

論文96-33B-4-23

8비트 데이터 정밀도를 가지는 다층퍼셉트론의 역전과 학습 알고리즘 (Learning of Multi-Layer Perceptrons with 8-bit Data Precision)

吳相勳*, 宋玠宣*

(Sang-Hoon Oh and Yoon-Seon Song)

요 약

이 논문에서는 데이터 정밀도가 8비트인 하드웨어에 구현된 다층퍼셉트론에서의 역전과 학습 알고리즘을 제안한다. 제안한 방법은 크로스-엔트로피(cross-entropy) 오차함수를 사용하여 출력노드의 오차신호가 시그모이드 함수의 기울기 때문에 줄어들지 않도록 한다. 그리고, 가중치 합 표현 범위를 16비트로 확장시켜서 가중치 합 계산과정의 넘침(overflow)을 방지하며, 그 결과를 다음 단계 연산을 위해 8비트 데이터로 변환시 적정한 범위를 설정하여 데이터 표현 비트의 축소에 따른 오차를 최소화 한다. 전방향 계산에서는 시그모이드의 포화특성을 이용하여 비트-변환 범위를 설정하였고, 역방향 계산에서는 역전과 신호의 확률밀도함수를 유도하여 비트-변환 범위를 설정하였다. 제안한 방법의 학습성능을 CEDAR^[1] 데이터베이스의 필기체 숫자 인식 문제 시뮬레이션으로 확인한 결과, 16비트 데이터 정밀도의 역전과 학습알고리즘과 비슷한 정도의 일반화 성능을 보였다.

Abstract

In this paper, we propose a learning method of Multi-Layer Perceptrons(MLPs) with 8-bit data precision. The suggested method uses the cross-entropy cost function to remove the slope term of error signal in output layer. To decrease the possibility of overflows, we use 16-bit accumulator for the summation of 8-bit weighted signals and convert the 16-bit weighted sum results into the 8-bit data with appropriate range. In the forward propagation, the range for bit-conversion is determined using the saturation property of sigmoid function. In the backward propagation, the range for bit conversion is derived using the probability density function of back propagated signal. In a simulation study to classify handwritten digits in the CEDAR^[1] database, our method shows similar generalization performance to the Error Back-Propagation learning with 16-bit precision.

I. 서 론

신경회로망의 주된 응용분야는 패턴인식이며, 다층퍼셉트론이 가장 많이 사용되는 신경회로망 모델이다. 신경회로망의 실시간 응용을 위해서 주로 아날로그 혹은 디지털 방식의 하드웨어로 신경회로망을 구현하는 데, 이 두가지 방법 중 디지털 방식이 동작에 대한 신뢰도

측면에서 월등히 뛰어나다. 다층퍼셉트론을 디지털 하드웨어로 구현하여 패턴인식 문제에 응용하려면, 학습에 필요한 적정 데이터 표현 정밀도(data precision)를 결정해야 한다. 디지털 하드웨어의 데이터 표현 정밀도가 N 비트이면, 모든 값들은 제한된 범위 내에서 N 비트 정밀도에 의해 표현된다. N 비트 숫자 사이의 '×' 연산은 반드시 절삭(truncation)을 수반하므로, 절대치가 작은 숫자의 '×' 결과는 '0'이 된다(하위넘침 오차: underflow error). 그리고, '+'와 '-' 연산은 N 비트 표현에 의한 제한된 범위를 넘을 수 있다(범람오차:

* 正會員, 韓國電子通信研究所
(ETRI)

接受日字:1995年5月21日, 수정완료일:1996年3月15日

overflow error). 이들이 신경회로망의 디지털 하드웨어 구현 시 발생하는 오차의 주요 원인이다^{12, 31}. 일반적으로 실제적인 패턴인식 문제를 학습하기 위해서는 16 비트 이상의 데이터 표현 정밀도가 필요하다^{14, 5, 6}. 만약, 16 비트 보다 낮은 데이터 표현 정밀도로 학습이 가능하다면, 패턴인식 다층퍼셉트론을 디지털 하드웨어에 훨씬 경제적으로 구현할 수 있을 것이다.

디지털 하드웨어로 구현된 신경회로망에서 학습에 필요한 데이터 표현 정밀도를 낮추기 위해서, Hoehfeld와 Fahlman은 가중치의 변경량이 정밀도 보다 작은 경우 그 가중치가 확률적으로 바뀌도록 하였다¹⁵. 그리고, Choi 등은 가중치를 임의로 변경시키는 무작위 탐색(random search) 방법을 제안하였다¹⁷. 이 방법들은 오히려 디지털 하드웨어를 복잡하게 만들므로 경제성이라는 원래 취지가 퇴색된다. Andes 등은 각 노드 값을 변경시키면서 출력오차의 변화를 조사하여 그 결과에 따라 가중치를 변경시키는 MR(Madaline rule) III을 제안하였으며¹⁸, Jabri와 Flower는 가중치를 하나씩 선택하여 조금씩 변경시킨 경우에 출력의 오차 변화를 조사하여 그 결과에 따라 가중치를 변경시키는 weight perturbation 방법을 제안하였다¹⁹. 그렇지만, 이 두 방법은 가중치 혹은 노드 값을 하나씩 변경시키면서 출력오차를 계산하는 과정을 모든 가중치 혹은 노드를 대상으로 하여 실시해야 하므로 학습시간이 너무 많이 필요하다. Sakaue 등은 이와 같은 방법들과 달리 가중오차함수(Weighted Error Function)를 제안하였으며, 인체체 알파벳의 학습시뮬레이션으로 8 비트 데이터 정밀도에 의한 학습성능이 일반적인 방법보다 향상됨을 보였다¹³. 그렇지만, 이 방법은 16비트 데이터 정밀도에 의한 학습결과와 보다는 여전히 나쁘다.

