

LVQ와 ADALINE을 이용한 학습 알고리듬
Learning Algorithm using a LVQ and ADALINE

Yoon, Seok - Hwan *
윤석환
Min, Joon - Young **
민준영
Shin, Yong - Back ***
신용백

Abstract

We propose a parallel neural network model in which patterns are clustered and patterns in a cluster are studied in a parallel neural network. The learning algorithm used in this paper is based on LVQ algorithm of Kohonen(1990) for clustering and ADALINE(Adaptive Linear Neuron) network of Widrow and Hoff(1990) for parallel learning.

The proposed algorithm consists of two parts. First, N patterns to be learned are categorized into C clusters by LVQ clustering algorithm. Second, C patterns that was selected from each cluster of C are learned as input pattern of ADALINE(Adaptive Linear Neuron). Data used in this paper consists of 250 patterns of ASCII characters normalized into 8x16 and 1124.

The proposed algorithm consists of two parts. First, N patterns to be learned are categorized into C clusters by LVQ clustering algorithm. Second, C patterns that was selected from each cluster of C are learned as input pattern of ADALINE(Adaptive Linear Neuron). Data used in this paper consists of 250 patterns of ASCII characters normalized into 8x16 and 1124 samples acquired from signals generated from 9 car models that passed Inductive Loop Detector(ILD) at 10 points. In ASCII character experiment, 191(179) out of 250 patterns are recognized with 3%(5%) noise and with 1124 car model data. 807 car models were recognized showing 71.8% recognition ratio. This result is 10.2% improvement over backpropagation algorithm.

* 한국전자통신연구소 책임연구원
** 상지대학교 병설 전문대학 교수
*** 아주대학교 산업공학과 교수

1. 서 론

페셉트론은 1957년 Frank Rosenbaltt에 의하여 처음 제안되었으나 출력 유니트가 선형 분리 가능한 패턴들만 분류할 수 있고[1], XOR와 같은 간단한 문제도 해결하지 못하고 극히 제한된 패턴에 대해서만 분류할 수 있었기 때문에 1970년대까지 더 이상의 발전을 할 수 없었다. 그 이후 David Parker(1985)와 LeCun(1986)등 여러 학자에 의해서 역전파 학습 알고리즘이 제안되었으며[2], 이 방법은 비선형으로 패턴을 분류할 수 있다는 장점으로 현재까지 널리 쓰이고 있다. 그러나 이 알고리즘은 다음과 같은 세가지 단점을 갖고 있다. 첫째, 학습 도중 지역 최소점(local minimum)이 될 경우에 연결강도의 변화가 없어서 패턴을 더 이상 인식못하고 학습이 종료되며, 둘째, 은닉층의 수와 은닉층 안의 유니트수에 대한 정확한 정의가 되어 있지 않고, 세째, 한 유니트에서의 연결강도는 그 유니트를 중심으로 상위층과 하위층의 연결강도에 의존되어 결정되기 때문에 학습하는 데 소요되는 시간이 매우 길다[3]. 이와 반면에 비교사학습은 페셉트론이 현실적으로 목표패턴을 구하기 난해하기 때문에 입력 패턴들의 통계적 특징을 파악하여 유사한 패턴끼리 하나의 카테고리 안에 들어올 수 있도록 클러스터링하는 데에 그 목적이 있다[4]. 비교사 학습 알고리즘으로는 Kohonen(1990)의 SOFM(Self Organization Feature Maps)와 LVQ(Leatning Vector Quantization), Carpenter 와 Grossberg(1976)의 ART-1(Adaptive Resonence Theory-1)과 ART-2모델이 있으며, 최근 Pal et.al.의 GLVQ(Generalized Learning Vector Quantization)알고리즘등이 소개되고 있다[5].

