 제어에 적용하여 그 유효성을 검토한다. 아래의 그림 7은 진화연산을 이용한 동적 귀환 신경망의 저차원화 과정을 독립진자에 적용한 블록 다이어그램이다.

그림에서 독립진자 시스템의 다이내믹스는 아래와 같다.

$$\ddot{x} = \frac{F + m_p l [\ddot{\theta} \sin \theta - \dot{\theta}^2 \cos \theta]}{m_c + m_p} \quad (11)$$

$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta \left[\frac{-F - m_p l \ddot{x} \sin \theta}{m_c + m_p} \right]}{l \left[\frac{4}{3} - \frac{m_p \cos^2 \theta}{m_c + m_p} \right]} \quad (12)$$

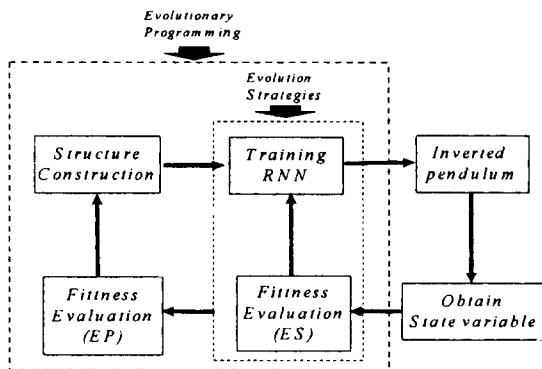


그림 7. 독립진자 시스템의 블록 다이어그램
Fig. 7. Block diagram of inverted pendulum system

단, 수레의 질량을 m_c , 질량이 m_p 이고 길이가 $2l$ 인 진자를 모델로 하였다.

동적 귀환 신경망의 최소 노드수를 5개, 최대 노드수를 10개로 하여 구조를 탐색하였다. 최소 노드수를 5개로 한 것은 신경망이 입력 ($x, \dot{x}, \theta, \dot{\theta}$)과 출력 (F)만을 가지도록 한 경우이다. 초기 수레의 위치는 기준점에서 1 m 떨어져 있고 수레에 15 N의 힘을 가했을 경우와 진자를 10° 기울였을 때 진자의 안정화 및 수레의 위치 제어를 목적으로 하였다.

그림 8은 초기에 10개의 노드와 100개의 연결강도를 가진 동적 귀환 신경망 구조로부터 진화를 시작해서 30세대가 진화된 후 최소의 노드수인 5개의 노드와 13개의 연결강도만을 가진 구조를 얻을 수 있었다.

그림 9는 저차원화 된 구조를 이용해서 독립진자의 안정화 및 위치 제어를 수행한 결과이다. 그림 (a)는 수레에 초기조건으로 15 N의 힘을 가한 경우이고, 그림 (b)는 초기조건으로 진자를 10° 기울인 경우에 대

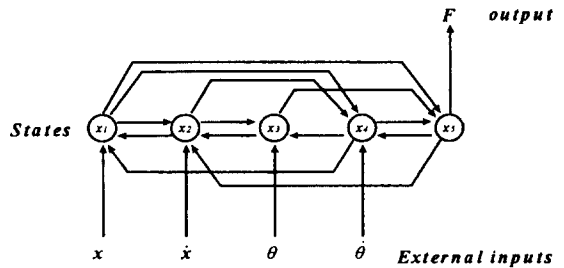
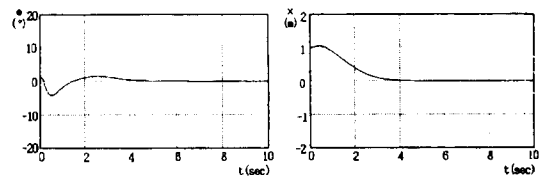
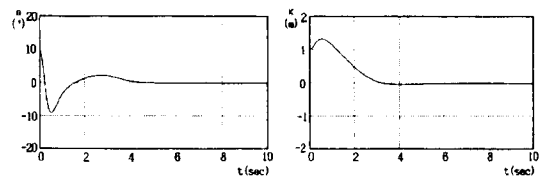


그림 8. 저차원화 된 구조
Fig. 8. Pruned structure



(a) 수레에 가한 초기 힘 : 15N
(a) Initial force applied to cart is 15N



(b) 진자의 초기 각 : 10°
(b) Initial angle of pendulum is 10°

그림 9. θ 와 x 의 수렴성
Fig. 9. Convergence of θ and x

하여 θ 와 x 의 수렴성을 각각 보인 것이다.

시물레이션 결과로부터 알 수 있는 바와 같이 초기 조건에 상관없이 도립진자가 빠른 시간 안에 목표값에 수렴하고 있음을 알 수 있다.

