Neural Comput & Applic
DOI 10.1007/s00521-010-0351-2

ORIGINAL ARTICLE

The effect of weight fault on associative networks

**Andrew Chi-Sing Leung · Pui Fai Sum ·
Kevin Ho**

Received: 30 June 2009 / Accepted: 12 February 2010

Abstract In the past three decades, the properties of associative networks has been extensively investigated. However, most existing results focus on the fault-free networks only. In implementation, network faults can be exhibited in different forms, such as open weight fault and multiplicative weight noise. This paper studies the effect of weight fault on the performance of the bidirectional associative memory (BAM) model when multiplicative weight noise and open weight fault present. Assuming that connection weights are corrupted by these two common fault models, we study how many number of pattern pairs can be stored in a faulty BAM. Since one of important feature of associative network is error correction, we also study the number of pattern pairs can be stored in a faulty BAM when there are some errors in the initial stimulus pattern.

Keywords Fault tolerance · Associative networks

The work presented in this paper is supported by a research grant from the City University of Hong Kong (Project No. 7002480).

A. C.-S. Leung (✉)
Department of Electronic Engineering,
The City University of Hong Kong, Kowloon Tong, Hong Kong
e-mail: eeleungc@cityu.edu.hk

P. F. Sum
Institute of Electronic Commerce,
National Chung Hsing University, Taichung, Taiwan

K. Ho
Department of Computer Science and Communication
Engineering, Providence University, Shalu, Taiwan

1 Introduction

In the past three decades, associative networks have been intensively studied in the past decade [1–5]. Besides, a wide range of applications [1, 2, 6–8] were proposed, such as document processing [6] and fault diagnosis [8]. One form of associative networks is the bivalent additive bidirectional associative memory (BAM) [3].

A BAM is a two-layer heteroassociator that stores a prescribed set of bipolar pattern pairs, namely library pairs. It consists of two layers of neurons, F_X and F_Y. Layer F_X has n neurons, and layer F_Y has p neurons. The recall process of the BAM is an iterative process between the two layers. The BAM model has some nice features. First, it performs both heteroassociative and autoassociative data recalls. Second, the stimulus input can be fed in layer F_X or layer F_Y. Besides, the BAM is stable during recall. That is, the state of the two layer converges to a fixed point in a finite number of iterations.

In the past, the research on BAM focuses on improve the capacity and capacity analysis. For example, in [9–13] various modified encoding methods were introduced. Besides, BAM has been intensively analyzed with a perfect laboratory environment consideration [5, 14–16]. Amari [5] theoretically showed that the memory capacity of a BAM is equal to $\frac{\min(n,p)}{2\log\min(n,p)}$ for random library pairs. That means, a library pair has a high chance to be stored as a fixed point when the number of library pairs is less than $\frac{\min(n,p)}{2\log\min(n,p)}$. In [15], the memory capacity of the BAM model with a forgetting learning rule was given. Besides, the guideline for choosing the forgetting constant was addressed. Simpson [14] empirically studied the memory capacity of the BAM model with high order interconnection but the theoretical memory capacity has not yet been derived. In [16], the stability and statistical properties of the second order BAM were examined. The result in [16] showed that the stability of the BAM model with high- order interconnection is not guaranteed. For this result, the statistical dynamics was introduced for estimating the capacity of the BAM model with high-order interconnection.

In neural networks, network faults can be exhibited in many different forms, such as weight noise and weight fault. The multiplicative weight noise results from the finite precision representation of trained weights in the implementation [17]. For example, to implement a neural network in digital circuits, the trained weights are usually obtained first by a high precision computer. Then, the trained weights are encoded in digital implementation, like on FPGA [18]. The encoding process will cause a precision problem as the number representation in FPGA is a low precision floating point format [19], which is different from the format used in a computer. In accordance with the studies in [20, 21], the rounding error is proportional to the magnitude of the number encoded. Therefore, a digital implementation of a computer-simulated neural network will lead to a problem identical to adding multiplicative weight noise to that neural network. For example, in the FPGA implementation, the loss of precision in the encoding interconnection weights is equivalent to the multiplicative noise added to the interconnection weights. Distinguishing from the multiplicative weight noise, in open weight fault [22–25], some weights are disconnected to the output layer. For example, in VLSI implementation, some physical faults, such as defects in the silicon, open circuits in metal, and holes in oxides used in transistors [26], may appear. Those implementation defects cause the weights to be failed.

Until now, most researchers focus on the capacity of associative memories and the recall performance of associative memories with initial noise input. Only a few articles [27] studied how the weight noise affects the recall performance. For instance, in [27], the recall performance of a median associative memory was studied in a deterministic way.

This paper focuses on the quantitative impact of weight fault to the BAM capacity in a statistics way. We will study how many number of pattern pairs can be stored as fixed points when weight fault presents. Since the memory capacity is not meaningful without considering the error correction capability, we also present the memory capacity of BAM with weight fault when there are some errors in the input pattern. The rest of this paper is organized as follows. Section 2 introduces the BAM model and multiplicative weight noise. In Sect. 3, the capacity analysis on BAM with multiplicative weight noise is used. Simulation examples are given in Sect. 4. Then, we conclude our work in Sect. 5.

2 Background

2.1 Introduction to BAM

A BAM [3] is a two-layer nonlinear feedback heteroassociative memory. It consists of two layers. Layer F_X has n neurons while layer F_Y has p neurons. A BAM is used to store library pairs $(\vec{X}_1, \vec{Y}_1), \ldots, (\vec{X}_m, \vec{Y}_m)$, where $\vec{X}_h \epsilon \{-1, 1\}^n$ and $\vec{Y}_h \epsilon \{-1, 1\}^p$. The m library pairs are encoded into a connection matrix W. The encoding equation, proposed by Kosko, is given by

$$W = \sum_{h=1}^{m} \vec{Y}_h \vec{X}_h^T \tag{1}$$

which can be rewritten as

$$w_{ji} = \sum_{h=1}^{m} x_{ih} y_{jh}, \tag{2}$$

where $X_h = (x_{1h}, x_{2h}, \ldots, x_{nh})^T$ and $Y_h = (y_{1h}, y_{2h}, \ldots, y_{ph})^T$.