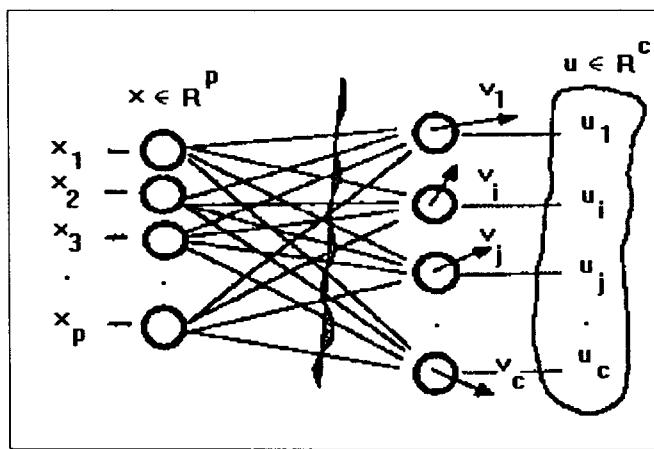
본 논문은 패턴을 학습도중 지역최소점이나 정상점(stationary point)이 이를 경우 더 이상의 학습을 하지 못하거나 학습이 완료된다 하여도 장시간의 학습시간이 소요된다는 점, 그리고 은닉층의 수와 유니트의 수를 결정하는 데 많은 시행착오를 해야하는 문제점이 있는 기존의 역전파 학습 알고리즘에 대체되는 학습 알고리즘을 제안한다. 본 알고리즘은 우선 N개의 패턴을 c개의 클러스터로 클러스터링을 한 다음 각 클러스터에서 i번째 패턴 하나씩을 추출한 c개의 패턴을 선형 활성화된 출력노드를 갖는 ADALINE네트워크에 입력시켜 학습을 하는 알고리즘을 제안한다. 각 클러스터에서 i번째 패턴 하나씩을 추출한 이유는 ADALINE네트워크에서 학습을 할 때 유사성이 있는 패턴의 분류보다는 비유사성을 갖는 패턴의 분류가 학습시간이 짧게 걸릴 뿐만 아니라 분류율을 보다 높일 수 있기 때문이다.

2. 관련 네트워크

2.1 클러스터링 네트워크

클러스터링은 패턴의 유사성을 측정하여 유사한 패턴끼리 몇개의 그룹으로 조직화 하는 것으로 이러한 문제에 접근하는 이론으로는 Kohonen, Bezdek, Duda와 Hart, Tou와 Gonzalez, Hartigan, Jain과 Dubes에 의해서 연구되어왔다[5]. 본 논문에서 이용한 클러스터

링 알고리즘은 Kohonen(1990)의 LVQ알고리즘을 이용하였다. LVQ는 비교사학습에 근거한 패턴인식기의 일종으로서 Bayes결정경계면을 근사화하여 높은 분류능력을 가지며, 오류역전파(error backpropagation) 학습알고리즘에 비하여 학습 속도가 빨라 음성인식등과 같이 학습의 패턴수가 많은 경우에 적합한 패턴인식기로 알려져 있다[6]. LVQ네트워크의 프로토타입은 [그림 1]과 같다[5].



[그림 1] LVQ네트워크의 프로토타입

또한 LVQ알고리즘은 다음과 같다[5].

[Step 1] Given data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^p$.
 Fix c , T , and $\epsilon > 0$.
 [Step 2] Initialize $\mathbf{V}_0 = (\mathbf{v}_{1,0}, \dots, \mathbf{v}_{c,0}) \in \mathbb{R}^p$
 and learning rate $\alpha_0 \in (1, 0)$.
 [Step 3] For $t=1,2,\dots,T$:
 For $k=1,2,\dots,n$:

$$\|\mathbf{x}_k - \mathbf{v}_{i,t-1}\| = \min_{1 \leq j \leq c} \{\|\mathbf{x}_k - \mathbf{v}_{j,t-1}\|\}$$

b. Update the winner

$$\mathbf{v}_{i,t} = \mathbf{v}_{i,t-1} + \alpha_t (\mathbf{x}_k - \mathbf{v}_{i,t-1})$$

c. Next k .

d. Apply the 1-NP(nearest prototype) rule to the data using

$$u_{LVQ_{ik}} = \begin{cases} 1: & \|\mathbf{x}_k - \mathbf{v}_i\| \leq \|\mathbf{x}_k - \mathbf{v}_j\|, \quad 1 \leq j \leq c, \quad j \neq i \\ 0: & \text{otherwise.} \end{cases}$$

$$1 \leq i \leq c \quad \text{and} \quad 1 \leq k \leq n.$$

$U_{LVQ} = [u_{LVQ_{ik}}]$: $c \times n$ matrix that defines a hard or
 crisp partition of X using the 1-NP classifier assignment rule.

[Step 4] Compute $E_t = \sum_{k=1}^n \sum_{r=1}^c |\mathbf{v}_{rk,t} - \mathbf{v}_{rk,t-1}|$.

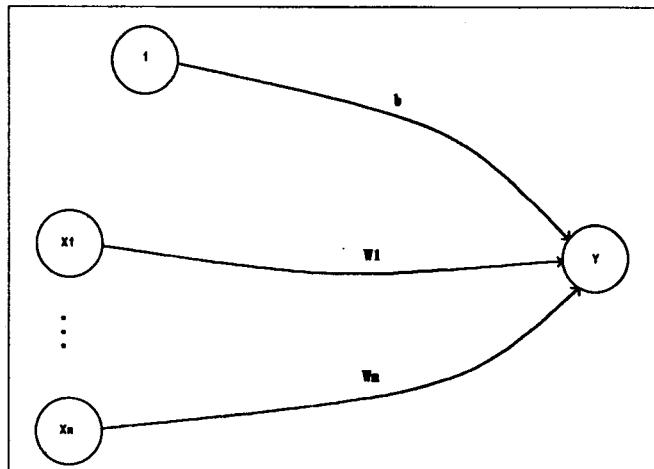
[Step 5] if $E_t \leq \epsilon$ then STOP; Else adjust learning rate α_t ; Next t .

