

바이어스항이 있는 GBAM 모델을 이용한 양방향 연상메모리 구현

Implementation of Bidirectional Associative Memories Using the GBAM Model with Bias Terms

임채환, 박주영
고려대학교 제어계측공학과

Chae-Hwan Leem and Jooung Park
Dept. of Control and Instrumentation Engineering, Korea University
E-mail : mirage@cie01.korea.ac.kr

Abstract

In this paper, we propose a new design method for bidirectional associative memories model with high error correction ratio. We extend the conventional GBAM model using bias terms and formulate a design procedure in the form of a constrained optimization problem. The constrained optimization problem is then transformed into a GEVP(generalized eigenvalue problem), which can be efficiently solved by recently developed interior point methods. The effectiveness of the proposed approach is illustrated by a design example.

Keywords : General bidirectional associative memory (GBAM), generalized eigenvalue problem (GEVP)

I. 서론

신경망을 기반으로 하는 양방향 연상 메모리에 관해서는 그동안 많은 연구가 있었는데, 특히 Shi 등은 기존의 Kosko BAM 모델[1]에서 하드리미터(hard limiters)를 최대연산자(max operators)로 대체시키는 GBAM(general bidirectional associative memory) 모델을 제안하고 이 모델의 연결강도행렬의 학습을 위하여 퍼셉트론 규칙을 사용하는 방법을 제안하였다[2]. GBAM 모델의 우수한 성능은 학습 방법에 기인한 것이라기보다는 모델의 기본 구조 때문인 것으로 판단된다.

본 논문에서는 이러한 판단에 따라서, 기존의 GBAM 모델에 바이어스항을 새롭게 추가하여 높은 성능 지수를 나타낼 수 있도록 하고, 또한 GBAM 모델에 대한 이론적 고찰을 바탕으로 효과적인 학습 알고리즘을 제안 하고자 한다. 제안된 학습 방법은 GEVP(generalized eigenvalue problems) 형태의 선형행렬부동식 기반의 최적화 문제(linear matrix inequalities based optimization problems)로 표현된다. 이러한 종류의 문제는 최근 개발된 내부점 방법(interior point methods)과 같은 관련 범용 소프트웨어에 의해 빠르고 편리하게 계산될 수 있는 장점을 갖는다.

본 논문에서 사용하는 주요 용어와 기호의 정의는 다음과 같다: R^n 과 $R^{n \times m}$ 은 각각, n 차원의 실수

벡터 공간과 $n \times m$ 차원의 실수 행렬 집합을 나타낸다. 벡터 $x \in R^n$ 의 i 번째 원소는 x_i 로 표기된다. 0 이 아닌 모든 $x \in R^n$ 에 대해서 $x^T A x > 0$ 이 성립하는 대칭 행렬 $A \in R^{n \times n}$ 는 양의 정부호 행렬이라 불리우며, $A > 0$ 로 표기된다. 또한, $A > B$ 은 $A - B$ 가 양의 정부호임을 의미한다. B^n 은 n 차원 이진 벡터의 집합 $\{-1, +1\}^n$ 을 나타낸다. B^n 에 속한 두 이진 벡터 x^* 와 x 사이의 해밍거리(Hamming distance)는 $H(x^*, x)$ 로 표기된다. 그리고, 이진 벡터와 패턴은 같은 의미를 갖는 동의어로 사용되었다.

II. 바이어스항이 있는 GBAM 모델

본 논문에서 제안하는 바이어스항이 추가된 GBAM 모델은 m 개의 뉴런 (X_1, \dots, X_m) 으로 이루어진 X 층과 n 개의 뉴런 (Y_1, \dots, Y_n) 으로 이루어진 Y 층으로 이루어진다. 각 층을 이루는 뉴런의 활성화 상태(firing state)는 이진수 “+1” 또는 “-1”로 표현되며, 이들은 각각 “활성화되었음” 또는 “활성화되지 않았음”을 의미한다. X 층 뉴런들의 활성화 상태는 X 층 활성화상태벡터 (X -layer firing state vector), $x = (x_1, \dots, x_1, \dots, x_m)^T \in B^m$ 에 의해 표현된다. 여기에