The recall process of the BAM model employs interlayer feedback. With an initial stimulus $\vec{X}^{(0)}$ in F_X, the net input $W\vec{X}^{(0)}$ in layer F_Y is obtained and then is thresholded. Afterward, a new state $Y^{(1)}$ in F_Y is obtained. The new state is passed back through W^T and is thresholded again, leading to a new state $X^{(1)}$ in F_X. The process repeats until the state of BAM converges. Mathematically, the recall process is:

$$\vec{Y}^{(t+1)} = \text{sgn}\left(W\vec{X}^{(t)}\right), \quad \text{and} \quad \vec{X}^{(t+1)} = \text{sgn}\left(W^T \vec{Y}^{(t+1)}\right), \tag{3}$$

where $\text{sgn}(\cdot)$ is the sign operator:

$$\text{sgn}(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \\ \text{state unchanged} & x = 0 \end{cases}$$

Using an element-by-element notation, the recall process can be written as:

$$y_i^{(t+1)} = \text{sgn}\left(\sum_{i=1}^{n} w_{ji} x_i^{(t)}\right), \text{ and}$$

$$x_j^{(t+1)} = \text{sgn}\left(\sum_{j=1}^{p} w_{ji} y_i^{(t+1)}\right), \tag{4}$$

where $x_i^{(t)}$ is the state of the ith F_X neuron and $y_j^{(t)}$ is the state of the jth F_Y neuron. The above bidirectional process produces a sequence of pattern pairs $(X^{(t)}, Y^{(t)}) : (X^{(1)}, Y^{(1)}), (X^{(2)}, Y^{(2)}), \ldots$. This sequence converges to one of the fixed points (X^f, Y^f) and this fixed point ideally should be one of the library pairs or nearly so. A fixed point (X^f, Y^f) has the following properties:

$$\vec{Y}^f = \text{sgn}(W\vec{X}^f) \quad \text{and} \quad \vec{X}^f = \text{sgn}(W^T \vec{Y}^f). \tag{5}$$

Hence, a library pair can be retrieved only if it is a fixed point. One of the advantages of Kosko's encoding method is the ability of incremental learning, i.e., the ability of encoding new library pairs to the model based on the current connection matrix only. With the Kosko's encoding method, when the number of library pairs is less than $\frac{\min(n,p)}{2\log\min(n,p)}$, a library pair can have a high chance to be stored as a fixed point.

2.2 Multiplicative weight noise

An implementation of a weight w_{ji} is denoted by $\tilde{w}_{ji}$. In multiplicative weight noise, each implemented weight deviates from its nominal value by a random percent, i.e.,

$$\tilde{w}_{b,ji} = (1 + b_{ji})\, w_{ji} \tag{6}$$

where b_{ji}'s are identical independent mean zero random variables with variance σ_b^2. The density function of b_{ji}'s is symmetrical.

2.3 Open weight fault

In weight fault model, a implemented weight is given by

$$\tilde{w}_{\beta,ji} = (1 + \beta_{ji}) w_{ji} \tag{7}$$

where β_{ji} 's are independent $\{-1, 0\}$ random variables, Prob $(\beta_{ji} = -1) = \epsilon$. and Prob $(\beta_{ji} = 0) = 1 - \epsilon$. A weight w_{ji} is opened, when $\beta_{ji} = \ -1$.

3 Analysis on BAM with weight fault

3.1 Memory capacity of BAM

We will investigate the BAM's memory capacity when the weight fault is presented.

(1) Assumption and notation: The following assumptions and notations are used.

- The dimensions, n and p, are large. Also, $p = r\ n$, where r is a positive constant.
- Each component of library pairs $(\vec{X}_h, \vec{Y}_h)'$s is $a \pm 1$ equiprobable independent random variable.
- $EU_{b,j,h}$ is the event that $\sum_i^n \tilde{w}_{b,ji} x_{ih}$ is equal to y_{jh} (the j-th component of the library pattern $\vec{Y}_h$). Also, $\overline{EU}_{b,j,h}$ is the complement event of $EU_{b,j,h}$.
- $EV_{b,i,h}$ is the event that $\sum_j^p \tilde{w}_{b,ji} y_{jh}$ is equal to x_{ih} (the i-th component of the library pattern $\vec{X}_h$). Also, $\overline{EV}_{b,i,h}$ is the complement event of $EV_{b,i,h}$.
- $EU_{\beta,j,h}$ is the event that $\sum_i^n \tilde{w}_{\beta,ji} x_{ih}$ is equal to y_{jh} (the j-th component of the library pattern $\vec{Y}_h$). Also, $\overline{EU}_{\beta,j,h}$ is the complement event of $EU_{\beta,j,h}$.
- $EV_{\beta,i,h}$ is the event that $\sum_j^p \tilde{w}_{\beta,ji} y_{jh}$ is equal to x_{ih} (the i-th component of the library pattern $\vec{X}_h$). Also, $\overline{EV}_{\beta,i,h}$ is the complement event of $EV_{\beta,i,h}$.

(2) Useful lemmas: With the above assumptions, we will introduce Lemmas 1–4. They will assist us to derive the memory capacity of BAM with weight fault.

Lemma 1 *The probability $Prob(\overline{EU}_{b,j,h})$ is approximately equal to*

$$Q\left(\sqrt{\frac{n}{(1+\sigma_b^2)m}}\right)$$

for $j = 1, \ldots, n$ *and* $h = 1, \ldots, m$, where $Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty \exp(\frac{-z^2}{2}) \mathrm{d}z$.

Proof Event $\overline{EU}_{b,j,h}$ means that $\mathrm{sgn}\left(\sum_{i=1}^n \tilde{w}_{b,ji} x_{ih}\right) \neq y_{jh}$. From (2) and (6), we have

$$\begin{aligned}\sum_{i=1}^n \tilde{w}_{b,ji} x_{ih} &= \sum_{i=1}^n w_{ji}(1+b_{ji}) x_{ih} \\ &= \sum_{i=1}^n \left(\sum_{h'}^m y_{jh'} x_{ih'}\right)(1+b_{ji}) x_{ih} \\ &= n y_{jh} + \sum_{i=1}^n \left(\sum_{h' \neq h}^m y_{jh'} x_{ih'}\right) x_{ih} \\ &\quad + \sum_{i=1}^n \left(\sum_{h'=1}^m y_{jh'} x_{ih'}\right) b_{ji} x_{ih}\end{aligned} \tag{8}$$