$$\alpha_t = \alpha_0 (1 - t/T) \text{ update learning } r$$

그러나 LVQ알고리즘의 경우 초기 중앙값(initial weight) $v_{i,0}$ 에 많은 영향이 있어서 초기 중앙값이 입력 데이터와 멀리 떨어진 값으로 설정되었을 경우(outside of convex hull)에 학습이 완료된 후의 결과가 만족할 만한 클러스터링이 안된다는 단점이 있다. 실제로 Pal은 LVQ알고리즘을 이용하여 Fisher의 IRIS데이터를 클러스터링 한 결과 초기 연결강도와 학습률에 따라 오분류 되는 패턴의 수가 17개에서 100개까지 나온 것으로 발표하였다[5].

2.2 ADALINE(Adaptive Linear Neuron)

ADALINE은 각각의 입력패턴에 연결강도를 곱하여 합한($\sum x_i w_i$)입력신호의 선형결합이다. 출력은 0과 1중의 하나로 결정된다. ADALINE의 학습은 입력-출력을 쌍으로 순차적으로 제시하여 원하는 출력값이 나오도록 연결강도를 수정하는 것으로 1959년 Widrow와 Hoff가 제안한 최소자승오차 알고리즘을 사용하였다[1]. ADALINE의 네트워크 구성은 [그림 2]와 같다[2].



[그림 2] ADALINE 네트워크 구성

ADALINE의 알고리즘은 다음과 같다.

[Step 1] Initialize weights.(Small random values)

Set learning rate α .

[Step 2] While stopping condition is false, do Step 3-7.

[Step 3] For each bipolar training pair s:t, do Step 4-6.

[Step 4] Set activations of input units, $i=1, 2, \dots, n$

$x_i = s_i$ (where, s_i : training input vector)

[Step 5] Compute net input to output unit :

$$y_{in} = b + \sum x_i w_i$$

[Step 6] Update bias and weights, $i=1, 2, \dots, n$:

$$b(\text{new}) = b(\text{old}) + \alpha(1 - y_{in})$$

$$w_i(\text{new}) = w_i(\text{old}) + \alpha(t - y_{in})x_i$$

[Step 7] Test for stopping condition:

If the largest weight change that occurred in Step 3 is smaller than a specified tolerance, then stop; otherwise continue

본 논문에서는 ADALINE의 출력값을 패턴의 레이블로 선형 결합시킨 학습 알고리즘을 개발하여 이를 적용하였다.

3. 데이터의 처리

3.1 ASCII문자 데이터

본 논문에서 이용한 데이터는 전 처리 과정을 통하여 얻은 250개의 ASCII코드에 해당하는 문자를 16×8 격자(128비트)로 추출한 이진 데이터이다. [그림 3]은 ASCII코드 '65'에 해당하는 알파벳 대문자 'A'를 나타내고 있다[7].

```
.....  
.....  
.....  
....1...  
.1111.  
.11.11.  
.11...11  
.11..11.  
.1111111  
.11...11  
.11..11  
.11...11  
.....  
.....
```

[그림 3] 정규화된 입력패턴(알파벳 'A')

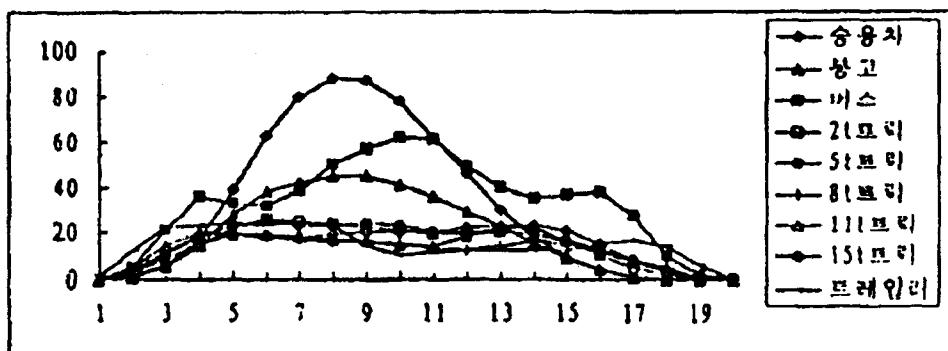
3.2 9개 차종의 기초자료

본 논문에서는 이용한 9개 차종의 기초자료는 고속도로관리시스템(Freeway Traffic management System, FTMS)구축의 일환으로 중부고속도로에 설치된 총 320Km의 구간 중 판교-구리간의 설치된 직경 1.8m크기의 원형검지기(ILD)로부터 1994년 11월 13일~1994년 11월 19일까지의 데이터이다. 이러한 원형검지기로부터 수집되는 자료는 통행차량이 검지기를 점유하는 동안 루프코일에서 발생되는 자장의 변화를 A/D변환장치에 의하여 변환된 정보로써 9개 차종에 관한 자료를 기초자료로 수집하였으며, 각 수집된 자료의 현황은 [표 1]과 같다[8].

