Proceedings of V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 349-354, April 2-5, 2001 - Rio de Janeiro - RJ - Brazil

Memória Associativa Holográfica

Policarpo B. Uliana, Rui Seara, José Carlos M. Bermudez
Universidade Federal de Santa Catarina
Departamento de Engenharia Elétrica
E-mail: poli@linse.ufsc.br

Abstract

This paper proposes a holographic associative memory model, which uses a data structure based on random patterns to store and recall the desired information. That approach for data storage discards any training process as well as does not need complex procedures to recall the output data. As the information is distributed in the whole memory, some part of it may be corrupted by noise or even deleted without any loss of essential information. Some application examples are presented and discussed. An assessment of the robustness and storage capacity of this memory class is also presented.

1. Introdução

Memórias associativas [1] são estruturas que permitem o armazenamento e a restauração (recuperação) de informações com base na associação de padrões.
Diferentemente das memórias convencionais, que acessam seus dados através de endereços físicos predefinidos, as memórias associativas armazenam conjunto de informações que podem ser representados (modelados) como pares de vetores ($\boldsymbol{x}_k, \boldsymbol{y}_k$), definidos por:

$$\boldsymbol{x}_k \in \Re^n \leftrightarrow \boldsymbol{x}_k = [x_{k1},..., x_{kn}]^{\mathrm{T}} \qquad ,x_{ki} \in \Re, \tag{1a}$$

$$\boldsymbol{y}_k \in \Re^m \leftrightarrow \boldsymbol{y}_k = [y_{k1},..., y_{km}]^{\mathrm{T}} \qquad ,y_{ki} \in \Re, \tag{1b}$$

onde $\Re^n$ e $\Re^m$ representam espaços vetoriais reais de dimensões n e m, respectivamente, e $k = 1,\dots, p$ caracteriza o número total de pares associados.
A memória associativa realiza uma transformação no espaço vetorial T: $\Re^n \rightarrow \Re^m$, mapeando o vetor $\boldsymbol{x}_k$ no vetor $\boldsymbol{y}_k$.
Se os vetores $\boldsymbol{x}_k$ e $\boldsymbol{y}_k$ forem iguais, a memória é denominada autoassociativa; se forem distintos (caso geral), a memória é denominada heteroassociativa [2].
As memórias autoassociativas são usadas, geralmente, em aplicações de identificação de padrões [3]. Por outro lado, as memórias heteroassociativas são mais abrangentes em termos de aplicações. Como nessas memórias o acesso pode ser realizado de forma recursiva, uma informação pode estar relacionada à outra, compondo uma cadeia de dados, de forma análoga ao que se presume ocorrer com a memória humana [2].
As memórias associativas vêm sendo amplamente empregadas em sistemas conexionistas (baseados em redes neurais artificiais (RNA)), visando, além do armazenamento, o processamento e a representação de dados de forma distribuída. Uma aplicação típica nesta área é em sistemas de processamento de linguagem natural [4].
As memórias associativas propostas na literatura têm ainda aplicação limitada, apresentando alguns inconvenientes de ordem prática, tais como:

- necessitam de um processo de treinamento que, além de demandar um grande esforço computacional, dificultam a aplicação de tais memórias em sistemas que operam em tempo real;
- para a maior parte das memórias, o dado recuperado (extraído) apresenta um alto nível de degradação, exigindo um procedimento complexo para a sua restauração.

Neste artigo, propõe-se uma nova abordagem de implementação de memórias associativas que empregam estruturas de dados baseadas em padrões aleatórios para armazenar as informações desejadas. Essa forma de armazenamento dispensa qualquer processo de treinamento e não requer procedimentos complexos de restauração de dados de saída. Como a informação está distribuída em toda a memória, parte de seu conteúdo pode ser corrompida por ruído ou mesmo apagada sem que informações relevantes sejam perdidas. Essa estrutura de organização e operação de memória é denominada memória associativa holográfica (MAH).