이 논문에서는 확률론적인 방법을 사용하지 않고서, 실제적인 문제의 학습 시 16비트 정밀도에 의한 학습성능을 8비트 정수연산을 수행하는 디지털 하드웨어에서 얻는 방법을 제안한다. N 비트 디지털 연산을 이용한 다층퍼셉트론의 학습과정에서 출력노드의 오차신호 계산 시 하위넘침 오차를 줄이기 위해서 크로스-엔트로피 오차함수를 사용하며^{10, 11}, 가중치 합 계산과정의 범람 오차를 줄이기 위해 누산기(accumulator)의 크기를 $2N$ 비트로 한다. 그리고, 가중치 합 계산결과($2N$ 비트)를 N 비트 숫자로 변환 시 변환오차를 최소화시키기 위해서, 시그모이드 함수의 포화특성과 역전파 신

호의 확률밀도함수를 사용한다. 마지막으로, 학습의 조기포화 현상을 피하기 위해서 가중치 표현범위를 학습의 진행에 따라 변경시킨다. 이 논문의 2장에서는 위와 같은 방법에 대하여 설명하겠다. 3장에서는, CEDAR 데이터 베이스의 필기체 숫자인식 문제의 시뮬레이션으로 제안한 방법의 8비트 데이터표현에 의한 학습성능이 일반적인 학습방법의 16비트 데이터 표현에 의한 학습성능과 거의 차이가 없음을 보이겠다. 그리고, 4장에서 결론을 맺겠다.

II. 다층퍼셉트론의 디지털 학습방법

L 층으로 이루어진 다층퍼셉트론에서, l 층은 N_l 개의 노드로 구성되어 있으며, 그 상태벡터는 $x^{(l)} = [x_1^{(l)}, x_2^{(l)}, \dots, x_{N_l}^{(l)}]$, $l = 0, 1, 2, \dots, L$,로 표시된다. 여기서, $x^{(0)}$ 와 $x^{(L)}$ 은 각각 입력상태 벡터와 출력상태 벡터를 나타낸다. 또한, $t = [t_1, t_2, \dots, t_{N_L}]$ 는 임의의 입력벡터에 대한 목표벡터를 나타낸다. 이 다층퍼셉트론에 학습패턴 x^b 가 입력되면, 전방향 계산에 의해 각 층의 노드 상태가

$$x_j^{(l)} = f(a_j^{(l)}) = \frac{2}{1 + \exp[-a_j^{(l)}]} - 1 \quad (1)$$

와 같이 결정된다. 여기서,

$$a_j^{(l)} = w_{j\mu}^{(l)} + \sum_{i=1}^{N_{l-1}} w_{ji}^{(l)} x_i^{(l-1)} \quad (2)$$

이며, $w_{j\mu}^{(l)}$ 은 $x_i^{(l-1)}$ 과 $x_j^{(l)}$ 사이의 연결 가중치, $w_{j\mu}^{(l)}$ 은 $x_j^{(l)}$ 의 bias를 나타낸다.

이렇게 $x_k^{(l)}$ 이 구해지면, 다층퍼셉트론의 학습정도는 일반적으로 m.s.e. 함수

$$E_m(x^b) = \frac{1}{2} \sum_{k=1}^{N_L} (t_k - x_k^{(L)})^2 \quad (3)$$

에 의해 측정된다¹¹². $E_m(x^b)$ 를 최소화 하기 위해서, 각 가중치들을 $E_m(x^b)$ 의 가중치에 대한 편미분 값에 따라 변경시킨다. 출력층 가중치 $w_{kj}^{(L)}$ 은

$$\Delta w_{kj}^{(L)} = -\eta \cdot \frac{\partial E_m}{\partial a_k^{(L)}} \frac{\partial a_k^{(L)}}{\partial w_{kj}^{(L)}} = \eta \delta_k^{(L)} x_j^{(L-1)} \quad (4)$$

에 따라 변한다. 여기서,

$$\delta_k^{(L)} = -\frac{\partial E_m \partial x_k^{(L)}}{\partial x_k^{(L)} \partial a_k^{(L)}} = (t_k - x_k^{(L)}) \frac{(1 - x_k^{(L)})(1 + x_k^{(L)})}{2} \quad (5)$$

은 출력노드 $x_k^{(L)}$ 에서 발생하는 오차신호이며, η 는 학습률이다. 아래층 가중치 $w_{ji}^{(b)}$ 는

$$\Delta w_{ji}^{(b)} = -\frac{\eta \partial E_m}{\partial a_j^{(b)}} \frac{\partial a_j^{(b)}}{\partial w_{ji}^{(b)}} = \eta \delta_j^{(b)} x_i^{(l-1)} \quad (6)$$