[표 1] 자료수집현황

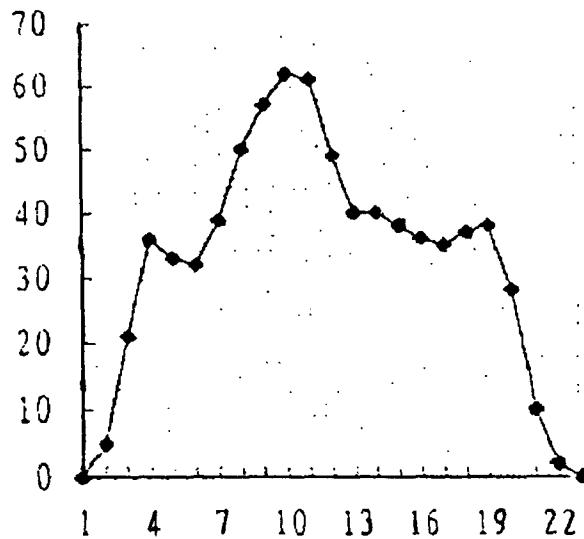
구 분	자료의 수	비 고
승 용 차	250 set	
봉 고	349 set	불량 2 set
버 스	46 set	
2톤 트럭	205 set	불량 2 set
5톤 트럭	83 set	
8톤 트럭	34 set	
11톤 트럭	35 set	
15톤 트럭	109 set	불량 1 set
트레일러	18 set	

이들 자료들은 2차원의 표현시 수직 성분은 자속의 변화이며 수평 성분은 샘플링 순서이다. 각 차종별 수집된 자료에 근거한 대표적인 파형의 특성은 [그림 4]와 같다.



[그림 4] 9개 차종의 샘플링된 파형

이 수집된 자료들은 이상적인 상태(점유하는 동안 등속도로 통행하는 경우)에서는 그 해당차종에 대한 고유의 차속변화에 대한 특징이 명확하여 차종을 식별하는데 매우 용이하게 접근할 수 있다. 그러나 실제 상황에서는 [그림 5]와 같이 점유시간중 부분적 지체에 의한 다양한 파형변화가 있을 수 있으며, 파형 변화가 일정하다 하더라도 차량 점유시간의 다양한 변화로 이상적인 상태에서 차량이 갖는 고유의 파형을 파괴하는 요인이 된다.



[그림 5] 비등속에 의한 파형

이러한 현상은 평상시 등속도로 통행되는 고속도로 보다는 통행속도의 완급이 빈번한 도시가로 상에서 자주발생될 수 있는 요인이 된다. 이러한 고유 파형의 파괴는 다음과 같은 상황에서 발생된다.

첫째, 고속도로의 경우 검지기가 설치된 지역에서 사고에 의한 급진적 정체시 혹은 급정거의 경우에 발생될 수 있다.

둘째, 도시가로망의 경우 검지기가 정지선 부근과 상류부(링크의 길이가 300m 기준시 약 250m 부근에 설치된 경우)에 설치된 경우 상류부 혼잡 및 정지선에서의 출발손실시간에 따른 속도의 완급변화에 따라 파형의 변화가 발생될 수 있다.

이러한 현상은 도로를 통행하는 차량들의 비등속도 통행에 의하여 나타날 수 있는 결과라 할 수 있다. 비 등속도 통행에 의한 파형의 변화는 어느 정도의 수준까지는 인식분야에 있어서 효과적으로 활용될 수 있는 소위 전처리과정의 다양한 방법들을 토대로 극복할 수는 있으나 근본적인 해결방법은 될 수 없는 요소이다.