2. Memórias Associativas

As memórias associativas podem ser implementadas basicamente de duas formas:

- ortogonalização e mapeamento: os vetores de entrada são ortogonalizados (ou classificados em grupos de padrões). Os vetores de saída são obtidos pela aplicação de uma função de mapeamento sobre os vetores de entrada ortogonalizados;
- transformações inversíveis: os pares de vetores de entrada e saída são combinados através de algum tipo de operação inversível, sendo todas as combinações acumuladas e representadas por um único vetor. Para a obtenção dos vetores primitivos armazenados, uma transformação inversa é aplicada.

Exemplos de memórias associativas encontradas na literatura:

2.1. Memória associativa linear (LAM)

Esta memória é composta de uma matriz $\mathbf{M}$ que mapeia as entradas diretamente nas saídas:

$$\boldsymbol{y} = \mathbf{M}\boldsymbol{x}\,. \tag{2}$$

A matriz $\mathbf{M}$ pode ser obtida diretamente da matriz de correlação entre os pares de entrada e saída [5]. Se os vetores de entrada forem ortogonais, os padrões de saída permitem a perfeita restauração dos padrões de entrada. Se os vetores não são ortogonais (o que normalmente ocorre), teremos erros na saída causados pela interferência entre os padrões.

2.2. Memória associativa de segunda ordem (SAM)

Neste tipo de memória, uma transformação não-linear é introduzida com o objetivo de reduzir o problema da não-ortogonalidade dos padrões de entrada. Dessa forma:

$$\boldsymbol{y} = \mathbf{M}T(\boldsymbol{x})\,, \tag{3}$$

onde $T(\boldsymbol{x})$ é uma transformação não-linear (por exemplo, uma transformação polinomial) e a matriz $\mathbf{M}$ é agora obtida via processo de otimização linear [6].

2.3. Memória recursiva autoassociativa (RAAM)

Em 1990, Pollack [7] apresentou uma estrutura de memória denominada *recursive auto-associative memory* (RAAM). Essa memória era obtida através de uma RNA de duas camadas, treinada via algoritmo *backpropagation*.

Na estrutura RAAM, dois vetores de dimensão *n* são apresentados a uma rede com 2*n* entradas, *n* neurônios na primeira camada e 2*n* neurônios na segunda camada. A rede é treinada para apresentar na saída os mesmos dois vetores de entrada. Após o treinamento, para um dado conjunto de vetores, observa-se que a primeira camada da rede realiza uma compactação dos dois vetores de entrada em um único vetor, enquanto a segunda camada realiza a descompactação. Dessa forma, a primeira camada de uma RAAM pode ser utilizada de forma recursiva a fim de armazenar uma seqüência de vetores em um único vetor. Essa seqüência pode ser restaurada através da segunda camada da RNA de forma também recursiva.

2.4. Memória baseada em convolução circular (HRR)

Em 1994, Plate [8] propôs uma estrutura denominada *holographic reduced representation* (HRR). Essa abordagem era baseada em operações de convolução circular e correlação circular para o armazenamento e a restauração dos dados, respectivamente. A convolução circular (representada por $\otimes$) é basicamente uma operação de convolução entre dois vetores, conservando a dimensão do vetor gerado [9]. A correlação circular (representada por $\oplus$) é a operação inversa da convolução circular, podendo-se demonstrar que, para certos casos (vetores de grande dimensão não correlacionados), a Eq. (4) pode ser satisfeita:

$$(\boldsymbol{y}_k \otimes \boldsymbol{x}_k) \oplus \boldsymbol{x}_i \cong \begin{cases} \boldsymbol{y}_i & ,k = i \\ 0 & ,k \neq i \end{cases} \tag{4}$$

Os pares de vetores são então armazenados através do procedimento descrito pela Eq. (5):

$$\boldsymbol{s} = \boldsymbol{x}_1 \otimes \boldsymbol{y}_1 + \boldsymbol{x}_2 \otimes \boldsymbol{y}_2 + \cdots + \boldsymbol{x}_p \otimes \boldsymbol{y}_p\,. \tag{5}$$

Assim, para recuperar um dos vetores, basta aplicar (4) em (5). Portanto,

$$\boldsymbol{y}_i = \boldsymbol{s} \oplus \boldsymbol{x}_i\,, \tag{6a}$$

$$\boldsymbol{x}_i = \boldsymbol{s} \oplus \boldsymbol{y}_i\,. \tag{6b}$$

3. Memória Associativa Holográfica (MAH)

A estrutura proposta no presente artigo está baseada na utilização de uma transformação inversível, atuando de forma similar à memória HRR [8], sendo aplicável para vetores de entrada e saída representados em uma base binária.