에 의해 변하며, 이때

$$\delta_j^{(b)} = -\frac{\partial E_m}{\partial x_j^{(b)}} \frac{\partial x_j^{(b)}}{\partial a_j^{(b)}} = \frac{(1 - x_j^{(b)})(1 + x_j^{(b)})}{2} \sum_{k=1}^{N_{l+1}} w_{kj}^{(l+1)} \delta_k^{(l+1)} \quad (7)$$

은 상위층의 가중치를 통하여 역전파된 오차신호이다. 이 과정을 모든 학습패턴 $x^p (p = 1, 2, \dots, P)$ 에 대하여 한번씩 수행한 것을 sweep 단위로 표시한다. 전체패턴에 대한 오차는

$$E_m = \frac{1}{P} \sum_{p=1}^P E_m(x^p) \quad (8)$$

으로 정의된다.

위와 같은 다층퍼셉트론의 학습알고리즘이 N비트 디지털 하드웨어로 구현되면, 모든 데이터는 제한된 범위 $\pm R$ 내에서 N비트 데이터 정밀도 $R/(2^{N-1}-1)$ 의 정수배로 표시되며 시그모이드 변환은 찾기표(look-up table)을 사용하여 수행된다. 이때, '×'연산은 절삭과정을 수반하고 '+' 연산은 범람을 일으킬 수 있다. 이 논문에서는 $x_j^{(b)}$ 과 t_k 의 최대치는 ± 1 , $w_{ji}^{(b)}$ 의 최대치는 $W_{max}^{(b)}$ 으로 하였다. 이때, 학습을 위해서는 '×'연산의 하위넘침 오차와 가중치 합 과정의 범람 오차를 줄여야 한다. 왜냐하면, 가중치 합 과정의 범람 오차는 시그모이드 변환 결과가 부정확하도록 하며, '×'연산의 하위넘침 오차는 가중치의 변경량이 '0'이 되도록 하기 때문이다.

1. 크로스-엔트로피 오차함수

위에서 설명한 역전파 알고리즘에서, 식 (5)의 오차신호 $\delta_k^{(L)}$ 은 목표값과 실제값의 차이에 시그모이드 활성화 함수의 기울기가 곱해진 형태이다. 이 오차신호에서 기울기 항은 최대치가 1/2 이므로 항상 오차신호를 줄여서 하위넘침이 발생할 가능성을 증가시킨다^[3]. 그리고 $t_k = 1$ 이고 $x_k^{(L)} \approx -1$ 인 경우 혹은 그 반대인 경우에, $x_k^{(L)}$ 은 이 부적절한 포화상태를 빠져나오기에

충분히 강한 오차신호를 발생시키지 못하여 역전파 학습에서 E_m 의 최소화를 지연시키게 한다. 따라서, 디지털 학습의 경우 기울기 항이 없이 출력노드의 실제값과 목표값의 차이에 의해서만 출력층의 오차신호가 발생하는 것이 좋을 것이다.

최근에 크로스-엔트로피 오차함수가

$$E_c(x^p) = -\sum_{k=1}^{N_l} [(1 + t_k) \ln(1 + x_k^{(L)}) + (1 - t_k) \ln(1 - x_k^{(L)})] \quad (9)$$

와 같이 정의되었다^[10, 11]. 이 오차함수를 이용하면 식 (5)의 오차신호는

$$\delta_k^{(L)} = (t_k - x_k^{(L)}) \quad (10)$$

로 바뀐다. 그 이외의 다른 식들은 E_m 을 이용한 역전파 알고리즘과 동일하다. 이 방법을 이용하면, 출력층의 오차신호에서 시그모이드의 기울기 항이 없으므로 하위넘침 오차를 줄일 수 있어서 디지털 학습에 적합하다. 또한, 출력신호의 시그모이드 기울기 항 때문에 나타나는 출력노드의 부적절한 포화현상도 줄일 수 있다^[10, 13]

2. 누산기의 크기 확장

다층퍼셉트론을 N 비트 디지털 하드웨어에서 학습시, 전방향과 역방향의 가중치 합 과정에서 범람 오차가 발생하기 쉽다. Sakaue 등은 시그모이드 변환의 포화특성 때문에 전방향 가중치 합 과정의 범람 오차를 무시하였다^[3]. 그렇지만, 가중치 합 계산 과정의 범람은 계산결과를 잘못되게 할 수 있으므로 무시하면 안된다. 따라서, 여기서는 N 비트의 가중입력 혹은 가중오차를 더할 때 범람을 방지하기 위해서 누산기의 크기를 2N 비트로 확장하였다. 이때, 누산기의 크기 확장은 하드웨어에 큰 부담을 주지 않을 것이다^[4].

