ISSN 1975-8359(Print) / ISSN 2287-4364(Online)
The Transactions of the Korean Institute of Electrical Engineers Vol. 64, No. 1, pp. 113~120, 2015
http://dx.doi.org/10.5370/KIEE.2015.64.1.113

Stacked Autoencoder를 이용한 특징 추출 기반 Fuzzy k-Nearest Neighbors 패턴 분류기 설계

Design of Fuzzy k-Nearest Neighbors Classifiers based on Feature Extraction by using Stacked Autoencoder

노 석 범* · 오 성 권†
(Suck-Bum Rho · Sung-Kwun Oh)

Abstract - In this paper, we propose a feature extraction method using the stacked autoencoders which consist of restricted Boltzmann machines. The stacked autoencoders is a sort of deep networks. Restricted Boltzmann machines (RBMs) are probabilistic graphical models that can be interpreted as stochastic neural networks. In terms of pattern classification problem, the feature extraction is a key issue. We use the stacked autoencoders networks to extract new features which have a good influence on the improvement of the classification performance. After feature extraction, fuzzy k-nearest neighbors algorithm is used for a classifier which classifies the new extracted data set. To evaluate the classification ability of the proposed pattern classifier, we make some experiments with several machine learning data sets.

Key Words : Stacked Autoencoders, Boltzmann Machine, Restrict Boltzmann Machine, Deep Networks, Fuzzy C-Means Clustering, Fuzzy k-Nearest Neighbors

1. 서 론

현대 사회는 스마트 폰과 태블릿 PC와 같은 다양한 모바일 기기의 사용 빈도가 폭발적으로 증가하고 있다. 이러한 모바일 기기의 폭발적인 사용증가로 인하여 개인 만들어 내고 있는 데이터의 종류와 양도 엄청나게 증가하고 있으며, 데이터의 종류 또한 다양해지고 있다. 개인이 만들어 내는 데이터의 종류와 양의 증가는 이를 이윤 창출에 이용하고자 하는 다양한 형태의 기업에 의해 분석되어 지고 분류되어져서 소비자들의 요구 사항을 파악하는 하나의 도구로 사용되어지고 있다. 이와 같은 데이터의 분석 및 분류는 기업뿐만 아니라 여러 분야에서 이용되어지고 있다. 특히 패턴 분류기는 정보생성과정을 거쳐 엄청난 양의 데이터 들이 발생하는 현시대에 의료, 경제, 고장 진단과 같은 여러 유용한 방면에 사용되어지고 있다.

다양한 종류의 데이터들의 패턴을 분석하여 분류하고자 하는 패턴 분류 기법은 많은 접근 방법으로 시도되고 있다. 이러한 패턴 분류 기법은 크게 통계적 기법과 뉴럴 네트워크를 이용한 방법, 규칙 기반 분류 기법으로 분류 되어질 수 있다[1].

이러한 다양한 데이터 패턴 분류 기법들 중에서 특히 Multi-layered Neural Networks가 영상 데이터와 음성 데이터 처리 및 분석 분야에 특히 많이 적용되어지고 있다[2]. 일반적인 Multi-layered Neural Networks은 네트워크의 층수가 증가되거나 혹은 한 층의 노드수가 증가하게 되면 네트워크의 일반화 성능이 떨어진다고 알려져 있다. 이를 극복하기 위한 목적으로 Deep Learning 구조 와 알고리즘이 최근 제안되었다[3]. 최근 연구되고 있는 Deep Learning은 인지와 의사 결정 (Decision making)과 같은 분야에 필요한 높은 수준의 추상화 성능을 보이는 아주 복잡한 함수의 학습에 우수한 성능을 보이고 있다[4, 5].

Deep Learning Networks는 다양한 형태의 노드들로 구성된 네트워크이며 이러한 네트워크를 구성하게 되는 대표적인 노드들의 형태는 Restricted Boltzmann Machine (RBM)이 있다[5].

본 논문에서는 RBM을 기본 노드로 하는 Autoencoder를 차곡차곡 쌓아 올려 Deep Neural Networks 형태를 이루는 Stacked Autoencoder를 이용하여 차원이 높은 데이터의 차원을 축소하여 제안된 패턴 분류기의 성능을 개선한다. 일반적으로 데이터의 차원이 높으면 패턴 분류기의 성능이 좋지 않게 되는 경향을 보이면, 또한 패턴 분류 과정에서 과도한 계산 량을 발생하게 만드는 원인으로 작용하게 된다.

Autoecoder는 입력 데이터와 출력 데이터를 최대한 유사하게 만드는 기능을 수행한다. Autoencoder는 1980년대 Hinton교수와 PDP 그룹에 의해 제안되어졌다[6]. 최근 들어 Autoencoder는 Deep Architecture 분야에서 다시 각광을 받고 있으며, 이 분야에서 사용되는 Autoencoder는 특별히 RBM들로 구성되어 진다. Autoencoder에 사용되는 RBM은 층층이 쌓여진 구조를 이루며, 학

† Corresponding Author : Dept. of Electrical Engineering, University of Suwon, Korea
E-mail : ohsk@suwon.ac.kr
* WI-A Corporation R&D Center.
Received : December 12, 2014; Accepted : December 21, 2014

습 시에는 아래층으로부터 위층으로 비교사 학습 (unsupervised learning) 형태로 학습이 진행 된다. RBM에 적용된 비교사 학습 (unsupervised learning) 은 네트워크의 초기 상태를 결정하고, 초기화가 끝난 네트워크는 위층으로부터 아래층으로 교사 학습을 통해 최종 학습된다. 이와 같이 최종 학습된 Autoencoder의 층수와 각 층의 노드 수를 조절하여 다 차원 문제를 설계자가 원하는 차원을 가진 입력 변수들로 특징 추출 할 수 있다. 위와 같이 Autoencoder를 통해 차원 축소 된 입력 공간에 위치하는 입력 데이터 집합을 Fuzzy k-Nearest Neighbor (FkNN) classifier를 이용하여 패턴 분류한다.

