The Behavior of Forgetting Learning in Bidirectional
Associative Memory

Chi Sing Leung
Lai Wan Chan
Department of Computer Science and Engineering,
Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Forgetting learning is an incremental learning rule in associative memo-
ries. With it, the recent learning items can be encoded, and the old learning
items will be forgotten. In this article, we analyze the storage behavior of
bidirectional associative memory (BAM) under the forgetting learning.
That is, “Can the most recent \( k \) learning item be stored as a fixed point?”
Also, we discuss how to choose the forgetting constant in the forgetting
learning such that the BAM can correctly store as many as possible of the
most recent learning items. Simulation is provided to verify the theoret-
ical analysis.

1 Introduction

Associative memory is a major class of neural networks with a wide range of
applications, such as content-addressable memory and pattern recognition
(Kohonen 1972; Palm 1980). An important feature of associative memory
is its associative nature, that is, the ability to recall the stored item based
on partial or noisy information. Associative memory encodes and recalls
library patterns or pattern pairs. If it encodes single patterns, it is called
autoassociative memory. If it encodes pattern pairs, it is called heteroas-
sociative memory. One form of heteroassociative memory is the bivalent
additive bidirectional associative memory (BAM) (Kosko 1988).

The BAM is used to store bipolar library pairs \((X_h, Y_h)\), where \(X_h = (x_{1h}, \ldots, x_{nh})^T\), \(Y_h = (y_{1h}, \ldots, y_{ph})^T\), \(h = 1, \ldots, m\), and \(m\) is the number of
library pairs. The components \(x_{ih}\) and \(y_{jh}\) are bipolar: either \(+1\) or \(-1\). There
are two layers in the BAM: layer \(F_X\) and layer \(F_Y\), with \(n\) and \(p\) neurons,
respectively. The connection matrix \(W\), proposed by Kosko, is

\[
W = \sum_{h=1}^{m} Y_h X_h^T.
\]
The retrieval process is an iterative feedback process that starts with a stimulus vector $X^{(0)}$ in $F_X$:

$$
Y^{(v+1)} = \text{sgn}\left[WX^{(v)}\right],
$$
$$
X^{(v+1)} = \text{sgn}\left[W^T Y^{(v+1)}\right],
$$

(1.2)

where $\text{sgn}$ is defined as

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

Kosko proved that for any real connection matrix, one of fixed points $(X_f, Y_f)$ can be obtained from this iterative process. Apparently such a fixed point is desired to be one of the library pairs. A fixed point has the following properties:

$$
X_f = \text{sgn}(W Y_f) \quad \text{and} \quad Y_f = \text{sgn}(W^T X_f).
$$

(1.3)

Hence, a library pair can be retrieved only if it is a fixed point. With the iterative process, the BAM can achieve both heteroassociative and autoassociative data recollections. The final state in $F_X$ represents the autoassociative recall. The final state in $F_Y$ represents the heteroassociative recall.

The storage behavior of the BAM under Kosko’s encoding method and its variants has been studied by various researchers (Haines and Nielsen 1988; Leung et al. 1995; Leung 1993; Wang et al. 1990). An advantage of Kosko’s encoding method is the ability of incremental learning; encoding new library pairs to the model based on only the current connection matrix. However, with Kosko’s encoding method, the BAM can only correctly store up to $\min(n, p)/2 \log \min(n, p)$ library pairs. When the total number of library pairs exceeds that value, all library pairs, including the old and new items, may not be stored as fixed points. To avoid this matter, we may require that the BAM has the ability to forget; that is, the model should be able to create space for the new library pairs. In modeling the forgetting behavior of the BAM, we would like to have three physiological requirements:

1. **Locality.** Each weight should be updated based on the values of the corresponding connected neurons.

2. **Being incremental.** The updated rule is based on only the current connection matrix and is independent of which library pairs have been stored.

3. **Boundedness.** The magnitude of the weights should be bounded.