Without loss of generality, we consider the library pair $(\vec{X}_h, \vec{Y}_h)$ having all components positive: $\vec{X}_h = (1, \ldots, 1)^T$ and $\vec{Y}_h = (1, \ldots, 1)^T$. This consideration is usually used [5] and does not affect our results. We can easily verify this by use of conditional probability. Now, (8) becomes

$$\begin{aligned}\sum_{i=1}^n \tilde{w}_{b,ji} x_{ih} &= n + \sum_{i=1}^n \left(\sum_{h' \neq h}^m y_{jh'} x_{ih'}\right) + \sum_{i=1}^n \left(\sum_{h'=1}^m y_{jh'} x_{ih'}\right) b_{ji} \\ &= n + \sum_{i=1}^n \alpha_{ji} + \sum_{i=1}^n \gamma_{ji},\end{aligned} \tag{9}$$

where α_{ji}'s are independent identical zero mean random variables (i.e., $E[\alpha_{ji}] = 0$, and $E[\alpha_{ji}\ \alpha_{ji'}] = 0$ for $i \neq i'$) and the variance of α_{ji}'s, denoted as $\mathrm{Var}[\alpha_{ji}]$, is equal to $(m-1)$. Since γ_{ji}'s are independent identical zero mean random variables and they are independent of $\left(\sum_{h'=1}^m y_{jh'} x_{ih'}\right)$'s. Hence, γ_{ji}'s are independent identical zero mean random variables, where $E[\gamma_{ji}] = 0$, $E[\gamma_{ji}\ \ \gamma_{ji'}] = 0$ for $i \neq i'$, $\mathrm{Var}[\gamma_{ji}] = \sigma_b^2 m$.

For large n, the summations, $\sum_{i=1}^n \alpha_{ji}$ and $\sum_{i=1}^n \gamma_{ji}$, tend to normal with variances equal to $(m-1)n$ and $\sigma_b^2 mn$, respectively. Hence, (9) becomes

$$\sum_{i=1}^{n} \tilde{w}_{b,ji} x_{ih} = n + \sum_{i=1}^{n} \alpha_{ji} + \sum_{i=1}^{n} \gamma_{ji} = n + \alpha + \gamma. \tag{10}$$

We have $E[\alpha\gamma] = E[\alpha]E[\gamma] = 0$, $E[\alpha + \gamma] = 0$ and $\mathrm{Var}[\alpha + \gamma] = (m-1)n + \sigma_b^2 mn$. Besides, the sum of the two normal random variables is still a normal random variable. Let $\varsigma = \alpha + \gamma$. Event $\overline{EU}_{b,j,h}$ means that $\varsigma < -n$. Hence, $\mathrm{Prob}(\overline{EU}_{b,j,h}) \approx Q\left(\sqrt{\frac{n}{(m-1)+\sigma_b^2 m}}\right)$. For large m, $Q\left(\sqrt{\frac{n}{(m-1)+\sigma_b^2 m}}\right) \approx Q\left(\sqrt{\frac{n}{(1+\sigma_b^2)m}}\right)$. (Proof completed).

Using the similar way, we can have Lemma 2.

Lemma 2 *The probability Prob* $(\overline{EV}_{b,i,h})$ *is approximately equal to*

$$Q\left(\sqrt{\frac{p}{(1+\sigma_b^2)m}}\right)$$

for $i = 1, \ldots, p$ *and* $h = 1, \ldots, m$.

Lemma 3 *The probability Prob* $(\overline{EU}_{\beta,j,h})$ is approximately equal to

$$Q\left(\sqrt{\frac{(1-\epsilon)n}{m}}\right)$$

for $j = 1, \ldots, n$ *and* $h = 1, \ldots, m$.

Proof Event $\overline{EU}_{\beta,j,h}$ means that $\mathrm{sgn}\left(\sum_{i=1}^{n} \tilde{w}_{\beta,ji} x_{ih}\right) \neq y_{jh}$. From (2) and (6), we have

$$\sum_{i=1}^{n} \tilde{w}_{\beta,ji} x_{ih} = n y_{jh} + \sum_{i=1}^{n} \left(\sum_{h' \neq h}^{m} y_{jh'} x_{ih'} \right) x_{ih} + \sum_{i=1}^{n} \left(\sum_{h'=1}^{m} y_{jh'} x_{ih'} \right) \beta_{ji} x_{ih} \tag{11}$$

Without loss of generality, we consider the library pair $(\vec{X}_h, \vec{Y}_h)$ having all components positive: $\vec{X}_h = (1, \ldots, 1)^T$ and $\vec{Y}_h = (1, \ldots, 1)^T$. Now, (11) becomes

$$\sum_{i=1}^{n} \tilde{w}_{\beta,ji} x_{ih} = n + \left(\sum_{i=1}^{n} (1 + \beta_{ji}) \left(\sum_{h' \neq h}^{m} y_{jh'} x_{ih'} \right) \right) + \left(\sum_{i=1}^{n} \beta_{ji} \right). \tag{12}$$

Let $\xi_{ji} = (1 + \beta_{ji})(\sum_{h' \neq h}^{m} y_{jh'} x_{ih'})$. So (12) becomes

$$\sum_{i=1}^{n} \tilde{w}_{\beta,ji} x_{ih} = n + \sum_{i=1}^{n} \xi_{ji} + \sum_{i=1}^{n} \beta_{ji}. \tag{13}$$

For large n, the summation $\varphi = \sum_{i=1}^{n} \xi_i$ tends to normal. Besides, the summation $\chi = \sum_{i=1}^{n} \beta_{ji}$ also tends to normal. After careful analysis, we have $E[\varphi + \chi] = -\epsilon n$, and Var $[\varphi + \chi] = (1 - \epsilon)(m - 1 + \epsilon)\, n$. Event $\overline{EU}_{\beta,j,h}$ means that $\varphi + \chi < -n$. Hence, $\mathrm{Prob}(\overline{EU}_{\beta,j,h}) \approx Q\left(\sqrt{\frac{(1-\epsilon)n}{(m-1+\epsilon)}}\right)$. For large m, $Q\left(\sqrt{\frac{(1-\epsilon)n}{m-1+\epsilon}}\right) \approx Q\left(\sqrt{\frac{(1-\epsilon)n}{m}}\right)$. (Proof completed).

Using the similar way, we can have Lemma 4.

Lemma 4 *The probability Prob* $(\overline{EV}_{\beta,j,h})$ *is approximately equal to*

$$Q\left(\sqrt{\frac{(1-\epsilon)p}{m}}\right)$$

for $j = 1, \ldots, p$ *and* $h = 1, \ldots, m$.

