ISSN 1975-8359(Print) / ISSN 2287-4364(Online)
The Transactions of the Korean Institute of Electrical Engineers Vol. 65, No. 6, pp. 1070~1079, 2016
http://dx.doi.org/10.5370/KIEE.2016.65.6.1070

빅 데이터 처리를 위한 증분형 FCM 기반 순환 RBF Neural Networks 패턴 분류기 설계

Design of Incremental FCM-based Recursive RBF Neural Networks Pattern Classifier for Big Data Processing

이 승 철* · 오 성 권†
(Seung-Cheol Lee · Sung-Kwun Oh)

Abstract - In this paper, the design of recursive radial basis function neural networks based on incremental fuzzy c-means is introduced for processing the big data. Radial basis function neural networks consist of condition, conclusion and inference phase. Gaussian function is generally used as the activation function of the condition phase, but in this study, incremental fuzzy clustering is considered for the activation function of radial basis function neural networks, which could effectively do big data processing. In the conclusion phase, the connection weights of networks are given as the linear function. And then the connection weights are calculated by recursive least square estimation. In the inference phase, a final output is obtained by fuzzy inference method. Machine Learning datasets are employed to demonstrate the superiority of the proposed classifier, and their results are described from the viewpoint of the algorithm complexity and performance index.

Key Words : Incremental fuzzy C-Means, Recursive least square estimation, RBF neural networks

1. 서 론

세계적으로 인공지능, 컴퓨팅 기술, 모바일 기술의 발달과 디지털 경제의 확산으로 인해 데이터가 기하급수적으로 늘어나고 있어 다양한 분야에서 빅 데이터를 주목하고 있다. 세계 경제 포럼은 떠오르는 10대 기술 중 하나로 빅 데이터 처리 기술을 선정하여 실시간으로 수집 가능한 데이터를 처리하기 위한 기술들의 중요성이 부각되고 있다. 또한 최근 우리나라 ICT (Information and Communications Technologies) 10대 핵심기술 중 하나로 빅 데이터를 선정하였다. 문자와 영상 데이터도 포함하고 있는 빅 데이터는 시간이 지남에 따라 누적되는 데이터의 양이 기하급수적으로 증가하여 테라바이트(Terabyte)를 넘어 페타바이트(Petabyte)에 이르고, 과거에 생성되던 데이터에 비해 생성주기가 짧다. 이러한 방대한 양의 데이터를 어떻게 처리할 것인지, 또한 어떻게 학습할 것인지에 대한 연구가 활발히 진행되고 있다. 이에 대한 연구로써 ICT 기술 중 하나인 인공지능(AI: Artificial Intelligence)을 꼽을 수 있다. 인공지능은 인간의 학습능력과 추론능력, 지각능력, 이해능력 등을 컴퓨터로 실현한 기술로써 최근 미국, 일본등과 같은 선진국에서는 인공지능이 미래사회를 지배할 것이라고 판단하고 있다. 우리나라에서도 인공지능 기술을 우리나라의 산업을 이끌어 나갈 중요한 기술로 여기고 있다. 인공지능 기술은 빅 데이터의 중요성이 부각되기 전부터 많은 연구가 진행 되었으나, 최근 빅 데이터가 사회적 이슈로 등장한 이후 더욱 더 많은 연구들이 진행되고 있다. 또한 과거에는 인공지능 기술을 바탕으로 다양한 산업분야로 적용하기에는 다소 어려움이 많았으나, 최근에는 많은 산업분야에서 인공지능 기술 중 하나인 신경망을 이용하여 음성인식, 영상처리, 전력감시와 같은 분야에서 응용 및 적용하고 있다. 더불어 빅 데이터가 사회적 이슈로 등장한 이후에는 금융, 포털 등 다양한 분야에서도 응용하여 적용하고 있는 추세이다. 하지만 아직까지 방대한 양의 데이터 처리 및 학습에 관한 연구는 고성능의 하드웨어를 기반 한 연구들만 진행될 뿐 방대한 양의 데이터를 효과적으로 처리하는 방법과 효율적으로 메모리를 사용하는 방법에 대한 연구는 미비하다. 또한 방대한 양의 데이터를 순차적으로 처리하는 방법에 관한 연구는 더욱 미비하다. 이에 따라 고성능의 하드웨어 사용이 불가능한 상황에서는 방대한 양의 데이터를 처리 및 학습하지 못하는 문제가 발생한다.

따라서 본 논문에서는 빅 데이터를 순차적으로 처리하고, 제한적인 상황에서 보다 효율적으로 컴퓨터 메모리를 사용하기 위해 지능형 알고리즘을 이용하여 패턴 분류기를 설계한다.[1][2] 지능형 알고리즘으로 다차원 문제, 강인한 네트워크 그리고 예측능력이 우수하다고 알려진 방사형 기저함수 신경회로망(RBFNN:

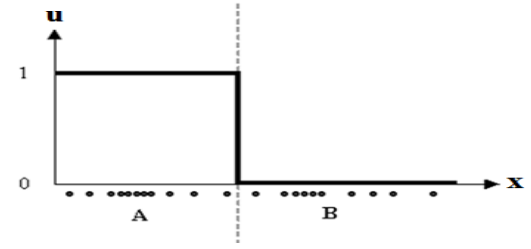
† Corresponding Author : Dept. of Electrical Engineering, The University of Suwon, Korea
E-mail : ohsk@suwon.ac.kr
* Dept. of Electronic Engineering, The University of Suwon, Korea
Received : March 2, 2016; Accepted : April 8, 2016

Radial Basis Function Neural Networks)을 이용한다.[3][4] 구조는 조건부, 결론부 그리고 추론부로 구성되어 있고, 조건부에서 활성함수로 주로 사용되던 가우시안 함수 대신에 증분형 FCM (Incremental Fuzzy C-Means) 클러스터링 알고리즘을 이용하여 데이터 특성을 반영하고[5][6], 방대한 양의 데이터를 순차적으로 처리한다.[7][8] 그리고 결론부에서 다항식은 기존의 상수항을 확장한 1차 선형식(Linear)을 사용한다. 다항식의 계수는 순환최소자승법(RLSE: Recursive Least Square Estimation)을 이용하여 순차적으로 추정한다.[9][10] 순환최소자승법은 데이터를 순차적으로 처리 가능하여 방대한 양의 데이터에 대한 다항식 계수를 추정할 때 효과적이다.