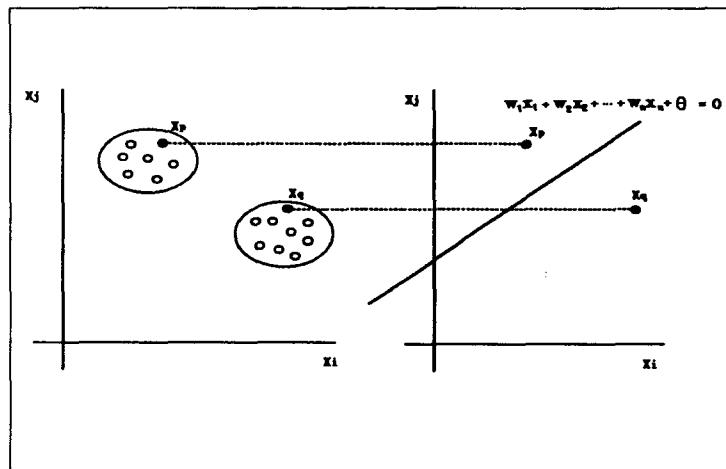
본 연구에서는 이러한 비등속도 통행으로 인한 부분적 증가분의 지체요인을 배제한 정상적인 등속도로 고속도로를 통행하는 차량들에 준하여 수집된 자료들을 이용하였다. 또한 하나의 보완장치로서 비등속도 통행에 의한 변화를 어느정도 완화시킬 수 있으며 자료특성의 동질성을 유지시킬 수 있는 방법으로 자속변화 및 점유시간의 변화에 대하여 각각을 정규화하는 전처리 과정을 이용하였다.

4. 병렬 신경망 모델

4.1 기본 원리

패턴 $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset R^p$ 를 LVQ 클러스터링 알고리즘을 이용하여 c -class로 클러스터링을 하면 각 클러스터에는 $\{n_1, n_2, \dots, n_c\}$, $\sum_i n_i = n$ 개의 패턴이 분포하게 된다. 여기서 각 클러스터에는 상호 유사성이 있는 패턴끼리 구성된다. LVQ에서 이루어진 클러스터링은 각 클러스터의 중심값이 입력노드와 출력노드 사이를 연결해 주는 연결강도가 되며, 이 연결강도를 중심으로 유클리디안 거리가 짧은 패턴이 한 클러스터로 속하게 된다. 패턴을 구분하는데 있어서 패턴간의 거리가 멀리 떨어져 있을 경우 즉, 각 패턴들간의 유사성이 없는 경우에 신경망에서 패턴을 구분하는 것이 인식률을 향상 시킬 뿐만 아니라 학습 시간도 현저히 줄일 수가 있다([그림 6] 참조). 예로써 ASCII코드값 179에서 245까지의 semi-graphic코드 66개의 패턴을 단순 퍼셉트론(simple perceptron)에 입력하였을 경우 인식률이 43%에 불과하며, 반복횟수를 증가하여도 더 이상의 인식은 할 수가 없었다. 그러나 알파벳 대문자 'A'에서 'Z'까지의 패턴과 소문자 'a'에서 'z'까지의 패턴 48개를 학습시켰을 경우는 반복횟수 7,921회만에 100%학습된 결과가 나왔다.

본 논문에서 제안한 알고리즘의 기본원리는 N 개의 패턴을 c 개의 클러스터로 나눈 후에 패턴끼리의 거리가 멀리 떨어져 있는 패턴끼리 추출하여(c 개) 이 패턴을 간단한 네트워크(ADALINE 네트워크)에 입력하여 학습률을 향상시키고 학습의 시간을 단축하는 데 있다.