(3) Memory Capacity: Now, we start to estimate the memory capacity. For multiplicative weight noise, let the probability that a library pair $(\vec{X}_h, \vec{Y}_h)$ is fixed point be $P*_b$:

$$\begin{aligned} P*_b &= \mathrm{Prob}\left(EU_{b,1h} \cap \cdots \cap EU_{b,nh} \cap EV_{b,1h} \cap \cdots \cap EU_{b,ph}\right) \\ &= 1 - \mathrm{Prob}\left(\overline{EU}_{b,1h} \cup \cdots \cup \overline{EU}_{b,nh} \cup \overline{EV}_{b,1h} \cup \cdots \cup \overline{EV}_{b,ph}\right) \\ &\geq 1 - p\mathrm{Prob}(\overline{EU}_{b,jh}) - n\mathrm{Prob}(\overline{EV}_{b,ih}). \end{aligned} \tag{14}$$

From Lemmas 1 and 2, (14) becomes

$$P*_b \geq 1 - pQ\left(\sqrt{\frac{n}{(1+\sigma_b^2)m}}\right) - nQ\left(\sqrt{\frac{p}{(1+\sigma_b^2)m}}\right). \tag{15}$$

Letting $P_B = pQ\left(\sqrt{\frac{n}{(1+\sigma_b^2)m}}\right)$ and $P_A = nQ\left(\sqrt{\frac{p}{(1+\sigma_b^2)m}}\right)$, we get

$$P*_b \geq 1 - P_B - P_A. \tag{16}$$

If z is large,

$$Q(z) \approx \exp\left\{ -\frac{z^2}{2} - \log z - \frac{1}{2}\log 2\pi \right\}, \tag{17}$$

which is quite accurate for $z > 3$. Using the approximation (17),

$$\begin{aligned} P_A &= \exp\left\{ \log p - \frac{n}{2(1+\sigma_b^2)m} - \frac{1}{2}\log\frac{n}{(1+\sigma_b^2)m} - \frac{1}{2}\log 2\pi \right\} \\ &= \exp\left\{ \log r + \log n - \frac{n}{2(1+\sigma_b^2)m} - \frac{1}{2}\log\frac{n}{(1+\sigma_b^2)m} - \frac{1}{2}\log 2\pi \right\} \\ &= \exp\left\{ \log n - \frac{n}{2(1+\sigma_b^2)m} - \frac{1}{2}\log\frac{n}{(1+\sigma_b^2)m} + \text{constant} \right\} \end{aligned} \tag{18}$$

Clearly, if $m < \frac{n}{2(1+\sigma_b^2)\log n}$, P_A tends zero as n tends infinity. Similarly, we can get that as $p \to \infty$ and $m < \frac{p}{2(1+\sigma_b^2)\log p}$, $P_B \to 0$. To sum up, for large n and p, If

$$m < \frac{\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}, \tag{19}$$

then $P*_b \to 1$. That means for the multiplicative weight noise, if the number m of library pairs is less than $\frac{\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}$, a library pair is with a very high chance to be a fixed point. So the memory capacity of BAM with multiplicative weight noise is equal to

$$\frac{\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}. \tag{20}$$

For open weight fault, let the probability that a library pair $(\vec{X}_h, \vec{Y}_h)$ is fixed point be $P*_\beta$:

$$\begin{aligned} P*_\beta &= \text{Prob}\left(EU_{\beta,1h} \cap \cdots \cap EU_{\beta,nh} \cap EV_{\beta,1h} \cap \cdots \cap EU_{\beta,ph}\right) \\ &= 1 - \text{Prob}\left(\overline{EU}_{\beta,1h} \cup \cdots \cup \overline{EU}_{\beta,nh} \cup \overline{EV}_{\beta,1h} \cup \cdots \cup \overline{EV}_{\beta,ph}\right) \\ &\geq 1 - p\text{Prob}(\overline{EU}_{\beta,jh}) - n\text{Prob}(\overline{EV}_{\beta,ih}). \end{aligned} \tag{21}$$

Based on Lemmas 3 and 4, we can also prove that for open weight fault, if the number m of library pairs is less than $\frac{(1-\epsilon)\min(n,p)}{2\log\min(n,p)}$, a library pair is with a very high chance to be a fixed point. So the memory capacity of BAM with open weight fault is equal to

$$\frac{(1-\epsilon)\min(n,p)}{2\log\min(n,p)}. \tag{22}$$

3.2 Error correction

In this section, we will investigate the memory capacity of BAM with weight fault when the initial input is a noise version X_h^{noise} of a library pattern $\vec{X}_h$. Let X_h^{noise} contains $\rho\, n$ bit errors, where ρ is the input noise level. If

$$\vec{Y}_h = \text{sgn}(\tilde{W}\vec{X}_h^{noise}) \tag{23}$$

$$\vec{X}_h = \text{sgn}(\tilde{W}^T\vec{Y}_h) \tag{24}$$

$$\vec{Y}_h = \text{sgn}(\tilde{W}\vec{X}_h), \tag{25}$$

then a noise version $\vec{X}_h^{noise}$ of $\vec{X}_h$ can successfully recall the correct the desire library pair $(\vec{X}_h, \vec{Y}_h)$. Similarly, we hope that a noise version $\vec{Y}_h^{noise}$ of $\vec{Y}_h$ can successfully recall the correct the desire library pair $(\vec{X}_h, \vec{Y}_h)$. We will study under what condition of m, the probability of successful recall tends to one.

(1) Notations:

- Define $EU_{b,jh}^{noise}$ be the event that $\sum_i^n \tilde{w}_{b,ji}x_{ih}^{noise}$ is equal to y_{jh} (the j-th component of the library pattern $\vec{Y}_h$). Also, $\overline{EU}_{b,jh}^{noise}$ is the complement event of $EU_{b,jh}^{noise}$.
- Also, define $EV_{b,ih}^{noise}$ be the event that $\sum_j^p \tilde{w}_{b,ji}y_{jh}^{noise}$ is equal to x_{ih} (the i-th component of the library pattern $\vec{X}_h$). Also, $\overline{EV}_{b,ih}^{noise}$ is the complement event of $EV_{b,ih}^{noise}$.
- Define $EU_{\beta,jh}^{noise}$ be the event that $\sum_i^n \tilde{w}_{\beta,ji}x_{ih}^{noise}$ is equal to y_{jh} (the j-th component of the library pattern $\vec{Y}_h$). Also, $\overline{EU}_{\beta,jh}^{noise}$ is the complement event of $EU_{\beta,jh}^{noise}$.
- Also, define $EV_{\beta,ih}^{noise}$ be the event that $\sum_j^p \tilde{w}_{\beta,ji}y_{jh}^{noise}$ is equal to x_{ih} (the i-th component of the library pattern $\vec{X}_h$). Also, $\overline{EV}_{\beta,ih}^{noise}$ is the complement event of $EV_{\beta,ih}^{noise}$.