To achieve these requirements, we introduce a decay factor $\alpha_f \in (0, 1)$, called the forgetting constant, in the original Kosko’s encoding method,

$$
W_t = \alpha_f W_{t-1} + Y_t X_t^T,
$$

(1.4)
where $W_{(0)}$ is a zero matrix and $(X_t, Y_t)$ is the new library pair that we want to encode into the BAM. Equation 1.4 represents a simple form of forgetting learning. It is the discrete-time version of the original continuous-time forgetting rule in Kokso (1992). According to the decay factor, the forgetting learning can encode the recent library pairs as fixed points with a high chance and can delete some old stored library pairs from the model. Also, with our suggestion on $\alpha_f$ (see Section 3), the magnitude of the weights is upper bounded by $O(n/\log n)$ in the worst case and upper bounded by $O(\sqrt{n})$ in the probabilistic sense (see Section 4). However, similar to some conventional backpropagation rules, real-valued computations are involved in equation 1.4. The hardware implementation of equation 1.4 is more difficult than that of Kosko’s encoding scheme. We will use simulation to investigate the behavior of equation 1.4 using the fixed-point computation machine described in Section 3.

The storage behavior of a Hopfield network under some similar forgetting rules was studied numerically or theoretically by Nadal et al. (1986), Hemmen et al. (1988), and Parisi (1986). They showed that the previous $\lambda n$ library patterns can be stored in a Hopfield network when a small number of errors are allowed in the recalling patterns, where $n$ is the number of neurons in the Hopfield network and $\lambda$ is a positive constant less than one. (For a brief review, see Hemmen and Kuhn 1991).

The learning within bounds (Hopfield 1982) was proposed for the Hopfield network and studied numerically by Parisi (1986). Its updating rule is

$$W_{(t)} = \phi \left( W_{(t-1)} + X_t X_t^T \right),$$

where $\phi(x) = x$ for $|x| < A\sqrt{n}$; otherwise $\phi(x) = \text{sgn}(x)A\sqrt{n}$. Since we are now discussing the Hopfield model, the second term of the right-hand side in equation 1.5 is the outer product of the library pattern itself. Let $k'$ be the number of most recently stored patterns that are satisfactorily kept in the memory. Parisi (1986) numerically showed that $k'$ can be proportional to $n$ when a small number of errors are allowed in the retrieved items. Using the nonrigorous replica method, (Hemmen et al. 1988) showed that with a suitable constant $A \approx 0.35$ the recent $0.04n$ previous library patterns can be kept in the memory when a small number of errors are allowed in the retrieved items. Since learning within bounds involves only integer-valued weights, its hardware implementation is easier than that of equation 1.4. However, the replica method for averaging over the disorder in the system due to the different possible realizations of nominal patterns is strictly valid only for a temperature $T' > 0$ (Forrest and Wallace 1991). In fact, the “$T' = 0$” in Hemmen et al. (1988) violates this criterion. It should also be noted that for the BAM, the learning within bounds can be used only when $p = n$. To our knowledge, for the BAM the selection role about the constant $A$ in the learning within bounds has not been reported when $p \neq n$. 

Downloaded from http://www.mitpressjournals.org/doi/pdfplus/10.1162/neco.1997.9.2.385 by guest on 24 April 2021
In Nadal et al. (1986), the marginalist rule was proposed for the Hopfield network. Its updating rule is
\[ W(t) = W(t-1) + \delta^{t-1}X_tX_t^T, \]  
(1.6)
where \( \delta \) is a constant larger than one. The concept of the marginalist rule is that the acquisition intensity for learning the last pattern should be greater than the background intensity. Hence, the last pattern can be stored with high chance. Nadal et al. (1986) numerically showed that \( k' \) also can be proportional to \( n \) for the marginalist rule when a small number of errors are allowed in the retrieved patterns. However, the role of choosing \( \delta \) (with which the number of recent library items being nearly correctly stored is as large as possible) was not mentioned in Nadal et al. (1986).

Although equations 1.4 and 1.6 are similar after some mathematical treatments, their effects on the magnitude of the weights are different. With equation 1.6, the magnitude exponentially increases. On the other hand, with equation 1.4, the magnitude is upper bounded by \( O(n \log n) \) in the worst case and upper bounded by \( O(\sqrt{n}) \) in the probabilistic sense. Note that equation 1.4 can also be written as the sum of the old matrix and an updating matrix:
\[ W(t) = W(t-1) + \Delta W(t), \]  
(1.7)
where
\[ \Delta W(t) = (\alpha_f - 1)W(t-1) + Y_tX_t^T. \]

Instead of using the classical nonrigorous replica method and allowing error in the retrieved pattern, our main goal is to study the storage behavior of the BAM under the simple forgetting rule (cf. equation 1.4) when error is not allowed in the retrieved pair. Also, we discuss the role of choosing the forgetting constant \( \alpha_f \).

