

아날로그 셀룰라 병렬 처리 회로망(CPPN)을 이용한 Pattern Classification

오태완, 이해정, 김형석
전북대학교 전자정보공학부

Pattern Classification with the Analog Cellular Parallel Processing Networks

Taewan Oh, Hyejung Lee, Hyongsuk Kim
Division of Electronics and Information Engineering
Chonbuk National University
E-mail : control@mail.chonbuk.ac.kr

Abstract

A fast pattern classification algorithm with Cellular Parallel Processing Network-based dynamic programming is proposed. The Cellular Parallel Processing Networks is an analog parallel processing architecture and the dynamic programming is an efficient computation algorithm for optimization problem. Combining merits of these two technologies, fast pattern classification with optimization is formed. On such CPPN-based dynamic programming, if exemplars and test patterns are presented as the goals and the start positions, respectively, the optimal paths from test patterns to their closest exemplars are found. Such paths are utilized as aggregating keys for the classification. The pattern classification is performed well regardless of degree of the nonlinearity in class borders.

I. 서론

컴퓨터의 지각 능력이 인간을 아직 능가하지 못하는 중요한 요인의 하나는 그 기술적 핵심인 영상 패턴의 분류 및 인식 기술이 충분히 성숙하지 못한 점 때문이라고 할 수 있다. 기존의 대표적인 패턴 분류 기술들

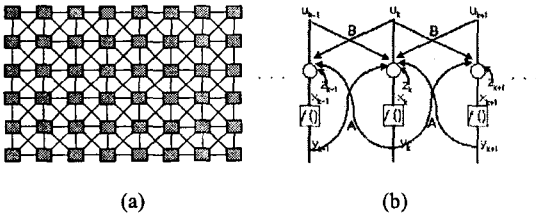
로는 확률 통계기반 기술, 퍼지기반 기술 및 신경회로망 기반 기술들로 크게 분류할 수 있다. 특히 신경회로망 기반 방법 [1]은 이와 같은 표본 패턴에 대한 분석이나 퍼지 규칙 생성이 불필요한 기술이다. 신경회로망은 인간과 유사한 방법으로 사전의 분석이나 지식이 없이도 표본 패턴간의 상관관계를 학습을 통해서 파악하고, 패턴 분류를 위한 규칙을 신경회로망에 자동으로 생성시키게 된다. 그러나, 이 방법의 단점은 많은 학습 시간 소요와 지역 최소치에 빠져 학습이 어려울 수 있다는 것이다.

제안한 패턴 분류 방법은 CPPN(Cellular Parallel Processing Networks, CPPN) [2][3]의 고속 병렬 처리와 지역적인 단순 연산을 통해 최적화 해법을 찾는 동적계획법(Dynamic Programming) [4]을 사용하는 방식이다. 이 CPPN 을 이용한 알고리즘은 표본 패턴을 사용한다는 점에서 기존의 신경회로망과 유사하지만, 학습을 하지 않는다는 점에서는 크게 다르다. 따라서, 기존의 신경회로망이 갖는 학습성공의 불확실성 문제가 없이 성공적인 분류를 항상 보장할 수 있으며, 학습도 불필요하다.

II. 셀룰라 병렬 처리 회로망(CPPN)

CPPN 은 그림 1(a)와 같이 2 차원 배열을 갖는 아

날로그 병렬처리 회로로서, 각 격자점에는 아날로그 연산처리 셀들이 배치되어 있다. 이 셀들은 주위의 8 개 셀들과의 회로적 연결을 가지며, 주변 셀들과 긴밀하게 정보 교환을 통하여 처리하는 구조이다. 그림 1(b)는 1차원 CPPN의 구조인데 x 는 각 셀의 출력 상태이며, y 는 셀의 출력이고 $f(\cdot)$ 는 셀의 상태값을 출력으로 변환하는 비선형 출력함수로서 시그모이드 함수를 주로 사용한다. 또한, u 는 입력을 표시한다. 그림 1(b)에서 각 연산 셀들의 입력으로는 자신의 위치에 해당하는 입력값 뿐 아니라 인접 셀들의 입력값들을 시냅틱 가중치들과의 컨볼루션에 의하여 입력하는데, 시냅틱 가중치 값들에 따라 다양한 종류의 연산이 가능하게 된다. 이 시냅틱 가중치 벡터를 **B** 템플릿이라고 부른다. 또한, 인접한 셀들의 출력들도 제한시켜 입력하는데, 이 때의 시냅틱 가중치 벡터를 **A** 템플릿이라고 한다. 그리고, 각 셀에는 제작시의 불균일을 고려하여 z 로 표현된 bias 값을 인가할 수 있게 되어 있다. 각 셀의 출력은 입력된 모든 정보들을 합산하여 출력함수 $f(\cdot)$ 에 의해 연산한 결과이다.