패턴 분류기로 사용되는 FkNN은 Bezdek이 제안한 Fuzzy C-Means Clustering[7] 기법을 이용하여 k개의 Nearest Neighbor들의 하중값을 결정하게 된다. 본 연구에서 제안된 패턴 분류기의 패턴 분류 성능을 비교, 평가하기 위하여 데이터의 차원이 매우 높은 MNIST 데이터 집합을 이용하여 고차원 문제에서 제안된 패턴 분류기의 성능을 평가한다.

2. Restricted Boltzmann Machine

일반적인 Boltzmann machines (BMs)은 학률 과정 유니트 (Stochastic Processing Unit)를 기본 노드로 가지며, 각 노드들이 양방향성 네트워크를 이루는 네트워크이다. 이와 같은 Boltzmann Machine 네트워크는 뉴럴 네트워크 모델로 이해되고 있다. 일반적으로 볼츠만 머신은 알려져 있지 않은 확률 분포를 통해 얻어진 샘플 데이터들을 이용하여 확률 분포의 중요한 측면을 추정하기 위하여 사용되어진다. 그러나 볼츠만 머신을 이용한 확률 분포 추정은 매우 어렵고 시간이 많이 소모된다. 이러한 문제점을 극복하기 위하여 네트워크의 구조를 제한한 Restricted Boltzmann Machine이 제안 되었다[8].

RBM 네트워크의 구조는 그림 1과 같다.

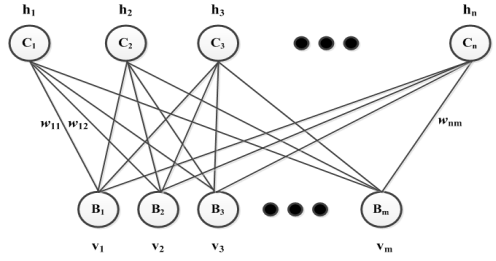

그림 1 Restricted Boltzmann Machine 네트워크의 구조
Fig. 1 Structure of Restricted Boltzmann Machine Network

그림 1에 보인 BRM 네트워크는 m개의 visible 노드 $\boldsymbol{V}=(V_1,\ V_2,\ \cdots,\ V_m)$와 n개의 hidden 노드 $\boldsymbol{H}=(H_1,\ H_2,\ \cdots,H_n)$로 구성되어졌다. 가장 간단한 이진 RBM에서, 확률 변수 $(\boldsymbol{V},\boldsymbol{H})$는 $(\boldsymbol{v},\boldsymbol{h})\in\{0,1\}^{m+n}$을 가지고, 결합 확률 분포(Joint Probability Distribution)로는 (1)로 정의되는 Gibbs 분포를 따른다.

$$p(\boldsymbol{v},\boldsymbol{h})=\frac{1}{z}e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{1}$$

여기서, $z=\sum_{\boldsymbol{v},\boldsymbol{h}}e^{-E(\boldsymbol{v},\boldsymbol{h})}$

$$E(\boldsymbol{v},\boldsymbol{h})=-\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}h_iv_j-\sum_{j=1}^{m}b_jv_j-\sum_{i=1}^{n}c_ih_i \tag{2}$$

그림 1에서 보인 바와 같이 노드 사이의 연결은 가시층(visible layer)과 은닉층(hidden layer) 사이에만 존재하며, 단일 층에서는 노드간 연결이 존재하지 않는다. 이와 같은 그래프의 구조에 따라 가시층의 확률 변수의 상태가 주어진 은닉층의 확률 변수의 조건부 확률은 서로 독립이다. 마찬가지로 은닉층의 확률 변수의 상태가 주어진 가시층의 확률 변수의 조건부 확률 또한 서로 독립이다.

이와 같은 조건부 확률로 인하여 가시층의 확률 변수 $\boldsymbol{v}$의 주변 분포 (marginal distribution) 은 (3)과 같이 정의 된다.

$$p(\boldsymbol{v})=\frac{1}{z}\sum_{\boldsymbol{h}}p(\boldsymbol{v},\boldsymbol{h})=\frac{1}{z}\sum_{\boldsymbol{h}}e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{3}$$

RBM를 초기화시키기 위한 비교사 신호 학습은 log-likelihood 함수를 최대화 하는 것이다. 이때 필요한 log-likelihood 함수는 (4)와 같다.

$$\ln\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{v})=\ln p(\boldsymbol{v}|\boldsymbol{\theta})=\ln\frac{1}{Z}\sum_{\boldsymbol{h}}e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{4}$$

주어진 log-likelihood 함수를 최대화 시키는 $\boldsymbol{\theta}$값을 (5)를 통해 얻을 수 있다.

$$\frac{\partial\ln\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{v})}{\partial\boldsymbol{\theta}}=-\sum_{\boldsymbol{h}}p(\boldsymbol{h}|\boldsymbol{v})\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial\boldsymbol{\theta}}+\sum_{\boldsymbol{v},\boldsymbol{h}}p(\boldsymbol{v},\boldsymbol{h})\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial\boldsymbol{\theta}} \tag{5}$$

위에 주어진 log-likelihood 함수를 최대화 하는 파라미터 $\boldsymbol{\theta}$를 얻기 위하여 일반적으로 contrastive divergence 알고리즘을 사용한다. 표 1은 Contrastive divergence 알고리즘의 의사 코들 보인다.