서 x_i 는 X 층에 있는 i 번째 뉴런의 활성상태를 나타내는 이진수이다. 마찬가지로, Y 층 뉴런들의 활성상태는 Y 층 활성상태벡터 (Y -layer firing state vector), $y = (y_1, \dots, y_j, \dots, y_n)^T \in B^n$ 로 표현된다. 같은 층 내의 뉴런끼리는 연결이 존재하지 않으며, X 층과 Y 층 상호간에는 연결강도 행렬 $A = [a_{ij}] \in R^{n \times m}$ 과 $B = [b_n] \in R^{m \times n}$ 을 통한 연결이 존재한다. 여기에서 연결강도 a_{ij} 는 X 층의 j 번째 뉴런 X_j 의 활성상태가 Y 층의 i 번째 뉴런 Y_i 의 활성화에 미치는 영향의 크기를 나타내며, 연결강도 b_n 는 Y 층의 i 번째 뉴런 Y_i 의 활성상태가 X 층의 j 번째 뉴런 X_j 의 활성화에 미치는 영향의 크기를 나타낸다.

X 층의 뉴런들의 활성상태 $x = (x_1, \dots, x_m)^T$ 가 Y 층의 i 번째 뉴런 Y_i 의 활성화에 미치는 영향은 다음과 같다:

$$q(i|x) = \sum_{j=1}^m a_{ij}x_j + a_{i0}$$

여기서 a_{i0} 는 본 논문에서 새롭게 도입한 바이어스항이다. 이들을 원소로 갖는 벡터 즉, $q(x) = (q(1|x), \dots, q(n|x))^T = Ax$ 는 X 층 활성상태 벡터 x 가 Y 층의 활성화에 주는 영향을 나타내게 된다. $y \in B^n$ 가 $x \in B^m$ 로부터 받는 지지도(supporting function)는, $q(x)$ 와 y 의 내적 $\langle y, q(x) \rangle$ 으로 정의되고 $S(y|x)$ 로 다음과 같이 표기된다:

$$S(y|x) = \langle y, q(x) \rangle = \sum_{i=1}^n y_i q(i|x) = \langle y, Ax \rangle$$

마찬가지로, Y 층의 뉴런들의 활성상태가 X 층의 j 번째 뉴런 X_j 의 활성화에 미치는 영향은

$$q(j|y) = \sum_{i=1}^n b_{ji}y_i + b_{j0}$$

이며, b_{j0} 는 바이어스항이다. $x \in B^m$ 가 $y \in B^n$ 로부터 받는 지지도 $S(x|y)$ 는 다음과 같이 정의된다:

$$S(x|y) = \langle x, q(y) \rangle = \sum_{j=1}^m x_j q(j|y) = \langle x, By \rangle$$

바이어스항이 추가된 GBAM 모델의 회상(recall) 과정에서 사용되는 업데이트(update) 규칙은 다음과 같다:

$$\begin{cases} x^{s+1} = \arg \max_{x' \in H_1(x^s)} S(x'|y^s) \\ y^{s+1} = \arg \max_{y' \in H_1(y^s)} S(y'|x^{s+1}) \end{cases}$$

여기에서 $H_1(x^s)$ 는 $x^s \in B^m$ 로부터 해밍거리가 1 이내에 있는, 즉 $HD(x, x^s) \leq 1$ 을 만족하는 모든 이진벡터 $x \in B^m$ 로 이루어지는 해밍 볼(Hamming ball)이고, $H_1(y^s)$ 도 같은 방법으로 정의된다. 초기 조건으로 x^0 만이 주어지는 경우에는 y^0 는 다음과 같은 방식으로 생성된다:

$$y_i^0 = \begin{cases} 1 & \text{if } q(i|x) = (Ax)_i > 0 \\ -1 & \text{otherwise.} \end{cases}$$

그 이후의 x^s 및 y^s 들은 위에서 설명된 업데이트 규칙에 따라 갱신되어진다(초기조건으로 y^0 만 주어지는 경우에도 같은 원리가 적용된다.). 지금까지 기존의 GBAM 모델[2]에 바이어스항을 추가한 모델의 활성화 벡터에 따른 지지도와 업데이트 규칙을 살펴보았다.