Considerando $\mathbf{E}(k)$ um conjunto de matrizes retangulares, cujas linhas são formadas por vetores independentes normalizados (ortonormais), obtidos de uma base de números aleatórios, o produto de quaisquer duas dessas matrizes, denominadas matrizes aleatórias, (sendo uma delas transposta) é dado por:

$$\mathbf{E}(k)\mathbf{E}^{\mathrm{T}}(i) \cong \begin{cases} \mathbf{I} & ,k = i \\ 0 & ,k \neq i \end{cases} \tag{7}$$

Consideremos agora que cada vetor de entrada $(\boldsymbol{x}_k)$ possa estar associado a uma matriz $(\mathbf{E}_x(k))$, pertencente a um conjunto de matrizes aleatórias que satisfaz a expressão (7). Assim,

$$\boldsymbol{x}_k \Rightarrow \mathbf{E}_x(k)\,. \tag{8}$$

Consideremos também que cada vetor de saída $(\boldsymbol{y}_k)$ possa ser associado a uma matriz $(\mathbf{Y}_k)$ que carrega as mesmas informações desse vetor em sua diagonal principal. Assim,

$$\boldsymbol{y}_k \Leftrightarrow \mathbf{Y}_k\,. \tag{9}$$

A expressão que descreve a operação de armazenamento dos vetores de saída na memória associativa holográfica é dada por:

$$\mathbf{S} = \mathbf{Y}_1\mathbf{E}_x(1) + \mathbf{Y}_2\mathbf{E}_x(2) + \cdots + \mathbf{Y}_p\mathbf{E}_x(p), \qquad (10)$$

onde $\mathbf{S}$ é a matriz que conterá uma representação holográfica de todos os vetores de saída.

Para recuperar o vetor $\boldsymbol{y}_i$ dado o vetor $\boldsymbol{x}_i$, utiliza-se a matriz aleatória associada $\mathbf{E}_x(i)$ e, através da aplicação da expressão (7) em (10), obtém-se:

$$\mathbf{Y}_i = \mathbf{S}\mathbf{E}_x^{\mathrm{T}}(i)\ . \qquad (11)$$

O vetor de saída desejado, $\boldsymbol{y}_i$, é derivado diretamente dos elementos da diagonal principal da matriz $\mathbf{Y}_i$, como demonstrado no Apêndice I.

Assim, um conjunto de dados (caracterizados pelos pares de vetores de entrada e saída) é representado por uma única matriz $(\mathbf{S})$, com o auxílio das relações (10) e (11), compondo o que denominamos memória associativa holográfica (MAH).

Como demonstrado no Apêndice II, a capacidade de memorização da MAH é diretamente proporcional à dimensão da matriz $\mathbf{S}$ e inversamente proporcional à dimensão do vetor de saída $\boldsymbol{y}_k$. Por exemplo, uma matriz de dimensão 100×100 poderia armazenar aproximadamente 80 pares de vetores de dimensão 10, com uma probabilidade de extração correta superior a 99,9%.

4. Aplicações da MAH

A MAH pode ser utilizada em uma ampla gama de aplicações [9]. A seguir descreveremos de forma sucinta algumas dessas aplicações.

4.1. Armazenamento de seqüências:

A MAH pode armazenar seqüências de sinais, como por exemplo, a seqüência *a*,*b*,*c*,*d*, que pode ser representada pela seguinte expressão:

$$\mathbf{S} = \mathbf{AE}(1) + \mathbf{BE}(2) + \mathbf{CE}(3) + \mathbf{DE}(4)\ . \qquad (12)$$

4.2. Armazenamento de conjuntos de seqüências:

Conjuntos de seqüências, como por exemplo [*a*,*b*,*c*,*d*], [*f*,*g*,*a*] e [*b*,*c*,*h*], podem ser armazenados por:

$$\begin{aligned}\mathbf{S} = {} & (\mathbf{AE}_1(1) + \mathbf{BE}_1(2) + \mathbf{CE}_1(3) + \mathbf{DE}_1(4))\mathbf{E}_2(1) + \\ & (\mathbf{FE}_1(1) + \mathbf{GE}_1(2) + \mathbf{AE}_1(3))\mathbf{E}_2(2) + \\ & (\mathbf{BE}_1(1) + \mathbf{CE}_1(2) + \mathbf{HE}_1(3))\mathbf{E}_2(3).\end{aligned} \qquad (13)$$

4.3. Representação de variáveis:

Em muito casos, desejamos armazenar relações do tipo:

$$x = a,\ \ y = b,\ \ z = c\ ,$$

onde x, y, z são variáveis internas ao sistema, e a, b, c representam valores para um dado instante.