Under some assumptions, we will prove the following theorem in the next section.

**Theorem 1.** Under the forgetting learning (cf. equation 1.4), if a BAM is trained with \( t \) library pairs and \( k \) is less than
\[ \log \left( \frac{(1-\alpha_f^2) \min(n,p)}{2 \log \min(n,p)} \right) \frac{1}{2 \log \frac{1}{\alpha_f}} \]  
(1.8)
then the probability of the \( (t-k) \)th library pair \((X_{t-k}, Y_{t-k})\) being a fixed point tends to 1, as \( n \to \infty, p \to \infty \).
The Behavior of Forgetting Learning in Bidirectional Associative Memory 389

By theorem 1, given $\alpha_f$, we can determine the number of the most recent library pairs that the BAM can correctly store. Hence, the best value of $\alpha_f$ can be determined such that the BAM can correctly store as many of the most recent library pairs as possible.

In the next section, we will analyze the storage behavior of the BAM under forgetting learning. We first prove theorem 1. Hence, we can determine whether the most recent $k$ library pair, $(X_{t-k}, Y_{t-k})$, can be stored as a fixed point. We then discuss how to choose the forgetting constant such that $\log \frac{(1-\alpha_f^2) \min(n, p)}{2 \log \frac{1}{\alpha_f}}$ is as large as possible (see corollary 1). In Section 3, the simulation is given for the verification of the analysis. Section 4 addresses the magnitude of the weights in deterministic and probabilistic senses. A brief conclusion comprises Section 5.

2 Properties of Forgetting Learning

The following assumptions and notations are used:

- The dimensions, $n$ and $p$, are large. Also, $p = rn$, where $r$ is a positive constant.
- Each component of the library pairs $(X_h, Y_h)$ is a $\pm 1$ equiprobable independent random variable.
- $EU_{j,t-k}$ is the event that the $j$th component of $\text{sgn}(W^{(t)}X_{t-k})$ is equal to the $j$th component of $Y_{t-k}$. Also, $\overline{EU}_{j,t-k}$ is the complement event of $EU_{j,t-k}$.
- $EV_{i,t-k}$ is the event that the $i$th component of $\text{sgn}(W^{(t)}Y_{t-k})$ is equal to the $i$th component of $X_{t-k}$. $\overline{EV}_{i,t-k}$ is the complement event of $EV_{i,t-k}$.

**Lemma 1.** Chebychev’s inequality: For any random variable $\chi$ and $u \geq 0$,

$$\text{Prob}(\chi \geq u) \leq \inf_{\tau \geq 0} e^{-\tau u} E(e^{\tau \chi}).$$

**Lemma 2.** Let $v_1, v_2, \ldots, v_N$ be independent $\pm 1$ equiprobable random variables and $S_N = \sum_{i=1}^{N} v_i$,

$$E \left[ \exp \left\{ \tau \frac{S_N}{\sqrt{N}} \right\} \right] \leq E \left[ \exp \left\{ \tau \hat{S} \right\} \right] = \exp \left\{ \frac{\tau^2}{2} \right\}.$$
for \( \tau > 0 \), where \( \hat{S} \) is a standard normal variable. In general,

\[
E \left[ \exp \left\{ \tau \beta \frac{S_N}{\sqrt{N}} \right\} \right] \leq \exp \left\{ \frac{\beta^2 \tau^2}{2} \right\}
\]

where \( \beta \) is a real number.

**Lemma 3.** The probability \( \text{Prob}(E \bigcup_{j, t-k}) \) is less than

\[
\exp \left\{ -\frac{(1 - \alpha_f^2)\alpha_f^{2h} n}{2} \right\}
\]

for \( j = 1, \ldots, p \).