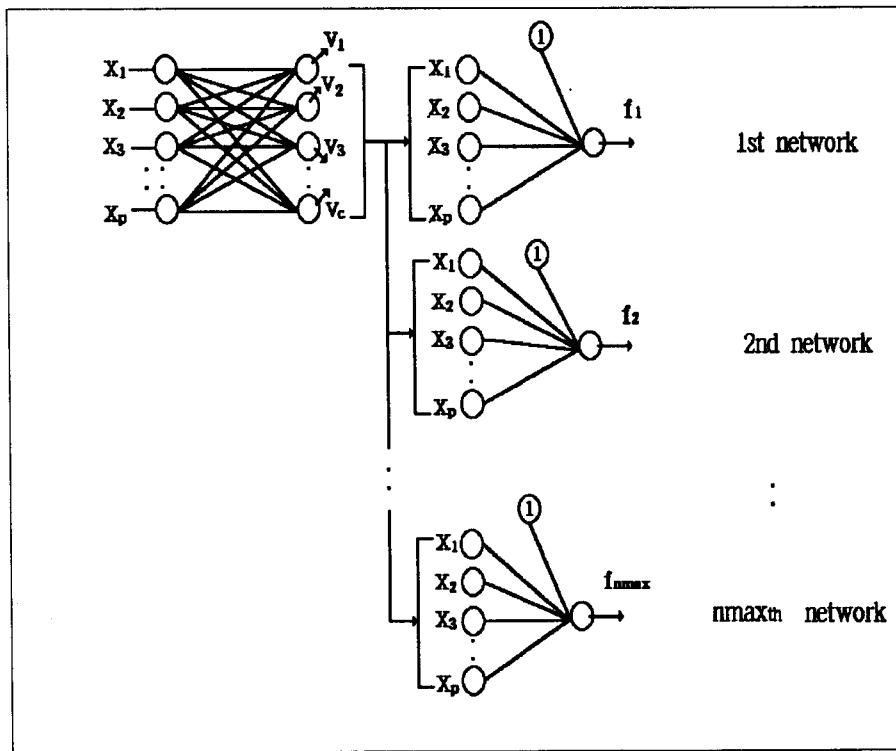
[그림 6] 알고리즘의 기본원리

4.2 네트워크구성

본 알고리즘은 각 클러스터에서 i 번째 $1 \leq i \leq n_{\max}$ (n_{\max} :maximum number of patterns in each cluster) 패턴들을 추출하여 c 개씩 n_{\max} 개로 재구성하였고, 이 재구성한 C 개의 패턴

을 입력층의 노드 p 개와 출력층의 노드 1개의 ADALINE으로 학습을 시킨다.

본 논문에서 제안한 네트워크의 구성은 [그림 7]과 같다.



[그림 7] 제안한 네트워크 구성

4.3 학습알고리즘

본 논문에서 이용한 알고리즘 중 ADALINE 학습알고리즘은 출력값을 패턴의 레이블로 선형결합 시킨 학습알고리즘을 개발하여 이를 적용하였다. 즉, Widrow와 Hoff(1960)가 제안한 출력값은 0과 1사이의 출력값만이 나올 수 있었으나 본 논문에서는 다음과 같이 학습알고리즘을 수정하여 적용하였다.

[Step 1] Initialize weights(Small random values).

Set learning rate a .

Given data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$.

Determine the teaching data : $T_i = i$ (PatternLabel), $i = 1, \dots, n$

[Step 2] While $T_i \neq y_{in}$, do Step 3-4.

[Step 3] Compute net input to output unit :

$$y_{in} = b + \sum_i x_i w_i$$

if ($y_{in} < 0$) then $f = -y_{in} \times \text{coordination rate}$
 if ($y_{in} > 0$) then $f = y_{in} \times \text{coordination rate}$

[Step 4] Update bias and weights, $i = 1, 2, \dots, n$:
 $b(\text{new}) = b(\text{old}) + \alpha(1 - y_{in})$
 $w_i(\text{new}) = w_i(\text{old}) + \alpha(T_i - y_{in})x_i$

[Step 5] Test for stopping condition:

If the largest weight change that occurred in Step 3 is smaller than a specified tolerance, then stop ; otherwise continue

여기서 조정률(coordination rate)은 교사신호가 1에서 n개(패턴의 수)로 되어 있기 때문에 출력값이 그 범위를 초과하였을 경우 이를 1과 n사이로 출력할 수 있도록 경험적으로 주게 되어 있다. 본 논문에서는 9개 차종 데이터를 학습할 때에는 0.0013을 주었으며, ASCII문자를 학습할 경우에는 0.00013의 값을 주었다.

4.4 인식 알고리즘

학습이 완료된 다음 정규화된 미지(unknown)의 패턴 x_{unknown} 이 입력되었을 때 인식하는 알고리즘은 우선 입력된 패턴과 가장 가까운 클러스터를 발견하고, 그 클러스터에 속해 있는 패턴들 중 가장 오차가 적은 패턴의 위치 i ($1 \leq i \leq n_{c, \max}$, where, $n_{c, \max}$: c 클러스터에 속해있는 학습된 패턴의 수)를 찾는다. 다음 i 번째 ADALINE네트워크의 입력패턴으로 입력하여 연결강도와 곱하여 출력값을 산출한다.

본 알고리즘은 다음과 같다.

[Step 1] Find the nearest cluster

$$\|x_{\text{unknown}} - v_i\| = \min_{1 \leq j \leq c} \{\|x_{\text{unknown}} - v_j\|\}$$

[Step 2] Find the location i ($1 \leq i \leq n_{\max}$) which is nearest learned pattern vector x_{ci}

$$i = \min_{1 \leq j \leq n_{\max}} \{\|x_{\text{unknown}} - x_{cj}\|\}$$

[Step 3] Calculate the output

$$f = b + \sum_{k=1}^n x_{\text{unknown}} w_k$$

5. 실험 및 결과분석

본 알고리즘에 적용한 데이터는 전처리과정을 통하여 얻은 250개의 ASCII코드에 해당하는 문자를 16×8 로 정규화한 데이터와 ILD에서 얻은 9개 차종별 기초데이터 1,124개이다. ASCII문자 데이터인 경우 250개의 패턴을 LVQ를 이용하여 6개의 클러스터로 나누었으며, 9개 차종 데이터는 448개의 학습 패턴을 9개의 클러스터로 나누었다. 여기서 각 패턴의 래

이들은 ASCII문자의 경우 각 ASCII코드값이 되며, 9개 차종 데이터는 [표 2]와 같다.