표 1 Contrastive divergence 알고리즘의 의사코드
Table 1 Pseudo code of Contrastive divergence algorithm

Procedure: Contrastive Divergence
Input: RBM$(V_1, V_2, \cdots, V_m, H_1, H_2, \cdots, H_n)$,
and Training batch S
Ouput: gradient approximation $\triangle w_{ij}, \triangle b_j, \triangle c_i$
for $i=1,2,\cdots,n,\ j=1,2,\cdots,m$
for all the $\boldsymbol{v}\in S$ do
$\boldsymbol{v}^{(0)}\leftarrow\boldsymbol{v}$
for $t=0,1,\cdots,k-1$ do
for $i=1,2,\cdots,n$ do sample $h_i^{(t)}\sim p(h_i|\boldsymbol{v}^{(t)})$
for $j=1,2,\cdots,m$ do sample $v_j^{(t+1)}\sim p(v_j|\boldsymbol{h}^{(t)})$
for $i=1,2,\cdots,n,\ j=1,2,\cdots,m$ do

$\triangle w_{ij}\leftarrow\triangle w_{ij}+p(H_i=1|\boldsymbol{v}^{(0)})\cdot v_j^{(0)}-p(H_i=1|\boldsymbol{v}^{(k)})\cdot v_j^{(k)}$
$\triangle b_j\leftarrow\triangle b_j+v_j^{(0)}-v_j^{(k)}$
$\triangle c_i\leftarrow\triangle c_i+p(H_i=1|\boldsymbol{v}^{(0)})-p(H_i=1|\boldsymbol{v}^{(k)})$

3. Autoencoders

그림 2는 일반적인 autoencoders의 architecture를 보인다. 그림 2에 보인 autoencoders의 architecture를 설명하기 위하여 아래와 같은 정의가 필요하다.

1. $\boldsymbol{F}$, $\boldsymbol{G}$ 는 일반적인 집합이다.
2. n, p는 자연수를 나타내며 조건 $0<p<n$을 만족한다.
3. $\boldsymbol{A}$ 는 $\boldsymbol{G^p}$에서 $\boldsymbol{F^n}$ 로의 함수들의 클래스를 나타낸다.
4. $\boldsymbol{B}$ 는 $\boldsymbol{F^n}$에서 $\boldsymbol{G^p}$ 로의 함수들의 클래스를 나타낸다.
5. $\boldsymbol{X}=\{x_1, x_2, \cdots, x_m\}$은 $\boldsymbol{F^n}$의 학습 벡터들의 집합을 의미한다.
   $\boldsymbol{Y}=\{y_1, y_2, \cdots, y_m\}$은 $\boldsymbol{F^n}$의 학습 벡터들에 대응되는 목표 벡터들을 나타낸다.
6. $\triangle$는 $\boldsymbol{F^n}$에서 정의된 비유사 함수 (dissimilarity function) 이다.

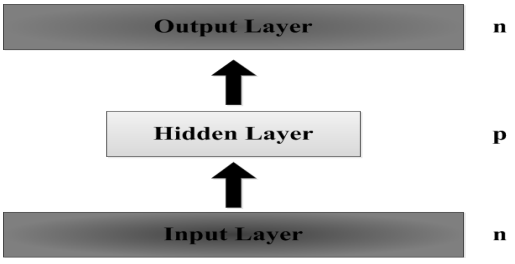

그림 2 Autoencoder의 Architecture
Fig. 2 Architecture of Autoencoder

그림 2에 보인 Autoencoders의 최종 출력 식은 (6) 과 같다.

$$Y = A \circ B(X) \tag{6}$$

여기서, 임의의 함수 A, B는 $A \in \boldsymbol{A}$, $B \in \boldsymbol{B}$이고, autoencoder 변환의 입력 벡터는 x_t이며 autoencoder 변환의 출력 벡터는 (6)과 같이 Y이다.

최적의 autoencoder 변환을 위한 최적해 A, B는 (7)과 같이 정의된 비유사 함수를 최소화 시켜야 한다.

$$\begin{aligned} \min_{A,B} E(A,B) &= \min_{A,B} \sum_{t=1}^{m} E(x_t) \\ &= \min_{A,B} \sum_{t=1}^{m} \triangle(A \circ B(x_t), x_t) \end{aligned} \tag{7}$$

여기서, $\triangle(a_i, b_i) = \| a_i - b_i \|^2$이다.

3.1 Stacked Autoencoders

Stacked Autoencoder는 위에서 설명한 autoencoder들을 미리 설계자가 결정한 개수만큼 스택처럼 쌓아 올려 네트워크를 형성한다. 이와 같이 형성된 네트워크는 입력 데이터들에서 중요한 특징들을 추출할 수 있다.