**Proof of Lemma 3.** Without loss of generality, we consider that the library pair \((X_{t-k}, Y_{t-k})\) has all positive components; that is, \(X_{t-k} = (1, \ldots, 1)^T\) and \(Y_{t-k} = (1, \ldots, 1)^T\). Then,

\[
\text{Prob}(E \bigcup_{j, t-k}) = \text{Prob} \left( \sum_{h=1, h \neq t-k}^t \alpha_f^{t-h} \frac{S_{j,h}}{\sqrt{n}} > \alpha_f^k \frac{\sqrt{n}}{\sqrt{n}} \right).
\]

(2.1)

where

\[
S_{j,h} = y_{jh} \sum_{i=1}^n x_{ih}.
\]

From lemma 2,

\[
E \left[ \exp \left\{ \tau \alpha_f^{t-h} \frac{S_{j,h}}{\sqrt{n}} \right\} \right] \leq \exp \left\{ -\frac{\alpha_f^{2t-2h} \tau^2}{2} \right\}.
\]

By Chebyshev’s inequality (cf. lemma 1),

\[
\text{Prob}(E \bigcup_{j, t-k}) \leq \inf_{\tau \geq 0} \exp\{-\tau \mu\} \exp \left\{ \frac{\sum_{h=1, h \neq t-k}^t \alpha_f^{2t-2h} \tau^2}{2} \right\}
\]

\[
\leq \inf_{\tau \geq 0} \exp\{-\tau \mu\} \exp \left\{ \frac{\sum_{h=0}^\infty \alpha_f^{2h} \tau^2}{2} \right\}
\]

\[
\leq \inf_{\tau \geq 0} \exp\{-\tau \mu\} \exp\left\{ \frac{\tau^2}{2(1 - \alpha_f^2)} \right\}
\]
where \( u = \alpha_f \sqrt{n} \). Clearly, the right-hand side becomes minimum when \( \tau = u(1 - \alpha_f^2) \).

With such a value of \( \tau \),

\[
\text{Prob}(E_U_{j,t-k}) \leq \exp \left\{ -\frac{(1 - \alpha_f^2)u_f^{2j}n}{2} \right\}
\]

for \( j = 1, \ldots, p \).

**Lemma 4.** The probability \( \text{Prob}(E_V_{i,t-k}) \) is less than

\[
\exp \left\{ -\frac{(1 - \alpha_f^2)p_f^{2i}p}{2} \right\}
\]

for \( i = 1, \ldots, n \).

**Proof of Lemma 4.** The proof is similar to that of lemma 3.

With lemmas 3 and 4, theorem 1 can be proved in the following way: We denote the probability that \((X_{t-k}, Y_{t-k})\) is a fixed point as \( P_* \). Then,

\[
P_* = \text{Prob} \left( E U_{1,t-k} \cap \cdots \cap E U_{p,t-k} \cap E V_{1,t-k} \cap \cdots \cap E V_{n,t-k} \right)
\]

\[
= 1 - \text{Prob} \left( E U_{1,t-k} \cup \cdots \cup E U_{p,t-k} \cup E V_{1,t-k} \cup \cdots \cup E V_{n,t-k} \right)
\]

\[
\geq 1 - p \text{Prob} \left( E U_{1,t-k} \right) - n \text{Prob} \left( E V_{1,t-k} \right). \quad (2.2)
\]

One should be aware that

\[
P_* \neq \left( 1 - \text{Prob}(E U_{1,t-k}) \right)^{p} \left( 1 - \text{Prob}(E V_{1,t-k}) \right)^{n} \quad (2.3)
\]

because the above events, \( E U_{j,t-k}'s \) and \( E V_{i,t-k}'s \), are not mutually independent.

With lemmas 3 and 4, it is easy to see that in the following condition the right-hand side of equation 2.2 tends to one as \( n \) tends to \( \infty \) (also \( p \to \infty \) due to \( p = \frac{2}{n} \) and \( r \) is a constant).

\[
k < \min \left( \frac{\log (1 - \alpha_f^2)n}{2 \log \frac{1}{\alpha_f}}, \frac{\log (1 - \alpha_f^2)p}{2 \log \frac{1}{\alpha_f}} \right) = \frac{\log (1 - \alpha_f^2) \min(n, p)}{2 \log \frac{1}{\alpha_f}}.
\]