3. 전방향 계산과 시그모이드의 포화특성

전방향의 가중치 합 $a_j^{(b)}$ 계산은 식 (2)와 같다. 그런데, N 비트 디지털 하드웨어에서 $-1 \leq x_i^{(l-1)} \leq +1$, $-W_{max}^{(b)} \leq w_{ji}^{(b)} \leq W_{max}^{(b)}$ 이므로, $w_{ji}^{(b)} x_i^{(l-1)}$ 은 정밀도 $\frac{W_{max}^{(b)}}{(2^{N-1}-1)}$ 의 정수배로 표시된다. $a_j^{(b)}$ 의 계산시 2N 비트 누산기는 N비트 $w_{ji}^{(b)} x_i^{(l-1)}$ 데이터의 합을 $\frac{W_{max}^{(b)}}{2^{N-1}-1} (2^{2N-1}-1) \approx 2^N W_{max}^{(b)}$ 까지 표시할 수 있다. 따라서, 식 (2)에서

$$a_j^{(l)} \leq (1 + N_{l-1}) W_{max}^{(L)} \quad (11)$$

이므로, $2^N \geq (1 + N_{l-1})$ 이면, 전방향 가중치 합을 범람 없이 수행할 수 있다.

범람을 없애기 위해 $2N$ 비트 누산기를 사용하여 $a_j^{(l)}$ 을 표시하였으나, 찾기표(look-up table)를 이용한 N 비트 시그모이드 변환을 수행하기 위해서는 $a_j^{(l)}$ 을 N 비트 데이터로 변환시켜야 한다. 그런데, 시그모이드 변환은 양측면의 포화특성이 있으므로 $|a_j^{(l)}| > 10$ 인 데이터는 상위비트를 절삭하여 $|a_j^{(l)}| = 10$ 이 되도록 하여도 시그모이드 변환결과는 거의 오차가 없다. 따라서, $2N$ 비트로 표시된 $a_j^{(l)}$ 을 위와 같이 절삭 후 ± 10 범위 내에서 N 비트로 변환시키면, 데이터 표현비트의 축소로 인한 오차를 시그모이드 변환의 관점에서 줄일 수 있다.

4. 역방향 계산과 확률밀도함수

전방향 계산에 의해 모든 층의 노드 값이 결정되면, 각 층의 가중치들을 학습법칙에 따라 변경시키기 위해서 출력층의 오차신호가 식 (7)처럼 역전파되어야 한다. 이때, 크로스 엔트로피 함수를 사용하여 출력노드의 오차신호가 식 (10)과 같이 계산되면 $-2 \leq \delta_k^{(L)} \leq +2$ 이다. 그러면,

$$\sum_k N_k w_{kj}^{(L)} \delta_k^{(L)} \leq 2 \times N_L \times W_{max}^{(L)} \quad (12)$$

이며, $w_{kj}^{(L)} \delta_k^{(L)}$ 은 정밀도 $\frac{2 W_{max}^{(L)}}{(2^N - 1)}$ 의 정수배이다. 따라서, $2N$ 비트 누산기는 $2^{N+1} W_{max}^{(L)}$ 까지 표시가능하므로 $2^N \geq N_L$ 이면 범람 없이 오차신호의 역전파가 가능하다.

N 비트 가중치를 변경시키기 위해서는 $2N$ 비트 누산기에 표시된 $\sum_{k=1}^{N_k} w_{kj}^{(L)} \delta_k^{(L)}$ 을 N 비트 데이터로 변환시켜야 하는데, 변환오차를 최소화시키기 위해서는 변환시의 N 비트 데이터 최대값을 잘 설정해야 한다. 이를 위해서 다음과 같이 $\sum_{k=1}^{N_k} w_{kj}^{(L)} \delta_k^{(L)}$ 의 확률밀도 함수를 구하였다. 먼저, $w_{kj}^{(L)}$ 은 i.i.d.(independent, identically distributed)이고 $[-W_{max}^{(L)}, +W_{max}^{(L)}]$ 에서 균일분포라고 가정하자. 그리고, $\delta_k^{(L)}$ 역시 i.i.d.이고 $[-2, +2]$ 에서 균일분포라고 가정하자. 그러면, 중앙극한정리(central limit theorem)에 의해 $\sum_{k=1}^{N_k} w_{kj}^{(L)} \delta_k^{(L)}$ 은 근사적으로 평균이 0이고 분산이

$$\sigma^2 = \sum_{k=1}^{N_k} E \{ w_{kj}^{(L)} \delta_k^{(L)} \}^2 = \left(\frac{2 W_{max}^{(L)}}{3} \right)^2 N_L \quad (13)$$

인 정규분포를 가진다^[15]. 따라서, $2N$ 비트 데이터인 $\sum_{k=1}^{N_k} w_{kj}^{(L)} \delta_k^{(L)}$ 을 N 비트 데이터로 변환 시 최대값을 $\pm 2\sigma$ 로 하였다. $l \leq L-2$ 인 층에도 똑같은 절차를 적용하면 된다.

그리고 식 (7)에서 역전파된 신호가 시그모이드의 기울기와 곱해지는 데, 시그모이드의 기울기 역시 1/2을 최대값으로 지니도록 정량화된 찾기표(look up table)를 사용하면 시그모이드의 기울기 계산 시 발생하는 '×'연산에 의한 하위넘침 오차를 방지할 수 있다.

