

패턴인식의 MLP 고속학습 알고리즘

(A Fast-Learning Algorithm for MLP in Pattern Recognition)

이 태 승[†] 최 호 진^{**}

(Tae-Seung Lee) (Ho-Jin Choi)

요 약 MLP(multilayer perceptron)는 다른 패턴인식 방법에 비해 여러 가지 훌륭한 특성을 가지고 있어 패턴인식에서 폭넓게 사용되고 있다. 그러나 MLP의 학습에 일반적으로 사용되는 EBP(error backpropagation) 알고리즘은 학습시간이 비교적 오래 걸린다는 단점이 있다. 패턴인식에 사용되는 학습 데이터는 풍부한 중복특성을 내포하고 있으므로 패턴마다 MLP의 내부변수를 갱신하는 온라인 계열의 학습방식이 속도의 향상에 상당한 효과가 있다. 일반적인 온라인 EBP 알고리즘에서는 내부변수 갱신시 고정된 학습률을 적용한다. 고정 학습률을 적절히 선택함으로써 패턴인식 응용에서 상당한 속도개선을 얻을 수 있지만, 학습률이 고정되고 학습이 진행됨에 따라 학습패턴 영역이 달라지는 학습과정의 각 단계에 효과적으로 대응하지 못하는 문제가 있다. 이 문제에 대해 본 논문에서는 학습과정을 세 단계로 정의하고, 각 단계별로 필요한 패턴만을 학습에 반영하는 패턴별 가변학습속도 및 학습생략(ILVRS) 방법을 제안한다. ILVRS의 기본개념은 다음과 같다. 학습단계마다 학습에 필요한 패턴의 부분이 달라지므로 이를 구별하여 학습에 적용할 수 있도록 (1)패턴마다 발생하는 오류치를 적절한 범위 이내로 제한하여 가변 학습률로 사용하고, (2)학습이 진행됨에 따라 불필요한 부분의 패턴을 학습에서 생략한다. 제안한 ILVRS의 성능을 입증하기 위해 본 논문에서는 패턴인식 응용의 한 갈래인 화자증명을 실험하고 그 결과를 제시한다.

키워드: 다층신경망, 오류역전파 알고리즘, 최대기울기감소 방법, 고속학습 알고리즘, 화자증명, 패턴인식

Abstract Having a variety of good characteristics against other pattern recognition techniques, Multilayer Perceptron (MLP) has been used in wide applications. But, it is known that Error Backpropagation (EBP) algorithm which MLP uses in learning has a defect that requires relatively long learning time. Because learning data in pattern recognition contain abundant redundancies, in order to increase learning speed it is very effective to use online-based learning methods, which update parameters of MLP pattern by pattern. Typical online EBP algorithm applies fixed learning rate for each update of parameters. Though a large amount of speedup with online EBP can be obtained by choosing an appropriate fixed rate, fixing the rate leads to the problem that the algorithm cannot respond effectively to different learning phases as the phases change and the learning pattern areas vary. To solve this problem, this paper defines learning as three phases and proposes a Instant Learning by Varying Rate and Skipping (ILVRS) method to reflect only necessary patterns when learning phases change. The basic concept of ILVRS is as follows. To discriminate and use necessary patterns which change as learning proceeds, (1) ILVRS uses a variable learning rate which is an error calculated from each pattern and is suppressed within a proper range, and (2) ILVRS bypasses unnecessary patterns in learning phases. In this paper, an experimentation is conducted for speaker verification as an application of pattern recognition, and the results are presented to verify the performance of ILVRS.

Key words: multilayer perceptron, MLP, error backpropagation, EBP, steepest gradient descent, fast learning algorithm, speaker verification, pattern recognition

[†] 학생회원: 한국항공대학교 항공전자공학과
thestaff@hitel.net

^{**} 종신회원: 한국정보통신대학교 공학부 교수
hjchoi@icu.ac.kr

논문접수: 2001년 11월 21일

심사완료: 2002년 2월 14일

1. 서론

패턴인식이란 현실의 개체를 몇 가지 특징을 통해 추상화하고 이를 확률적 모델로 정의한 뒤 모델과 관련된 데이터가 관측될 때 이 모델의 사전적 지식을 활용하여

모델의 선택을 결정하는 처리를 말한다. 데이터의 관측과 모델의 선택이 동시에 일어날 확률은 두 사건의 교집합의 확률이고, 이 확률은 Bayes의 법칙에 따라 아래와 같이 표현될 수 있다.

$$P(\Theta, \mathbf{x}) = P(\Theta | \mathbf{x})p(\mathbf{x}) = p(\mathbf{x} | \Theta)P(\Theta)$$

$$P(\Theta | \mathbf{x}) = \frac{p(\mathbf{x} | \Theta)P(\Theta)}{p(\mathbf{x})} \quad (1)$$

여기서, \mathbf{x} 는 관측 데이터, Θ 는 선택할 모델을 나타낸다. 식(1)의 의미는 관측 데이터 \mathbf{x} 가 주어질 때 모델 Θ 를 선택할 확률(이를 사후확률이라고 한다.)은 모델 Θ 에 한정하여 얻어진 데이터 발생확률(이를 유사도라고 한다.)과 모델 Θ 자체의 확률(이를 사전확률이라고 한다.)을 전체 데이터 \mathbf{x} 가 발생할 확률로 나누었을 때 평가될 수 있다는 것이다.

식(1)의 유사도 또는 사후확률을 계산하기 위해 다양한 방법이 사용된다. 이들을 크게 하부확률분포 정보를 이용하는 파라메트릭(parametric) 방식과 이를 이용하지 않는 논파라메트릭(nonparametric) 방식으로 나누는데, 최대 유사도 예측(Maximum Likelihood Estimation)과 Bayesian 예측 방법이 전자에 속하고 Pazen 윈도우, k-최근접 예측 방법, 신경망이 후자에 속한다[1]. 한편, 특징 데이터를 추출할 때 효과적인 패턴인식을 위해 관측 데이터 공간을 모델을 가장 잘 표현하는 특징 공간으로 변환하는 PCA(principal component analysis)나 LDA(linear discriminant analysis) 등의 방법이 추가로 사용되기도 한다[1].

여러 가지 형태를 갖는 신경망 중에서도 최대 기울기 감소(Steepest Gradient Descent) 방법에 기초하여 패턴 데이터를 학습하는 MLP(multilayer perceptron)는 기존에 사후확률을 계산하기 위해 널리 사용되던 학습 및 인식 방법들에 비해 다음과 같은 특성을 갖는다[2][3].

- 논파라메트릭 방식이기 때문에 문제에서 가정해야 하는 하부확률분포가 필요없다.
- 학습되는 각 모델 사이의 차이를 최대한 구별하는 거부학습능력이 있기 때문에 인식오류 가능성을 최소화한다.
- 학습모델별로 +1, 0(또는 -1)의 학습목표치를 사용할 때 LDA와 유사한 특징공간 변환능력을 갖는다.

MLP는 0개 이상의 은닉 뉴런층과 1개의 출력 뉴런층으로 구성되며, 은닉층에서 MLP의 학습 복잡도를 결정하고 출력층에서 학습모델의 개수를 결정한다. MLP에서 학습을 위해 사용하는 오류역전파(error back-propagation, 이하 EBP) 알고리즘은 최대 기울기 감소

방법을 바탕으로 한 것으로, 신경망의 현재출력과 목표 출력 사이의 오류를 출력층에서 은닉층으로 역방향으로 전파하면서 내부변수를 조정하는 방법으로 최종적인 목표치를 달성한다[3].