각 층 노드들의 파라미터 $\boldsymbol{\theta}$는 그림 2와 같이 은닉층을 거쳐 출력충의 출력과 입력 데이터와의 유사성을 비교하여 입력 데이터와 가장 유사한 출력 데이터를 출력하게 하는 파라미터를 결정한다.

하나의 은닉층의 파라미터가 결정되면, 출력층을 제거 하고 학습된 은닉층의 출력을 입력 데이터로 하여 또 다른 하나의 은닉층과 출력층을 가진 autoencoder를 설계한다.

이와 같은 과정을 수차례 설계자가 미리 결정한 횟수만큼 반복하여, 네트워크 형성한다.

그림 3은 Stacked Autoencoder의 Architecture를 보인다.

입력 데이터의 주요 특징들을 추출하기 위하여 Stacked Autoecoder 네트워크를 이용하고자 한다.

주어진 입력데이터 벡터들의 집합에서 주요한 특징들을 추출하기 위하여 Stacked Autoencoder 네트워크의 은닉층의 출력을 사용하고자 한다.

은닉층 p_2에서의 출력을 패턴 분류기 FkNN의 최종 입력 데이터로 사용한다.

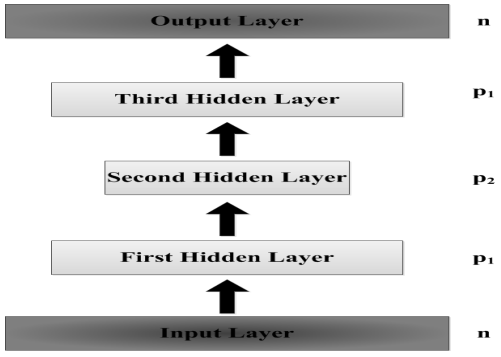

그림 3 Stacked Autoencoder Architecture
Fig. 3 Architecture of Stacked Autoencoder

4. Fuzzy k-Nearest Neighbors Classifier

통계적 패턴분류기의 일종인 k-Nearest Neighbors 패턴 분류기와 변형된 k-Nearest Neighbors 패턴 분류기는 다양한 데이터들로 이루어진 패턴 분류의 영역에서 우수한 성능을 보이는 것으로 알려져 있다 [9]. 일반적으로 kNN 알고리즘은 local learning 방법의 일종으로 알려져 있다. kNN 알고리즘은 학습 데이터 패턴 전체를 저장하고, 새로운 입력이 주어졌을 경우, 주어진 새로운 입력과 가장 가까운 k개의 데이터 패턴을 이용하여 새롭게 주어진 입력 패턴을 분류하게 된다.

일반적으로 데이터 패턴 주어진 쿼리 패턴 (query pattern)과의 연관성은 아래식과 같은 데이터 패턴과 주어진 쿼리 패턴사이의 유클리디언 거리에 기반을 두어 결정된다.

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sum_{l=1}^{m} (a_l - b_l)^2 \tag{8}$$

여기서, m은 입력 변수의 차원을 나타낸다.

정의된 지역 영역에 속한 데이터 패턴 (학습 패턴)들에 하중 값을 부여하여야 한다. 일반적으로 데이터 패턴에 하중 값을 부여하기 위한 방법으로 가우시안 함수를 kernel 함수로 사용하는 경우가 많다. 그러나 우리는 가우시안 함수를 kernel 함수로 사용하지 않고, Bezdeck이 제안한 퍼지 클러스터링 알고리즘에 적용된 상대적 거리를 이용한 함수를 하중값을 부여하기 위한 하중 함수로 사용할 것이다.

4.1 Crisp k-Nearest Neighbors 기법

대표적인 non-parametric 알고리즘인 k-Nearest Neighbors(kNN) 기법은 다양한 분류 문제에 적용되어져왔다. 일반적으로, KNN 기법은 알려져 있지 않은 데이터 패턴이 주어질 경우, 주어진 데이터 패턴과 가장 가까운 위치에 있는 k개의 데이터 패턴을 선정하고, 선정된 데이터 패턴의 출력 패턴을 비교하여 새롭게 주어진 데이터 패턴의 출력 값을 추정한다. Crisp k-Nearest Neighbors 기법을 설명하기 위하여 입력 패턴 집합 X와 입력 패턴에 대응하는 출력 패턴의 집합 Y를 (9)와 (10)과 같이 정의한다. 분류 문제에서 출력 패턴의 집합은 연속적인 값이 아닌 이산적인 값을 가진다.

$$\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}\}, \ \boldsymbol{x_k} \in R^m \tag{9}$$

$$\boldsymbol{Y} = \{y_1, y_2, \cdots, y_n\}, \ y_k \in N \tag{10}$$

여기서, N은 자연수의 집합을 의미하며, n은 데이터 패턴의 수를 의미한다. 학습 데이터의 집합이 (9)와 같이 정의 되어졌다고 가정하고, 분류해야할 쿼리 패턴 $\boldsymbol{x_q}$가 주어지면, 학습 데이터 집합 X의 요소들 중에서 $\boldsymbol{x_q}$와 가까운 거리에 있는 k개의 데이터 패턴을 찾는다.

이때 데이터 패턴들 간의 거리는 (11)과 같이 정의된 weighted Euclidean distance를 이용하여 계산한다.