Para esta representação, basta associar uma matriz aleatória para cada variável. Assim,

$$\mathbf{S} = \mathbf{AE}(x) + \mathbf{BE}(y) + \mathbf{CE}(z)\ . \qquad (14)$$

4.4. Representação de predicados:

Para representar predicados do tipo: "*João come peixe*", formamos uma tripla:

Tipo_de_ação (sujeito, predicado).

Se, para cada tipo de ação, associarmos uma matriz aleatória, teremos:

$$\mathbf{S} = \mathbf{M}_{Joao}\mathbf{E}(sujeito)\mathbf{E}(comer) + \mathbf{M}_{Peixe}\mathbf{E}(predicado)\mathbf{E}(Joao)\ , \qquad (15)$$

onde $\mathbf{M}_{Joao}$ será uma matriz associada ao vetor binário que representa o símbolo "*João*", e $\mathbf{E}(sujeito)$ será uma matriz aleatória associada ao símbolo "*sujeito*", sendo que a mesma composição será repetida para os demais símbolos.

4.5. Representação de quadros:

Informações definidas por quadros (*frames*), [10], podem ser representadas como sentenças lógicas da seguinte forma:

quadro 1: (atributo 1, atributo 2, atributo 3...).

Exemplo:

quadro *frutas*: (nome, cor, gosto)=
{(maçã, vermelha, doce), (limão, verde, ácido)}.

Para armazenar este quadro na MAH, temos:

$$\begin{aligned}\mathbf{S} = {} & \mathbf{M}_{maca}\mathbf{E}(gosto)\mathbf{E}(doce) + \mathbf{M}_{maca}\mathbf{E}(cor)\mathbf{E}(vermelha) + \\ & \mathbf{M}_{doce}\mathbf{E}(gosto)\mathbf{E}(maca) + \mathbf{M}_{vermelha}\mathbf{E}(cor)\mathbf{E}(maca) + \\ & \mathbf{M}_{limao}\mathbf{E}(gosto)\mathbf{E}(acido) + \mathbf{M}_{limao}\mathbf{E}(cor)\mathbf{E}(verde) + \\ & \mathbf{M}_{acido}\mathbf{E}(gosto)\mathbf{E}(limao) + \mathbf{M}_{verde}\mathbf{E}(cor)\mathbf{E}(limao).\end{aligned} \qquad (16)$$

Para recuperar uma dada informação, basta utilizar as matrizes aleatórias associadas às informações relativas aos dados armazenados. Dessa forma, a questão "Cor da maçã?" equivale à:

$$\mathbf{M}_{vermelha} = \mathbf{S}\mathbf{E}^{\mathrm{T}}(maca)\mathbf{E}^{\mathrm{T}}(cor)\ . \qquad (17)$$

5. Resultados Experimentais

Como uma das aplicações mais complexas da MAH consiste no armazenamento de quadros, essa aplicação foi utilizada em alguns casos reais que empregavam esse tipo de representação. A MAH foi utilizada para a representação de quadros que continham símbolos associados a vetores binários de dimensão 10. Foram utilizadas matrizes aleatórias com dimensão 100×100, cujos elementos foram obtidos de um processo aleatório com distribuição normal (média zero e desvio padrão 0,1). A cada símbolo associou-se uma dessas matrizes aleatórias. A memória foi utilizada então para o armazenamento de um conjunto de quadros, como por exemplo:

a cor do pino é azul,
a cor da bola é preta,
o tamanho do pino é pequeno.