각 내부변수의 현재 값에 대해 목표치에 가장 빠르게 접근할 수 있는 변위는 아래와 같이 계산된다.

$$\frac{\partial e}{\partial w_{ij}} = \frac{\partial e}{\partial s_i} \frac{\partial s_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \quad (2)$$

여기서, e 는 임의의 오류측정 함수이고, w_{ij} 는 j번째 뉴런과 i번째 뉴런을 연결에 가중을 가하는 내부변수치, s_i 는 i번째 뉴런의 동작치, net_i 는 i번째 뉴런의 가중된 입력의 총합을 나타낸다. 이 식에서 계산된 변화량을 이전의 내부변수에 적용하면 목표치에 더욱 가까운 값을 도출할 수 있으며, 이를 아래의 방법으로 실현한다.

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial e(t)}{\partial w_{ij}(t)} \quad (3)$$

여기서, t 는 내부변수의 특정 상태 시각을 나타내며, η 는 적용할 변화량의 크기를 결정하는 상수(이를 학습률 또는 갱신간격이라고 한다.)를 가리킨다.

식(3)에서 분명하게 드러나 있듯이 목표치를 향한 가중치 변화는 학습률 η 에 상당한 영향을 받는다. η 가 너무 크면 학습이 빠르게 이루어지지만 출력치가 목표치 근방에서 진동하면서 수렴하지 못할 가능성이 증가한다. 반면에 η 가 너무 작으면 수렴하기까지 너무 많은 갱신회수가 필요해지며, 경우에 따라서는 w 벡터가 로컬 미니마(local minima)에 빠져 목표치에 도달하지 못하는 수도 있다. 따라서 가장 적절한 학습속도를 알아내기 위해 η 를 여러 값으로 변경해가며 반복 실험하여 수렴이 보장되는 최대의 수치를 선택한다.

일반적으로 위와 같이 최대의 학습속도를 사용하더라도 MLP의 학습에는 상당한 시간이 소요된다. 이는 최대 기울기 감소 방법이 현재의 w_{ij} 에 대한 지역적인 정보만 사용하는 것에서 연유한다. MLP의 학습속도를 개선하려는 시도는 크게 두 방향으로 이루어졌다. 첫 번째 방향은 식(3)의 경험과 실험결과를 활용한 것으로, 출력치가 목표치에서 멀 경우에는 η 를 크게 하고 가까운 경우에는 작게 하는 것이다. 이것은 다시 내부변수 전체에 일괄적으로 영향을 미치는 전역 η 를 변경하는 방법 [4]과 각 내부변수마다 식(2)의 변화에 따라서 지역 η 를 변경하는 방법 [5]으로 나뉜다. 두 번째 방향은 최적화 이론을 활용한 것으로, 내부변수의 2차 미분정보를 사용한다. 이러한 부류로는 모멘텀을 사용하여 이전의

학습양상을 현재 갱신에 반영하거나[4], 뉴턴(Newton)의 최적화 이론[6] 또는 이를 변형한 알고리즘[7][8]을 이용하여 목표치로 가장 빠르게 수렴할 수 있는 내부변수 갱신치를 계산하는 방법이 있다.

한편, 식(3)의 내부변수 갱신은 두 가지 방식으로 이루어진다. 하나는 모든 학습데이터를 제시한 후 그에 따른 변경치들의 평균을 적용하는 방법이고, 다른 하나는 학습데이터를 하나씩 제시할 때마다 변경치를 적용하는 방법이다. 전자를 오프라인(또는 일괄적) 방법이라고 하고, 후자를 온라인(또는 확률적) 방법이라고 부른다. 상기 소개한 학습속도 개선방법은 모두 오프라인 방식으로 분류된다. 양쪽 모두 모든 학습데이터를 제시하는 한 주기를 에폭(epoch)이라고 하고, 에폭마다 목표치와 현재 MLP의 출력치의 차이인 오류치를 검사하여 학습의 수행여부를 결정한다.

패턴인식에서는 모델마다 다수의 패턴이 포함되고 학습은 각 모델을 구별할 수 있도록 진행된다. 일반적으로 모델 내의 패턴 사이의 차이는 모델 간 패턴 사이의 차이보다 훨씬 작다. 이런 이유로 인해 패턴인식에서 MLP를 사용할 경우 오프라인 학습보다 - 대규모 데이터를 학습하는 경우 심지어 위에서 소개한 여러 개선 알고리즘보다도 - 온라인 학습이 훨씬 빠른 속도로 이루어진다. 온라인 학습의 빠른 속도는 다음과 같은 이유에 기인한다[9].

- 모델 내의 모든 패턴이 서로 유사하기 때문에 패턴마다 식(3)에 의한 최대 기울기 계산에 기여하므로 모델에 포함된 패턴수가 많을수록 에폭 단위의 학습속도가 빨라진다.
 - 식(3)에 의해 계산되는 최대 기울기가 90°이내일 경우 오류를 최소화하는 방향으로 학습이 진행된다.
 - 로컬 미니마에 빠질 가능성을 크게 줄인다. 이러한 특성은 모델 내의 모든 패턴마다 내부변수의 갱신이 이루어질 때 중심위치에서 상대적으로 멀리 떨어진 패턴에 의해 진행방향과 다른 임의적 이동이 발생하기 때문이다.
 - 함수 근사화 응용 등과는 달리 패턴인식에서는 오류 목표가 상대적으로 엄격하지 않다. 패턴인식에서 표준 온라인 EBP 알고리즘을 이용하는 학습의 경우 초기에는 오류치 하락률이 급격하다가 일정수준부터는 완만해지는데, 인식을 하라키 크지 않은 수준에서 오류 목표를 최대한 느슨하게 잡으면 초기의 급격한 오류치 하락 특성에 의해 빠른 학습속도가 달성된다.
- 위와 같은 온라인 학습방법의 이점을 활용하기 위해 뉴턴 최적화 계열의 오프라인 학습속도 향상 알고리즘

에 온라인 특성을 결합한 방법이 소개되었다[7][10].

이처럼 온라인 학습방법이 오프라인 방법에 비해 상당한 유리점을 내포하고 있지만, 그럼에도 불구하고 여전히 개선할 수 있는 여지가 남아있다. 기존의 온라인 학습방법에서는 고정 학습률을 사용했는데, 이것은 온라인 학습에서 모델 내의 패턴이 맡는 역할이 학습이 진행됨에 따라 달라지는 현상을 이용하지 못하는 것이다. 온라인 학습시 MLP는 3단계의 학습단계를 밟는다. 즉, (1)먼저 모델의 중심위치를 학습하고 (2)모델의 분산을 학습한 다음 (3)모델분포의 윤곽을 학습한다. 후자 쪽으로 진행해 갈수록 학습에 관여하는 패턴의 부분이 일부로 한정되는데, 고정 학습률은 이러한 변화를 수용하지 못하므로 최적 학습을 달성하기까지 소요되는 시간이 길어진다.

이러한 사실을 기반으로 하여 본 논문에서는 온라인 학습방법에서 고정 학습률의 단점을 개선하여 패턴마다 적용되는 가변 학습률을 도입하고 학습의 진행단계에 따라 불필요한 패턴의 학습을 생략하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서 고정 학습률을 사용하는 온라인 EBP 학습의 문제점을 설명한 뒤에 대한 해결책을 3절에서 제안한다. 제안한 방법이 패턴인식에서 유효한지 확인하기 위해 화자증명 실험을 통한 입증을 4절에서 시도한 후, 5절에서 본 논문의 주장을 마무리한다.