한편, 그림 10과 11은 진화하는 동안의 최대 적합도의 변화와 신경망의 구조를 탐색한 결과를 나타낸 것이다. 그림으로부터 알 수 있는 바와 같이 신경망 제어기의 진화초기의 조건(노드수: 10개, 연결강도수:

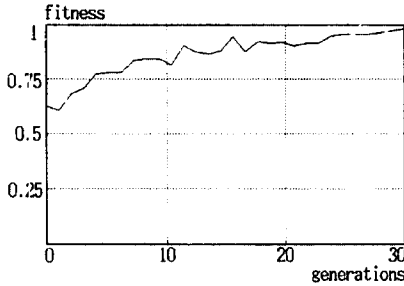
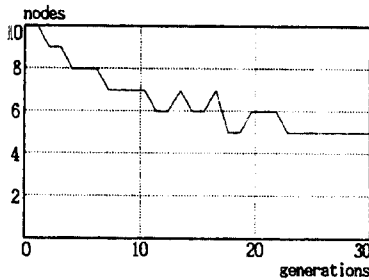
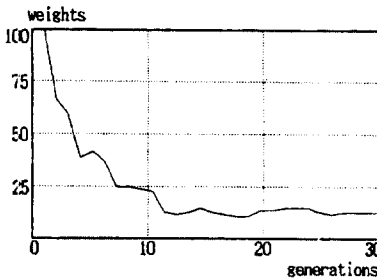


그림 10. 최대 적합도의 변화
Fig. 10. Change of best fitness



(a) 노드수의 변화(최소 노드수: 5)
(a) Change of node(min. number of node : 5)



(b) 연결강도 수의 변화(최소 연결강도 수: 13개)
(b) Change of weight(min number of weight : 13)

그림 11. 30세대 진화 후 탐색된 노드와 연결강도의 변화
Fig. 11. Change of node and weight after 30 generation evolution

100개)으로부터 진화하기 시작하여, 최대 적합도가 점점 증가함에 따라 노드수와 연결강도의 수는 점점 감소하여, 30세대 이후에는 최소 노드수인 5개의 노드와 13개의 연결강도만을 가진 구조로 저차원화 된 동적 귀환 신경망이 얻어질 수 있음을 확인할 수 있다.

4.2 매니퓰레이터의 image-based 비주얼 서보잉

본 절에서는 3장에서 제안한 수법을 RV-M2 로봇 매니퓰레이터의 비주얼 서보잉에 적용하여 그 유효성을 검토한다. 아래의 그림 12는 진화연산을 이용한 동적 귀환 신경망의 저차원화 과정을 비주얼 서보잉에 적용한 블록 다이어그램이다.

비주얼 서보잉을 위한 대상물체는 가로×세로×높이가 50×50×30 mm인 직육면체로 대상물체의 초기 위치와 각각의 특징점에 대한 정보는 미리 알고 있고, 대상물체는 로봇 매니퓰레이터의 작업공간에서 정지해 있다는 가정하에 시물레이션을 실시하였다.

또한 동적 귀환 신경망의 최소 노드의 수를 13개, 최대 노드의 수를 21개로 하여 구조를 탐색하였다.

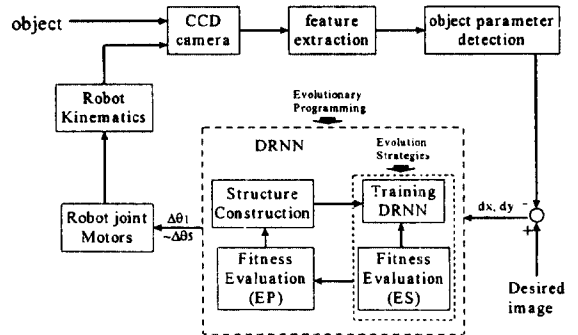


그림 12. 비주얼 서보잉 시스템의 블록 다이어그램
Fig. 12. Block diagram of visual servoing system

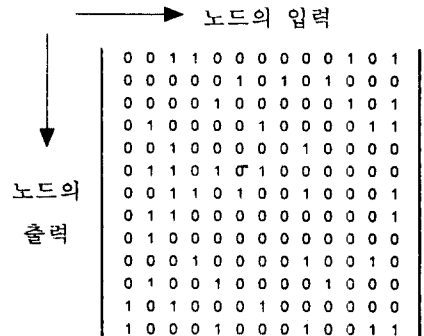


그림 13. 저차원화 된 구조 행렬
Fig. 13. Pruned structure matrix

최소 노드수를 13개로 한 것은 신경망의 입력(4개의 특징점에 대한 x, y 좌표 편차)과 출력(로봇 매니플레이터의 관절각)만을 가지도록 한 경우이다.

카메라가 부착된 로봇 매니플레이터의 엔드 이펙트를 이동시켜 실제의 이미지를 목표로 하는 이미지와 일치시키는 것을 목적으로 한다.