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{l=1}^{m} \frac{(a_l - b_l)^2}{\sigma_l^2}} \tag{11}$$

(6)에 의해 얻어진 쿼리 패턴과 학습 데이터 집합의 요소들 사이의 거리 정보를 이용하여 거리가 가까운 K개의 학습 데이터들은 찾는다.

집합 L은 주어진 쿼리 패턴과 거리가 가까운 데이터 패턴의 인덱스의 집합을 의미한다.

$$L = \{j : \boldsymbol{x_j} \text{ is one of the k nearest instances to query point } \boldsymbol{q}\} \tag{12}$$

근접 위치에 있는 K개의 학습 데이터들의 출력 label을 이용하여 주어진 쿼리 패턴의 출력 label 을 (13)과 같이 구한다.

$$\hat{y}(\boldsymbol{x_q}) = \underset{c}{\operatorname{argmax}} \sum_{j=1}^{k} f(y(\boldsymbol{x_{L\{j\}}}), c) \tag{13}$$

여기서, $f(y(\boldsymbol{x_{L\{j\}}}), c) = \begin{cases} 1, & \text{if } y(\boldsymbol{x_{L\{j\}}}) = c \\ 0, & \text{if } y(\boldsymbol{x_{L\{j\}}}) \neq c \end{cases}$

4.2 Fuzzy k-Nearest Neighbors 기법

Fuzzy k-Nearest Neighbors (FkNN) 기법은 기존의 crisp k-Nearest Neighbors기법에 퍼지 집합의 개념을 적용한 기법으로 주어진 쿼리 패턴과 학습 데이터 패턴들 간의 유사성 (similarity)의 정의에 기반을 두고 있다.

데이터 패턴들 간의 유사성은 (14)과 같이 정의한다.

$$w(\boldsymbol{x_{L\{i\}}}, \boldsymbol{q}) = \begin{cases} 1, & \text{if } \boldsymbol{x_{L\{i\}}} = \boldsymbol{q} \\ \dfrac{\left(\dfrac{1}{\| \boldsymbol{x_{L\{i\}}} - \boldsymbol{q} \|}\right)^{\frac{2}{p-1}}}{\displaystyle\sum_{j=1}^{k} \left(\dfrac{1}{\| \boldsymbol{x_{L\{j\}}} - \boldsymbol{q} \|}\right)^{\frac{2}{p-1}}}, & \text{if } \boldsymbol{x_{L\{i\}}} \neq \boldsymbol{q} \end{cases} \tag{14}$$

여기서, w는 데이터 $\boldsymbol{x_{L\{i\}}}$와 쿼리 패턴 $\boldsymbol{q}$와의 유사성을 의미하며, k는 Nearest neighbor들의 수, p는 퍼지화 계수 (fuzzification coefficient)를 의미한다.

(14)를 통해 얻어진 유사성 $w(\boldsymbol{x_{L\{i\}}}, \boldsymbol{q})$는 다음 조건을 만족한다.

$$0 < w(\boldsymbol{x_{L\{i\}}}, \boldsymbol{q}) \leq 1 \tag{15}$$

근접 위치에 있는 K개의 학습 데이터들의 출력 label을 이용하여 FkNN 기법을 적용하여 주어진 쿼리 패턴의 출력 label은 (16)과 같이 구한다.

$$\hat{y}(\boldsymbol{x_q}) = \underset{c}{\operatorname{argmax}} \sum_{j=1}^{k} f(y(\boldsymbol{x_{L\{j\}}}), c) \tag{16}$$

여기서, $f(y(\boldsymbol{x_{L\{j\}}}), c) = w(\boldsymbol{x_{L\{j\}}}, \boldsymbol{q})$.

여기서, $\hat{y}(\boldsymbol{q})$는 쿼리 패턴과 연관된 fuzzy k-Nearest Neighbor 기법을 이용하여 추정된 출력 label을 의미하며, $y(\boldsymbol{x})$는 데이터 패턴 $\boldsymbol{x}$와 연관된 알려진 출력 label을 의미한다.

4.3 Fuzzy C-Means Clustering 기법

Fuzzy k-Nearest Neighbors 패턴 분류기를 설계하는 과정에

서 새롭게 주어진 데이터와 가장 가까운 k개의 Nearest Neighbors을 정의하고, 정의된 데이터들에 대하여 주어진 데이터와 유사성을 이용하여 하중값을 결정할 필요가 있다. Fuzzy C-Means Clustering (FCM) 알고리즘을 이용하여 데이터들 간의 유사성을 결정한다.

FCM 클러스터링은 n개의 벡터 $x_i (i=1,..,n)$ 집합을 c개의 클러스터로 분할하고, 목적함수가 최소가 일 때 생성된 각 클러스터에서 중심 값을 찾는다. FCM 클러스터링의 목적함수는 (17)과 같다.

$$J(\boldsymbol{\mu}, \boldsymbol{v}) = \sum_{i=1}^{c}\sum_{k=1}^{N} \mu_{ik}^{m} (\| \boldsymbol{x_k} - \boldsymbol{v_i} \|)^2 \tag{17}$$

목적함수를 최소화하는 $\boldsymbol{\mu}$와 $\boldsymbol{v}$는 (18), (19)와 같이 얻을 수 있다.