A recuperação das informações é realizada com questões (Q) e respostas (R) do tipo:

Q: cor pino? *R: azul*
Q: cor pino amarelo? *R: não*
Q: cor pino azul? *R: sim*

A memória implementada foi utilizada para armazenar até 60 conjuntos de dados sem que nenhuma perda de informação fosse verificada.
Em casos de informações conflitantes (por exemplo, para as sentenças do tipo: "*a cor do pino é azul*", "*a cor do pino é vermelha*", perguntamos qual é a cor do pino?), a resposta tende a ser um valor não-válido e, com menor freqüência, tende a uma das respostas corretas.
Se for aplicado um decaimento exponencial (fator de esquecimento) às informações existentes a cada vez que uma nova informação é armazenada, as informações menos recentes irão sendo "esquecidas", o que torna a aplicação da MAH interessante para tratar informações em ambientes dinâmicos.

6. Avaliação da Robustez da MAH

Considerando que a MAH representa os dados de forma holográfica, a informação é preservada mesmo quando existe uma perda de parte significativa da memória ou quando a mesma é corrompida por ruído. Armazenando um único quadro representado por $\mathbf{S} = \mathbf{AE}$, onde a informação de entrada é dada por um vetor binário de dimensão 10, realizam-se os seguintes testes para avaliar a robustez da MAH:

i) adição de ruído aleatório a todas as posições da memória;

ii) destruição (apagamento) da informação armazenada em posições da memória selecionadas de forma aleatória.

Tabela 1 – Indicadores de robustez da MAH obtidos experimentalmente

Dimensões da MAH	Limite da SNR sem perda de informação	Limite de destruição sem perda de informação
50×50	-10,5dB	85%
100×100	-12,5dB	90%
150×150	-14,5dB	94%
200×200	-16,0dB	97%

A Tabela 1 ilustra o desempenho da MAH, para diversas dimensões de memória, sendo apresentados os limites de razão sinal-ruído (SNR) e os seus correspondentes limites de destruição do conteúdo da memória, sem que haja perda de informações relevantes. Observa-se, nesses resultados, que os dados podem ser fortemente corrompidos sem o comprometimento da informação armazenada na memória.

7. Plausibilidade Biológica

Observamos que a estrutura proposta pode ser implementada utilizando RNAs nas quais os pesos dos neurônios definem as "matrizes" aleatórias, e os estados de saída de um grande número de neurônios definem a "matriz" que representa a memória holográfica.
Além das diversas possíveis aplicações práticas para a MAH, pode-se extrapolar a concepção do modelo holográfico proposto. Consideremos então algumas características da MAH:

i) restauração associativa de dados;

ii) representação distribuída (holográfica) das informações;

iii) robustez à perda de elementos de armazenamento;

iv) possibilidade de esquecimento;

v) capacidade de tratamento de informações ambíguas;

vi) capacidade de processamento simbólico;

vii) possibilidade de implementação através de estruturas neurais.

Além disso, quando observamos os dados contidos nos elementos de memória (que são padrões aleatórios), não identificamos diretamente a informação ali armazenada. Essa só poderá ser recuperada com o auxílio da matriz aleatória utilizada na fase de armazenamento.
A comparação de todas estas características com aquelas conjecturadas para a memória humana nos leva a pensar que esse tipo de processamento holográfico possa lançar alguma luz sobre as obscuras questões de operação de nossa memória e de nosso cérebro como um todo.

8. Conclusões

A estrutura de memória associativa holográfica aqui discutida representa uma evolução em relação às estruturas existentes. Ela não necessita de nenhum processo de "limpeza" dos dados de saída, sendo assim uma evolução da HRR.
Os conceitos básicos da MAH são amplamente conhecidos, mas, para aplicá-los, tivemos de assumir que os símbolos manipulados pela memória são representados por duas formas distintas: um vetor binário e uma matriz aleatória associada.
Para uma aplicação prática, é necessário utilizar matrizes de grandes dimensões. Dessa forma, a restauração dos dados demandará um considerável esforço computacional. Por outro lado, esse processamento pode ser realizado em um único passo, por uma máquina com processamento paralelo de grande capacidade ou mesmo por uma rede neural dedicada.