그림 13의 행렬은 초기에 최대 21개의 노드와 441개의 연결강도를 가진 구조로부터 진화를 시작해서 30세대 진화 후에 13개의 노드와 43개의 연결강도만의 구조를 가진 저차원화 된 동적 귀환 신경망을 나타낸 것이다. 그림에서 '1'의 의미는 노드가 서로 연결되어 있음을 나타내고, '0'은 연결되어 있지 않음을 의미한다.

그림 14는 4가지의 학습패턴에 대하여 저차원화 된 구조를 이용하여 로봇 매니플레이터의 Image-based 비주얼 서보잉에 적용한 결과 목표로 하는 이미지와 실제의 이미지와의 x, y 좌표 편차의 변화량을 나타낸 것이다. 단, 그림에서 dx 는 특징점 x 좌표의 변화량의 평균이고, dy 는 특징점 y 좌표의 변화량의 평균을 각각 나타낸다.

한편, 그림 15와 16은 진화하는 동안의 최대 적합도의 변화와 신경망의 구조를 탐색한 결과를 나타낸 것이다. 그림으로부터 알 수 있는 바와 같이 신경망 제어기의 진화초기의 조건(노드수: 21개, 연결강도수: 441개)으로부터 진화하기 시작하여, 최대 적합도가 점점 증가함에 따라 노드수와 연결강도의 수는 점점

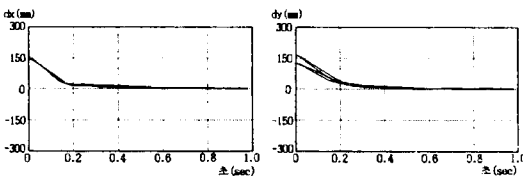


그림 14. 편차 dx, dy 의 변화
Fig. 14. Change of deviation dx and dy

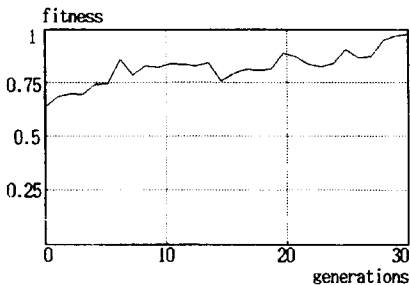
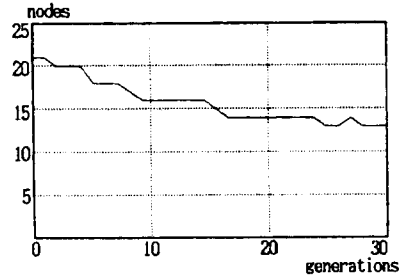
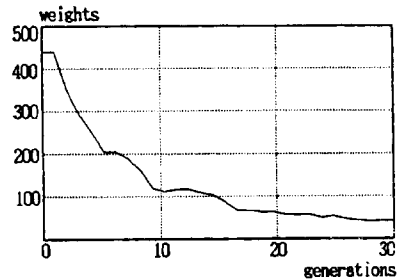


그림 15. 최대 적합도의 변화
Fig. 15. Change of best fitness



(a) 노드수의 변화(최소 노드 수: 13개)
(a) Change of node(min. number of node : 13)

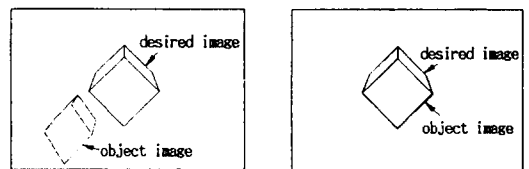


(b) 연결강도 수의 변화(최소 연결강도의 수: 43개)
(b) Change of weight(min. number of weight: 43)

그림 16. 30세대 진화 후 탐색된 노드와 연결강도의 변화
Fig. 16. Change of node and weight after 30 generation evolution

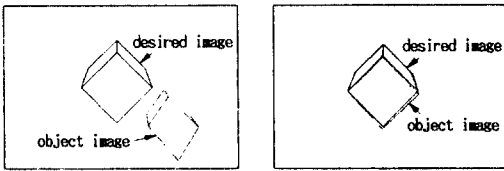
감소하여 30세대 이후에는 최소 노드수인 13개의 노드와 43개의 연결강도만을 가진 구조로 저차원화 된 동적 귀환 신경망이 얻어질 수 있음을 확인할 수 있다.