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} (\frac{\|\boldsymbol{x_k} - \boldsymbol{v_i}\|}{\|\boldsymbol{x_k} - \boldsymbol{v_i}\|})^{2/m-1}} \tag{18}$$

$$v_{ij} = \frac{\sum_{k=1}^{N} (u_{ik})^m x_{ki}}{\sum_{j=1}^{N} (u_{ik})^m} \tag{19}$$

여기서, $\boldsymbol{\mu}$는 데이터의 퍼지 집합 소속 멤버십 함수를 의미하며, $\boldsymbol{v}$는 해당 하는 퍼지 집합의 중심점을 의미한다.

일반적인 FCM의 적용과는 달리 FkNN의 응용에서는 주어진 입력 공간 전체를 분석하여 각 퍼지 클러스터의 중심 값을 (19)과 같이 결정할 필요가 없기 때문에 (19) 대신 새롭게 주어진 데이터와 가장 가까운 k개의 데이터 포인트들로 대치한다.

5. 실험 및 결과 고찰

본 연구에서는 제안된 분류기의 성능을 평가하기 위하여 제안된 분류기를 입력 데이터의 차원이 매우 큰 MNIST와 CIFIR-10 데이터를 이용하여 분류기로서의 성능을 평가 및 분석한다. Benchmark 데이터 집합을 이용하여 제안된 패턴 분류기의 성능과 특성을 기존 논문에서 이미 제안된 패턴 분류기의 성능과 비교, 평가 한다. 제안된 제어기의 분류 성능 평가를 위한 성능 평가 지수는(20), (21)과 같이 패턴 분류기의 오분류 비율 (misclassification rate) 로 한다.

$$\text{오분류율} = \frac{\sum_{i=1}^{N} f(y_i - \hat{y_i})}{N} \times 100 \tag{20}$$

여기서, N은 데이터의 수를 의미한다.

$$f(y_i - \hat{y_i}) = \begin{cases} 0, y_i \neq \hat{y_i} \\ 1, y_i = \hat{y_i} \end{cases} \tag{21}$$

표 2는 제안된 Stacked Autoencoder를 이용한 특징 추출 기반 퍼지 k-Nearest Neighbor 패턴 분류기의 설계를 위해 미리 설정되어져야 하는 파라미터들을 보인다.

표 2 제안된 패턴 분류기의 파라미터
Table 2 Selected numeric values of the parameters of the proposed technique

Parameter	Value
Number of Nearest Neighbors (k)	3, 5, 10, 20, 30, 50, 100, 150
Fuzzification Coefficient	2
Number of Hidden Layers (L)	5
Number of Learning Iteration	50

5.1 MNIST 데이터 집합

그림 4는 제안된 특징 추출 방법을 이용한 패턴 분류기의 성능을 평가하기 위하여 사용된 MNIST 데이터 집합의 일부 데이터를 표현한 것이다.

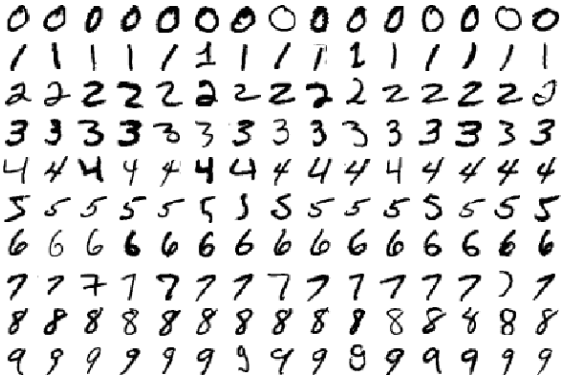

그림 4 MNIST 데이터
Fig. 4 Data of MNIST

MNIST 데이터는 손으로 쓰여진 숫자들을 모아 놓은 데이터 집합으로 하나의 데이터 패턴은 가로 28 픽셀, 세로 28 픽셀로 구성된 흑백 이미지이다. 데이터의 총 수는 70000개이며, 학습 데이터는 60000개로 구성되며, 테스트 데이터는 학습 데이터를 제외한 10000개의 데이터로 구성되어 있다.

위에 설명한 바와 같이 MNIST 데이터 집합의 입력 공간의 차원은 28×28로 784개의 입력 변수로 구성되어 있다 [10].

표 3은 MNIST 데이터 집합을 이용한 기존 패턴 분류기와 제안된 패턴 분류기의 패턴 분류 성능을 보인다.

표 3 제안된 패턴 분류기와 기존 패턴 분류기 성능 비교
Table 3 Comparison of Classification Performances of the proposed Classifier to Conventional Classifiers

Classifier	Ratio of Feature Seletion (%)	Test Error Rate (%)
Linear Classifier (1-layer NN)	100	12.0
Linear Classifier (1-layer NN)	100	8.4
pairwise linear classifier	100	7.6
K-nearest-neighbors, Euclidean (L2)	100	5.0
K-nearest-neighbors, Euclidean (L2)	100	***2.4***
K-NN, Tangent Distance	$\frac{256}{784}\times 100 = 32.65$	***1.1***
Proposed Classifier, k=3	$\frac{30}{784}\times 100 = 3.83$	3.79
Proposed Classifier, k=10	$\frac{30}{784}\times 100 = 3.83$	4.25
Proposed Classifier, k=30	$\frac{30}{784}\times 100 = 3.83$	***2.79***
Proposed Classifier, k=50	$\frac{30}{784}\times 100 = 3.83$	3.06
Proposed Classifier, k=100	$\frac{30}{784}\times 100 = 3.83$	3.77

그림 5는 제안된 특징 추출 기법을 이용한 패턴 분류기의 패턴 분류 성능과 입력 공간의 차원 축소 비율을 나타낸다.