With the above definition, we can follow the proofs of Lemmas 1 and 3 to get the following four lemmas.

Lemma 5 *The probability Prob $(\overline{EU}_{b,ih}^{noise})$ is approximately equal to*

$$Q\left(\sqrt{\frac{(1-2\rho)n}{(1+\sigma_b^2)m}}\right)$$

for $i = 1, \ldots, p$ and $h = 1, \ldots, m$.

Proof Let I be the index set of i's such that $x_{ih}^{noise} = x_{ih}$ and Let $\bar{I}$ be the index set of i's such that $x_{ih}^{noise} = -x_{ih}$. Since there are ρn errors in $\vec{X}_h^{noise}$, the sizes of $|I|$ and $|\bar{I}|$ are equal to $(1-\rho)$ n and ρn, respectively. Event $\overline{EU}_{b,j,h}^{noise}$ means that $\sum_i^n \tilde{w}_{b,ji} \neq y_{jh}$. From (2) and (6), we have

$$\begin{aligned} \sum_{i=1}^n \tilde{w}_{b,ji}x_{ih}^{noise} &= \sum_{i\epsilon I} w_{ji}(1+b_{ji})x_{ih} - \sum_{i\notin I} w_{ji}(1+b_{ji})x_{ih} \\ &= (1-2\rho)ny_{jh} + \sum_{i\epsilon I}\left(\sum_{h'\neq h}^m y_{jh'}x_{ih'}\right)x_{ih} \\ &\quad + \sum_{i\epsilon I}\left(\sum_{h'=1}^m y_{jh'}x_{ih'}\right)b_{ji}x_{ih} \\ &\quad + \sum_{i\notin I}\left(\sum_{h'\neq h}^m y_{jh'}x_{ih'}\right)x_{ih} \\ &\quad + \sum_{i\notin I}\left(\sum_{h'=1}^m y_{jh'}x_{ih'}\right)b_{ji}x_{ih} \end{aligned} \tag{26}$$

Following the proof of Lemma 1, we can have $\text{Prob}(\overline{EU}_{b,j,h}) \approx Q\left(\sqrt{\frac{n}{(m-1)+\sigma_b^2 m}}\right)$. For large m, $Q\left(\sqrt{\frac{n}{(m-1)+\sigma_b^2 m}}\right) \approx Q\left(\sqrt{\frac{n}{(1+\sigma_b^2)m}}\right)$. (Proof completed).

Using the similar way, we can have Lemmas 6 and 7.

Lemma 6 *The probability Prob $(\overline{EV}_{b,ih}^{noise})$ is approximately equal to*

$$Q\left(\sqrt{\frac{(1-2\rho)p}{(1+\sigma_b^2)m}}\right)$$

for $i = 1, \ldots, p$ and $h = 1, \ldots, m$.

Lemma 7 *The probability Prob $(\overline{EU}_{\beta,ih}^{noise})$ is approximately equal to*

$$Q\left(\sqrt{\frac{(1-2\rho)(1-\epsilon)n}{m}}\right)$$

for $i = 1, \ldots, p$ and $h = 1, \ldots, m$.

Lemma 8 *The probability Prob $(\overline{EV}_{\beta,ih}^{noise})$ is approximately equal to*

$$Q\left(\sqrt{\frac{(1-2\rho)(1-\epsilon)p}{m}}\right)$$

for $i = 1, \ldots, p$ *and* $h = 1, \ldots, m$.

For multiplicative weight noise, define $P{**}_b$ be the probability that a noise version with ρ fraction of errors can recall the desired library pair. It is not difficult to show that

$$P{**}_b \geq 1 - p\left(\text{Prob}(\overline{EU}_{b,jh}) + \text{Prob}(\overline{EU}^{noise}_{b,jh})\right) - n\left(\text{Prob}(\overline{EV}_{b,ih} + \text{Prob}(\overline{EV}^{noise}_{b,ih}))\right). \quad (27)$$

From Lemmas 1, 2 and 5, 6 for, large n and p, if

$$m < \frac{(1-2\rho)\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}, \quad (28)$$

then $P{**}_b \to 1$. That means, when there are ρn (or ρp) bit errors in the initial input, the capacity of BAM with multiplicative weight noise is equal to

$$\frac{(1-2\rho)\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}. \quad (29)$$

For open weight fault, define $P{**}_\beta$ be the probability that a noise version with ρ fraction of errors can recall the desired library pair. It is not difficult to show that

$$P{**}_\beta \geq 1 - p\left(\text{Prob}(\overline{EU}_{\beta,jh}) + \text{Prob}(\overline{EU}^{noise}_{\beta,jh})\right) - n\left(\text{Prob}(\overline{EV}_{\beta,ih}) + \text{Prob}(\overline{EV}^{noise}_{\beta,ih}))\right). \quad (30)$$

From Lemmas 3, 4 and 7, 8, for large n and p, if

$$m < \frac{(1-2\rho)(1-\epsilon)\min(n,p)}{2\log\min(n,p)}, \quad (31)$$

then $P{**}_\beta \to 1$. That means, when there are $\rho\ n$ (or ρp) bit errors in the initial input, the capacity of BAM with open weight fault is equal to

$$\frac{(1-2\rho)(1-\epsilon)\min(n,p)}{2\log\min(n,p)}. \quad (32)$$

3.3 Summary

We have shown that under multiplicative weight noise with noise variance equal to σ_b^2, the memory capacity is equal to $\frac{\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}$. Compared with the original fault-free model, the degradation factor is only equal to $\frac{1}{(1+\sigma_b^2)}$. When the initial stimulus contains ρn bit errors, a library pair can be correctly recalled, if the number of library pairs is less than $\frac{(1-2\rho)\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}$.