2. 고정 학습률 사용 온라인 학습의 문제

패턴인식에 사용되는 MLP의 출력 동작함수로는 바이너리 시그모이드(binary sigmoid) 또는 바이폴라 시그모이드(bipolar sigmoid) 형태를 사용한다. 전자의 동작치가 0~+1.0인데 비해 후자의 동작치는 -1.0~+1.0이다. 지금까지의 경험을 통해 바이폴라 시그모이드가 더 빠른 학습속도를 낸다는 사실이 밝혀졌다[3]. 일차 미분함수의 유도 편의성을 고려하여 일반적으로 사용되는 바이폴라 시그모이드 함수는 다음과 같다.

$$y(x) = \frac{2}{1+e^{-x}} - 1 \quad (4)$$

MLP의 각 패턴마다 학습상황을 평가할 오류함수는 다음과 같다.

$$e_{o,p}(t) = d_{o,p}(t) - y_{o,p}(t) \quad (5)$$

여기서, t 는 에폭이고, o 는 출력뉴런, P 는 학습패턴, d 는 학습 목표치, y 는 실제 출력치를 의미한다. 학습 목표치는 o 에 속한 모델의 경우 +1.0이고 그 외는 -1.0이

다. 에폭마다 학습성고를 평가할 오류함수 $e_{av}(t)$ 는 다음과 같다.

$$e_p(t) = \frac{1}{2} \sum_{o=1}^C e_{o,p}^2(t) \quad (6)$$

$$e_{av}(t) = \frac{1}{N} \sum_{p=1}^N e_p(t) \quad (7)$$

여기서, C 는 출력노드 개수를, N 은 전체 패턴 개수를 나타낸다. 오프라인 학습과 온라인 학습은 모두 식(7)의 값을 통해 학습중지 여부를 결정한다.

온라인 학습에서 각 패턴에 의해 계산되는 식(2)의 값은 식(5)의 영향을 크게 받는다. 즉, 현재 패턴이 만들어내는 출력뉴런의 값이 목표치와 큰 차이를 보이면 식(2)의 값이 커지고, 출력치가 목표치에 근접하면 식(2)의 값은 작아진다. 식(2) 값이 클수록 현재 학습에 미치는 영향이 크고, 학습에폭이 증가할수록 각 패턴의 식(2)의 값은 감소한다. 이 관계에 의해 온라인 학습에서 각 모델은 다음 3단계의 과정을 거쳐 학습된다.

- (1) 중심위치
- (2) 분산(대략경계)
- (3) 윤곽(상세경계)

학습 초기에는 각 모델 내의 모든 패턴이 큰 식(2) 계산치를 보이기 때문에 모든 패턴이 소속 모델의 중심 위치를 다른 모델의 중심위치와 구별하는 학습에 기여한다. 다수의 패턴에 의해 중심위치가 충분히 학습되었으면 그 다음으로 모델의 대략경계에 분포한 패턴에 의해 모델의 분산이 학습되게 되는데, 다른 분산을 갖는

두 모델의 경우 모델 간 구별경계는 분산이 낮은 쪽에 가까이 위치하게 된다. 그 뒤에는 각 모델의 상세한 경계 주변에 위치한 소수의 패턴에 의해 윤곽을 구분하는 학습이 이루어진다. 이 3단계의 학습은 학습목표의 허용 오류를 얼마나 낮게 주느냐에 따라 일부만 이뤄질 수도 있고 지나치게 세밀한 모델경계(사실상 잡음에 가까운)까지 학습할 수도 있다. 또한, 마지막 단계까지의 학습이 이뤄질 수 있으려면 MLP의 은닉층 및 은닉뉴런의 수가 모델을 표현하기에 충분한 수준이어야 한다.

그림 1에서 두 패턴집단의 MLP 학습에서 일어나는 위 3단계의 학습과정을 보여주고 있다. 여기서 (b)~(c)의 그림이 (1)~(3)의 단계와 대응한다. (b)~(c) 그림에서 검은색은 출력뉴런의 낮은 값을, 하얀색은 높은 값을 나타내므로 검은 영역이 두 집단의 경계가 된다. 학습이 시작되면 얼마 지나지 않아 (b)처럼 두 집단의 중심 위치를 가르는 경계가 학습된다. 그 뒤 상당량의 학습이 이뤄지고 (c)처럼 두 집단의 데이터 분산에 따른 대략경계가 밝혀진다. 그리고 마지막으로 (d)처럼 두 집단의 상세한 경계가 확립된다.

그림 2에서 이러한 학습과정의 원리를 집단을 구분하는 경계선(즉, 내부변수 값)에 미치는 패턴의 척력개념으로 설명하고 있다. 학습이 일어나기 전에는 모든 패턴이 상대집단의 중심위치를 향해 최대한의 척력을 갖는다(그림 2의 (a)~(b)). 중심위치가 결정되고 나면 그 경계선의 인근에 있는 패턴과 경계선 넘어 상대집단의 영역에 들어있는 패턴의 척력에 의해 집단의 분산정도가 경계선에 반영되고(그림 2의 (b)~(c)), 이어진 학습에

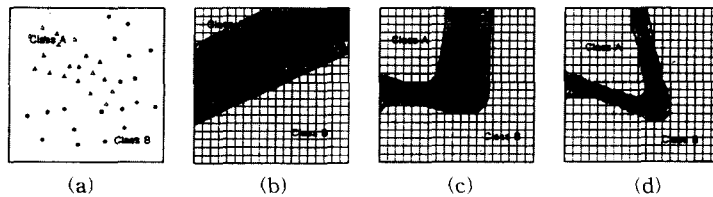


그림 1 (a) 2집단 데이터 분포, (b) 중심위치학습, (c) 분산학습, (d) 윤곽학습.
온라인 MLP 학습의 3단계

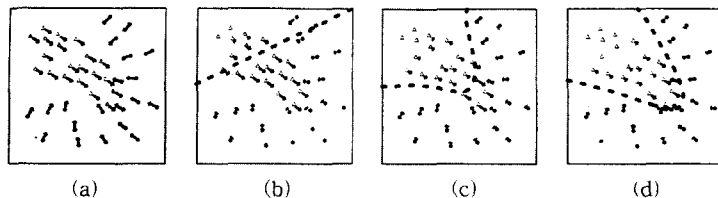


그림 2 온라인 학습의 패턴별 척력

의해 최종적인 윤곽이 결정된다(그림 2의 (c)~(d)).

상기와 같은 온라인 학습의 과정을 감안할 때 학습률을 고정하면 두 가지 문제가 발생한다.

먼저, 상기 학습과정에서 학습단계마다 요구되는 학습률의 크기가 달라진다. 큰 학습률로는 중심위치를 빠르게 찾은 뒤 신속하게 분산과 윤곽을 학습할 수 있지만 학습률이 문제에서 요구되는 수준보다 지나치게 크면 수렴위치 부근에서 진동하여 목표치 수렴이 어렵게 된다. 반면, 작은 학습률로는 분산과 윤곽을 정밀하게 학습할 수 있지만 중심위치를 찾는 데 많은 시간이 소요될 수 있고, 자칫 이 과정에서 로컬 미니마에 빠질 우려도 있다. 이 때문에 대부분의 패턴인식 응용에서 최단의 학습속도와 최고의 인식률을 동시에 얻지 못하며, 학습률을 달리한 반복실험을 통해 두 지표 사이에 타협점을 찾는 것이 일반적이다[11].

그리고, 학습단계가 진행될수록 일부분의 패턴만 유효하므로 유효하지 않은 패턴에 의한 불필요한 학습과 이에 따른 방해가 일어날 수 있다. 즉, 모델의 중심, 분산, 윤곽은 해당 위치의 패턴에 의해 지배적으로 학습되므로 분산이 학습될 때는 중심 위치에 분포한 패턴의 필요성이, 윤곽이 학습될 때는 중심과 분산 위치에 분포한 패턴의 필요성이 낮아지는 것이다. 더군다나 경우에 따라서는, 출력뉴런에서 입력의 가중된 합이 충분히 커서 0에 가까운 1차 미분함수치가 계산됨에 따라 발생하는 학습속도 저하를 회피할 목적으로 목표치를 함수 최대치의 90% 수준으로 지정하기도 하는데[3], 이 경우 불필요해진 중심이나 분산 위치의 패턴이 계속해서 학습에 관여함으로써 이후 학습단계에 부정적인 영향을 미칠 수 있다.