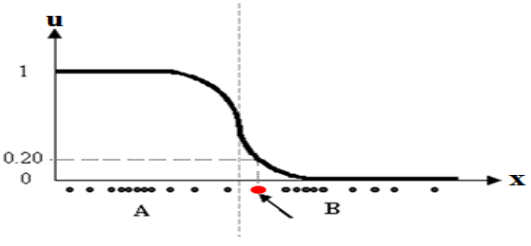
본 논문 2장에서는 데이터를 순차적으로 처리하기 위한 증분형 FCM과 순환최소자승법에 대해 설명한다. 증분형 FCM은 일반적인 FCM과 비교하여 설명하고, 순환최소자승법은 일반적인 최소자승법과 비교하여 설명한다. 3장에서는 방대한 양의 데이터를 효과적으로 학습하기 위한 증분형 FCM 기반 순환 방사형 기저함수 신경회로망의 구조 및 설계 과정에 대해 설명한다. 그리고 4장에서는 Machine Learning 데이터인 Pima, Magic 그리고 Shuttle 데이터를 사용하여 증분형 FCM 기반 순환 방사형 기저함수 신경회로망의 성능을 평가한다. 마지막으로 5장에서는 결론에 대하여 설명한다.

2. 데이터를 순차적으로 처리하기 위한 증분형 FCM 및 순환최소자승법

본 장에서는 데이터를 순차적으로 처리 가능한 증분형 FCM과 일반적인 FCM을 비교 설명하고, 방대한 양의 빅 데이터를 학습하기 위한 순환최소자승법과 일반적인 최소자승법을 비교 설명한다.



(a) Hard C-Means



(b) Fuzzy C-Means

그림 1 HCM과 FCM의 차이점

Fig. 1 Differences between HCM and FCM

2.1 일반적인 FCM과 증분형 FCM

FCM(Fuzzy C-Means) 클러스터링 알고리즘은 하나의 클러스터에 속해져 있는 데이터 점의 소속 정도를 열거한 데이터 분류 알고리즘으로, HCM(Hard C-Means) 클러스터링 알고리즘을 개선하기 위해서 제안되었다. HCM은 0과 1, 이진 논리에 의해서 분리된 데이터가 그룹에 속해 있는지 아닌지 판별한다. 하지만, FCM은 0과 1사이의 소속정도에 의해서 나타난 소속감의 정도를 가지고 주어진 데이터 점이 몇 개의 그룹에 속할 수 있다는 퍼지 분할을 사용한다.

일반적인 FCM은 n개의 입력변수 집합을 c개의 퍼지 그룹들로 분할하고 목적함수가 최소가 되도록 각 클러스터의 중심점을 데이터 전체를 이용하여 찾는 알고리즘이고, 본 논문에서 사용하는 증분형 FCM은 데이터의 일부를 이용하여 초기 중심점을 찾고, 추가적으로 들어오는 데이터에 따라 중심점을 변경하는 알고리즘이다.[8] 증분형 FCM은 데이터 전체를 한번에 처리하는 것이 아닌 추가적으로 들어오는 데이터마다 처리하여 순차적으로 중심점 변경이 가능하다. 또한 제한적인 컴퓨터 메모리를 효율적으로 사용할 수 있기 때문에 방대한 양의 데이터 처리에도 효과적이다.[4] 하지만 데이터를 업데이트 형식으로 사용함에 따라 연산속도는 일반적인 FCM보다 느리다. 반면 일반적인 FCM은 데이터를 한번에 모두 사용하여 Adaptive FCM과 같이 목적함수가 최소가 되도록 파라미터를 조절할 수 있는 기술이 응용될 수 있지만, 데이터 전체를 한번에 사용하기 때문에 순차적으로 중심점 변경이 불가능하고, 방대한 양의 데이터 처리를 위해서는 필요 충분한 컴퓨터 메모리가 필요하다.[12] 식 (1)은 일반적인 FCM의 목적함수이고, 식 (2)는 증분형 FCM의 목적함수다.[5]

$$J(u_{ik}, v_i) = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{\beta} d^2(x_k, v_i) \tag{1}$$

$$J(u_{ik}, v_i) = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{\beta} d^2(x_k, v_i) + \sum_{i=1}^{c} u_{iN+1}^{\beta} d^2(x_{N+1}, v_i) \tag{2}$$

여기서, u_{ik}는 0과 1 사이의 소속정도를 나타내는 값으로 $i(i=1, ..., c)$번째 클러스터에 속해져 있는 x_k의 $k(k=1, ..., n)$번째 데이터의 소속정도를 나타낸다. v는 $i(i=1, ..., c)$번째 클러스터 중심 벡터이다. β는 퍼지화 계수를 나타내며 $\beta \in [1, \infty]$와 같은 범위를 가지고 있다. 또한 d는 유클리디안 거리를 나타낸다. 식 (1)의 목적함수를 최소화하기 위해서 다음과 같이 목적함수를 세분화 시켜야 한다.

$$v_{il}^{(r)} = \frac{\sum_{k=1}^{N} (u_{ik})^{\beta} \cdot x_{kl}}{\sum_{k=1}^{N} (u_{ik})^{\beta}} \tag{3}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)}} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)}} \tag{4}$$

세분화된 식 (3), (4)를 이용하여 $u_{i,N+1}$을 식 (5)를 새롭게 정의 할 수 있고, 최종적으로 증분형 FCM의 중심점 v_{il}^{N+1}은 식 (6)과 같이 새롭게 정의할 수 있다.

$$u_{i,N+1}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_{N+1}-v_j\|}{\|x_{N+1}-v_i\|}\right)^{2/(\beta-1)}} \tag{5}$$

$$v_i^{N+1}=\frac{\sum_{k=1}^{N+1}(u_{ik})^{\beta}\cdot x_{kl}}{\sum_{k=1}^{N+1}(u_{ik})^{\beta}}=\frac{\sum_{k=1}^{N}(u_{ik})^{\beta}\cdot x_{kl}+u_{i,N+1}\cdot x_{kl}}{\sum_{k=1}^{N}(u_{ik})^{\beta}+u_{i,N+1}} \tag{6}$$

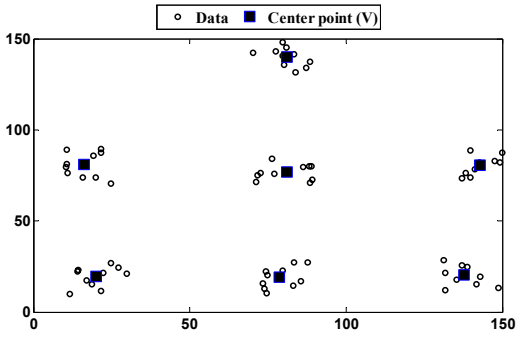
$$=\frac{v_{il}+\frac{u_{i,N+1}}{\sum_{k=1}^{N}u_{ik}}\cdot x_{kl}}{1+\frac{u_{i,N+1}}{\sum_{k=1}^{N}u_{ik}}}$$

그림 2는 일반적인 FCM과 증분형 FCM의 중심점을 비교 분석한 것이다. (a)는 데이터 전체를 한번에 모두 사용하는 일반적인 FCM의 중심점을 나타내고, (b)는 데이터 일부를 사용하여 초기 중심점을 찾고, 순차적으로 새롭게 들어오는 데이터를 이용하여 중심점을 변화시키는 증분형 FCM의 중심점을 나타낸다. 그림으로는 중심점의 차이를 확인할 수 없기 때문에 표 1에서 중심점의 값을 나타내었다.