For open weight fault with fault rate equal to $\epsilon\epsilon$, the memory capacity is equal to $\frac{(1-\epsilon)\min(n,p)}{2\log\min(n,p)}$. Compared with the original fault-free model, the degradation factor is only equal to $(1-\epsilon)$. When the initial stimulus contains ρn bit errors, a library pair can be correctly recalled, if the number of library pairs is less than $\frac{(1-\epsilon)(1-2\rho)\min(n,p)}{2\log\min(n,p)}$.

4 Simulation

4.1 Memory capacity

The memory capacity of BAMs under multiplicative weight noise and open weight fault will be experimentally investigated. We consider that $n = p = 512$. For each m, we randomly generate 1,000 sets of library pairs. The Kosko's rule is then used to encode the matrices.

For the multiplicative weight noise, we add the multiplicative weight noise to the matrices. The variances σ_b^2 of weight noise are set to 0, 0.2, 0.4. Figure 1 shows the percentage of a library pair being successfully stored. From our analysis, i.e., (20), for $n = p = 512$, a BAM can store up to 41, 34, and 29 pairs for σ_b^2 equal to 0, 0.2, and 0.4, respectively. From Fig. 1, all the corresponding successful rates are very high. Also, there are sharply decreasing changes in successful rate for $\{m > 41, \sigma_b^2 = 0\}$, $\{m > 34, \sigma_b^2 = 0.2\}$, and $\{m > 29, \sigma_b^2 = 0.4\}$. To sum up, the simulation result is consistent with our analysis (20).

For the open weight fault, we add the random weight fault to the connection weight. The fault rates ϵ are set to 0.1, 0.2, 0.3. Figure 2 shows the percentage of a library pair being successfully stored. From our analysis, i.e., (22), for $n = p = 512$, a BAM can store up to 41, 32, and 28 pairs for ϵ equal to 0, 0.2 and 0.3, respectively. From Fig. 2, all the corresponding successful rates are very high. Also, there are a sharply decreasing changes in successful

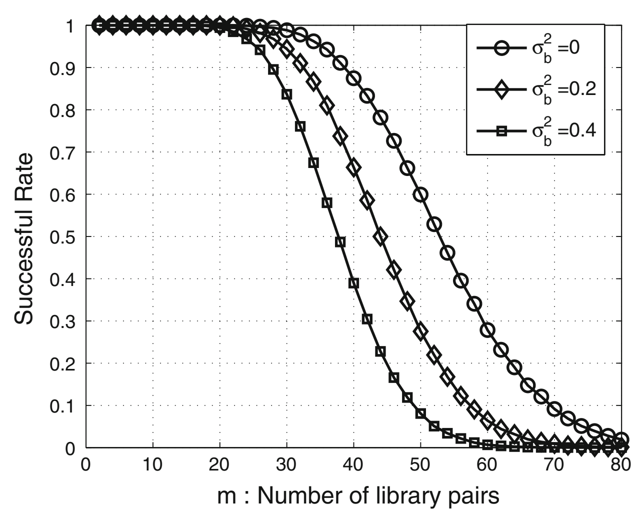

Fig. 1 Successful rate of a library pair being a fixed point under multiplicative weight noise. For each value of m, we generate 1,000 sets of library pairs

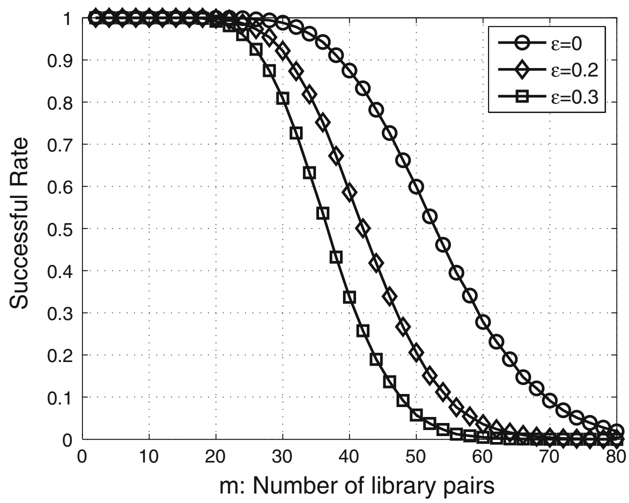

Fig. 2 Successful rate of a library pair being a fixed point under open weight fault. For each value of m, we generate 1,000 sets of library pairs

rate for $\{m > 41, \epsilon = 0\}$, $\{m > 32, \epsilon = 0.2\}$, and $\{m > 28, \epsilon = 0.3\}$. To sum up, the simulation result is consistent with our analysis (22).

4.2 Error correction

We will experimentally investigate the recall performance of faulty BAMs when the initial stimulus is a noise version of a library pattern. The dimension is equal to 512. We considers three input error levels, $\rho = 0.003125, 0.0625, 0.125$. For each m, we randomly generate 1,000 sets of library pairs. The Kosko's rule is then used to encode the matrices.

For the multiplicative noise, we add the multiplicative weight noise to the matrices. For each library pair, we generate ten noise versions. We then feed the noise versions as initial stimulus input and check whether the desire library can be recalled or not. Figure 3 shows the successful recall rate. From our analysis, i.e., (28), for the dimension $n = p = 512$ and weight noise level $\sigma_b^2 = 0.2$, a BAM can store up to 32, 30, and 25 pairs for the input error level ρ equal to 0.03125, 0.0625, and 0.125, respectively. For other weight noise levels, we obtained similar phenomena.

For the open weight fault, we add the random weight fault to the connection weight. The fault rates ϵ are set to 0.1, 0.3. Figure 4 shows the successful recall rate. From our analysis, i.e., (31), for the dimension $n = p = 512$ and open weight fault rate $\epsilon = 0.1$, a BAM can store up to 34, and 27 pairs for the input error level ρ equal to 0.03125 and 0.125, respectively. From Fig. 4, all the corresponding successful rates are high. Also, there are sharply decreasing changes in successful recall rates for $\{m > 34, \rho = 0.03125\}$, and $\{m > 27, \rho = 0.125\}$. For fault rates, we obtained similar results.

4.3 Low precision floating point

In the digital implementation, such as FPGA, we may use low precision floating representation to encode the interconnect weights. As shown in [20], the effect of precision error is similar to the multiplicative noise. Let t is the number of bits allotted to the mantissa. The corresponding rounding error ϵ on a weight w can be modeled as $\tilde{w} = w(1 + \epsilon)$, where ϵ is an independent random variable uniformly distributed in $(-2^{-t}, 2^{-t})$ with variance $\frac{2^{-2t}}{3}$. From the above analysis, when BAM weights (under outer product rule) are encoded by the low precision floating point, the capacity is at least equal to

$$\frac{\min(n, p)}{2(1 + \frac{2^{-2t}}{3}) \log \min(n, p)}. \tag{33}$$

For instance, if we use 3 bits mantissa, the capacity becomes

$$\frac{\min(n, p)}{2(1 + 0.0052) \log \min(n, p)}. \tag{34}$$

Note that even we use a floating point format with 1 bit mantissa, degradation on the capacity is less than 8%.