Thus, theorem 1 is obtained. From theorem 1, the most recent \( k \) library pair, \((X_{t-k}, Y_{t-k})\), can be stored as a fixed point with high probability if

\[
k < \frac{\log(1 - \alpha_f^2) \min(n, p)}{2 \log \frac{1}{\alpha_f}}.
\]
Given $p$ and $n$, let
\[
 f(\alpha_f) = \log \left(\frac{1-\alpha_f^2 \min(n, p)}{2 \log \min(n, p)}\right) / 2 \log \frac{1}{\alpha_f}. \tag{2.4}
\]

The most interesting point is how to choose the value of $\alpha_f \in (0, 1)$ such that $f(\alpha_f)$ is maximal. We denote the value of $\alpha_f$, with which $f(\alpha_f)$ is maximal, as $\alpha_{f, \text{max}}$.

Clearly, $f(\alpha_f)$ is a continuous function of $\alpha_f \in (0, 1)$. Also,
\[
 \frac{d f(\alpha_f)}{d \alpha_f} \bigg|_{\alpha_f=0^+} > 0
\]
and
\[
 \frac{d f(\alpha_f)}{d \alpha_f} \bigg|_{\alpha_f=1^-} < 0.
\]

Hence, $f(\alpha)$ has at least one local maximum when $0 < \alpha_f < 1$. A typical plot of $f(\alpha_f)$ is shown in Figure 1.

Using simple numerical method, we obtain Table 1, which summarizes the values of $\alpha_{f, \text{max}}$ at different values of $\min(n, p)$. From the table, as $\min(n, p)$
The Behavior of Forgetting Learning in Bidirectional Associative Memory

Table 1: Summary of $\alpha_{f,\text{max}}$ and $f(\alpha_{f,\text{max}})$ Values, Found from Numerical Method.

<table>
<thead>
<tr>
<th>min$(n, p)$</th>
<th>$\alpha_{f,\text{max}}$</th>
<th>$f(\alpha_{f,\text{max}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.613</td>
<td>0.6012</td>
</tr>
<tr>
<td>32</td>
<td>0.742</td>
<td>1.223</td>
</tr>
<tr>
<td>64</td>
<td>0.837</td>
<td>2.3453</td>
</tr>
<tr>
<td>128</td>
<td>0.902</td>
<td>4.3611</td>
</tr>
<tr>
<td>256</td>
<td>0.943</td>
<td>7.9967</td>
</tr>
<tr>
<td>512</td>
<td>0.9675</td>
<td>14.599</td>
</tr>
<tr>
<td>1024</td>
<td>0.982</td>
<td>26.673</td>
</tr>
<tr>
<td>2048</td>
<td>0.9900</td>
<td>48.906</td>
</tr>
<tr>
<td>4096</td>
<td>0.9945</td>
<td>90.079</td>
</tr>
<tr>
<td>8192</td>
<td>0.9970</td>
<td>166.721</td>
</tr>
</tbody>
</table>

Table 2: Summary of $\alpha_{f,\text{max}}$ and $f(\alpha_{f,\text{max}})$ Values, Found from Equation 2.5.

<table>
<thead>
<tr>
<th>min$(n, p)$</th>
<th>$\alpha_{f,\text{max}}$</th>
<th>$f(\alpha_{f,\text{max}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.2407</td>
<td>0.35</td>
</tr>
<tr>
<td>32</td>
<td>0.6412</td>
<td>1.13</td>
</tr>
<tr>
<td>64</td>
<td>0.8042</td>
<td>2.29</td>
</tr>
<tr>
<td>128</td>
<td>0.9810</td>
<td>4.33</td>
</tr>
<tr>
<td>256</td>
<td>0.9393</td>
<td>7.98</td>
</tr>
<tr>
<td>512</td>
<td>0.9663</td>
<td>14.59</td>
</tr>
<tr>
<td>1024</td>
<td>0.9814</td>
<td>26.67</td>
</tr>
<tr>
<td>2048</td>
<td>0.9989</td>
<td>48.91</td>
</tr>
<tr>
<td>4096</td>
<td>0.9945</td>
<td>90.08</td>
</tr>
<tr>
<td>8192</td>
<td>0.9970</td>
<td>166.72</td>
</tr>
</tbody>
</table>

increases, $\alpha_{f,\text{max}}$ increases. Also, for large min$(n, p)$, $\alpha_{f,\text{max}}$ is nearly equal to one. Based on this phenomenon, we can obtain the approximation of $\alpha_{f,\text{max}}$:

$$\alpha_{f,\text{max}} \approx \sqrt{1 - \exp \left\{ -\log \frac{\min(n, p)}{2 \log \min(n, p)} + 1 \right\}}.$$ (2.5)