온라인 학습의 효율을 높이기 위해서는 위와 같은 문제점을 해결해야 한다.

3. 패턴별 가변학습속도 및 학습생략(ILVRS) 방법

2절에서 언급한 문제를 해결하기 위해 본 논문에서는 온라인 학습에 적용할 두 가지 개선방법을 제안한다.

첫 번째는 학습률과 학습단계 사이의 부조화를 해결하는 방법이다.

학습률은 중심위치, 분산, 윤곽의 순서로 큰 값에서 작은 값으로의 변화가 필요하다. 한편, 윤곽을 학습하는 단계에서 모델영역이 한쪽으로 치우친 외형을 갖는 경우(즉, 중심위치에서 경계까지의 위치가 방위에 따라 큰 차이를 보이는 경우) 일시적으로 큰 학습률을 적용하여 경계의 기형적인 요철을 학습할 필요도 있다. 따라서 학

습률은 이런 점을 고려하여 모델의 학습단계와 패턴마다 변동될 수 있어야 한다.

온라인 학습방법을 사용할 때 보편적으로 선택되는 학습률의 범위는 1~0.0001이다[11]. 이 범위에서 상한을 넘어서면 내부변수의 값이 발산하기 쉽고, 하한을 넘어서면 학습이 불필요하게 길어진다. 2절에서 언급한 대로 학습속도를 빠르게 하기 위해 모델의 목표치를 90%로 줄여서 사용하는 경우 식(7)의 $e_{\alpha, \beta}^2(t)$ 는 0~+3.24 사이의 값을 갖게 된다. 이러한 $e_{\alpha, \beta}^2(t)$ 의 값은 현재 패턴의 오류수준을 나타낼 뿐 아니라 적절한 학습률 범위에 들어가기 때문에 현재 패턴의 학습률로 사용하기에 적당하다. 단, 실험결과 온라인 학습에서 3이상의 학습률을 사용할 경우 발산하기 쉬우므로 $e_{\alpha, \beta}^2(t)$ 의 계산치를 다시 아래 식에 집어넣어 값을 0~2 사이로 억제한다.

$$y(x) = \frac{4}{1+e^{-x}} - 2 \quad (8)$$

학습률로 사용할 $e_{\alpha, \beta}^2(t)$ 는 출력뉴런의 값 중 가장 큰 값을 선택한다. 이 학습률은 초기 에폭에서는 평균적으로 큰 값을 유지하다가 에폭이 진행될수록 작아지지만 돌출된 모델경계에서는 일시적으로 큰 값이 되기도 한다.

두 번째 방법에서는 학습과정이 진행됨에 따라 불필요해지는 패턴을 처리한다.

EBP에서 이뤄지는 주요 계산은 패턴의 오류계산, 오류 역전파, 내부변수 갱신이다. 현재 학습단계에서 현재 패턴의 기여도는 오류치 $e_{\alpha, \beta}^2(t)$ 를 통해 알 수 있으므로, $e_{\alpha, \beta}^2(t)$ 의 값이 학습 전에 설정하는 식(7)의 최종목표 오류치보다 작은 경우 현재 패턴이 학습에 기여하는 바가 적다고 판단하여 오류 역전파와 내부변수 갱신 계산과정을 생략할 수 있다. 만일 이후 다른 패턴의 학습으로 현재 패턴의 기여도가 높아진다면 이 상태는 오류치에 의해 발견되기 때문에 다시 학습에 참여할 수 있게 된다.

학습속도를 높이는 효과 외에도 이 방법의 적용으로 온라인 학습방법에서 추가로 요구되는 메모리나 계산이 거의 없다는 점도 큰 장점이다. 본 논문에서는 이 두 방법을 패턴별 가변학습속도 및 학습생략(Instant Learning by Varying Rate and Skipping) 방법이라고 부른다.

4. 실험

3절에서 제안한 ILVRS 방법의 증명을 위해 표준 오프라인 EBP와 이것의 속도향상 버전인 VLR(variable learning rate)[4] 및 RPROP(resilient propagation)[5], 그리고 표준 온라인 EBP의 성능을 ILVRS 방법과 비

교하여 제시한다. 비교방법은 먼저 간단한 삼각함수 근사화(function approximation) 실험을 통해 기존에 보고되었던 위 알고리즘들의 성능비교 결과[4]를 확인한 뒤, 음성데이터를 이용한 화자증명(speaker verification)에서 화자를 증명시스템에 등록하고 증명시험을 수행하는 인식기로서 MLP를 사용할 때 화자를 등록하는 MLP 학습의 속도차이를 비교하여 대규모 데이터를 사용하는 패턴인식의 경우에서 ILVRS의 성능우위를 입증한다. [4]에서 수행한 실험결과에 따르면 패턴인식에서 RPROP의 성능은 오프라인 계열의 여타 속도개선 알고리즘 중에서 최고수준으로 나타났으므로, 본 논문의 화자증명 실험결과에서 RPROP는 ILVRS의 좋은 비교대상이 될 수 있다.

4.1 비교 알고리즘

전술한 바와 같이 두 실험에 사용하는 MLP의 학습 알고리즘을 달리하여 제안한 ILVRS의 학습속도의 비교우위를 보인다. 서론에서 표준 EBP를 설명했고 3절에서 ILVRS를 설명했으므로 이 절에서는 VLR과 RPROP 알고리즘만 요약한다.

먼저, VLR은 식(2)와 식(3)의 학습방법을 약간 수정한 형태이다.

(1) 초기 MLP 출력치와 오류치를 계산한다.

(2) 현재 학습률을 사용하여 내부변수를 갱신하고 다음 예폭에서 다시 새로운 출력과 오류를 계산한다.

(2-1) 새로 계산한 오류가 이전 예폭의 오류를 일정한 비율(1.04) 이상 초과할 경우 내부변수 갱신을 취소하고 학습률을 일정한 비율(0.7)로 감소시킨다.

(2-2) 그렇지 않다면 내부변수의 갱신상태를 그대로 두고 학습률을 일정한 비율(1.05)로 증가시킨다.

다음으로 RPROP는 전역 학습률을 변경하는 대신 식(2)의 정보를 이용하여 내부변수 별로 학습률을 변경한다. 즉, 이전 예폭과 현재 예폭의 식(2) 값이 같은 부호를 갖는 경우에는 아직 최적 오류에 도달하지 않았으므로 학습률을 증가시켜서 내부변수를 갱신하고, 다른 부호를 갖는 경우에는 최적 오류를 지나친 것이므로 이전 갱신을 취소하고 학습률을 감소시킨 후 내부변수를 다시 갱신한다. 이 과정을 아래에 의사코드로 나타내었다.

```

if (  $\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0$  ) then {
     $\Delta w_{ij}(t) = \min(|\Delta w_{ij}(t-1)| * \eta^+, \Delta_{max})$ 
     $\Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta w_{ij}(t)$ 
     $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$ 
}

```

```

else if (  $\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0$  ) then {
     $\Delta w_{ij}(t) = \max(|\Delta w_{ij}(t-1)| * \eta^-, \Delta_{min})$ 
     $w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij}(t-1)$ 
     $\frac{\partial E}{\partial w_{ij}}(t) = 0$ 
}
else if (  $\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0$  ) then {
     $\Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta w_{ij}(t-1)$ 
     $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$ 
}

```

여기서, \min 과 \max 는 각각 최소, 최대 요소를 선택하는 함수이고, η^+ 와 η^- 는 각각 학습률의 증가(1.2), 감소비율(0.5)을, Δ_{max} 와 Δ_{min} 은 각각 부동소수점 연산의 오버플로우와 언더플로우를 막기 위한 상한(1.0)과 하한($1.0e-6$)을 나타낸다. RPROP에서는 초기 학습률 변화율 Δ_0 의 선택이 알고리즘 속도에 큰 영향을 주므로 여러 값으로 변경하면서 최적의 값을 알아내야 한다. 한편, 위 알고리즘에서는 이전 예폭과 현재 예폭의 식(2) 값이 다른 부호를 가질 때 이전 갱신을 취소하는데, 이 부분을 아래처럼 변경하면 학습속도의 증가효과가 있다.

$$w_{ij}(t+1) = w_{ij}(t) - \delta(t) * \Delta w_{ij}(t-1), \quad 0 < \delta(t) < 1 \quad (9)$$

여기서 δ 는 0~1 사이의 임의수를 발생시키는 함수이다. 이 방법은 내부변수 갱신취소에 임의성을 적용하여 로컬 미니마에 빠질 가능성을 낮춘다.