(a) 2차원 연결 (b) 1차원 CPPN 셀들의 연결

여기서, 템플릿 A 및 B에 의한 입력값과 내부 bias 값 z 를 고려한 회로의 동 특성식(dynamic)은 (1)처럼 1차 미분방정식으로 표현된다.

$$\frac{dx(i,j)}{dt} = -x(i,j) + \sum_{k \in N_i} A(i,j,k) y_k + \sum_{k \in N_i} B(i,j,k) U(k,l) + z_{ij} \quad (1)$$

식 (1)에서 $x(i,j)$ 는 셀 (i,j)의 출력측 상태를 의미하고 템플릿 $A(i,j,k)$ 와 $B(i,j,k)$ 는 각각 셀 (i,j)와 셀 (k,l)간의 출력측 및 입력측의 템플릿이며, Z_{ij} 는 셀 (i,j)의 bias이다. 이 셀의 출력 함수는

$$y(i,j) = f(x(i,j)) \quad (2)$$

와 같다.

III. CPPN을 이용한 동적계획법의 구현

동적계획법은 각 노드에서의 지역적인 최소값 연산을 통해서 전체적으로 최적의 경로를 계산하는 효율적 계산 알고리즘이다.

만약, $D(i,j)$ 와 $D(k,l)$ 를 각각 노드 (i,j)와 노드 (k,l)로부터 목적지에 이르는 데 소요되는 최소 거리라고 하자. 또, $d_{ij,kl}$ 를 노드 (i,j)로부터 노드 (k,l)에 이르는 지역적인 최소 거리라고 가정하면, $D(i,j)$ 는 $D(k,l)$ 와 $d_{ij,kl}$ 를 이용해서 다음과 같이 표현할 수 있다.

$$D(i,j) = \min \{ d_{ij,kl} + D(k,l), (k,l) \in R(i,j) \} \quad (3)$$

여기서, $R(i,j)$ 는 노드 (i,j) 주변의 모든 노드들의 집합이다. 식 (3)에서의 연산은 \min 과 $+$ 가산 연산인데, 일반적으로 \min 연산은 \max 연산보다 회로가 복잡하므로, 이를 위해서 기준값을 I_{max} 라 하고 $y(i,j)$ 를 I_{max} 와 $D(i,j)$ 간의 보수 (complement)로 표현하면, $y(i,j)$ 는

$$y(i,j) = I_{max} - D(i,j) \quad (4)$$

으로 표현된다. 식 (4)를 (3)에 대입하고, I_{max} 가 상수이므로, 식 (4)는

$$y(i,j) = -\min \{ d_{ij,kl} - y(k,l); (k,l) \in R(i,j) \} \\ = -\min \{ -(y(k,l) - d_{ij,kl}); (k,l) \in R(i,j) \} \quad (5)$$

로 정리된다. 식 (5)에서 부의 부호가 \min 연산식 밖으로 나오면 \min 이 \max 로 바뀌게 되므로,

$$y(i,j) = \max \{ y(k,l) - d_{ij,kl}; (k,l) \in R(i,j) \} \quad (6)$$

와 같아진다. 또한, 목적지에서는 자신의 위치까지의 거리 $D(i,j)$ 가 0이므로, 이 관계를 이용하면, 노드의 위치가 목적지 여부에 따라, 다음과 같이 정리할 수 있다.