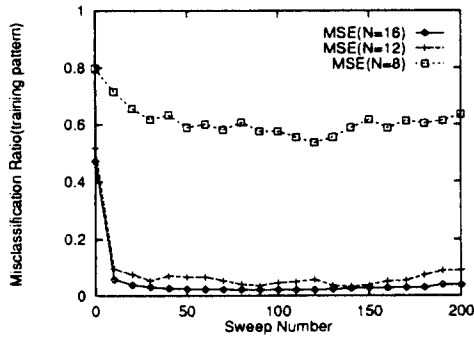
5. 가중치의 범위조정

다층퍼셉트론을 학습시키기 위해서는 먼저 가중치들을 임의의 값으로 초기화 시켜야 한다. 이때, 학습의 초기포화 현상을 피하기 위해서는 초기가중치들의 범위가 충분히 작아야 한다^[11]. 한편, N 비트 디지털 하드웨어에서 $w_{\mu}^{(l)}$ 은 정밀도 $\frac{W_{max}^{(l)}}{(2^N - 1)}$ 의 정수배만을 가질 수 있다. 따라서, $W_{max}^{(l)}$ 가 학습완료 시점을 고려하여 큰 값을 가지도록 한 경우에는 작은 값을 가지는 초기가중치들이 가중치의 정밀도 보다 작게되어, 정량화 시킨 초기가중치들은 거의 대부분 '0'이 된다. 이 문제를 해결하기 위해서 학습초기에 $W_{max}^{(l)}$ 는 충분히 작게하여 정밀도가 초기가중치들을 표시할 수 있도록 하고서, 학습의 진행에 따라 가중치의 크기가 커지면 $W_{max}^{(l)}$ 를 일정비를 확장시켰다. 만약 한 $w_{\mu}^{(l)}$ 이 범람해도 $W_{max}^{(l)}$ 을 확장시킨다면, $W_{max}^{(l)}$ 의 확장에 수반되는 $w_{\mu}^{(l)}$ 정밀도 저하의 영향이 클 것이다. 따라서, $w_{\mu}^{(l)}$ 이 학습에 의해 어느 정도 커진 후 $W_{max}^{(l)}$ 이 확장되도록 하기 위해서, 한 입력패턴에 대하여 학습 후 범람이 발생한 $w_{\mu}^{(l)}$ 이 1% 이상이면 $W_{max}^{(l)}$ 를 1.5배 확장시켰다.

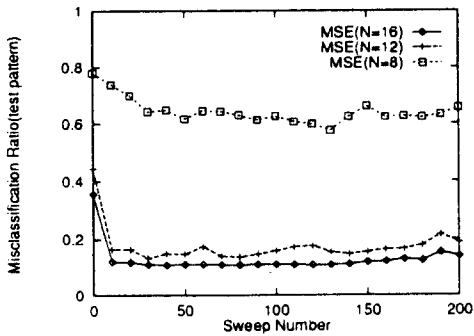
III. 시뮬레이션

이 논문에서 제안한 다층퍼셉트론의 N 비트 디지털 학습방법을 일반적인 방법 및 가중오차함수 방법과 비교하기 위해서 필기체 숫자인식 문제를 시뮬레이션 하였다. CEDAR 데이터베이스^[11]에서 2,308개의 필기체 숫자를 선택하여 크기를 12×12 로 정규화한 후 학습

에 사용하였다. 이때, 패턴의 각 픽셀은 16개의 정량화된 값을 가진다. 다층퍼셉트론은 144 입력, 30 중간층, 10 출력 노드들로 구성되었으며, 학습률은 0.1을 사용하였다. 일반화 성능은 역시 정규화된 2,213개의 시험 패턴으로 평가하였다. 시뮬레이션을 각 학습방법에 따라 4번 수행하여 그 결과의 평균치를 그림으로 그렸다.



(a)



(b)

그림 1. 일반적인 디지털 방법을 이용한 필기체 숫자 인식의 학습결과(N: 데이터 표현 비트 수) (a) 학습패턴에 대한 오인식률 (b) 시험패턴에 대한 오인식률

Fig. 1. Simulation results of handwritten digit recognition using the conventional EBP method with N bit precision.

(a) Misclassification ratio for the training patterns (b) Misclassification ratio for the test patterns

1. 일반적인 N 비트 학습 방법

2장에서 설명한 다층퍼셉트론의 일반적인 학습이 N 비트로 정량화된 데이터 및 연산을 사용하여 수행된 경우를 시뮬레이션하였다. $W_{max}^0 = 14.8$ 로 하고서, $[-1, +1]$ 에서 균일분포를 가지는 정량화된 가중치로 다층퍼셉트론을 초기화 하였다. 그리고 가중치 합과

정의 범람을 줄이기 위해서 가중치 신호들은 $(N-1)$ 비트를 가지도록 하였다. 즉, '×' 연산결과는 $(N-1)$ 비트 데이터로 표시된다. 그림 1은 $N=8, 12, 16$ 인 경우의 학습결과를 보여준다. $N=12, 16$ 일 때는 학습이 이루어졌으나, $N=8$ 인 경우는 데이터 정밀도 때문에 학습결과가 매우 나쁘다. 그 원인은 그림 2를 보면 알 수 있다.