참고문헌[2]에서 제안한 GBAM 모델을 위한 학습 규칙만을 보면, 원형 패턴 쌍들 사이에 다음의 조건, $HD(x^{(p)}, x^{(q)}) \geq 3, HD(y^{(p)}, y^{(q)}) \geq 3, \forall p \neq q$, 이 만족되지 않을 경우에는 적용할 수 없다는 등의 단점이 존재한다. 이에 따라, 본 논문에서는 이러한 단점을 갖지 않고 우수한 오차 수정 능력을 보장할 수 있도록 바이어스항을 추가로 도입하고 새로운 학습 방법론을 GBAM 모델에 대한 이론적 고찰을 바탕으로 제안하고자 한다.

III. GEVP기반의 학습방법

본 논문에서는 바이어스항이 추가된 GBAM에 관한 이론적 고찰을 바탕으로 주어진 원형 패턴 쌍을 안정하게 그리고 최대한 높은 오차 수정능력을 갖도록 저장시키는 학습 방법을 개발하기 위하여, 다음의 정리를 제시한다.

정리 : 저장하고자 하는 패턴 쌍 $(x^*, y^*) \in B^{m+n}$ 이 주어졌을 때, GBAM의 연결강도행렬 $A \in R^{n \times m}$, $B \in R^{m \times n}$ 와 바이어스항들이 일정한 양수 $d_j, j = 1, \dots, m$ 와 $g_i, i = 1, \dots, n$ 에 대하여 다음의 부등식을 만족한다고 하자:

$$\left\{ \begin{array}{l} x_j^* (\sum_{i=1}^n b_{ji}y_i^* + b_{j0}) > 2d_j \max_i |b_{ji}| \\ j = 1, \dots, m, \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} y_i^* (\sum_{j=1}^m a_{ij}x_j^* + a_{i0}) > 2g_i \max_j |a_{ij}| \\ i = 1, \dots, n. \end{array} \right. \quad (2)$$

이 같은 경우에, GBAM은 반드시 다음과 같은 성질을 갖는다:

- ① 패턴 쌍 (x^*, y^*) 는 안정한 상태로 저장된다.
- ② 초기조건으로 Y 층이 $0 \leq H(y^0, y^*) \leq d_j$ 를 만족하는 벡터 $y^0 \in B^n$ 를 활성벡터로 가지면, 다음 시간 스텝에서 X 층의 j 번째 뉴런은 올바른 활성상태를 갖는다 (즉, $x_j^1 = x_j^*$).
- ③ 초기조건으로 X 층이 $0 \leq H(x^0, x^*) \leq g_i$ 를 만족하는 벡터 $x^0 \in B^m$ 를 활성벡터로 가지면, 다음 시간 스텝에서 Y 층의 i 번째 뉴런은 올바른 활성상태를 갖는다 (즉, $y_i^0 = y_i^*$).

위의 정리로부터, 각 d_j 와 g_i 를 최대화시키는 A 와 B 를 연결강도행렬로 갖는 GBAM은 주어진 패턴 쌍 (x^*, y^*) 을 안정하게 저장하고, 우수한 오차 수정 능력을 가질 수 있음을 알 수 있다. 따라서, GBAM의 설계는 다음과 같은 최적화 문제로 표현되

어질 수 있다:

문제 : 기억시켜야 할 원형 패턴 쌍 $\{(x^{(k)}, y^{(k)}) | k = 1, \dots, K\}$ 이 주어졌을 때,

1) 각 $j \in \{1, \dots, m\}$ 에 대하여

$$\begin{aligned} \max \quad & d_j (> 0) \\ \text{s.t.} \quad & x_j^{(k)} (\sum_{i=1}^n b_n y_i^{(k)} + b_{j0}) > 2 d_j \max_i |b_{ji}|, \\ & k = 1, \dots, K \end{aligned}$$