$$y(i,j) = \begin{cases} I_{max} & \text{if } (i,j) \text{ is designated as the goal.} \\ \max \{ y(k,l) - d_{ij,kl}; (k,l) \in R(i,j) \} & \text{otherwise.} \end{cases} \quad (7)$$

여기서, I_{max} 값은 큰 값이므로 식 (7)을 간략화하면 $y(i,j) = \max \{ u(i,j), (y(k,l) - d_{ij,kl}, (k,l) \in R(i,j)) \}$ (8)

이고, $u(i,j)$ 는

$$u(i,j) = \begin{cases} I_{max} & \text{if } (i,j) \text{ is designated as a goal} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

로 표현할 수 있다. 이 식에서 확인할 수 있는 바와 같이 (1)에서의 \min 함수가 회로구현이 용이한 \max 함수로 표현되었음을 알 수 있다. 여기서, 식 (8)로 표현된 변형된 동적계획법의 원리는 목표점에 부여한 기준값 I_{max} 에서 지역적 거리값 만큼을 감한 결과가 모든 노드로 전파되게 되며, 각 노드는 이들 중에서 최대값을 취하

는 연산을 수행한다. 우리는 이 전파되는 값을 거리 기준값(Distance Reference, DR)이라 명명한다. 또한, 이 거리 기준값이 전파되어 0 보다 큰 영역을 DR 영역이라고 한다. DR 필드를 따라 식 (4)를 통해 최적 경로를 구할 수 있으며, (i,j)를 최적 경로상의 한 점이라면 최적 경로상의 다음 점은 식 (10)를 만족하는 점 (k,l)로 결정하면 된다.

$$\left\{ \begin{array}{l} \text{Cell } (i, j) \in R(k, l) ; \text{ for the } (i, j) \text{ which is the} \\ \text{point on the predetermined optimal path} \quad (a) \\ y(k, l) = y(i, j) + d_{ij,kl} \quad (b) \end{array} \right. \quad (10)$$

하나의 칩에 많은 셀을 구현하기 위해 본 논문에서 사용한 방법은 식 (8)을 단순한 여러 개의 연산들로 분해하여 각기 다른 층에서 처리하게 하는 것이다. 이를 위해서 식 (8)의 양변에 y_{ij} 를 더하면 식 (8)은

$$y_{ij} = \max \left\{ \frac{y_{ij} + y_{kl} - d_{ij,kl}}{2}, (k, l) \in R(i, j) \right\} \quad (11)$$

가 된다. 식 (11)에서 각각의 연산을 다층구조로 나누어 처리하게 할 수 있다. 즉, max 연산을 수행하는 층을 거리 계산층(Distance Computation layer, DC)이라고 하고, 괄호 내부의 연산을 하는 층을 중간층(Intermediate layer)이라고 하면, 각 층에서는 각각

$$y_{DC}(i, j) = \max \{ y_r(k, l), (k, l) \in R(i, j) \} \quad (12)$$

과

$$y_I(k, l) = \frac{1}{2}(y_{DC}(i, j) + y_{DC}(k, l)) - \frac{d_{ij,kl}}{2} \quad (13)$$

의 연산을 한다. 여기서, $y_{DC}(ij)$ 과 $y_I(ij)$ 는 거리 계산(DC) 층과 중간(I) 층의 출력이다. 경로 결정을 위한 조건식 (10(b))는 DC 층에서의 y 값이므로 이 값을 (13)식에 대입하면, 경로 결정 조건식은 (14) 식으로 단순화된다.

$$y_I(k, l) = y_{DC}(i, j) \quad (14)$$

이 식의 의미는 DC 층의 출력과 같은 값을 갖는 I 층의 셀이 있다면 그 방향이 path의 방향이다. 이와 같이 path 여부를 판단하는 일은 경로 결정층(PF layer)라고 하는 또 하나의 층에서 수행한다.

IV. CPPN의 최적 경로 계산을 이용한 패턴 분류

제안한 패턴 분류는 CPPN의 병렬처리 시스템과 동적계획법의 최적화 특성을 결합한 고속의 최적 경로 계산 알고리즘을 이용한 것이다. 이를 위해서 특징변수

의 공간을 CPPN 셀로 표현하고 최적 경로의 목적지 위치를 표본 패턴에 대한 특징값들의 위치로 표시한다. 또한, 테스트 패턴의 위치를 시작점으로 표시하여, 목표 위치 셀까지 이르는 최적 경로를 구하여 각 테스트 패턴이 속할 적절한 그룹을 선택하게 하는 것이다.