Equation 2.5 gives us a guideline to choose the value of $\alpha_f$. Table 2 shows $\alpha_{f,\text{max}}$ at different values of min$(n, p)$ based on equation 2.5. From the two tables, equation 2.5 is a good approximation of $\alpha_{f,\text{max}}$ when min$(n, p)$ is large.

Substituting equation 2.5 into theorem 1, we can obtain the following corollary.
Corollary 1. Under the forgetting learning with

$$\alpha_{f,\text{max}} \approx \sqrt{1 - \exp\left\{-\log \frac{\min(n, p)}{2 \log \min(n, p)} + 1\right\}}$$

if a BAM is trained with \( t \) library pairs and \( k \) is less than

$$\frac{\min(n, p)}{2e \log \min(n, p)^2}.$$  \hspace{1cm} (2.6)

then the probability of the \((t-k)\)th library pair \((X_{t-k}, Y_{t-k})\) being a fixed point tends to one, as \( n \to \infty, p \to \infty \).

From corollary 1, the storage ability of the forgetting learning is similar to that of Kosko’s encoding method. With Kosko’s encoding method, the BAM can encode only up to \( \min(n, p)/2 \log \min(n, p) \) library pairs. Further encoding of the new library pairs will damage the entire system. On the other hand, the forgetting learning can encode any number of library pairs and always keep the most recent \( \min(n, p)/2e \log \min(n, p) \) library pairs in the model. Hence, the BAM with the forgetting learning is similar to short-term memory, which always extracts the most recent information from the environment.

3 Simulation

3.1 Optimal Forgetting Constant. We have carried out a simulation to verify equation 2.5 and corollary 1. The dimensions are 512. We randomly generate the library pairs and use the forgetting rule to encode them with different forgetting constants. Figure 2 shows the percentage of the most recent \( k \) library pair being successfully stored. Note that \( k = 0 \) means the current library pair.

Suppose that we use 90% as the threshold. From Figure 2, if the value of \( \alpha_f \) is too large (such as 0.990), all library pairs (both old and new items) may not be stored as fixed points. When we use a very small value of \( \alpha_f \) (such as 0.940), the number of the most recent items being stored as fixed points becomes small.

In general, the case of \( \alpha_f = 0.9663 \) (based on Table 2) is better than other cases. Also, there is a sharply decreasing change for \( k > 14 \) when \( \alpha_f = 0.9663 \). This result is consistent with our theoretical work presented in Tables 1 and 2.

3.2 Comparison with Learning Within Bounds. Additionally, we have carried out another simulation to compare the performance of equation 1.4 with that of the learning within bounds (Hopfield 1982). Note that for the
Figure 2: The percentage of the most recent previous library pair being successfully stored. Here, error is not allowed in the retrieved items and $n = p = 512$. The forgetting rule is equation 1.4, and the forgetting constant is varied.

BAM, learning within bounds can be used only when $p = n$. To our knowledge, for the BAM the selection role about the constant $A$ in the learning within bounds has not been reported when $p \neq n$.

The dimensions are 512. For equation 1.4, the forgetting constant is based on equation 2.5. For learning within bounds, the constant $A$ is 0.35, which is the optimal value in Hemmen et al. (1988).

Figure 3 shows the percentage of the most recent $k$ previous library pair being successfully stored when error is not allowed in the retrieved items. For equation 1.4, the BAM can store the recent 14 library pairs. However, for learning within bounds, the BAM can store only the recent 5 library pairs. Hence, the performance of equation 1.4 is much better than that of learning within bounds when error is not allowed in the retrieved items.