실험에서 표준 EBP의 오프라인 버전은 OffEBP로, 온라인 버전은 OnEBP로 표기한다. OffEBP와 OnEBP 실험에서는 최적 학습률을 결정하기 위해 0.01~10 사이의 학습률에 대해 실험한다. VRL은 최초 학습률의 크기에 상관없이 일정한 학습속도를 보인다. RPROP은 초기 학습률 변화율 Δ_0 에 따라 학습속도가 달라지므로 Δ_0 를 0.1~10.0 사이의 값으로 변경하면서 실험한다. RPROP는 기존의 버전을 OldRPROP로, 식(9)을 적용한 버전을 NewRPROP로 표기한다. ILVRS는 학습률을 가변하는 방법만 적용하는 버전을 ILVR로, 여기에 불필요한 패턴의 계산을 생략하는 기능을 추가한 버전을 ILVRS로 표기한다.

4.2 삼각함수 근사화 실험

4.2.1 실험데이터 및 MLP

함수 근사화는 일정한 값 범위를 갖는 x 와 그 범위

에 대한 함수 $f(x)$ 가 주어졌을 때 MLP가 이들 사이의 내부관계를 학습하게 하는 MLP 응용이다. 이 실험에서는 $(x | 0 \leq x \leq 1)$ 과 $f(x) = \sin(2\pi \cdot x)$ 가 주어지는데, x 의 간격을 0.05로 하였으므로 입력 데이터와 출력 데이터는 각각 21개가 된다.

이 실험에 사용하는 MLP는 입력층, 은닉층, 출력층을 갖는 구조로 되어 있으며, 각기 1, 5, 1개의 뉴런으로 구성된다. 은닉층 및 출력층의 모든 뉴런은 동작함수로 식(4)를 가지며, 모든 내부변수는 학습 전 $-0.5 \sim +0.5$ 의 임의수치로 초기화된다. 21개의 x 데이터가 모두 입력되었을 때를 1에폭으로 하고, 식(7)의 계산치가 0.005 이하가 되는 때를 학습목표로 한다. 임의수치를 이용하여 내부변수를 초기화하면 학습시도마다 결과가 달라지므로 속도측정치는 30회의 실험결과를 평균한 것을 기록한다.

4.2.2 결과 및 분석

그림 3~6은 학습방법마다 에폭수에 영향을 미치는 매개변수를 변화시키면서 최소 에폭수를 찾는 과정을 보인다. 이들 그림에서 알 수 있듯이 이러한 매개변수의 영향이 상당하므로 MLP를 활용하는 문제마다 정밀하게 탐색할 필요가 있다. 그림 7에서는 6개 학습방법의 최저 에폭수 비교결과를 보인다.

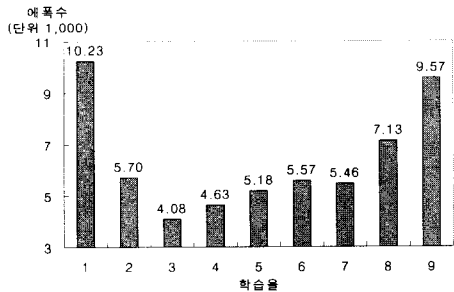


그림 3 OffEBP: 학습률에 따른 에폭수 변화

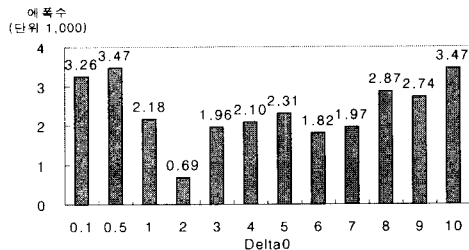


그림 4 OldRPROP: Δ_0 에 따른 에폭수 변화

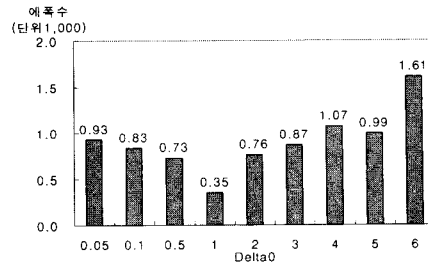


그림 5 NewRPROP: Δ_0 에 따른 에폭수 변화

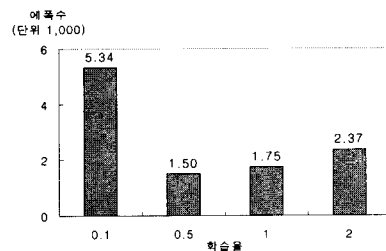


그림 6 OnEBP: 학습률에 따른 에폭수 변화

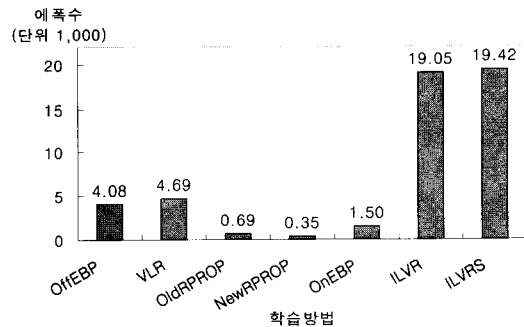


그림 7 학습방법별 에폭수 비교결과

그림 7의 결과는 MLP 구조의 차이에 의한 절대수치의 편차를 제외하면 [4]에서 제시된 순위와 일치한다. 실험에서 가장 빠른 학습방법은 NewRPROP로 나타났으며 가장 느린 방법은 ILVRS로 나타났다. 이 결과는 그림 8에서 보듯이 에폭별 오류의 감소율이 OffEBP, VLR, ILVRS, OnEBP에서는 0.01이하에서 완만한 데 비해 RPROP에서는 일정 시점에서 급격히 감소하기 때문인 것으로 보인다. 이렇듯 함수 최적화 응용에서는 RPROP가 우수한 특성을 갖고 있지만, OnEBP와 ILVRS의 오류 감소율이 오류치 0.1 이하에서는 다른 어떤 방법보다도 급격하다는 점을 눈여겨볼 필요가 있다. 이 특성은 오류

목표가 엄격하지 않은 패턴인식에서 이점으로 부각된다.