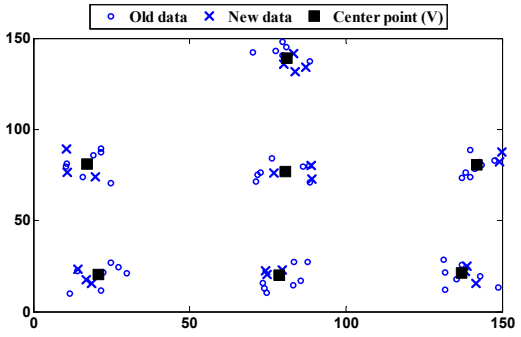
표 1 일반적인 FCM과 증분형 FCM의 중심값
Table 1 Center points of general FCM and incremental FCM

	중심값			
Cluster	일반적인 FCM		증분형 FCM	
1	142.83	80.70	141.77	80.56
2	137.62	20.28	136.91	21.12
3	16.19	80.96	17.10	80.85
4	78.67	19.03	76.60	19.94
5	19.96	19.53	20.74	20.49
6	81.06	140.07	80.97	139.23
7	81.06	76.83	80.63	76.97

표 1과 같이 일반적인 FCM과 증분형 FCM의 중심점을 비교하였을 때 큰 차이가 없는 것을 확인할 수 있다. 중심점에는 큰 차이가 없지만 데이터를 이용하는 방법이 다르다는 차이점이 있다.[7] 증분형 FCM은 데이터 일부분을 이용하여 중심점을 찾고, 이후 데이터가 업데이트 될 때마다 새로운 중심점을 찾기 때문에 일반적인 FCM으로는 불가능한 데이터 순차적 처리를 할 수 있다. 또한 증분형 FCM은 데이터를 한번에 모두 사용하지 않아도 되기 때문에 컴퓨터 메모리 사용에도 효율적이다. 증분형 FCM 알고리즘의 단계는 다음과 같고, 단계에 대한 내용은 3장에서 순환 최소자승법과 함께 자세히 설명한다.[5] [6]



(a) 일반적인 FCM의 중심점



(b) 증분형 FCM의 중심점

그림 2 일반적인 FCM과 증분형 FCM의 중심점 비교
Fig. 2 Comparison of center points of general FCM and incremental FCM

[단계 1] 클러스터의 개수(c)를 정하고 퍼지화 계수(β)를 선택한다. 그리고 초기 소속행렬 $U^{(r)}$을 초기화 한다.

$$U^{(r)}=\left\{u_{ik}\in[0,1],\ \sum_{i=1}^{c}u_{ik}=1\,\forall l, 0<\sum_{k=1}^{N}u_{ik}<n\,\forall i\right\} \tag{7}$$

[단계 2] 식 (3)을 이용하여 클러스터의 중심 v를 계산한다.

[단계 3] 계산된 중심 v와 식 (4)를 이용하여 소속행렬 $U^{(r+1)}$을 계산한다.

[단계 4] 식 (8)을 계산하고, 만약 $\Delta>\varepsilon$이면 $r=r+1$로 정하고 단계 2로 되돌아간다. $\Delta\le\varepsilon$이면 다음 단계로 넘어간다. (여기까지는 일반적인 FCM 알고리즘의 순서와 동일하다.)

[단계 5] N개의 데이터를 이용하여 단계 1 ~ 4를 거쳐 소속행렬을 계산한 후, 순차적으로 들어오는 $N+1$ 번째 데이터의 소속행렬 $u_{i,N+1}$을 식 (5)를 이용하여 계산한다.

[단계 6] 단계 5에서 계산된 새로운 소속행렬 $u_{i,N+1}$와 식 (6)을 이용하여 새로운 중심점을 계산한다. 그리고 단계 3으로 되돌아가서 알고리즘을 반복하고, 새로운 데이터가 없을 때 알고리즘을 종료한다.
(단계 3으로 되돌아가면 그 이후로는 단계 4는 무시한다.)

2.2 최소자승법과 순환최소자승법

최소자승법(LSE : Least Square Estimation)과 순환최소자승법(RLSE : Recursive Least Square Estimation)은 파라미터를 학습하는 방법 중 하나이다.[10] 최소자승법과 순환최소자승법의 가장 큰 차이점은 데이터를 이용하는 방법이 다르다는 것이다. 최소자승법은 데이터 전체를 한번에 모두 사용하여 파라미터를 학습하고, 순환최소자승법은 전체 데이터의 일부를 사용하여 초기 파라미터를 학습한 후, 남은 데이터를 순차적으로 이용하여 파라미터를 업데이트한다. 방대한 양의 데이터를 최소자승법을 이용하여 학습할 경우, 기하급수적으로 커지는 행렬에 의해 컴퓨터 메모리 부족과 같은 문제를 발생시킬 수 있다. 하지만 순환최소자승법을 이용할 경우, 메모리 부족 문제를 해결할 수 있다. 최소자승법은 식 (9)와 같이 정의되고, 순환최소자승법은 다음과 같이 정의된다.

$$\theta_N = (A^T A)^{-1} A^T Y \tag{9}$$

식 (9)에서 θ_N은 N번째 데이터까지를 이용한 파라미터를 나타내고, A와 Y행렬은 식 (10)과 같이 표현된다.

$$A = \begin{bmatrix} u_{11} \cdots u_{c1} \cdots & x_{11}u_{11} \cdots & x_{k1}u_{11} \cdots & x_{k1}u_{c1} \\ \vdots \quad \vdots \quad \vdots & \cdots & \vdots \quad \vdots & \vdots \\ u_{1N} \cdots u_{cN} \cdots & x_{1N}u_{1N} \cdots & x_{kN}u_{1N} \cdots & x_{kN}u_{cN} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{10}$$

여기서, c는 클러스터 수, k는 입력 차원 수를 의미한다. N번째 데이터까지는 최소자승법을 이용하여 초기 파라미터를 학습하고, $N+1$번째 데이터부터는 순환최소자승법을 이용하여 θ를 업데이트 한다. $N+1$번째까지의 파라미터는 다음과 같이 표현 가능하다.[9]

$$\theta_{N+1} = \left(\begin{bmatrix} A \\ a^T \end{bmatrix}^T \begin{bmatrix} A \\ a^T \end{bmatrix} \right)^{-1} \begin{bmatrix} A \\ a^T \end{bmatrix}^T \begin{bmatrix} Y \\ y \end{bmatrix} \tag{11}$$