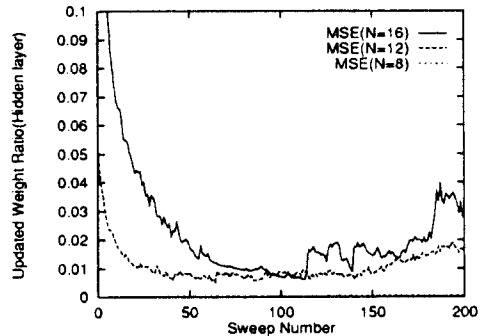


그림 2. 일반적인 디지털 학습에서 중간층 가중치의 변경비율(N: 데이터 표현 비트 수)

Fig. 2. The ratio of updated weights in the hidden layer using the conventional EBP method with N -bit precision.

그림 2는 한 시뮬레이션에서 중간층 가중치의 변경 비율을 그린 것인데, $N=8$ 인 경우 중간층 가중치가 거의 변하지 못하는 것을 알 수 있다. 이것은 중간층으로 역전파된 오차신호가 하위넘침 오차에 의해 '0'에 가까운 값을 가지기 때문이다.

2. 가중오차합수를 이용한 N 비트 학습

학습에 필요한 데이터 비트 수를 낮추기 위해서, Sakaue 등이 제안한 가중오차합수는

$$E = G \sum_{k=1}^{N_k} |g_k(t_k - x_k^{(L)})|^2 \quad (14)$$

이다¹³⁾. 여기서,

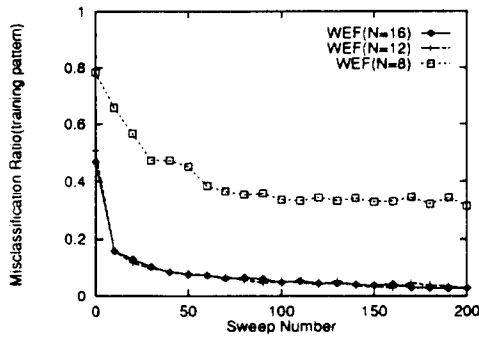
$$G = \begin{cases} 0, & \text{if } |t_k - x_k^{(L)}| < T_G \text{ for all } k, \\ 1, & \text{otherwise} \end{cases} \quad (15)$$

이고,

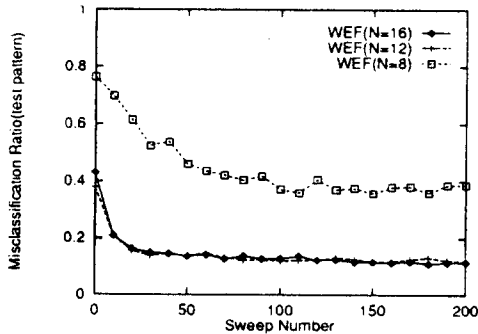
$$g_k = \begin{cases} 2, & \text{if } |t_k - x_k^{(L)}| > T_{\kappa}, \\ 1, & \text{otherwise} \end{cases} \quad (16)$$

이다. 이 방법을 $T_{\kappa} = 1, T_G = 0$ 으로 하고서 다층퍼

셉트론의 필기체 숫자 학습에 적용시켰다.



(a)



(b)

그림 3. 가중오차함수를 이용한 필기체 숫자인식의 디지털 학습결과(N: 데이터 표현 비트 수)

(a) 학습패턴에 대한 오인식률 (b) 시험패턴에 대한 오인식률

Fig. 3. Simulation results of handwritten digit recognition using the weighted error method with N -bit precision.

(a) Misclassification ratio for the training patterns (b) Misclassification ratio for the test patterns

초기가중치는 일반적인 학습방법(3.1절)과 같이 하였으며, '×' 연산은 N 비트 데이터를 가지도록 절삭하였다. 가중치 합 과정의 범람 오차를 없애기 위해서 $2N$ 비트 누산기를 사용하고 그 결과 중 N 비트 이상의 값은 N 비트에서 표현할 수 있는 최대값으로 절삭시켰다. 그림 3은 $N=8, 12, 16$ 인 경우의 학습결과인데, $N=8$ 인 경우의 학습결과가 일반적인 방법(그림 1) 보다는 많이 향상되었으며, 중간층 가중치의 변경비율은 그림 4에서 보는 바와 같이 일반적인 방법(그림 2) 보다 커졌다. 그렇지만, $N=8$ 인 경우의 학습결과는 여전히 $N=16$ 인 경우보다 매우 나쁘다.

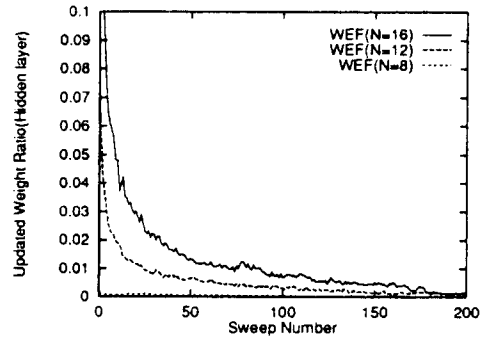


그림 4. 가중오차함수를 이용한 디지털 학습에서 중간층 가중치의 변경비율(N: 데이터 표현 비트 수)

Fig. 4. The ratio of updated weights in the hidden layer using the weighted error method with N -bit precision.