[표 2] 9개 차종 데이터의 패턴 레이블

차종	레이블
11톤	0 - 13
15톤	14 - 56
2톤	57 - 137
5톤	138 - 170
8톤	171 - 184
봉고	185 - 322
버스	323 - 340
승용차	341 - 440
트레일러	441 - 447

9개 차종 데이터인 경우 학습패턴은 각 차종별 표본크기의 30%를 추출하여 같은 차종끼리 순차적으로 레이블을 결정지었다.

각 클러스터로 나눈 결과는 [표 3]과 [표 4]와 같다.

[표 3] 6개 클러스터에 포함된 ASCII코드 문자 패턴

[표 4] 9개 클러스터에 포함된 차종 패턴

클러스터	패턴 레이블														
클러스터 0	237 282 320 332 343 345 347 348 352 374 380 381 383 384														
	395 398 403 405 406 412 416 418 421 426 427 434 436														
클러스터 1	342 388 389 396 399														
클러스터 2	190 196 198 201 204 205 212 213 214 219 230 234 235 242														
	243 246 248 252 253 254 257 258 263 264 268 270 272 274														
	275 277 281 283 284 285 287 289 291 294 302 303 306 315														
	317 321 323 325 326 327 328 330 331 333 334 335 336 337														
	338 339 340														
클러스터 3	359 362 390 408 422 431														
클러스터 4	74 80 88 101 103 105 122 126 130 158 165 185 187 188														
	189 191 192 193 194 195 197 199 200 203 206 207 208 209														
	210 211 215 216 217 218 220 221 222 223 224 225 226 227														
	228 229 231 232 233 236 238 239 241 244 245 247 249 250														
	255 256 259 260 261 262 265 266 267 269 271 273 276 278														
	279 280 286 288 293 295 296 297 298 299 300 301 304 305														
	307 308 309 311 312 313 314 316 318 319 322 324 329														
클러스터 5	349 350 356 358 364 369 370 373 375 376 379 382 386 391														
	393 397 400 402 410 413 420 423 424 430 433 437 438 439														
클러스터 6	341 344 346 351 353 354 355 357 360 361 363 368 371 372														
	377 385 387 392 401 404 407 409 417 419 429 432 435 440														
클러스터 7	365 366 367 378 394 411 414 415 425 428														
클러스터 8	0 1 2 3 4 5 6 7 8 9 10 11 12 13														
	14 15 16 17 18 19 20 21 22 23 24 25 26 27														
	28 29 30 31 32 33 34 35 36 37 38 39 40 41														
	42 43 44 45 46 47 48 49 50 51 52 53 54 55														
	56 57 58 59 60 61 62 63 64 65 66 67 68 69														
	70 71 72 73 75 76 77 78 79 81 82 83 84 85														
	86 87 89 90 91 92 93 94 95 96 97 98 99 100														
	102 104 106 107 108 109 110 111 112 113 114 115 116 117														
	118 119 120 121 123 124 125 127 128 129 131 132 133 134														
	135 136 137 138 139 140 141 142 143 144 145 146 147 148														
	149 150 151 152 153 154 155 156 157 159 160 161 162 163														
	164 166 167 168 169 170 171 172 173 174 175 176 177 178														
	179 180 181 182 183 184 186 202 240 251 290 292 310 441														
	442 443 444 445 446 447														

본 논문에서 학습을 시킨 ADALINE네트워크는 ASCII문자 데이터인 경우 128개 9개 차종 데이터인 경우 188개의 네트워크를 별별로 학습을 시킨 결과 ASCII 문자데이터인 경우 모든 ADALINE네트워크에서 학습이 100% 완료되었고, 9개 차종 데이터에서는 [표 5]와 같은 결과가 나왔다.

[표 5] 9개 차종 학습 데이터의 에러율

Error 발생 번째 네트워크	Error 율(%)
2	66.67
3	77.78
4	88.89
5	100.00
6	37.50
9	71.43
21	100.00
22	100.00
23	100.00
24	100.00
26	100.00
27	100.00
28	100.00

학습이 완료된 후 ASCII문자 데이터인 경우에는 250개 패턴 전체에 대하여 문자별로 3%, 5%의 노이즈를 추가하여 인식을 시켰으며, 9개 차종 데이터인 경우에는 1124개의 데이터를 입력하여 인식을 시켰다. 또한 같은 데이터를 이용하여 역전파 학습 알고리즘에 적용하여 그 인식률을 비교한 결과 두 데이터 모두 역전파 학습 알고리즘 보다는 인식률의 향상이 있었음을 알 수 있다. 여러번의 시행착오를 거쳐 은닉층 2개층과 각 은닉층의 유니트 수가 9개, 18개의 유니트로 구성하였을 경우 에러율이 2.8%로 최소가 되는 모델을 선정하였다. 그 결과는 [표 6] 및 [표 7]과 같다.