2) 각 $i \in \{1, \dots, n\}$ 에 대하여

$$\begin{aligned} \max \quad & g_i (> 0) \\ \text{s.t.} \quad & y_i^{(k)} (\sum_{j=1}^m a_{ij} x_j^{(k)} + a_{i0}) > 2 g_i \max_j |a_{ij}|, \\ & k = 1, \dots, K. \end{aligned}$$

이 문제들에서 최적의 해를 구성하는 $A = [a_{ij}] \in R^{n \times m}$ 와 $B = [b_{ij}] \in R^{m \times n}$ 가 우리가 구하고자 하는 GBAM의 연결강도 행렬 A, B 이다. 그리고, $b_{j0} (j=1, \dots, m), a_{i0} (i=1, \dots, n)$ 는 GBAM의 바이어스항들이다.

위의 문제에 등장하는 제약조건들은, 추가적인 보조변수를 도입함으로써 선형행렬부등식(linear matrix inequalities, LMIs)관련 GEVP 형태의 부등식으로 변환될 수 있다. 선형 행렬 부등식과 관련된 최적화와 관련하여 중요한 주제 중 하나인 GEVP는, 다음과 같은 일반형을 갖는 최적화 문제를 의미한다[4]:

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & \lambda B(z) > A(z) \\ & B(z) > 0 \\ & C(z) > 0. \end{aligned}$$

여기에서, $A(z), B(z)$ 와 $C(z)$ 는 변수 z 에 대하여 에프린 함수인 대칭행렬들이다.

GEVP 형태의 최적화 문제는 내부점 방법(interior point method)에 의하여 효과적으로 풀릴 수 있음은 널리 알려져 있으며, 최근에는 이를 풀 수 있는 MATLAB의 Toolbox도 등장하였다.

앞에서 GBAM의 설계를 위하여 제시한 최적화 문제들 중 제약 조건 부분들은, 보조변수 $q_j > 0$ 와 $p_i > 0$ 를 이용하여 정리하면, 최종적으로 GBAM 설계문제에 등장하는 1) 항 및 2) 항의 최적화 문제들은 각각 다음과 같은 GEVP 기반 설계 문제들로 치환될 수 있다:

1) 각 $j \in \{1, \dots, m\}$ 에 대하여,

$$\begin{aligned} \min \quad & -d_j (< 0) \\ \text{s.t.} \quad & (-d_j)(2q_j) + (\sum_{i=1}^n b_n y_i^{(k)} + b_{j0}) x_j^{(k)} > 0 \quad (3) \\ & k = 1, \dots, K, \\ & -q_j < b_{ji} < q_j, i = 1, \dots, n. \end{aligned}$$

2) 각 $i \in \{1, \dots, n\}$ 에 대하여,

$$\min \quad -g_i (< 0)$$

$$\begin{aligned} \text{s.t.} \quad & (-g_i)(2p_i) + (\sum_{j=1}^m a_{ij} x_j^{(k)} + a_{i0}) y_i^{(k)} > 0 \quad (4) \\ & k = 1, \dots, K, \\ & -p_i < a_{ij} < p_i, j = 1, \dots, m. \end{aligned}$$

최적화를 이용하여 GBAM을 설계하는 과정에서 반드시 고려해야 할 추가적인 주의 사항 중 하나는, 연결 강도의 크기가 지나치게 큰 값을 갖지 않도록 상한선을 두는 것이 필요하다는 점이다. 이러한 필요는, 위의 GEVP 문제들의 제약 조건 부분에 양의 상수 L, U 를 이용하여 다음과 같은 선형 행렬 부등식 조건을 추가함으로써 충족시킬 수 있다:

$$L < q_j < U, j = 1, \dots, m, \quad (5)$$

$$L < p_i < U, i = 1, \dots, n. \quad (6)$$

그리고, 바이어스항에 대한 제약조건도 추가하였다:

$$-U < a_{i0} < U, i = 1, \dots, n, \quad (7)$$

$$-U < b_{j0} < U, j = 1, \dots, m. \quad (8)$$

조건 (5), (6), (7), (8)을 각각 GEVP 문제 (3), (4) 에 추가한 경우에도, 위의 문제들은 여전히 GEVP 형태를 유지하게 된다. 이상의 과정을 통하여 본 논문에서 제시하는 GBAM을 위한 설계 방법론이, $m + n$ 개의 GEVP 문제를 푸는 형태로 확립된다.