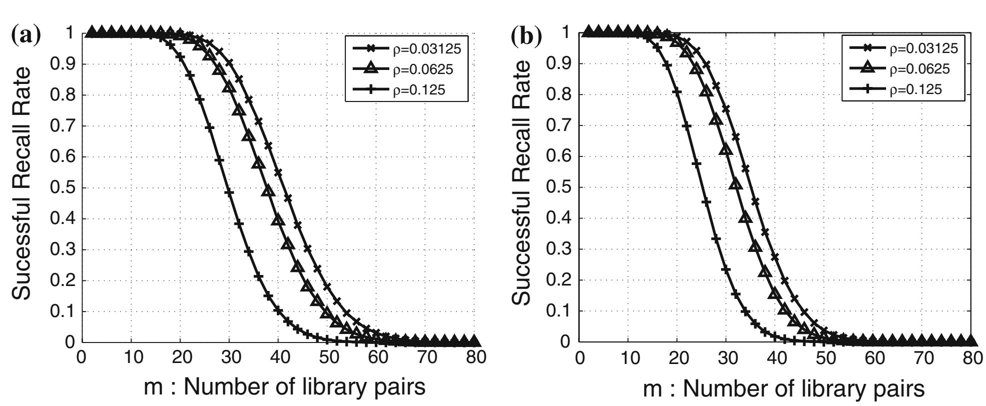

Fig. 3 Successful recall rate from a noise input. For each value of m, we generate 1,000 sets of library pairs. For each library pattern, we generate 10 noise versions. **a** $n = p = 512$, weight noise level $\sigma_b^2 = 0.2$. **b** $n = p = 512$, weight noise level $\sigma_b^2 = 0.4$

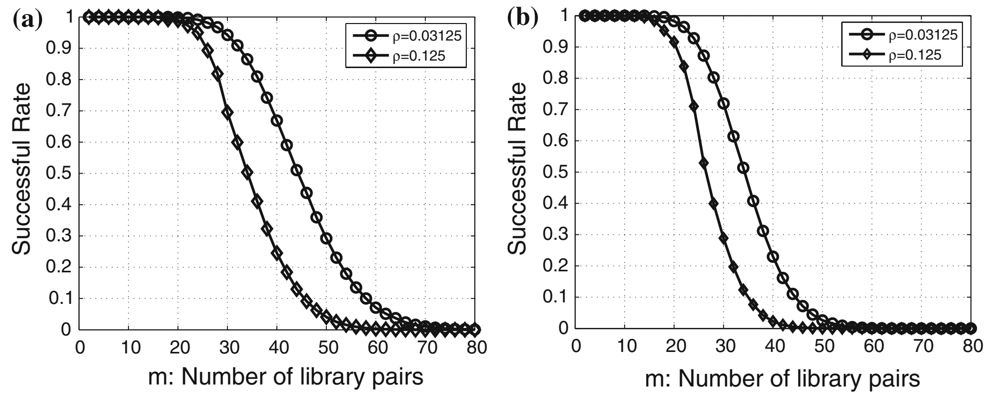

Fig. 4 Successful recall rate from a noise input. For each value of m, we generate 1,000 sets of library pairs. For each library pattern, we generate 10 noise versions. **a** $n = p = 512$, open weight fault rate $\epsilon = 0.1$. **b** $n = p = 512$, open weight fault rate $\epsilon = 0.3$

The memory capacity of BAMs under precision error will be experimentally investigated to verify the above analysis. We consider three cases of dimensions, $n = p = 512$, $n = p = 1{,}024$, $n = p = 2{,}048$. In the simulation, we use a floating point format with 1 bit mantissa and 4 bit exponent to encode the BAM weights. It is because if we assign more bits to the mantissa, the degradation becomes very small and it cannot observe from the simulation.

Figure 5 shows the percentage of a library pair being successfully stored. From Fig. 5, the degradation on the recall performance due to the precision error is very small and it agrees with our expectation.

5 Conclusion

This paper examined the statistical storage behavior of BAM with multiplicative weight noise and open weight fault. Compared with the original fault-free BAM, the degradation factor in the memory capacity is equal to $\frac{1}{1+\sigma_b^2}$