여기서, $P_N = (A^T A)^{-1}$ 이라고 정의하면, P_{N+1}은 식 (12)와 같이 정의된다.

$$P_{N+1} = \left(\begin{bmatrix} A \\ a^T \end{bmatrix}^T \begin{bmatrix} A \\ a^T \end{bmatrix} \right)^{-1} = \left([A^T \; a] \begin{bmatrix} A \\ a^T \end{bmatrix} \right)^{-1} = (A^T A + aa^T)^{-1} \tag{12}$$

앞서 정의한 P_N과 식 (12)를 통해 식 (13)을 유도할 수 있고, P_{N+1}도 식 (14)와 같이 재정의 할 수 있다.

$$P_N^{-1} = P_{N+1}^{-1} - aa^T \tag{13}$$

$$P_{N+1} = \left(P_N^{-1} + aa^T \right)^{-1} \tag{14}$$

그리고, P_N과 P_{N+1}로 정의된 식을 통해 식 (15)와 같이 θ_N과 θ_{N+1}을 정의 할 수 있다.

$$\theta_N = P_N A^T Y, \quad \theta_{N+1} = P_{N+1}(A^T Y + ay) \tag{15}$$

정의된 θ_N을 통해서 다음 식 (16)을 유도할 수 있고, θ_{N+1}을 재정의 할 수 있다.

$$A^T Y = P_N^{-1} \theta_N \tag{16}$$

$$\begin{aligned} \theta_{N+1} &= P_{N+1}\left(P_N^{-1}\theta_N + aY\right) = P_{N+1}\left[\left(P_{N+1}^{-1} - aa^T\right)\theta_N + aY\right] \\ &= \theta_N + P_{N+1}a\left(Y - a^T\theta_N\right) \end{aligned} \tag{17}$$

식 (14)에서 정의된 P_{N+1}을 행렬의 역변환 이론을 적용하면 식 (18)과 같이 표현 할 수 있다.

$$P_{N+1} = P_N - P_N a(I + a^T P_N a)^{-1} a^T P_N = P_N - \frac{P_N aa^T P_N}{1 + a^T P_N a} \tag{18}$$

최종적으로 식 (19), (20)을 통해 업데이트 되는 θ_{N+1}을 정의 할 수 있다.

$$P_{N+1} = P_N - \frac{P_N a_{N+1} a_{N+1}^T P_N}{1 + a_{N+1}^T P_N a_{N+1}} \tag{19}$$

$$\theta_{N+1} = \theta_N + P_{N+1} a_{N+1}\left(y_{N+1} - a_{N+1}^T \theta_N\right) \tag{20}$$

그림 3은 순환최소자승법을 이용하여 파라미터를 업데이트하는 과정을 나타낸다. 전체 데이터 중 m번째 데이터까지는 최소자승법을 이용하여 초기 파라미터를 추정하고, 추가적으로 들어오는 데이터인 $m+1$번째부터는 순환최소자승법을 이용하여 파라미터를 업데이트한다.

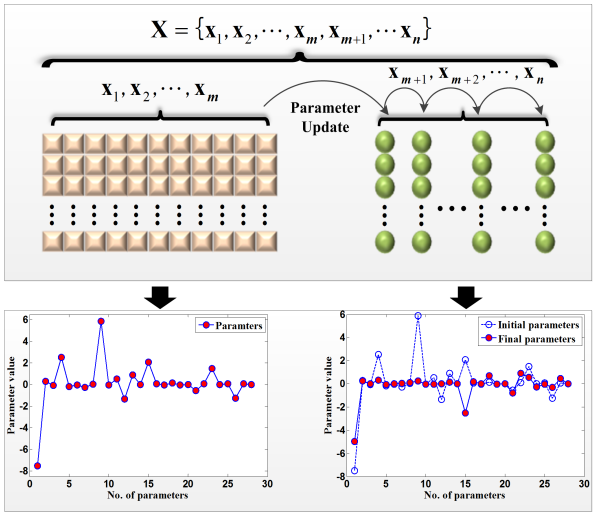


그림 3 파라미터 업데이트 과정

Fig. 3 Process of parameters update

3. 빅 데이터 학습을 위한 증분형 FCM 기반 순환 RBFNN 패턴 분류기 설계

본 장에서는 조건부, 결론부, 추론부 세 가지의 기능적 모듈로 네트워크 구조를 가지는 일반적인 RBFNN 패턴 분류기에 대해 설명하고, 방대한 양의 데이터를 효율적으로 학습하기 위한 증분형 FCM 기반 순환 RBFNN 패턴 분류기에 대해 설명한다.

3.1 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 구조

조건부, 결론부, 추론부 세 가지의 기능적 모듈로 네트워크 구조를 가지는 일반적인 RBFNN(Radial Basis Function Neural Networks)은 Multi-dimension 문제해결, 강인한 네트워크 특성, 예측 능력이 우수하다고 알려져 있다.[1][2] 입력 데이터는 조건부의 각 노드에 연결되고, 조건부의 출력과 결론부에서 구한 연결가중치를 이용하여 추론부에서 최종 출력을 구한다.[3] 조건부의 활성함수로는 일반적으로 가우시안 함수가 사용되며, 결론부의 연결가중치는 상수항으로 정의되고 최소자승법(LSE)을 통해 구한다. 본 논문에서는 방대한 양의 데이터를 순차적으로 처리 및 학습하기 위해 일반적인 RBFNN을 확장하여 증분형 FCM 기반 순환 RBFNN 패턴 분류기를 설계한다. 그림 4는 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 구조를 나타낸다. 구조는 조건부, 결론부, 추론부로 일반적인 RBFNN과 동일하다. 하지만, 조건부에서 활성함수로 가우시안 함수 대신에 증분형 FCM 클러스터링 알고리즘을 사용하고, 결론부의 연결가중치는 상수항을 확장하여 식 (21)~(23)과 같이 다항식 형태로 구성되고 파라미터는 순환최소자승법 통해 추정된다. 조건부와 결론부에서 사용되는 증분형 FCM과 순환최소자승법에 관한 내용은 아래에서 상세히 다룬다.