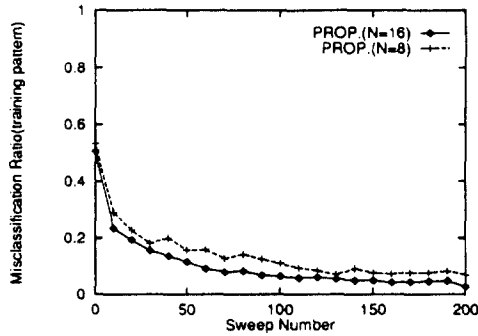
3. 제안한 N 비트 학습방법

2.1-2.5 절에서 제안한 N 비트 학습방법을 다층퍼셉트론의 필기체 숫자인식 문제에 적용시켰다. $W_{max}^B = 10^{-1}$ 으로 하고서, $[-10^{-3}, +10^{-3}]$ 에서 균일분포를 가지는 정량화된 가중치로 초기화 후 학습시켰다. 그림 5는 $N=8, 16$ 인 경우의 학습결과를 보여준다. 여기서, $N=8$ 의 학습패턴에 대한 인식률은 $N=16$ 인 경우보다 조금 못하나, $N=8$ 의 시험패턴에 대한 인식률은 $N=16$ 인 경우와 거의 비슷하다.

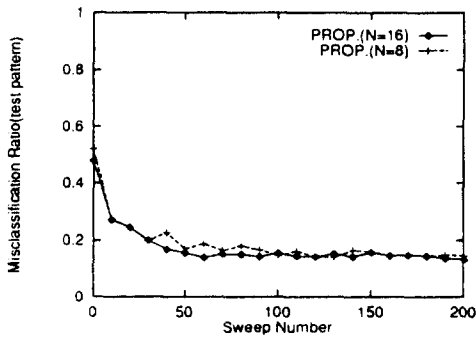
그림 6은 한 시뮬레이션에서 중간층 가중치의 변경비율을 그린 것인데, 제안한 방법이 $N=8$ 인 경우에도 다른 방법에 비해서 학습이 잘되는 이유는 그림 6에서 보는 바와 같이 가중치 변경비율이 다른 방법들 보다 많이 향상되었기 때문이다. 특히, 가중치의 범위 조절 방법은 학습초기의 가중치 정밀도에 영향을 주어 학습초기의 가중치 변경비율이 향상되었다. 물론, 제안한 방법에서 $N=8$ 인 경우 중간층 가중치의 변경비율이 $N=16$ 인 경우보다 현저히 낮으나, 이 정도의 변경비율로도 $N=16$ 인 경우와 비슷한 일반화 성능(시험패턴에 대한 인식률)을 얻음을 그림 5(b)에서 볼 수 있다.

다층퍼셉트론을 디지털 신경망 칩으로 구성된 하드웨어에 구현 시 제안한 방법을 사용하면 16비트 가중치 대신 8비트 가중치를 저장하는 메모리를 사용하므로 디지털 신경망 칩의 실리콘 면적을 크게 줄일 수 있다. 물론, 제안한 방법은 여전히 16비트 누산기와 그에 수반되는 16비트 덧셈기가 필요하나 전체 칩 면적

에서 이들이 차지하는 부분은 메모리 크기에 비하면 미미한 부분이다.



(a)



(b)

그림 5. 제안한 방법을 이용한 필기체 숫자인식의 디지털 학습결과(N: 데이터 표현 비트 수)
(a) 학습패턴에 대한 오인식률 (b) 시험패턴에 대한 오인식률

Fig. 5. Simulation results of handwritten digit recognition using the proposed method with N bit precision.

(a) Misclassification ratio for the training patterns (b) Misclassification ratio for the test patterns

누산기의 16비트 데이터를 8비트 데이터로 변환하는 부분은 디지털 신경망 칩에서 소프트웨어로 처리하는 것이 신경망 칩의 유연성을 높이는 데 좋을 것이다. 그리고 가중치 범위 조절 역시 그 영향이 시그모이드 찾기표를 다시 신경망 칩에 저장하고 가중치들은 다시 정규화 하는 과정을 수반하므로 소프트웨어로 처리하면 된다. 디지털 신경망 칩에서 가장 큰 면적을 차지하는 부분은 가중치를 저장하는 메모리 부분이다. 따라서, 디지털 신경망 칩의 구현 시 제안한 방법에 따라 16비트 데이터 정밀도를 8비트로 낮추어 메모리가 차지하

는 면적을 줄이는 것이 일반적인 방법 혹은 가중오차 함수를 이용한 방법에서 16비트나 12비트 데이터 정밀도를 사용토록 하는 것보다 훨씬 더 실리콘 면적을 작게 차지할 것이다.