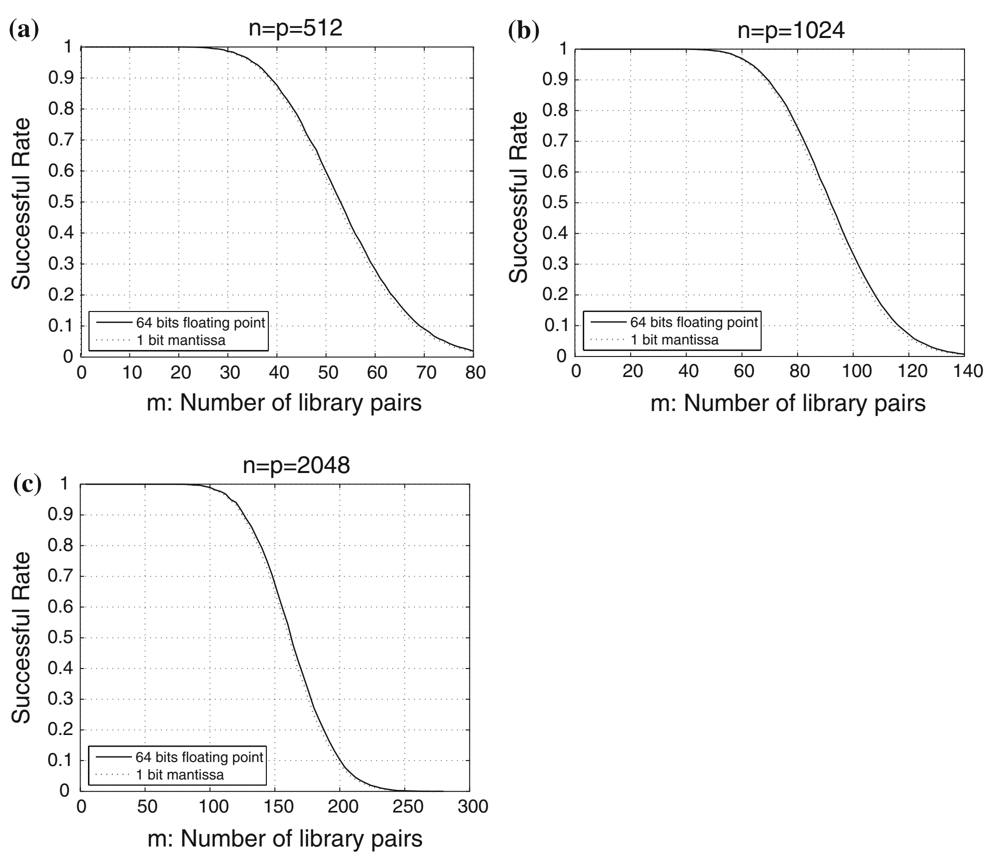

Fig. 5 Successful rate of a library pair being a fixed point under floating error precision error multiplicative weight noise. For each value of m, we generate 100 sets of library pairs. **a** $n = p = 512$. **b** $n = p = 1{,}024$, $n = p = 2{,}048$

when multiplicative weight noise presents. When open weight fault presents, the degradation factor is equal to $(1 - \epsilon)$. Since we expect BAM has certain error correction ability, we have investigated the capacity of BAM with weight noise when the initial input is a noise version of a library pattern. For multiplicative weight noise, we show that if $m < \frac{(1-2\rho)\min(n,p)}{2(1+\sigma_b^2)\log\min(n,p)}$, a noise version with ρn (or ρp) errors has a high chance to recall the desire library pair. For multiplicative weight noise, we show that if $m < \frac{(1-2\rho)(1-\epsilon)\min(n,p)}{2\log\min(n,p)}$, a noise version with $\rho\, n$ (or ρp) errors has a high chance to recall the desire library pair. Computer simulations have been carried out to verify our analysis. Besides, we have found that the degradation on the recall performance is very small when the floating precision error exists. This small degradation proves the robustness of the outer product rule. The results presented here can be extended to Hopfield network. By adopting the approach set above, we can easily obtain the result in Hopfield network by replacing min(n, p) with n in the above equations.

Acknowledgements The work presented in this paper is supported by a research grant from the City University of Hong Kong (Project No. 7002480).

References

1. Kohonen T (1972) Correlation matrix memories. IEEE Trans Comput 21:353–359
2. Palm G (1980) On associative memory. Biolog Cybern 36:19–31
3. Kosko B (1988) Bidirectional associative memories. IEEE Trans Syst Man Cybern 18:49–60
4. Haines K, Hecht-Nielsen R (1988) A bam with increased information storage capacity. In: Proceedings of the 1988 IEEE international conference on neural networks, pp 181–190
5. Amari S (1988) Statistical neurodynamics of various versions of correlation associative memory. In: Proceedings of the 1988 IEEE international conference on neural networks, pp 181–190
6. Oapos SEM, Austin J (1994) An application of an associative memory to the analysis of document images. In: Proceedings of the British machine vision conference
7. Chau FT, Cheung B, Tam KY, Li LK (1995) Application of a bidirectional associative memory (bam) network in computer assisted learning in chemistry. Comput chem 18(4):359–362
8. Ruz-Hernandez JA, Sanchez EN, Suarez DA (2006) Designing an associative memory via optimal training for fault diagnosis. In: Proceedings of IJCNN'06, pp 4338–4345
9. Wang YF, Cruz JB, Mulligan JH (1990) Two coding strategies for bidirectional associative memory. IEEE Trans Neural Netw 1:81–92
10. Leung CS (1993) Encoding method for bidirectional associative memory using projection on convex sets. IEEE Trans Neural Netw 4:879–991
11. Leung CS (1994) Optimum learning for bidirectional associative memory in the sense of capacity. IEEE Trans Syst Man Cybern 24:791–796
12. Lenze B (2001) Improving Leung's bidirectional learning rule for associative memories. IEEE Trans Neural Netw 12:1222–1226
13. Shen D, Cruz JB (2005) Encoding strategy for maximum noise tolerance bidirectional associative memory. IEEE Trans Neural Netw 16:293–300
14. Simpson PK (1990) Higher-ordered and intraconnected bidirectional associative memories. IEEE Trans Syst Man Cybern 20:637–653
15. Leung CS, Chan LW (1997) The behavior of forgetting learning in bidirectional associative memory. Neural Comput 9:385–401
16. Leung CS, Chan LW, Lai E (1997) Stability and statistical properties of second-order bidirectional associative memory. IEEE Trans Neural Netw 8:267–277
17. Burr JB (1991) Digital neural network implementations. Neural Netw Concepts Appl Implement III:237–285
18. Himavathi S, Anitha D, Muthuramalingam A (2007) Feedforward neural network implementation in fpga using layer multiplexing for effective resource utilization. IEEE Trans Neural Netw 18:880–888
19. Antony MM, Savich W, Areibi S (2007) The impact of arithmetic representation on implementing mlp-bp on fpgas: a study. IEEE Trans Neural Netw 18:240–252
20. Kaneko T, Liu B (1970) Effect of coefficient rounding in floating-point digital filters. IEEE Trans Aerosp Electron Syst AE-7:995–1003
21. Liu B, Kaneko T (1969) Error angalysis of digital filter realized with floating-point arithemetic. Proc IEEE 57:1735–1747
22. Phatak DS, Koren I (1995) Complete and partial fault tolerance of feedforward neural nets. IEEE Trans Neural Netw 6:446–456
23. Emmerson MD, Damper RI (1993) Determining and improving the fault tolerance of multilayer perceptrons in a pattern-recognition application. IEEE Trans Neural Netw 4:788–793
24. Zhou ZH, Chen SF (2003) Evolving fault-tolerant neural networks. Neural Comput Appl 11:156–160
25. Sing LC, John S (2008) A fault-tolerant regularizer for rbf networks. IEEE Trans Neural Netw 19:493–507
26. Bolt G (1991) Fault models for artificial neural networks. In: 1991 IEEE international joint conference on neural networks, vol 2, pp 1373–1378
27. Sossa H, Barron R, Vazquez RA (2007) Study of the influence of noise in the values of a median associative memory. In: ICANNGA 2007 (2), NCS 4432