IV. 모의 실험 및 결과

본 논문에서 성능지수로 사용된 오차수정율의 정의는 다음과 같다. 오차수정율 $P(x^{(k)}, r)$ 은 원형 패턴 $x^{(k)}$ 에서 r 비트 떨어진 해밍 볼(Hamming ball)의 패턴에서 시작해서 원형 패턴 쌍 $(x^{(k)}, y^{(k)})$ 에 수렴하는 확률을 의미한다. 그리고, 평균 오차수정율은 이들의 평균을 의미한다. 즉, $Prob(r) = (\sum_{k=1}^K P(x^{(k)}, r))/K$ 이다.

제안된 설계 방법의 성능을 평가하기 위한 예제는 참고문헌[3]에서 제시한 예제로 그림 1의 패턴 쌍을 저장하는 GBAM의 설계문제이다. 그림 1에서 저장되어야 할 패턴의 쌍은 4개이고, $\{(x1, y1), (x2, y2), (x3, y3), (x4, y4)\}, K=4$ X 층과 Y 층은 각각 20개의 뉴론 ($m = n = 20$)으로 이루어져 있다. 각각의 칸은 활성 상태를 나타내는 것으로서 검은색은 “+1”, 흰색은 “-1”을 나타낸다.

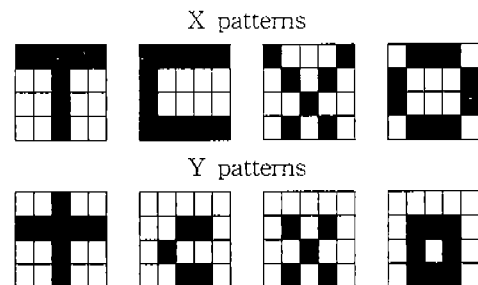


그림 1. 저장되어야 할 패턴의 쌍

본 논문에서 제안된 방법에 따라, $L = 1, U = 2$ 로 고정시키고 MATLAB LMI Control Toolbox[5]의 "gevp" 함수를 사용하여 연결강도행렬 A, B 와 바이어스항을 구하였다.

이렇게 설계된 바이어스항이 추가된 GBAM의 X 층에 이진 벡터를 초기 조건으로 주고 이들이 최종적으로 어떠한 패턴 쌍으로 수렴하는지를 관찰하여 표 1에 정리하였다. 표에 기입되어 있는 수치의 해석은 예를 들어, x_1 줄의 해밍거리(HD)=4인 칸과 만나는 95/100은 x_1 에서 해밍거리가 4인 이진 벡터가 가질 수 있는 모든 경우의 수 중에서 임의의 100개를 X 층의 초기 조건으로 주었을 때, 최종적으로 패턴 쌍 (x_1, y_1) 으로 수렴하는 경우의 수가 95 가지임을 의미한다.

표 1. 본 논문의 방법론에 따른 오차 수정율

	해밍 거리 (HD)				
	0	1	2	3	4
x_1	1/1	20/20	100/100	100/100	95/100
x_2	1/1	20/20	100/100	98/100	98/100
x_3	1/1	20/20	100/100	100/100	100/100
x_4	1/1	20/20	100/100	100/100	99/100
Sum.	4/4	80/80	400/400	398/400	392/400
%	100.0	100.0	100.0	99.5	98.0

또한, 잠음이 섞여 있는 초기조건을 X 층에 입력으로 하였을 때 수렴하는 과정을 그림 2에 나타내었다. 두 경우 모두 X 층의 패턴만을 나타낸 것이고, 가장 가까운 원형패턴과 초기조건에 해밍거리(HD)는 각각 3이다. 왼쪽 처음의 패턴이 입력이고 update과정을 통해 점차 오른쪽으로 이동해 최종적으로 그림 1의 저장된 패턴으로 수렴한다.