가령, i 번째 클래스의 집합을 P_i 라하고 이 집합내의 j 번째 패턴을 $p_i(j)$ 라고 하면,

$$p_i(j) \in P_i, \quad 1 \leq i \leq M, 1 \leq j \leq N_m \quad (15)$$

이다. 여기서, M은 클래스 번호이고, N_m 은 각 클래스에 속하는 각 패턴의 번호이다. 또한, 테스트 패턴을 P_i 라고 하고 학습용 표본 패턴을 $P_i(j)$ (j th pattern in the i th class)라고 하며, (i,j)위치의 표본 패턴에서 테스트 패턴까지의 거리를 $d_{i,j}$ 라고 하자. 또한, 테스트 패턴과 가장 가까운 거리에 있는 표본 패턴의 클래스를 k라 하고, 이 때의 거리를 $D_{i,k}$ 라고 하면, $D_{i,k}$ 는

$$D_{i,k} = \min \{ d_{i,j}, 1 \leq i \leq M, 1 \leq j \leq N_m \} \quad (16)$$

로 계산하며, 이 테스트 패턴은 k 번째 클래스에 속한다고 결정한다. 즉,

$$p_i \in P_k \quad (17)$$

가 된다.

제안한 알고리즘의 구체적인 구현 방법으로, 첫 번째 단계에서는 I 층에 셀 간의 거리값을 인가하는 과정이고, 다음 단계는 DC 층과 PF 층에 각각 표본 패턴과 테스트 패턴의 위치를 지정해주는 과정이다. 위치 지정을 위해서 DC 층에 위치한 모든 표본 패턴에 대해서 동일한 크기의 큰 거리 기준값 I_{max} 를 인가하고, PF 층에 위치한 모든 테스트 패턴의 위치를 큰 값 (+1)로 인가한다.

각 패턴의 위치가 지정되면 CPPN은 DC 층에 인가된 I_{max} 를 I 층에 있는 셀 간의 거리값만큼 감하면서 전파시킨다. 한편, 이 거리 기준값들이 테스트 패턴의 위치에 이르면 테스트 패턴에서부터 경로가 생성되기 시작한다. 이 때의 경로는 테스트 패턴의 위치나 형성된 경로에서만 이어갈 수 있도록 해야 하며, 클래스의 식별값을 전달하기 위한 통로로써 이용된다. 즉, 이 경로상의 위치에는 셀 간 거리값 0을 인가하며 경로 밖의 셀 간 거리값은 큰 값 +1을 부여한 후, 각 표본 패턴의 위치에 표본 패턴에 식별값을 인가하면, 경로상에서는 감소없이 테스트 패턴에 전달되게 된다. 이 식별값은 테스트 패턴이 어느 클래스에 해당하는지 알려준다.

V. 시뮬레이션

제안한 알고리즘에 의한 패턴 분류의 능력을 보였
다. 또한, 클래스 간에 고도의 비선형 경계선을 갖는 문
제에 대해서도 이 알고리즘이 잘 동작할 수 있음을 보
였다. 그림 2 는 4 개의 클래스를 갖는 문제로써, 그림
2(a)-2(b)는 각 클래스의 표본 패턴들의 위치와 테스트
패턴들의 위치이다. 여기서, 동일한 값의 거리 기준값들
을 각 표본 패턴의 위치에 인가하면, 이 값들은 그림
2(c)와 같이 셀 간의 거리값만큼 감소하면서 전파되게
된다. 이 거리 기준값이 테스트 패턴의 위치에 이르면,
테스트 패턴의 위치에서 발원한 경로선이 그림 2(d)와
같이 표본 패턴의 위치에 이르게 된다. 그림 2(e)는 그
림 2(d)의 경로 위로 전파된 표본 패턴들의 식별값을
보여주며, 그림 2(f)는 분류된 테스트 패턴의 결과로써
식별이 쉽도록 심볼로 표현한 것이다.