[표 6] ASCII코드 문자 인식 결과

적용 알고리즘	ASCII 코드 문자 인식률	
	3% 노이즈	5% 노이즈
Backpropagation	-	-
Parallel Neural Model	76.4%	71.6%

(주) - 는 학습이 안된 상태

ASCII문자 데이터인 경우 역전파 학습알고리즘에 적용하였을 경우 학습의 에러율이 3.2에서 3.9까지 나와 학습을 하지 못하는 것으로 나타났다.

[표 7] 9개 차종 데이터 인식 결과

차 종	입력 패턴수	Backpropagation		Parallel Neural Model	
		인식 패턴수	인식율	인식 패턴수	인식율
11톤	35	24	0.685714	6	0.171429
15톤	108	0	0.000000	84	0.777778
2톤	203	0	0.000000	161	0.793103
5톤	83	41	0.493976	50	0.602410
8톤	34	0	0.000000	30	0.882353
봉고	347	339	0.976945	307	0.884726
버스	46	42	0.913043	7	0.152174
승용차	250	234	0.936000	153	0.612000
트레일러	18	12	0.666667	9	0.500000
총 계	1124	692	0.615658	807	0.717972

인식 결과 역전파 학습 알고리즘인 경우에는 봉고, 버스 승용차에서 좋은 인식률을 보였으나, 트럭류에 대해서는 인식률이 저조함을 알 수 있다. 그러나 전체적인 인식률을 보면 본 논문에서 제안한 알고리즘에 적용하였을 경우 10%정도 인식률이 향상되었음을 알 수 있다.

6. 결론

본 논문에서는 비교사학습인 LVQ 클러스터링 알고리즘과 교사학습인 ADALINE 알고리즘을 결합하여 병렬로 학습시키는 방법을 제안하였다. 본 알고리즘의 특징은 다음과 같다.

첫째, 일반적으로 교사학습, 특히 역전파 학습 알고리즘인 경우 학습시간이 너무 많이 소요되는 단점이 있는 반면에 본 알고리즘은 전방향 학습이면서 병렬로 학습을 하기 때문에 학습시간의 단축될 수가 있다.

둘째, N개의 패턴이 모두 하나의 네트워크에 입력되는 것이 아니라 이 패턴들을 구분지어 각각 다른 네트워크에서 학습을 시키기 때문에 인식률이 향상될 수가 있다. 실험 결과 9개 차종 데이터인 경우 약 10%의 인식률의 향상을 보였다.

본 논문은 그 응용범위를 ASCII 문자와 9개 차종구분 데이터에 한정시켰고, 또한 통계학 또는 수학적인 증명과정이 미흡하기 때문에 앞으로의 연구에서는 보다 다양한 패턴과 학습 과정의 증명이 필요할 것으로 생각된다.

참 고 문 헌

- [1] 김태수, 신경망 이론과 응용 (I),(II), 하이테크정보, 1994.
- [2] Laurene Fausett, "Fundamentals of Neural Networks - Architectures, Algorithms

- and Applications," Prentice-Hall, Inc., 1994.
- [3] Terence D. Sanger, "Optimal Unsupervised Learning in a Single-layer Linear Feedforward Neural Network," *Neural Networks*, Vol. 2, 1989, pp. 459-473.
 - [4] J. Hertz, A. Krogh, R. G. Palmer, "Introduction to the Theory of Neural Computation," Addison Wesley Publishing, 1991.
 - [5] Nikhil R. Pal, James C. Bezdek, Eric C. K. Tsao, "Generalized Clustering Networks and Kohonens Self-Organizing Scheme," *IEEE Trans. on Neural Networks*, Vol. 4, No. 4, 1993, pp. 549-557.
 - [6] 김대극, 이상희, 김백섭, "변형된 CNN에 기초한 LVQ2의 초기벡터 결정방법", 정보과학 회논문지, Vol. 21, No. 9, pp. 1675-1682, 1994. 9.
 - [7] 민준영, 조형기, "Generalized Clustering Network을 이용한 전방향 학습알고리즘," 정보처리논문지, 1995. 9., pp. 619-625.
 - [8] 조형기, 민준영, 최종욱, "클러스터링 방법을 이용한 차종인식 모형," 정보처리논문지, Vol. 3, No. 2, pp. 369-380, 1996. 3.
 - [9] Youngjik Lee and Sang-Hoon Oh, "Input Noise Immunity of Multilayer Perceptrons," ETRI Journal, VOL. 16, NO. 1, Apr. 1994, pp. 35-43.