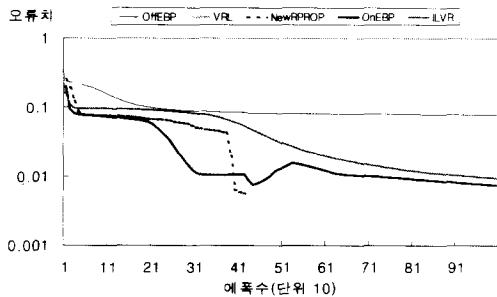


그림 8 학습방법별 에폭에 따른 오류 감소추이

4.3 화자증명 실험

4.3.1 화자증명 시스템

화자증명이란 개인별 목소리의 차이를 이용하여 신원을 주장하는 화자가 이전에 시스템에 등록된 화자인지 가려내는 음성처리를 가리킨다[12]. 개인에 따라 음성이 다른 이유는 성대에서 발생한 음원이 구강과 비강으로 이루어진 성도를 따라 공진하면서 음향을 만들어낼 때 성도의 물리적 구조(길이, 용적 등)가 사람마다 다르기 때문이다. 일반적으로 동일한 언어단위(문장, 단어, 음소 등)를 발성을 기준으로 비교할 때 각 화자의 고유특성이 잘 드러나기 때문에 본 논문의 화자증명 시스템은 음소를 기반으로 화자를 구별하는 방식을 채택한다[13]. 화자를 시스템에 등록하는 방법은 화자가 발성한 고립단어(앞뒤에 묵음을 두고 발성한 음성)에서 음소를 추출한 뒤 각 음소에 대해 미리 준비된 배경화자의 음소음성과 MLP를 통해 비교 학습함으로써 실현한다. 신원을 주장하는 의뢰화자를 증명하는 방법은 의뢰화자가 발성한 고립단어에서 음소를 추출하고 이들을 등록시 학습된 MLP에 입력하여 출력된 값을 적절한 문턱값과 비교함으로써 실현한다. 본 논문에서는 화자를 구별하는 언어단위로 한국어 지속음 (/a/, /e/, /ɛ/, /o/, /u/, /i/, /ɪ/, /ɯ/, 비음)을 사용한다.

화자증명 시스템의 기본구조와 처리흐름을 그림 9에 도시했으며, 이 그림의 각 부분의 처리를 요약하면 다음과 같다.(음성인식과 관련한 내용은 [14]에 자세히 정리되어 있다.)

(1) 음성분석 및 특징추출

- 16bit 16kHz로 샘플링된 입력음성을 20ms 오버랩 시킨 30ms 길이의 프레임으로 나눈다.
- 각 프레임에 대해 16차 Mel 간격 필터뱅크를 추출하여 고립단어 및 지속음 검출에 사용한다. 필터뱅크 계수는 전체 스펙트럼 포락에 미치는 성량의 영

향을 제거하기 위해 1kHz까지의 계수를 평균하여 모든 계수에서 차감한 뒤, 다시 모든 계수의 평균이 0이 되도록 조정한다.

- 각 프레임에 대해 50차의 0~3kHz 대역 균등간격 Mel 필터뱅크를 추출하여 화자증명에 사용한다. 이 음성특징은 2차 포먼트에 더 많은 화자정보가 집중된다는 연구결과[15]에 의한 것이다. 필터뱅크 계수는 전체 스펙트럼 포락에 미치는 성량의 영향을 제거하기 위해 1kHz까지의 계수를 평균하여 모든 계수에서 차감한 뒤, 다시 모든 계수의 평균이 0이 되도록 조정한다.
- (2) 고립단어 검출 및 언어단위 검출
 - 각 지속음과 묵음을 화자독립 방식으로 검출하도록 학습된 MLP를 사용하여 고립단어를 검출한다.
 - (3) 언어단위별 등록화자 학습
 - 화자등록 단계일 경우 MLP와 배경화자 데이터를 이용하여 등록화자를 학습한다.
 - (4) 언어단위별 화자점수 평가
 - 화자증명 단계일 경우 각 지속음 프레임을 해당 MLP에 입력하고 그 점수를 평균한다.
 - (5) 등록어 및 화자점수 문턱값 비교
 - (4)에서 계산한 평균값과 사전 설정한 문턱값을 비교하여 거부 및 수락여부를 결정한다.

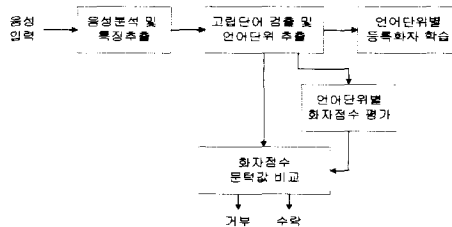


그림 9 화자증명 시스템 개략도

4.3.2 실험데이터

이 비교실험에 사용할 데이터는 한국인 남녀 40명의 4연숫자 발성을 녹음한 것이다. 여기서 4연숫자라 함은 아라비아 숫자 0~9에 해당하는 /goN/, /il/, /i/, /sam/, /sa/, /o/, /yug/, /cil/, /pa/, /gu/ 음을 연속해서 4자리를 발성한 것을 의미한다. 각 화자가 총 35개의 서로 다른 숫자음 배열을 4회씩 발성하였고, 발성은 16kHz 주기의 16bit 크기로 녹음되었다. 4회 발성 중 3회를 각 화자의 등록음성을 사용하고 나머지 1회를 증명시험 음성으로 사용한다. 화자증명에서의 인식성능은 등록화자가 증명시도를 했을 때 미리 정한 문턱값에 대해 거부되는 비율(False Reject Rate)과 사칭화자가 증명시도를 했을

때 수락되는 비율(False Accept Rate)로 구분하여 평가한다. 따라서 화자사칭을 위한 음성이 필요한데, 이 실험에서는 40명 가운데 현재 등록된 화자를 제외한 나머지 39명의 4회 숫자음 발생에서 1회 발생을 이 목적으로 사용한다. 결과적으로 화자당 35회의 FR 시도와 1,560회의 FA 시도를 평가하게 되고, 실험 전체로 봤을 때는 1,400회의 FR 시도와 54,600회의 FA 시도를 평가하게 된다.

한편, 등록화자 학습시 등록화자와의 차이학습에 필요한 배경화자는 위의 40명 외의 남녀 29명의 데이터를 사용한다. 모든 화자에 대해 9개 지속음에 대해 배경화자 당 10개의 음성패턴을 PBW(phoneme balanced word)에서 무작위 추출했다. 따라서 배경화자의 데이터 개수는 지속음 당 290개가 된다.

4.3.3 MLP

이 비교실험에 사용하는 MLP는 입력층, 은닉층, 출력층을 갖는 구조로 되어 있으며, 각기 50, 2, 1개의 뉴런으로 구성된다. 은닉층 및 출력층의 모든 뉴런은 동작 함수로 식(4)를 갖는다. 입력 데이터는 -1.0~+1.0로 평균화되며, 학습속도 향상을 위해 각 출력뉴런의 목표치는 등록화자에 +0.9, 배경화자에 -0.9를 지정한다. 모든 내부변수는 학습 전 -0.5~+0.5의 임의수치로 초기화된다. 학습시 두 모델의 패턴은 교대로 MLP에 제시되는데, 거의 대부분의 경우 두 모델의 학습패턴수가 일치하지 않으므로 많은 쪽 패턴이 모두 제시될 때까지 적은 쪽 패턴을 반복해서 제시하여 1에폭을 채운다. 최대 학습에폭은 로컬 미니마에 빠지는 경우를 고려하여 1000회로 제한하고, 학습목표는 식(7)의 계산치가 0.01 이하가 되는 동시에 충분한 학습이 이뤄지도록 이 값의 변화율[3]이 0.01 이하가 되는 것으로 한다.

4.3.4 결과 및 분석

학습속도는 등록화자 40명이 35개 어휘에 대해 등록할 때마다 이뤄지는 MLP 학습을 통해 측정했다. 따라서 측정회수는 총 1,400회가 되며, 각 측정시 이뤄지는 평균 MLP 학습회수는 각 어휘에 포함된 평균 지속음 개수와 같은 6.9회이다. 마찬가지로 오류율 역시 등록화자 40명이 35개 어휘에 대해 등록할 때마다 측정했으며, 측정방법은 4.2절에서 서술한 바와 같으나 FRR 또는 FAR 대신 문턱값을 조정하여 이 두 지표가 동일해지는 EER(equal error rate)을 기록했다.