그림 17과 18은 직육면체의 대상물에 대한 로봇 매니플레이터의 추종특성을 나타낸 것으로서 그림 17은 학습시킨 패턴에 대한 시뮬레이션 결과이고, 그림 18은 학습시키지 않은 패턴에 대한 시뮬레이션 결과를 각각 나타낸다. 시뮬레이션 결과로부터 저차원



(a) 초기 이미지 (a) Initial image
(b) 최종 이미지 (b) Final image

그림 17. 학습시킨 패턴에 대한 시뮬레이션 결과
Fig. 17. Simulation result of learning pattern



(a) 초기 이미지 (b) 최종 이미지
(a) Initial image (b) Final image

그림 18. 학습시키지 않은 패턴에 대한 시물레이션 결과
Fig. 18. Simulation result of no learning pattern

화 된 제어기를 이용하여 학습시킨 데이터뿐만 아니라 학습시키지 않은 데이터에 대해서도 잘 추종해 갈 수 있다.

5. 결 론

본 논문에서는 진화프로그래밍과 진화전략을 이용하여 동적 귀환 신경망의 구조를 저차원화 하는 방법을 제안하였다. 노드의 삭제시에 진화 프로그래밍의 확률적 선택방식과 입력 연결강도의 총합이 가장 작은 뉴런을 소거하는 방식을 융합하였고, 연결강도의 추가시에는 연결되어 있지 않은 다른 노드로 갈 수 있는 경우의 수를 고려하여 미리 지정한 확률함수에 따라서 노드를 추가하였다. 그리고 추가된 노드와 다른 노드와의 연결방법은, 서로 영향을 미칠 수 있는 모든 연결강도 중에서 확률적으로 선택하여 연결하였다. 또한 진화 프로그래밍의 돌연변이는 진화전략에서 구한 적합도 함수와 진화 프로그래밍의 적합도 함수값을 동시에 고려하여 좋은 개체를 보존하면서 동적 귀환 신경망의 구조를 저차원화 하였다. 제안한 방법을 도립진자 시스템의 안정화 및 위치 제어와 로봇 매니플레이터의 비주얼 서보잉에 적용하였을 때 각각 5개의 노드와 13개의 연결강도, 13개의 노드와 43개의 연결강도만을 가진 구조로서 완전 연결된 구조보다 더욱 좋은 성능을 얻을 수 있었다.

참고문헌

[1] T. Fukuda, T. Kohno, and T. Shibata, "Learning Scheme for Recurrent Neural Network by Genetic Algorithm," *日本機械學會論文集*, Vol. 59, No. 564, pp. 34-40, 1993. 8.
[2] 조철현, 공성곤, "유전 알고리즘을 이용한 전방향 신경망 제어기의 구조 최적화," *전자공학회 논문지*,

Vol. 33-B, No. 12, pp. 95-105, 1996. 12.
[3] C. L. Giles, C. W. Omlin, "Pruning Recurrent Neural Networks for Improved Generalization Performance," *IEEE Trans. Neural Networks*, Vol. 5, No. 5, pp. 848-851, 1994.
[4] H. Nishida, Y. Matsumoto, Y. Yamamoto, "A Pruning Method of Recurrent Neural Networks," *日本計測自動制御學會論文集*, Vol. 32, No.3, pp. 379-388, 1996.
[5] Chin-Teng Lin, C. S. George Lee, *Neural Fuzzy Systems*, Prentice Hall, 1996.
[6] P. J. Angeline, G. M. Saunders, J. B. Pollack, "An Evolutionary Algorithm that Constructs Recurrent Neural Networks," *IEEE Trans. Neural Networks*, Vol. 5, No. 1, pp.54-64, 1994. 1.
[7] 이동욱, 심귀보, "진화전략을 이용한 도립진자의 안정화 및 위치제어," *한국퍼지 및 지능 시스템학회 논문지*, pp. 71-79, 1996. 12.
[8] Z. Michalewicz, *Genetic Algorithms+Data Structures=Evolution Programs 3rd ed*, Springer-Verlag, 1995.
[9] 김대준, 이동욱, 심귀보, "진화연산을 이용한 리커런트 뉴럴 네트워크의 저차원화에 대한 연구," *로보틱스·제어계측·자동화 종합학술대회*, pp. 28-31, 1997. 3. 14.
[10] 김대준, 이동욱, 심귀보, "저차원화된 리커런트 뉴럴 네트워크를 이용한 비주얼 서보잉," *한국퍼지 및 지능 시스템학회 춘계학술대회 논문집(97 KFIS)*, pp. 259-262, 1997. 3. 29.



김대준 (Dae Joon Kim) 준회원

1996년: 중앙대학교 제어계측공학과 졸업 (공학사)

현 재: 중앙대학교 대학원 제어계측학과 석사과정 재학중 로보틱스 및 지능정보시스템 연구실

주관심분야: 로보틱스, 뉴로-퍼지 및 소프트 컴퓨팅, 진화알고리즘, 인공생명, 지능제어, 로봇비전 등



심귀보 (Kwee Bo Sim) 정회원

제 6권 제 4호 참조

현 재: 중앙대학교 제어계측공학과 부교수