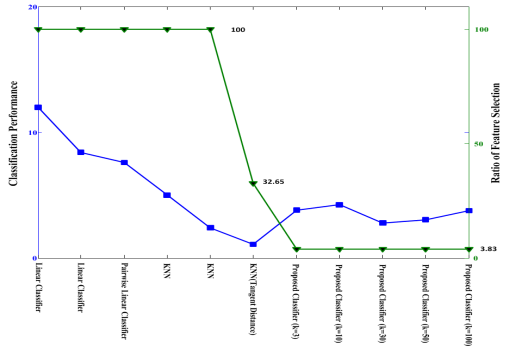

그림 5 패턴 분류기의 패턴 분류 성능 과 차원 축소 비율
Fig. 5 Pattern Classification Performance and Ratio of Dimension Reduction

표 3에 보인 실험 결과와 같이 제 Stacked Autoencoders를 이용한 특징 추출 기법을 적용한 패턴 분류기의 성능은 모든 입력 변수를 사용하는 다수의 패턴 분류기에 비해 우수한 결과를 보임을 알 수 있다. 그러나 최고의 패턴 분류 성능을 보이는 tangent distance를 사용하는 KNN에 비해서는 패턴 분류 성능이 좋지 못함을 알 수 있다. 그러나 차원 축소 측면에서는 tagent distance KNN에 비해 제안한 방법이 우수함을 알 수 있다. 즉, 입력 데이터의 차원을 $28\times 28 = 784$에서 Stacked Autoencoders를 이용하여, 30개로 줄여, k-Nearst Neighbor 패턴 분류기의 계산 량을 대폭 개선하였다.

5.2 CIFAR-10 데이터 집합

그림 6은 제안된 특징 추출 방법을 이용한 패턴 분류기의 성능을 평가하기 위하여 사용된 CIFAR-10 데이터 집합의 일부 데이터를 표현한 것이다.

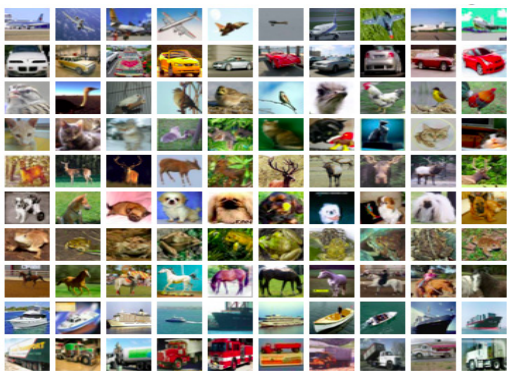

그림 6 CIFAR-10 데이터
Fig. 6 Data of CIFAR-10

CIFAR-10 데이터 집합은 물체 인식에 사용하기 위하여 만들어진 컴퓨터 비젼 데이터 집합이다. CIFAR-10 데이터는 60000개의 32×32컬러 이미지 데이터들의 집합이며 10개의 클래스들로 구성되어져 있다. 각 클래스는 6000개의 데이터 집합으로 구성되어진다.

표 4는 CIFAR-10 데이터 집합을 이용한 기존 패턴 분류기와 제안된 패턴 분류기의 패턴 분류 성능을 보인다.

표 4 CIFAR-10 데이터를 이용한 제안된 패턴 분류기와 기존 패턴 분류기 성능 비교
Table 4 Comparison of Classification Performances of the proposed Classifier to Conventional Classifiers using CIFAR-10

Classifier	Preprocessing	Test Error Rate (%)
PCANet		21.33
Learning Smooth Pooling Regions for Visual Recognition		19.98
Convolutional Kernel Networks		17.82
Proposed Classifier, k=3	Feature Extraction n=30	26.7
Proposed Classifier, k=5	Feature Extraction n=30	29.91
Proposed Classifier, k=10	Feature Extraction n=30	31.96
Proposed Classifier, k=20	Feature Extraction n=30	34.07
Proposed Classifier, k=50	Feature Extraction n=30	34.66

그림 7은 제안된 특징 추출 기법을 이용한 패턴 분류기의 패턴 분류 성능과 입력 공간의 차원 축소 비율을 나타낸다.

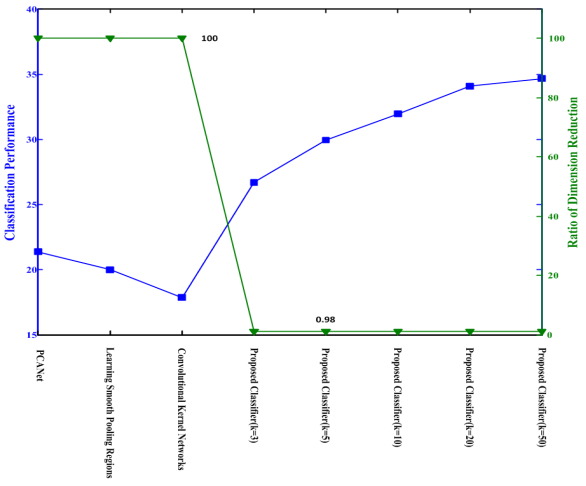

그림 7 패턴 분류기의 패턴 분류 성능 과 차원 축소 비율
Fig. 7 Pattern Classification Performance and Ratio of Dimension Reduction