Figure 4 shows the percentage of the most recent $k$ previous library pair being successfully stored when 5% errors are allowed in the retrieved items. For equation 1.4, the BAM can store the recent 24 library pairs. For learning within bounds, the BAM can store the recent 27 library pairs. Hence, the performance of learning within bounds is only a little better than that of equation 1.4 when a small number of errors is allowed in the retrieved items.
Figure 3: The percentage of the most recent \( k \) previous library pair being successfully stored. Here, error is not allowed in the retrieved items and \( n = p = 512 \). Two forgetting rules are used: equation 1.4 and learning within bounds. The forgetting constant \( \alpha_f \) in equation 1.4 is based on equation 2.5. The constant \( A \) in the “learning within bounds” is 0.35, which is the optimal value suggested in Hemmen et al. (1988). From this simulation, the performance of equation 1.4 is much better than that of the “learning within bounds” when error is not allowed in the retrieved items.

Figure 4: The percentage of the most recent \( k \) previous library pair being successfully stored. Here, 5% errors in the retrieved items are allowed and \( n = p = 512 \). Two forgetting rules are used: equation 1.4 and “learning within bounds.” The forgetting constant \( \alpha_f \) in equation 1.4 is based on equation 2.5. The constant \( A \) in the “learning within bounds” is 0.35, which is the optimal value suggested in Hemmen et al. (1988). From this simulation, the performance of the learning within bounds is only a little better than that of equation 1.4 when a small number of errors are allowed in the retrieved items.
3.2.1 Behavior Under Fixed-Point-Number Machine. The real-valued computation in equation 1.4 creates some difficulties in the implementation. Here, we investigate the behavior of equation 1.4 when the precision of the weights, as well as the precision of the computation, is limited. Three schemes of the precision are considered. The dimensions are 512. We randomly generate the library pairs and use the forgetting rule under limited precision to encode them.

In the first scheme, the precision is \( \pm 19.999 \) (4\( \frac{1}{2} \) digits machine). Due to the limitation of the precision, \( \alpha_f \) is set to 0.966. This means that all computations (the intermediate and final results) in equation 1.4 are rounded to a fixed-point number with three decimal places between \( \pm 19.999 \). The simulation result is shown in Figure 5. From Figure 5, the performance under this precision is quite similar to that under float-point number precision (see Fig. 2 also).

The second precision is \( \pm 19.99 \) (3\( \frac{1}{2} \) digits machine), and \( \alpha_f \) is set to 0.96. The simulation result is shown in Figure 6. From Figure 6, the BAM can still store 14 recent library pairs when error is not allowed in the retrieved items. When 5% errors are allowed, it can store only about 22 recent library pairs.

The third precision is \( \pm 9.9 \) (2 digits machine), and \( \alpha_f \) is set to 0.9. The simulation result is shown in Figure 7. From Figure 7, the performance of the BAM is greatly reduced, and it can store only about 10 recent library pairs. The main cause of such reduction comes from the incorrect \( \alpha_f \) value. From theorem 1, when \( \alpha_f \) is set to 0.9, the BAM can store about 10 recent library pairs.
Figure 6: The percentage of the most recent $k$ previous library pair being successfully stored. Here, equation 1.4 is used. The precision is $\pm 19.99$.

Figure 7: The percentage of the most recent $k$ previous library pair being successfully stored. Here, equation 1.4 is used. The precision is $\pm 9.9$.

To sum up, when sufficient precision is used (for $n = p = 512$, it is about 3 digits), corollary 1 still holds. When the precision limits the value of $\alpha_f$, we can use theorem 1 to predict the performance.
4 Magnitude of the Weights

4.1 The Worst Case Bound. Let \( w_{ij} \) be the weight between the \( j \)th neuron in \( F_Y \) and the \( i \)th neuron in \( F_X \). After \( t \) library pairs have been encoded using equation 1.4,

\[
 w_{ij} = \sum_{h=1}^{t} \alpha_f^{t-h} x_{ih} y_{jh}. \tag{4.1}
\]

In the worst case, \( x_{ih} = y_{jh} \) for all \( h \) (or \( x_{ih} = -y_{jh} \) for all \( h \)),

\[
 |w_{ij}| \leq \sum_{h=1}^{t} \alpha_f^{t-h} \leq \frac{1}{1 - \alpha_f}. \tag{4.2}
\]