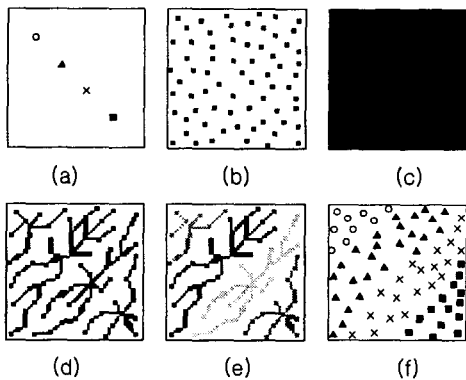


그림 2. 여러 종류의 클래스를 갖는 패턴들의 분류
(a)표본 패턴들의 위치 (b)테스트 패턴 (c)전파된 기준
값 (d)테스트 패턴으로부터 표본 패턴에 이르는 경로
(e)경로를 따라 전파된 클래스의 식별값 (f)테스트 패턴
들의 분류 결과

일반적인 패턴 분류 문제에 있어서 클래스 간의 경
계가 비선형성이면 분류가 매우 어렵다. 그림 3(a)-3(b)
는 클래스 간의 경계가 비선형적인 표본 패턴과 테스트
패턴의 위치를 보여준다. 또한, 그림 3(c)-3(f)는 각각 전
파된 거리 기준값, 각 테스트 패턴들에서 구해진 최적
경로 및 패턴의 식별값, 분류된 결과들을 보여준다.

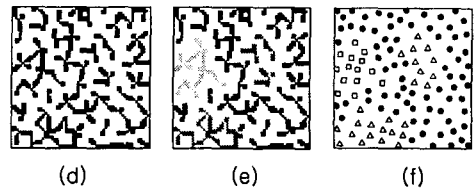
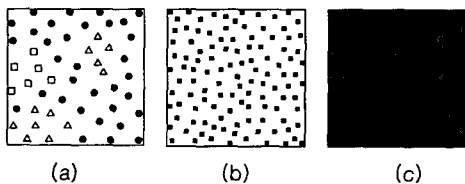


그림 3. 클래스 간의 경계가 매우 비선형인 패턴들에
대한 분류 (a) 표본 패턴들의 위치 (b) 테스트 패턴 (c)
전파된 기준값 (d) 테스트 패턴으로부터 표본 패턴에
이르는 경로 (e) 경로를 따라 전파된 클래스의 식별값
(f) 테스트 패턴들의 분류 결과

VI. 결론

동적계획법을 CPPN 의 아날로그 병렬처리 알고리
즘과 결합하여 패턴을 분류하는 방법을 제안하였다.
이 알고리즘은 CPPN 의 고속 병렬처리 능력과 동적계
획법의 최적화 능력을 함께 활용한 것이다.

이 패턴 분류 알고리즘에서는 표본 패턴과 테스트
패턴의 위치가 각각 최적 경로 계산문제에 있어서의 목
적지와 시작점 위치로 간주된다. 제안한 분류 방법에서
는 거리 기준값과 표본 패턴의 식별값들의 정보가 전파
되는 과정에서의 클래스 간 경쟁과 그 결과로 나타나는
최적화 특성을 이용하는 것이다. 또한, 이 알고리즘에서
는 표본 패턴들의 특징이나 사전 지식이 없이도 패턴을
분류할 수 있다는 점도 장점이다.

참고문헌

- [1] Murphey, Y. L. Yun Luo, "Feature Extraction for a Multiple Pattern Classification Neural Network System," Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol. 2, pp. 220-223, 2002.
- [2] L. O. Chua, and L. Yang, "Cellular Neural Networks: theory," IEEE Tr. on Circuits Systems, vol.35, pp.1257-1272, 1988.
- [3] L. O. Chua, and L. Yang, "Cellular Neural Networks: applications," IEEE Tr. on Circuits Systems, vol. 35, pp. 1273-1290, 1988.
- [4] R. Bellman, "Dynamic Programming," Princeton, NJ: Princeton Univ. Press, 1957.