삼각함수 근사화 실험과 마찬가지로 그림 10~13은 학습속도와 인식률에 영향을 미치는 매개변수를 변화시키면서 두 지표의 추이를 탐색하는 모습을 보인다. 각 학습방법에서 최단의 학습시간을 기록한 결과는 그림 14에서 정리한다. 이 그림에서는 이전 그림들과 달리 예

폭수 외에 학습시간을 추가로 기록하고 있는데, 이는 ILVRS의 경우 ILVR에 비해 예폭수에는 차이가 거의 없지만 학습패턴 생략방식으로 인한 실질적인 시간단축 효과가 있기 때문에 이를 보이기 위한 것이다. 이 외에도 학습방법 별로 1에폭에 필요한 계산량이 다르기 때문에 시간을 기준으로 한 비교가 중요한 의미를 갖는다. 시간의 측정에 사용된 컴퓨터는 1.4GHz AMD™ 마이크로프로세서를 탑재한 PC 머신이다.

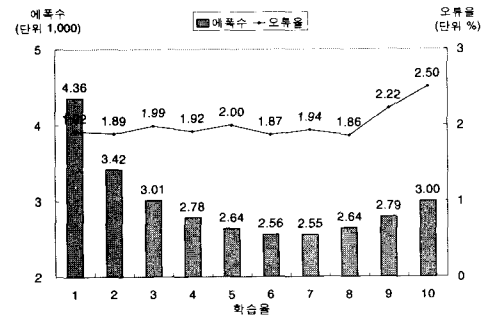


그림 10 OffEBP: 학습률에 따른 예폭수 및 오류율 변화

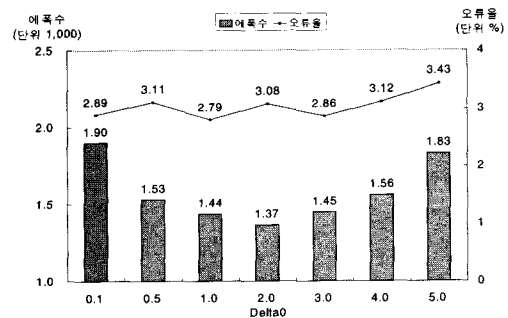


그림 11 OldRPROP: Δ에 따른 예폭수 및 오류율 변화

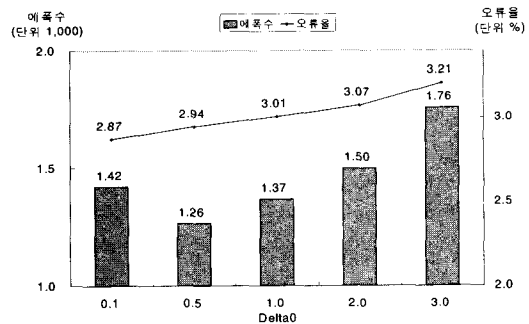


그림 12 NewRPROP: Δ에 따른 예폭수 및 오류율 변화

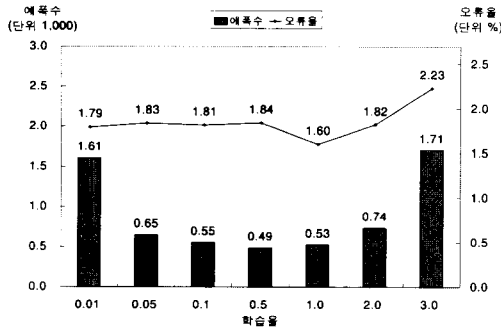


그림 13 OnEBP: 학습률에 따른 에폭수 및 오류율 변화

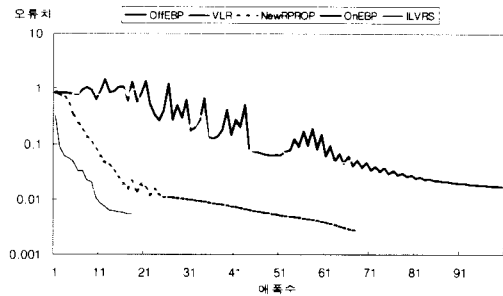


그림 15 학습방법별 에폭에 따른 오류 감소추이

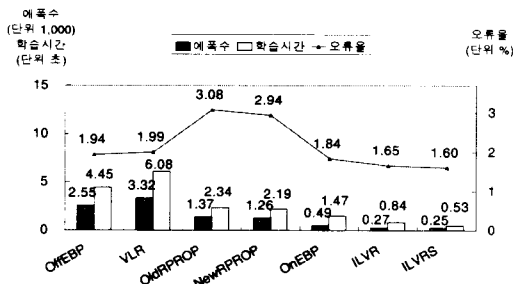


그림 14 학습방법별 에폭수, 학습시간, 오류율 비교결과

삼각함수 근사화 실험과는 달리 패턴인식에서는 학습 속도 뿐 아니라 인식률도 중요한 고려의 대상이 된다. 그림 14의 결과에서 볼 수 있듯이 VLR을 제외한 RPROP, OnEBP, ILVRS 방법이 나열된 순서대로 OffEBP보다 더 빠른 학습속도를 보여주고 있고, 오류율은 VLR, RPROP 방법을 제외한 기타 방법에서 나열된 순서대로 향상되고 있다. RPROP가 빠른 학습속도에 비해 인식률이 상대적으로 크게 저조한 문제는 차후에 연구해야 할 과제이다.

그림 14의 결과에 의하면 RPROP, OnEBP, ILVRS가 OffEBP에 비해 대략 2~9배까지 학습속도 면에서 우수한 것으로 나타난다. 이러한 결과는 그림 15의 에폭에 따른 오류치 감소추이에서 보듯이 이들 방법이 초기 에폭기간 동안 급격하게 오류를 감소시키기 때문인 것으로 보인다. 즉, 집단의 대략적인 위치를 빠르게 학습한 다음에 상세한 영역정보를 학습하기 때문에 처음부터 완전한 학습곡선을 그리는 OffEBP에 비해 빠른 학습속도를 보이는 것이다. RPROP 대 OnEBP 및 ILVRS 사이의 학습속도 차이는 역시 RPROP가 오프라인 계열의 학습방법이므로 초기위치 학습에 상대적으로 긴 시간이 필요한 것으로 나타난다.

OnEBP와 ILVRS는 어느 수준까지는 비슷한 학습곡선을 그리다가 ILVRS가 먼저 학습중지 조건을 만족시키는데, 이것은 2절에서 지적한 바와 같이 OnEBP의 고정 학습률과 불필요한 학습으로 인한 세부경계 학습의 비효율성을 ILVRS에서 해결했기 때문이다. 다시 말해, 에폭별 오류치는 동일한 추세로 줄어들지만, 식(7)에서 알 수 있듯이 이 값은 전체 학습패턴에 대한 평균치이기 때문에 세부영역의 학습상태를 알려주진 못한다. 세부영역 학습의 추이는 에폭에 대한 오류치보다는 패턴별 오류치를 관찰함으로써 알 수 있으며, 그림 16에서 이러한 OnEBP와 ILVRS의 패턴별 오류치를 보여주고 있다. 여기서 가로축의 눈금이 580개의 패턴마다 새겨져 있고 이것이 1에폭을 나타낸다. 그림에서 보듯이 같은 에폭 내에서도 패턴마다 오류치가 다른데, OnEBP는 이러한 차이를 전체적으로 완만하게 줄여나가는 데 비해 ILVRS는 초기 에폭기간 동안에는 급격한 학습행동을 보이다가 어느 순간부터 완만한 학습행동을 보인다. 즉, ILVRS가 복잡한 영역의 학습을 OnEBP보다 더 빨리 학습하고, 이에 따라 학습목표가 전반적으로 더 빠르게 만족될 수 있어 학습시간이 짧아지는 것이다.

한편, 학습방법에 따라 학습속도가 차이나는 원인을 로컬 미니마에 빠질 확률의 측면에서 고찰하는 일도 의미가 있다. 이것은 로컬 미니마에 빠지면 결국 학습시간의 증대로 이어지기 때문이다. 그림 17은 전체 실험에서 MLP를 학습시킨 회수에 대해 최대 에폭(1000)에 도달한 회수를 백분율로 나타낸 것이다. 가장 학습시간이 긴 VLR도 평균 MLP 학습 에폭수가 대략 580이었으므로 최대 에폭까지 오류목표를 만족하지 못한다는 것은 로컬 미니마에 빠진 것으로 보아도 무방하다. 그림 17의 추세는 그림13의 에폭수 추세와 거의 일치하는 것을 알 수 있고, 특히 온라인 계열의 학습방법에서는 로컬 미니마에 빠질 확률이 극히 낮다는 사실이 확인된다.