Clearly, when \( \alpha_f \) is a constant, the magnitude of \( w_{ij} \) is upper bounded by \( O(1) \). If \( \alpha_f \) is selected based on equation 2.5, then

\[
 |w_{ij}| \leq \frac{1}{1 - \sqrt{1 - \exp\left\{-\log\frac{\min(n,p)}{2\log\min(n,p)} + 1\right\}}} = \frac{1}{\frac{\sqrt{2}\log\min(n,p)}{\min(n,p)} + \frac{1}{8}\left(\frac{2\log\min(n,p)}{\min(n,p)}\right)^2 + \text{high order terms}}. \tag{4.3}
\]

Hence, for large \( n \) and \( p \) (note that \( p = rn \)),

\[
 |w_{ij}| \leq O\left(\frac{n}{\log n}\right). \tag{4.5}
\]

4.2 The Probabilistic Bound. If \( x_{ih} \) and \( y_{jh} \) are \( \pm 1 \) equiprobable independent random variables, \( w_{ij} \) (based on equation 1.4) can be expressed as

\[
 w_{ij} = \sum_{h=1}^{t} \alpha_f^{t-h} z_{ih}, \tag{4.6}
\]

where \( z_h \)'s are \( \pm 1 \) equiprobable independent random variables. Clearly, the distribution of \( w_{ij} \) is symmetric. Let \( \eta \) be a standard normal random variable with variance one. It is not difficult to show that

\[
 E(\exp\{\tau z_h\}) < E(\exp\{\tau \eta\}).
\]

With lemma 1,

\[
 \text{Prob}\left\{|w_{ij}| > \sqrt{\min(n,p)} A \right\} \leq 2 \inf_{\tau \geq 0} \exp\left\{-\tau A \sqrt{\min(n,p)}\right\}.
\]
\[
\times \exp \left\{ \frac{\sum_{h=1}^{t} a_j^{2h} - 2h^2}{2} \right\}
\leq 2 \inf_{\tau \geq 0} \exp \left\{ -\tau A \sqrt{\min(n, p)} \right\}
\times \exp \left\{ \frac{\tau^2}{2(1 - a_j^2)} \right\},
\]

where \( A \) is a positive constant. The right-hand side becomes minimum when
\[
\tau = A \sqrt{\min(n, p)} (1 - a_j^2).
\]

With such a value of \( \tau \),
\[
\text{Prob}\left\{ |w_{i,j}| > A \sqrt{\min(n, p)} \right\} \leq 2 \exp \left\{ -\frac{(1 - a_j^2)A^2 \min(n, p)}{2} \right\}. \tag{4.7}
\]

When \( a_j \) is a constant, the probability \( \text{Prob}\{|w_{i,j}| > A \sqrt{\min(n, p)}\} \) exponentially tends to zero as \( n \to \infty \). If \( a_j \) is selected based on equation 2.5,
\[
\text{Prob}\left\{ |w_{j,j}| > A \sqrt{\min(n, p)} \right\} \leq 2 \exp \left\{ -eA^2 \log \min(n, p) \right\}. \tag{4.8}
\]

It also tends to zero as \( n \to \infty \).

To sum up, the magnitude of the weight is upper bounded by
\[
O(\sqrt{\min(n, p)}) = O(\sqrt{n})
\]
in the probabilistic sense when equation 1.4 is used.

5 Conclusion

We have examined the statistical storage behavior of the BAM under the forgetting learning. Also, we have derived the formula for choosing the forgetting constant:
\[
\alpha_{f, \text{max}} \approx \sqrt{1 - \exp \left\{ -\log \frac{\min(n, p)}{2 \log \min(n, p)} + 1 \right\}}.
\]

With this value, the forgetting learning rule can encode any number of library pairs and always keep the most recent
\[
\frac{\min(n, p)}{2e \log \min(n, p)}
\]
library pairs in the BAM. Computer simulations have been carried out to verify our theoretical work. The magnitude of the weights was also discussed.
The results presented here can be extended to a Hopfield network. By adopting the approach we set out, we can easily obtain the result in a Hopfield network by replacing \( \min(n, p) \) with \( n \) in all of the equations in this article.

**Acknowledgments**

We are very thankful to the reviewers for their valuable suggestions for improving this article.

**References**


Received May 30, 1995; accepted April 25, 1996.