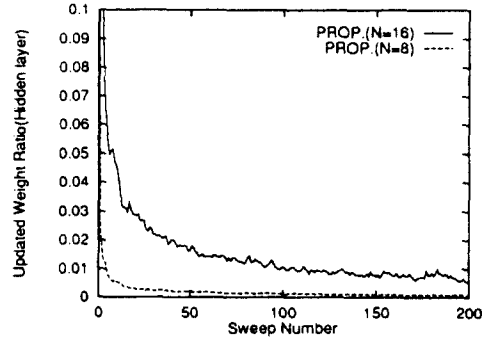


그림 6. 제안한 방법을 이용한 디지털 학습에서 중간층 가중치의 변경비율(N: 데이터 표현 비트 수)

Fig. 6. The ratio of updated weights in the hidden layer using the proposed method with N -bit precision.

IV. 결 론

이 논문에서는, N 비트 디지털 하드웨어에 다층퍼셉트론이 구현된 경우 학습에 필요한 적정 데이터 표현 N 을 낮추는 방법을 제안하였다. 제안한 방법은 가중치 합 과정의 범람 오차를 줄이기 위해서, 누산기의 크기를 $2N$ 비트로 확장하였다. 그리고, $2N$ 비트 가중치 합 계산결과를 N 비트 데이터로 변환 시 시그모이드의 포화특성과 역전과신호의 확률밀도함수를 이용하였다. Underflow 오차를 줄이기 위해서는, 크로스 엔트로피 오차함수를 사용하여 출력오차신호의 기울기 항을 제거하였고 중간층 오차신호의 계산 시 시그모이드의 기울기 항은 착시표(look-up table) 방법을 사용하였다. 마지막으로, 학습초기의 가중치 범위를 작게 하고서 학습의 진행에 따라 그 범위를 확장시켜서, 학습의 조기포화 현상 예방 및 학습초기의 가중치 변경비율을 향상시켰다.

제안한 방법을 필기체 숫자인식 문제의 학습을 통하여 확인한 결과, $N=8$ 인 경우 학습패턴에 대한 인식률은 $N=16$ 인 경우보다 조금 못하며, $N=8$ 의 시험패턴에 대한 인식률은 $N=16$ 인 경우와 비슷하다. 그리고, 일반적인 방법 및 가중오차함수 방법은 $N=8$ 인

경우의 학습성능이 $N=16$ 인 경우 보다 매우 나빴다. 제안한 방법을 이용하면 다층퍼셉트론을 구현하는 디지털 하드웨어에서 가중치 메모리 등이 차지하는 실리콘 면적이 크게 줄어들므로 디지털 하드웨어의 집적도를 향상시킬 수 있을 것이다.

참 고 문 헌

- [1] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pat. Ana. Mach. Int.*, accepted to appear as a correspondance.
- [2] C. Alippi and M. E. Nigri, "Hardware requirements for digital VLSI implementation of neural networks," *Proc. IJCNN'91 Singapore*, vol. II, pp. 1878, Nov. 1991.
- [3] S. Sakaue, T. Kohda, H. Yamamoto, S. Maruno, and Y. Shimeki, "Reduction of required precision bits for back propagation applied to pattern recognition," *IEEE Trans. Neural Networks*, vol. 4, pp. 270-275, Mar. 1993.
- [4] J. L. Holt and J. N. Hwang, "Finite precision error analysis of neural network hardware implementation," *IEEE Trans. Comp.*, vol. 42, pp. 281-290, Mar. 1993.
- [5] M. Hoehfeld and S. E. Fahlman, "Learning with limited numerical precision using the cascade correlation algorithm," *Research Note CMU CS-91-130, Carnegie Mellon Univ.*, pp. 1-17, 1991.
- [6] T. Baker and D. Hammerstrom, "Characterization of artificial neural network algorithm," *Proc. ISCAS'89*, pp. 78-81, 1989.
- [7] J. J. Choi, S. Oh, and R. J. Marks II, "Training layered perceptrons using low accuracy computation," *Proc. IJCNN'91 Singapore*, pp. 554-559, Nov. 1991.
- [8] D. Andes, B. Widrow, M. Lehr, and E. Wan, "MR III: a robust algorithm for training analog neural networks," *Proc. IJCNN Jan. 15-19, 1990, Washington, DC, USA*, vol. I, pp. 533-356.
- [9] M. Jabri and B. Flower, "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 154-157, Jan. 1992.
- [10] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, vol. 5, pp. 465-471, 1992.
- [11] M. D. Richard and R. P. Lippmann, "Neural network classifier estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461-483, 1991.
- [12] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.
- [13] S.-H. Oh and Y. Lee, "A modified error function to improve the error back-propagation algorithm for Multi-Layer Perceptrons," *ETRI Journal*, vol. 17, no. 1, pp. 11-22, April 1995.
- [14] Y. Lee, S. H. Oh, and M. W. Kim, "An analysis of premature saturation in back-propagation learning," *Neural Networks*, vol. 6, pp. 719-728, 1993.
- [15] A. Papoulis, *Probability, Random variables, and Stochastic Processes*. McGraw-Hill, New York, 2nd edition, 1984.

저 자 소 개

吳 相 勳(正會員) 第 28卷 B編 4號 參照

宋 玗 宣(正會員) 第 33卷 B編 2號 參照