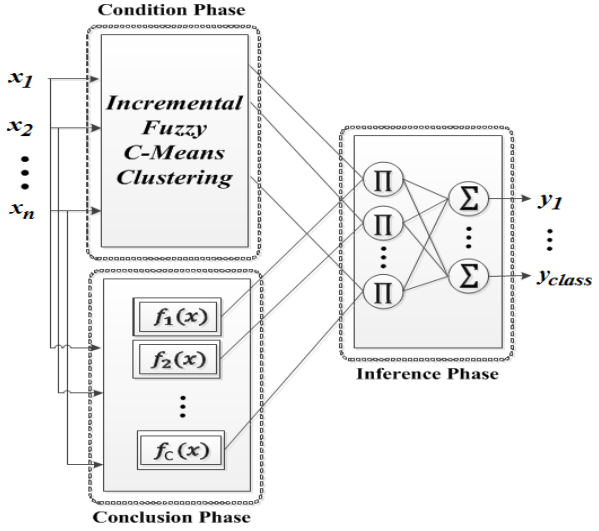


그림 4 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 구조
Fig. 4 Architecture of incremental FCM-based Recursive RBFNN pattern classifier

Type 1 : 상수항(Constant)

$$f_j(x_1,\cdots,x_k)=a_{j0} \tag{21}$$

Type 2 : 1차 선형식(Linear)

$$f_j(x_1,\cdots,x_k)=a_{j0}+\sum_{i=1}^{k}a_{ji}x_i \tag{22}$$

Type 3 : 2차 선형식(Quadratic)

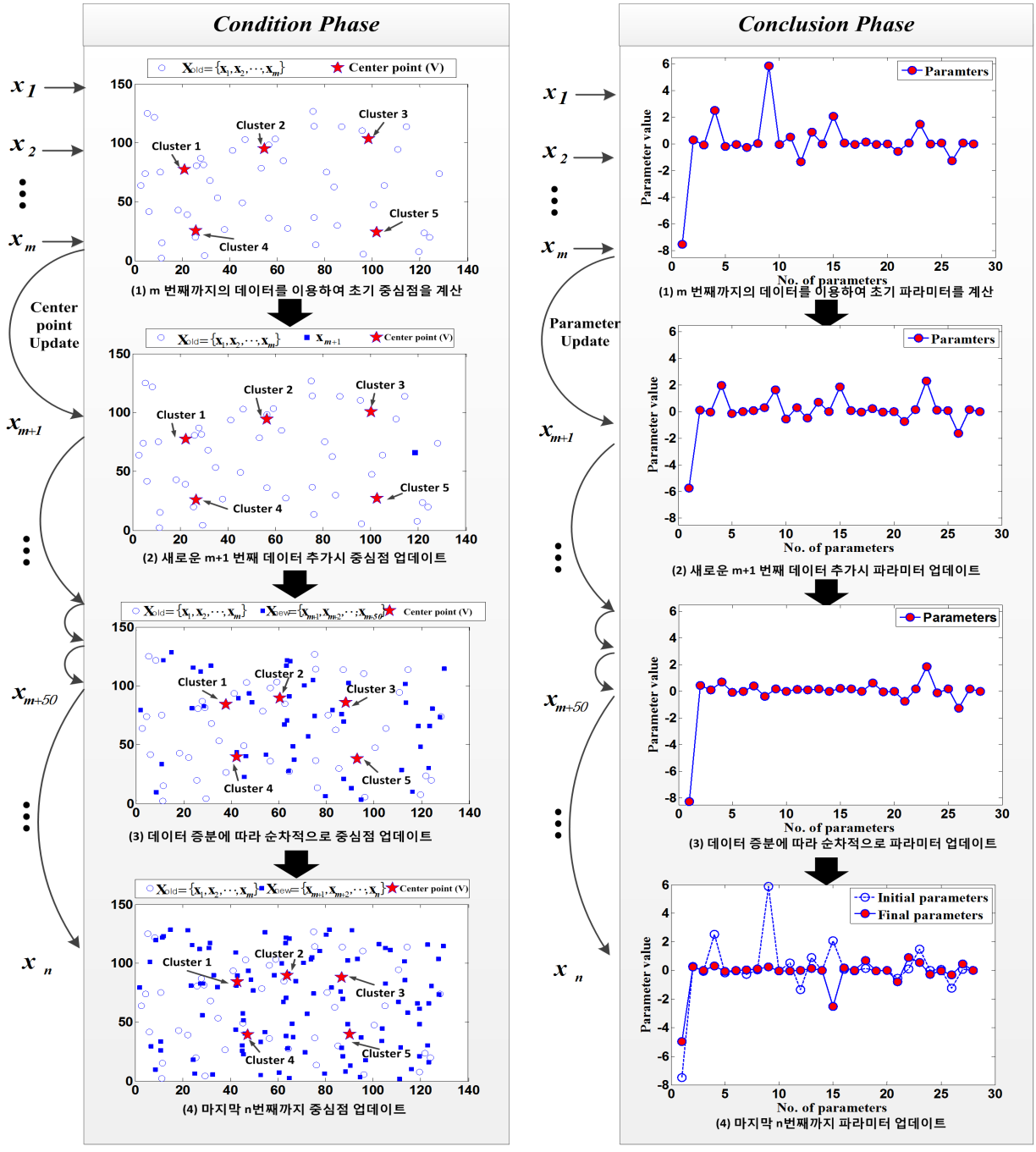
$$f_j(x_1,\cdots,x_k)=a_{j0}+\sum_{i=1}^{k}a_{ji}x_i+\sum_{i=1}^{k}a_{j(k+i)}x_i^2 +a_{(2k+1)}x_1x_2+\cdots+a_{(k(k+3)/2)}x_{(k-1)}x_k \tag{23}$$

증분형 FCM 기반 순환 RBFNN 패턴 분류기는 일반적인 RBFNN 패턴 분류기의 구조와 동일하여 다차원 문제해결, 강인한 네트워크 특성, 예측 능력이 우수하다는 장점을 갖고 있다. 또한 조건부에서 증분형 FCM 클러스터링 알고리즘을 사용하여 입력 데이터의 특성을 반영할 수 있고, 데이터를 순차적으로 처리가 가능하다. 그리고 다항식 형태로 확장된 결론부의 연결가중치를 m번째 데이터까지는 최소자승법(LSE)로 구하고, $m+1$번째부터 N번째까지는 순차적으로 순환최소자승법(RLSE)을 사용하여 파리미터를 업데이트하는 방식으로 방대한 양의 데이터를 효율적으로 학습이 가능하다.[4][9] 이와 같이 데이터를 순차적으로 처리함으로써 컴퓨터 메모리도 효율적으로 사용할 수 있다. 일반적으로 컴퓨터 메모리 사용량은 식 (24)와 같이 데이터 수에 비례하는데 증분형 FCM 기반 순환 RBFNN은 데이터 전체를 한번에 학습하는 것이 아닌 데이터 전체의 일부분을 이용하여 초기 학습하고 추가적으로 들어오는 데이터를 순차적으로 학습하기 때문에 컴퓨터 메모리 아웃과 같은 문제를 해결할 수 있다.

$$Size\,of\,memory \propto No.\,of\,data \tag{24}$$

3.2 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 설계 과정

방대한 양의 데이터를 순차적으로 처리 및 효과적으로 학습하기 위한 증분형 FCM 기반 순환 RBFNN 패턴 분류기는 전체 데이터에서 일부분만을 사용하여 일반적인 FCM과 최소자승법을 통해 초기 파라미터를 추정한다. 이후 추가적으로 들어오는 데이터를 순차적으로 학습하기 위해 조건부에서 다양한 FCM 방법 중, 증분형 FCM을 이용하고, 결론부에서는 조건부와 동일하게 순차적으로 파라미터를 업데이트하기 위해 순환최소자승법을 이용한다.[8][10] 그림 5는 초기 중심점 및 파라미터를 추정한 후, 추가적인 데이터를 순차적으로 처리함에 따라 변화되는 중심점과 파라미터를 나타낸다. 중심점의 변화는 조건부에서 수행하게 되고, 파라미터 변화는 결론부에서 수행한다. 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 설계 과정을 단계별로 나타내면 다음과 같다.