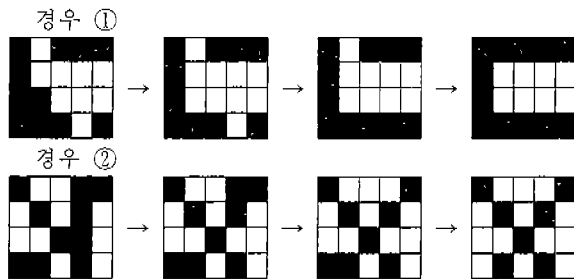


그림 2. 잠음이 섞여 있는 입력 패턴

이번에는 본 논문의 방법론에 의한 바이어스항이 있는 GBAM과 [2]의 방법으로 설계한 GBAM과 [3]의 방법에 따라 설계한 BAM의 오차수정율을 비교하여 그림 3에 나타냈었다.

그림 3에 보여진 결과로부터, 본 논문의 방법론에 따라 설계된 바이어스항이 추가된 GBAM이 여타 방법론에 의해 설계된 GBAM이나 BAM보다 전반적으로 우수한 오차 수정 능력을 가짐을 알 수 있다.

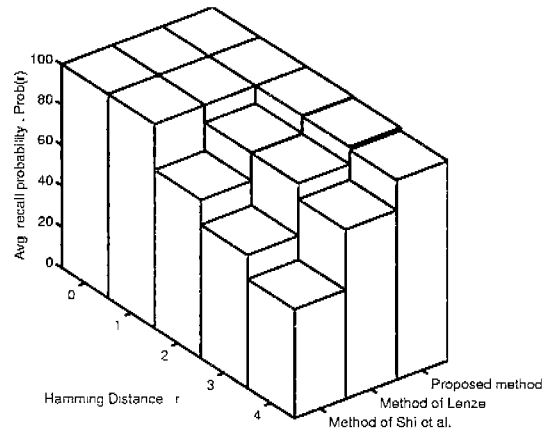


그림 3. 평균 오차수정율의 비교

V. 결 론

본 논문에서는, 높은 오차수정율을 갖는 GBAM을 설계하기 위하여 바이어스항이 추가된 GBAM을 고려하였다. 본문에서 고인대로 바이어스항이 추가된 GBAM은 기존의 GBAM과 유사한 구조를 가지며, 같은 update규칙을 사용하여 더 좋은 오차수정율을 얻을 수 있다. 그리고, 새로운 설계 방법론으로는 보조변수를 통한 GEVP의 최적화 문제를 제시하였다. 이러한 최적화 문제는 특수한 알고리즘이 아닌 MATLAB의 LMI Control Toolbox[5]와 같은 내부점 방법을 사용하는 보편화된 소프트웨어로 구현된다는 장점을 갖는다. 기존에 관련 연구에서 사용되어진 예제들 중 한가지를 사용하여 실험한 결과 기존의 방법으로 설계된 BAM 및 GBAM의 방법보다 바이어스항이 추가된 GBAM의 방법이 우수한 오차 수정 능력을 갖는다는 것으로 나타났다.

VI. 참 고 문 헌

- [1] B. Kosko, "Bidirectional Associative Memories," IEEE Trans. Syst., Man, and Cybern., vol. 18, pp. 49-60, 1988.
- [2] H. Shi, Y. Zhao, and X. Zhuang, "A general model for bidirectional associative memories," IEEE Trans. Syst., Man, Cybern.-Part B, vol. 28, pp. 511-519, 1998.
- [3] B. Lenze, "Complexity preserving increase of the capacity of bidirectional associative memories by dilation and translation," Neural Networks, vol. 10, pp. 1041-1048, 1998.
- [4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in systems and control theory*, Philadelphia, PA: SIAM, 1994.
- [5] P. Gahinet, A. Nemirovskii, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, Natick, MA: The Mathworks Inc., 1994.