표 4에 보인 실험 결과와 같이 제 Stacked Autoencoders를 이용한 특징 추출 기법을 적용한 패턴 분류기의 패턴 분류 성능은 모든 입력 변수를 사용하는 패턴 분류기에 비해 좋지 않은 결과를 보임을 알 수 있다. 패턴 분류 성능이 아닌 차원 축소 측면에서는 기존 패턴 분류기들에 비해 제안한 방법이 우수함을 알 수 있다. 즉, 입력 데이터의 차원을 $32\times32\times3=3072$에서 Stacked Autoencoders를 이용하여, 30개로 줄여, k-Nearst Neighbor 패턴 분류기의 계산 량을 대폭 개선하였다. 차원 축소 효율은 $\frac{30}{3072}\cdot100=0.98\%$이다.

6. 결 론

본 논문에서는 패턴 분류 분야에서 대두되고 있는 고차원 데이터 문제를 해결하기 위한 새로운 특징 추출 방법을 적용하여 패턴 분류 성능을 개선하고자 한다. 고차원 데이터를 저차원의 입력공간으로 투영하기 위한 방법인 특징 추출을 위해 Deep Networks의 일종인 Stacked Autoencoder 네트워크를 적용하였다. 적용된 Stacked Autoencoders 네트워크의 기본 노드는 Restricted Boltzmann Machine이다. Stacked Autoencoders 네트워크를 이용하여 고차원 입력 데이터가 가지고 있는 정보를 저차원 입력 공간에 투영된 특징들로 충분히 표현할 수 있는 특징들을 추출한다. 이와 같이 새롭게 얻어진 특징들은 저차원 데이터 이면서도 기존 고차원 입력 데이터 들이 가지고 있는 정도의 정보를 보유하고 있기 때문에 고차원 데이터로 인해 야기되는 문제를 감소 기키면서 패턴 분류기의 성능도 향상 시킬 수 있다. 제안된 특징 추출방법에 의해 추출된 입력 데이터를 이용한 패턴 분류기는 간단한 형태의 Fuzzy k-Nearest Neighbor 패턴 분류기를 사용한다.

이와 같은 특징 추출과 fuzzy k-nearest neighbor 패턴 분류기를 MNIST 데이터 집합에 적용하여 실험한 결과는 기존 패턴 분류기에 비해 패턴 분류 성능에서도 우수한 성능을 보였고 입력 데이터의 차원을 대폭 감소시킴으로서 패턴 분류 알고리즘의 계산 속도를 개선시킬 수 있었다.

감사의 글

This work was supported by GRRC program of Gyeonggi province [GRRC Suwon 2014-B2, Center for U-city Security & Surveillance Technology] and supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (NRF-2012R1A1B3003568)

References

[1] G. Pajares, M. Guijarro, A. Ribeiro, “A Hopfield Neural Network for combining classifiers applied to textured images,” Neural Networks, Vol.23, pp. 144-153, 2010.

[2] B. Chan, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, L. Carin, "Deep Learning with Hierarchical Convolutional Factor Analysis," IEEE Trans. on Pattern and Machine Intelligence, Vol. 35, No. 8, pp. 1958-1971, 2013.

[3] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504-507, 2006.

[4] Y. Bengio and Y. LeCun, “"Scaling learning algorithms toward AI,” in Large Scale Kernel Machine. Cambridge, MA: MIT Press, pp. 321-360, 2007.

[5] Y. Bengio, “Learning deep architectures for AI,” Foundations Trends Mach. Learn., vol. 2, no. 1, pp. 1 - 127, 2009.

[6] G.E. Hinton, S. Osindero, Y.W. The, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, no. 7, pp. 1527 - 1554, 2006.

[7] J. C. Bezdek, “Pattern Recognition with Fuzzy Objective Function Algorithms,” Plenum Press, NY, 1981.

[8] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations, pp. 194 - 281. MIT Press, 1986.

[9] W. Lam, C. K. Keung, C. X. Ling, "Learning good

prototypes for classification using filtering and abstraction of instances," Pattern Recognition, vol. 35, pp. 1491-1506. 2002

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

저 자 소 개

노 석 범(Suck-Bum Rho)

1994년 원광대 제어계측공학과 졸업. 1996년 동 대학 컴퓨터 공학과 졸업(석사). 2006년 동 대학 제어계측공하과 졸업(박사). 관심 분야 : 퍼지 모델링, 컴퓨터 지능, statistical learning, Deep Learning, Machine Learning

Tel : (063) 850-6344

E-mail : nado@wonkwang.ac.kr

오 성 권(Sung-Kwun Oh)

1981년 연세대학교 전기공학과 졸업, 동 대학원 석사(1983), 박사(1993). 1983-1989년 금성산전연구소(선임연구원). 1996-1997년 캐나다 Manitoba 대학 전기 및 컴퓨터공학과 Post-Doc. 1993-2004년 원광대학교 전기전자 및 정보공학부 교수. 2005～현재 수원대학교 전기공학과 교수, 2002～현재 대한전기학회, 제어로봇시스템학회, 퍼지및지능시스템학회 편집위원. 2012～현재 Information Sciences 편집위원. 관심분야는 퍼지 시스템, 퍼지-뉴럴 네트워크, 자동화 시스템, 고급 computational intelligence, 지능 제어 등.