(a) Incremental FCM 중심점 변화 (b) Recursive LSE를 이용한 파라미터 변화

그림 5 데이터 증분에 대한 중심점 변화 및 파라미터 변화

Fig. 5 Variation of center points and parameters for incremental data

[단계 1] 초기 학습 데이터 설정 및 초기 중심점 계산

[1-1] 학습 데이터($X = x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_n$) 중 $x_1, x_2, \cdots, x_m$의 데이터를 초기 학습 데이터로 설정하고, $x_{m+1}, x_{m+2}, \cdots, x_n$의 데이터를 추가 학습 데이터로 설정한다.

[단계 2] 증분형 FCM을 이용하여 순차적으로 중심점 업데이트

[2-1] 초기 학습 데이터($x_1, x_2, \cdots, x_m$)에 대한 초기 중심점을 일반적인 FCM을 통해 계산한다.

[2-2] 초기 중심점과 추가 학습 데이터인 x_{m+1} 데이터에 증분형 FCM을 적용하여 중심점을 업데이트 한다.

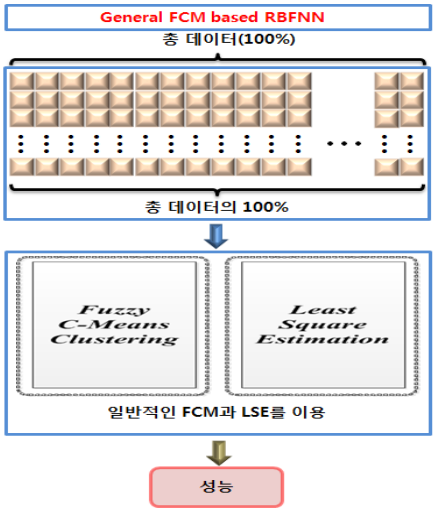
[2-3] 업데이트된 중심점과 x_{m+2} 데이터를 [2-2]을 이용하여 중심점을 업데이트 한다.

[2-4] 나머지 추가 데이터를 순차적으로 단계 4를 이용하여 중심점을 업데이트 한다.

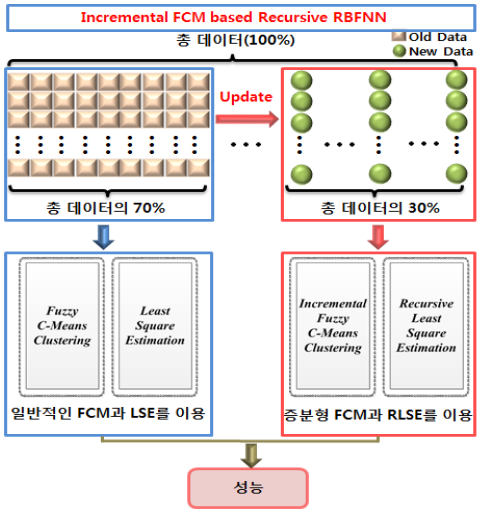
[2-5] 최종적으로 업데이트된 중심점을 이용하여 초기 학습 데이터에 대한 적합도를 계산한다.

[단계 3] 초기 파라미터 추정

[3-1] 초기 학습데이터에 대한 적합도와 최소자승법을 이용하여 초기 파라미터를 추정한다.



(a) 일반적인 FCM 기반 RBFNN 패턴 분류기 실험



(b) 증분형 FCM 기반 순환 RBFNN 패턴 분류기 실험

그림 6 실험의 전체 개요

Fig. 6 Outline of experiment

[단계 4] 순환최소자승법을 이용하여 순차적으로 파라미터 업데이트

[4-1] 추가 학습 데이터인 x_{m+1}에 대한 적합도를 계산하고, 적합도와 순환최소자승법을 통해 파라미터를 업데이트 한다.

[4-2] [4-1]을 이용하여 x_{m+2}부터 x_n까지 순차적으로 파라미터를 업데이트 한다.

[단계 5] 최종적으로 추정된 파라미터를 이용하여 패턴 분류기의 출력을 계산한다.

4. 실험 및 결과고찰

4.1 실험의 전체 개요

방대한 양의 데이터를 순차적으로 처리 및 학습하기 위한 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 성능 평가를 위해 Machine Learning 데이터(Pima, Magic, Shuttle DB)를 이용하고, 효과적인 학습을 위해 전체 데이터 중 70%는 일반적인 FCM 기반 RBFNN 패턴 분류기와 동일하게 하여 처리하고, 나머지 30%를 순차적으로 처리하는 방법을 이용한다. 또한, 전체 데이터를 한번에 모두 사용하는 일반적인 FCM 기반 RBFNN 패턴 분류기의 성능과 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 성능을 비교한다.

4.2 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 파라미터 설정

표 2는 본 실험에 사용된 데이터 및 파라미터 설정을 나타낸다. 데이터는 Pima, Magic, Shuttle DB를 사용하였고, Pima DB의 데이터 수는 768개, 입력 변수는 8개로 구성되어 있다. 그리고 Magic DB의 데이터 수는 19,020개, 입력 변수는 10개이고, Shuttle DB의 데이터 수는 58,000개, 입력 변수는 9개로 구성되어 있다. 분류기의 파라미터로는 퍼지화 계수, 다항식 형태, 규칙 수가 있다. 퍼지화 계수는 일반적으로 정의되어 있는 2.0으로 설정하였고, 다항식 형태는 1차 선형식으로 설정하였다. 또한, 객관적인 성능평가를 위해 K-fold cross validation을 이용하였다.