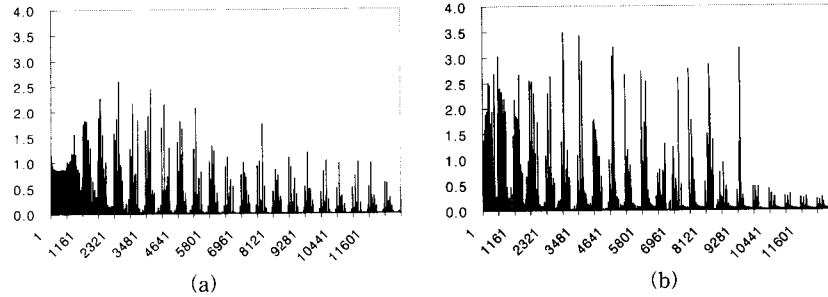


그림 16 (a) OnEBP와 (b) ILVRS의 패턴별 오류치 감소추이

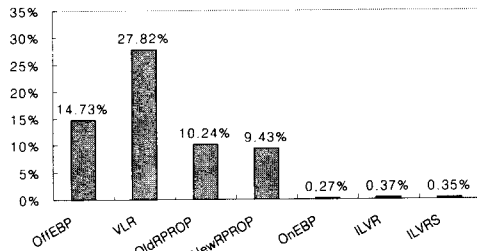


그림 17 학습방법별 로컬 미니마에 빠질 확률

ILVRS 방법의 이점을 요약하면 다음과 같다. 먼저 온라인 학습방법에 비해 추가로 요구되는 메모리와 계산량이 거의 없다. 그리고 학습시간 뿐 아니라 인식률 면에서도 최적에 가까운 수준을 기록한다.

5. 결론

MLP는 다른 패턴인식 방법에 비해 여러 가지 훌륭한 특성을 가지고 있어 패턴인식에서 폭넓게 사용되고 있다. 그러나 MLP의 학습에 일반적으로 사용되는 EBP 알고리즘은 학습시간이 비교적 오래 걸린다는 단점이 있다. 패턴인식에 사용되는 학습데이터는 풍부한 중복특성을 내포하고 있으므로 패턴마다 MLP의 내부변수를 갱신하는 온라인 계열의 학습방식이 속도의 향상에 상당한 효과가 있다. 일반적인 온라인 EBP 알고리즘에서는 내부변수 갱신시 고정된 학습률을 적용한다. 고정 학습률을 적절히 선택함으로써 패턴인식 응용에서 상당한 속도개선을 얻을 수 있지만, 학습률이 고정되고 학습이 진행됨에 따라 학습패턴 영역이 달라지는 학습과정의 각 단계에 효과적으로 대응하지 못하는 문제가 있다. 이 문제에 대해 본 논문에서는 학습과정을 세 단계로 정의하고, 각 단계별로 필요한 패턴만을 학습에 반영하는 ILVRS 방법을 제안했다.

ILVRS의 기본개념은 다음과 같다. 학습단계마다 학습에 필요한 패턴의 부분이 달라지므로 이를 구별하여 학습에 적용할 수 있도록 (1)패턴마다 발생하는 오류치를 적절한 범위 이내로 제한하여 학습률로 사용하고, (2) 학습이 진행됨에 따라 불필요한 부분의 패턴을 학습에서 생략한다. ILVRS의 성능을 입증하기 위해 오프라인 및 온라인 EBP와 기존에 제안된 VLR 및 RPROP 알고리즘의 학습속도를 비교하는 두 가지 실험을 실시하였다. 첫 번째 실험은 삼각함수 근사화 실험으로서 지금까지 보고된 이들 알고리즘의 성능우열을 확인하였고, 두 번째 실험은 화자증명 실험으로서 패턴인식에서 기존 학습방식에 대비한 ILVRS의 성능우위를 확인하였다.

ILVRS 방법의 이점을 정리하면, (1)온라인 학습방법에 비해 추가로 요구되는 메모리와 계산량이 거의 없고, (2)학습시간 뿐 아니라 인식률 면에서도 최적에 가까운 수준을 기록한다.

참고 문헌

- [1] R. O. Duda et. al., *Pattern Classification*, Wiley-Interscience, 2001.
- [2] H. Bourlard and N Morgan, "Hybrid Connectionist Models for Continuous Speech Recognition," *Automatic Speech and Speaker Recognition*, pp. 259-283, Kluwer Academic Publishers, 1996.
- [3] S. Haykin, *Neural Networks*, Prentice Hall, 1999.
- [4] H. Demuth and M. Beale, *Neural Network Toolbox*, The MathWorks, 2001.
- [5] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," *IEEE International Conference on Neural Networks*, pp. 586-591, Vol. 1, Mar 1993.
- [6] R. Fletcher, *Practical Methods of Optimization*, Wiley, 1987.
- [7] M. Moller, "Supervised Learning on Large Redundant Training Sets," *Proceedings of the*

1992 IEEE-SP Workshop Neural Networks for Signal Processing, pp. 79-89, Aug 1992.

- [8] S. Becker and Y. LeCun, "Improving the Convergence of Back-Propagation Learning with Second-Order Methods," in *Proceedings of the 1988 Connectionist Models Summer School*, pp. 29-37, 1988.
- [9] Y. Bengio, *Neural Networks for Speech and Sequence Recognition*, International Thomson Computer Press, 1995.
- [10] Y. LeCun et. al., "Automatic Learning Rate Maximization by On-Line Estimation of the Hessian's Eigenvectors," *Advances in Neural Information Processing Systems*, Vol. 5, pp. 156-163, 1993.
- [11] D. R. Wilson and T. R. Martinez, "The Need for Small Learning Rates on Large Problems," *International Joint Conference on Neural Networks*, Vol. 1, pp. 115-119, 2001.
- [12] J. M. Naik, "Speaker Verification: A Tutorial," *IEEE Communications Magazine*, Vol. 1, pp. 42-48, Jan 1990.
- [13] M. Savic and J. Sorensen, "Phoneme Based Speaker Verification," *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 165-168, 1992.
- [14] C. Becchetti and L. P. Ricotti, *Speech Recognition*, John Wiley & Sons, 1999.
- [15] P. Cristea and Z. Valsan, "New Cepstrum Frequency Scale for Neural Network Speaker Verification," *IEEE International Conference on Electronics, Circuits and Systems*, Vol. 3, pp. 1573-1576, 1999.



최 호 진

1982년 서울대학교 전자계산기공학과(학사). 1985년 영국 Univ. of Newcastle, Computing Laboratory(석사). 1995년 영국 Imperial College, Dept. of Computing(박사). 1982년 6월 ~ 1989년 8월 한국데이터통신(주) 정보통신연구소 선임연구원. 1995년 9월 ~ 1996년 11월 Imperial College, Planning Applications Research Centre 연구원. 1997년 3월 ~ 2002년 2월 한국항공대학교 항공전자공학과 교수. 2002년 2월 ~ 현재 한국정보통신대학교 공학부 교수. 관심분야는 인공지능, 논리프로그래밍, 구축조건 만족문제, 소프트웨어 공학.



이 태 승

1997년 한국항공대학교 항공전자공학과(학사). 2000년 한국항공대학교 항공전자공학과(석사). 2000년 ~ 현재 한국항공대학교 항공전자공학과 박사과정 재학중. 2000년 ~ 현재 (주)화음소 재직중. 관심분야는 음성인식, 패턴인식, 자연언어처리.

리, 인공지능