표 2 실험에 사용된 데이터

Table 2 Data used in experiment

Data information			
Data	Pima	Magic	Shuttle
No. of data	768	19,020	58,000
No. of inputs	8	10	9
No. of classes	2	2	7

4.3 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 성능 평가

증분형 FCM 기반 순환 RBFNN 패턴 분류기의 성능 평가를 위해, 식 (25)와 같이 패턴 분류율(Pattern Classification Rate)을

표 3 파라미터 설정
Table 3 Setting of parameters

Parameters	Values
Fuzzification coefficient	20
Polynomial type	Linear
No. of rules	Pima DB : [2 4 8 10] Magic DB : [4 8 10 15] Shuttle DB : [4 8 10 15]
K-fold cross validation	5

이용하여 나타낸다. 여기서, N은 총 데이터 수를 나타내고, $False$는 패턴 분류에 실패한 개수를 의미한다. 또한, 제안된 분류기에 필요한 메모리의 크기를 나타내기 위해 Big O 표기법을 이용한다.

$$PCR[\%] = \left\{1 - \left(\frac{False}{N}\right)\right\} \times 100 \tag{25}$$

표 4는 일반적인 FCM과 본 논문에서 사용한 증분형 FCM의 시간 및 공간 복잡도를 나타낸 표이고, 표 5는 일반적인 LSE와 본 논문에서 사용한 Recursive LSE의 시간 및 공간 복잡도를 나타낸 표이다.[8]

표 4 FCM의 시간 및 공간 복잡도
Table 4 Time and space complexity of FCM

	Time	Space
General FCM	$O(t \times c \times d \times n)$	$O((d+c) \times n)$
Incremental FCM	$O(t \times c \times d \times n)$	$O((d+c) \times (n \times r)), ※ r < 1$

여기서, t는 반복횟수, c는 클러스터 수(규칙 수), d는 입력의 차원 수, n은 데이터 수를 의미한다. 그리고 r은 초기 학습 데이터의 비율을 의미한다. 표 4와 같이 제안된 모델의 전반부에 사용되는 FCM에 대한 시간 및 공간 복잡도를 보았을 때, 일반적인 FCM과 증분형 FCM의 시간적인 부분은 동일하다[8]. 본 논문에서 사용한 증분형 FCM이 일반적인 FCM 보다 반복횟수가 증가하여 실제 실험에서 시간적인 부분이 다르게 적용될 수 있으나, 이는 FCM의 종료조건에 따라 변경될 수 있기 때문에 큰 의미가 없다[8]. 공간복잡도를 보면 초기 학습 데이터의 비율에 따라 증분형 FCM이 일반적인 FCM보다 작아지는 것을 확인할 수 있다. 결론적으로, 본 논문에서 사용한 증분형 FCM이 데이터의 양이 방대해질수록 효과적으로 데이터를 처리할 수 있다는 의미가 된다.

표 5 LSE의 시간 및 공간 복잡도
Table 5 Time and space complexity of LSE

	Time	Space
General LSE	$O(p^2 \times (n+p))$	$O(p^2 \times n^2)$
Recursive LSE	$O(p^2 \times (n+p))$	$O(p^2 \times (n \times r)^2), ※ r < 1$

표 5는 후반부 연결가중치 학습에 사용되는 LSE에 대한 시간 및 공간 복잡도를 나타낸다.[11] 여기서 n은 데이터 수를 나타내고, p는 파라미터 수를 나타낸다. 일반적인 LSE와 Recursive LSE의 시간 복잡도는 차이가 없다. 하지만 초기 학습 데이터 비율을 나타내는 r값에 따라 공간 복잡도를 줄일 수 있다. 즉, 데이터를 순차적으로 학습하는 제안된 분류기는 전체 데이터를 한번에 학습하는 분류기에 비해 방대한 양의 데이터를 효과적으로 학습할 수 있다.

표 6 Pima 데이터의 실험결과
Table 6 Results for the experiment of Pima dataset

Pattern Classification Rate (%)		
	General FCM based RBFNN	Incremental FCM based RBFNN
No. of rules	Training	Testing
2	77.94 ± 0.44	76.75 ± 2.31
4	78.50 ± 0.42	75.32 ± 3.18
8	80.46 ± 1.11	78.57 ± 0.69
10	81.50 ± 0.79	75.19 ± 3.48

표 6은 데이터 전체를 한번에 학습하는 일반적인 FCM 기반 RBFNN과 데이터를 순차적으로 처리 및 학습이 가능한 증분형 FCM 기반 RBFNN 패턴 분류기의 성능을 나타낸 표이다. 실험에 사용된 Pima DB는 학습데이터 614개, 테스트데이터 154개로 구성하여 실험하였다. 두 개의 패턴 분류기 성능을 비교하였을 때 큰 차이가 없는 것을 확인할 수 있다. 하지만 앞서 나타낸 표 4와 5를 통해 데이터를 순차적으로 처리 및 학습하는 제안된 분류기는 데이터 전체를 한번에 이용하는 분류기에 비해 공간 복잡도가 적다는 것을 알 수 있다.

표 7 Magic 데이터의 실험결과
Table 7 Result for the experiment of Magic dataset

Pattern Classification Rate (%)		
	General FCM based RBFNN	Incremental FCM based RBFNN
No. of rules	Training	Testing
4	82.18 ± 0.06	81.86 ± 0.41
8	82.69 ± 0.13	82.31 ± 0.35
10	83.17 ± 0.16	83.34 ± 0.19
15	83.34 ± 0.10	82.76 ± 0.29

표 7은 Magic DB를 이용하여 일반적인 FCM 기반 RBFNN과 증분형 FCM 기반 RBFNN 패턴 분류기의 성능을 나타낸 표이다. Magic DB의 데이터 수는 19,020개로 학습 데이터 15,216개 테스트 데이터 3,804개로 구성하여 실험하였다. Magic DB의 경우, 증분형 FCM 기반 RBFNN 패턴 분류기의 성능이 조금 더 우수하다. 또한 앞서 나타낸 표 4와 5를 통해 규칙 수가 증가할수록 제안된 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 공간 복잡도가 일반적인 FCM 기반 RBFNN에 비해 줄어든다. 즉, 데이터의 수가 많아지고 규칙수가 증가할수록 제안된 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 효율성이 나타난다.

표 8 Shuttle 데이터의 실험결과
Table 8 Results for the experiment of Shuttle dataset

Pattern Classification Rate (%)		
	General FCM based RBFNN	Incremental FCM based RBFNN
No. of rules	Training	Testing
4	95.68 ± 0.06	94.71 ± 0.23
8	97.32 ± 0.04	97.30 ± 0.16
10	97.49 ± 0.07	97.49 ± 0.05
15	98.26 ± 0.09	97.43 ± 0.21

표 8은 실험에 사용된 데이터 중 가장 많은 양의 데이터인 Shuttle DB를 사용하여 일반적인 FCM 기반 RBFNN과 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 성능을 나타낸 표이다. Shuttle DB의 데이터 수는 58,000개로 학습 데이터 46,400개 테스트 데이터 11,600개로 구성하여 실험하였다. Pima와 Magic DB를 이용한 실험 결과에서 확인할 수 있듯이 Shuttle DB를 이용한 실험에서도 두 개의 패턴 분류기 성능은 큰 차이가 없다. 하지만 다른 실험과 마찬가지로 규칙 수가 증가함에 따라 제안된 증분형 FCM 기반 순환 RBFNN 패턴 분류기의 공간복잡도는 일반적인 FCM 기반 RBFNN 패턴 분류기에 비해 크게 줄어들 수 있다. 이와 같은 결과로 볼 때 제안된 증분형 FCM 기반 순환 RBFNN 패턴 분류기는 방대한 양의 데이터를 학습 할 경우, 데이터를 한번에 모두 이용하는 방법과 비교하였을 때 성능의 저하 없이 공간 복잡도를 크게 줄일 수 있다.

5. 결 론

본 논문에서는 방대한 양의 데이터를 순차적으로 처리 및 효과적인 학습을 위해 증분형 FCM 기반 순환 RBFNN 패턴 분류기를 설계하였다. 성능 평가를 위해 Machine Learning 데이터인 Pima, Magic, Shuttle 데이터를 이용하였고, 객관적인 평가를 위해 5 fold cross validation을 이용하였다. 또한 성능 지수는 데이터 전체를 한번에 이용하는 일반적인 FCM 기반 RBFNN 패턴 분류기와 비교하였고, Big O 표기법을 통해 시간 및 공간 복잡도를 비교하였다. 실험을 통해 증분형 FCM 기반 순환 RBFNN 패턴 분류기와 일반적인 FCM 기반 RBFNN 패턴 분류기의 성능에 큰 차이가 없는 것을 확인할 수 있었다. 성능차이가 없음에도 불구하고, 제안된 분류기의 공간 복잡도를 줄여 방대한 양의 데이터를 처리 및 학습하는데 효과적이라는 것을 확인할 수 있었다. 실험에서는 전체 학습 데이터의 30%를 순차적으로 처리하였으나 그 이상을 순차적으로 처리할 경우, 그 효과는 더욱 더 높아질 것이라 판단된다. 하지만, 초기 학습 데이터의 비율에 따라 오버피팅 현상이 일어날 수 있으므로 초기 학습 데이터 비율을 결정하는 기준을 찾기 위한 연구가 필요하다. 향후, 방대한 양의 데이터를 한 개씩 순차적으로 처리하는 방법이 아닌 몇 개의 데이터를 군집화하여 순차적으로 처리하는 방법을 연구할 예정이고, 증분형 학습을 통해 발생할 수 있는 오버피팅 문제를 해결하기 위한 방법도 연구할 예정이다.

감사의 글

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning [grant number: NRF-2015R1A2A1A15055365] and also supported by the GRRC program of Gyeonggi province [grant number: GRRC Suwon 2015-B2, Centre for U-city Security & Surveillance Technology].

References

[1] S. K. Oh, W. Pedrycz, B. J. Park, "Polynomial-based Radial Basis Function Neural Net works realized with the Aid of Particle Swarm Optimization," Fuzzy Sets and Systems, Vol. 163, pp. 54-77, 2011.

[2] W. D. Kim, D. J. Lee, and S. K. Oh, "Structural Design of FCM-based Fuzzy Inference System", CICS, pp. 63-64, 2009.

[3] W. D. Kim, S. K. Oh, H. K. Kim, "Structural Design of FCM-based Fuzzy Inference System : A Comparative Study of WLSE and LSE", The Transactions of the Korean Institute of Electrical Engineers, Vol. 59, No. 5, pp. 981-989, 2010.

[4] Yangtao Wang, Lihui Chen, Jian-Ping Mei, "Incremental Fuzzy Clustering With Multiple Medoids for Large Data", IEEE TRANSACTIONS ON FUZZY SYSTEM, Vol. 22, No. 6, pp. 1557-1568, 2014.

[5] Richard J. Hathaway, James C. Bezdek, "Extending fuzzy and probabilistic clustering to very large data sets", Computational Statistics & Data Analysis, Vol. 51, pp. 215-234. 2006.

[6] Junjie Wu, Hui Xiong, Chen Liu, Jian Chen, "A Generalization of Distance Functions for Fuzzy c-Means Clustering With Centroids of Arithmetic Means", IEEE TRANSACTIONS ON FUZZY SYSTEM, Vol. 20, No. 3, pp. 557-571. 2012.

[7] Jonathon K. Parker, Lawrence O. Hall, "Acceleration Fuzzy-C Means Using an Estimated Subsample Size", IEEE TRANSACTIONS ON FUZZY SYSTEM, Vol. 22, No. 5, pp. 1229-1244. 2014.

[8] Yangtao Wang, Lihui Chen, Jian-Ping Mei, "Incremental Fuzzy Clustering With Multiple Medoids for Large Data", IEEE TRANSACTIONS ON FUZZY SYSTEM, Vol. 22, No. 6, pp. 1557-1568, 2014.

[9] Feiyan Chen, Feng Ding, "The filtering based maximum likelihood recursive least squares estimation for multiple-input single-output systems", Applied Mathematical

Modelling, 2015.
[10] Cheng Wang, Tao Tang, “Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems", Applied Mathematics Letters, Vol. 29, pp. 36-41. 2014.
[11] Henry Cohn, Robert Kleinberg, Balazs Szegendy and Chris Umans, “Group-theoretic Algorithms for Matrix Multiplication”, IEEE Computer Society, pp. 379-388, 2005.
[12] Srisuda Aphaipanan, Yuttana Kidjaidure “Action Recognition with Adaptive RBFNN”, IEEE Information and Communication Technology, pp. 1-5, 2014.

저 자 소 개

이 승 철 (Seung-Cheol Lee)

2014년 : 수원대학교 전기공학과 졸업
2014년~현재 : 동 대학원 석사과정
관심분야 : 뉴럴 네트워크, 퍼지 추론 시스템, 패턴 분류
Phone : +82-31-222-6544
E-mail : lsc225@suwon.ac.kr

오 성 권 (Sung-Kwun Oh)

1981년 : 연세대학교 전기공학과 공학사
1983년~1989년 : 금성산전연구소(선임연구원)
1993년 : 연세대학교 전기공학과 공학박사
1996년~1997년 : 캐나다 Manitoba 대학 전기 및 컴퓨터 공학과 Post-Doc
1993년~2004년 : 원광대학교 전기전자 및 정보공학부 교수
2005년~현재 : 수원대학교 전기공학과 교수
2002년~현재 : 대한전기학회, 퍼지 및 지능시스템학회 편집위원
2013년~현재 : Information Sciences 편집위원
관심분야 : 퍼지 시스템, 퍼지-뉴럴 네트워크, 자동화 시스템, 고급 Computational Intelligence, 지능제어 등.
Phone : +82-31-229-8162
E-mail : ohsk@suwon.ac.kr