Apêndice I. Princípio de Operação

Nesta seção, descreveremos o princípio de operação da memória associativa holográfica para o armazenamento e a recuperação de um par de vetores $\boldsymbol{a}$ e $\boldsymbol{b}$, assim definidos:

$$\boldsymbol{a} = [a_1, a_2, ..., a_n]$$
$$\boldsymbol{b} = [b_1, b_2, ..., b_n]$$

onde os elementos a_i e b_i assumem apenas dois valores $[+1, -1]$ (representação binária na base de Rademacher [11]).
A partir destes vetores, definimos duas matrizes $\mathbf{A}$, $\mathbf{B}$, onde:

$$\mathbf{A} = \left[\mathbf{a}_{ij}\right] = \begin{cases} a_i & , i = j \\ 0 & , i \neq j \end{cases} \qquad , i, j = 1, 2, \ldots, n \tag{A1}$$

$$\mathbf{B} = \left[\mathbf{b}_{ij}\right] = \begin{cases} b_i & , i = j \\ 0 & , i \neq j \end{cases} \qquad , i, j = 1, 2, \ldots, n \tag{A2}$$

Além disso, associamos aos vetores $\boldsymbol{a}$ e $\boldsymbol{b}$ duas matrizes $\mathbf{E}_a$ e $\mathbf{E}_b$, respectivamente. Essas matrizes são constituídas de vetores aleatórios (normalizados) com uma distribuição Gaussiana de média zero.
De posse destas matrizes, definimos então a matriz $\mathbf{S}$, dada por:

$$\mathbf{S} = \mathbf{A}\mathbf{E}_b + \mathbf{B}\mathbf{E}_a \,. \tag{A3}$$

Assim, dada a matriz aleatória associada a um dos vetores, extrai-se o outro vetor através de uma das seguintes expressões:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^{\mathrm{T}} \tag{A4}$$

$$\mathbf{B} \cong \mathbf{S}\mathbf{E}_a^{\mathrm{T}} \,. \tag{A5}$$

A fim de demonstrar a validade das expressões acima, substituímos a equação (A3) em (A4), obtendo:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^{\mathrm{T}} = \mathbf{A}\mathbf{E}_b\mathbf{E}_b^{\mathrm{T}} + \mathbf{B}\mathbf{E}_a\mathbf{E}_b^{\mathrm{T}} \,. \tag{A6}$$

O produto de $\mathbf{E}_b$ por sua transposta leva à:

$$\mathbf{E}_b\mathbf{E}_b^{\mathrm{T}} \cong \mathbf{I} \,. \tag{A7}$$

Da mesma forma, o produto de $\mathbf{E}_a$ por $\mathbf{E}_b^{\mathrm{T}}$ leva à:

$$\mathbf{E}_a\mathbf{E}_b^{\mathrm{T}} \cong 0 \,. \tag{A8}$$

Agora, aplicando (A7) e (A8) em (A6), obtém-se:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^{\mathrm{T}} = \mathbf{A}\mathbf{I} + \delta \,, \tag{A.9}$$

onde δ é uma matriz de erro aleatório sobreposto ao sinal recuperado.
A demonstração de (A5) pode ser obtida de forma análoga.
Para recuperar o vetor original de entrada $\boldsymbol{a}$ (ou $\boldsymbol{b}$) a partir da matriz $\mathbf{A}$ (ou $\mathbf{B}$), basta aplicar o seguinte procedimento:

$$a_i = \begin{cases} 1 & , a_{ii} \geq \mu \\ 0 & , -\mu \leq a_{ii} < \mu \\ -1 & , a_{ii} < -\mu \end{cases} \quad , i = 1, 2, \ldots, n \tag{A10}$$

onde μ é um valor ($0 \leq \mu < 1$) que define um limiar de erro de recuperação de dados. A condição de erro é sinalizada por um elemento zero no vetor de saída.
A generalização deste procedimento, para um número qualquer de vetores, torna-se uma simples extensão do que foi demonstrado.

Apêndice II. Capacidade de Armazenamento

Nesta seção, analisaremos a capacidade de armazenamento de informação da memória associativa holográfica.
É conhecido que, para um vetor aleatório (normalizado) de dimensão n com distribuição Gaussiana de média zero, o seu desvio padrão (σ) é dado por:

$$\sigma = \frac{1}{\sqrt{n}} \,. \tag{B1}$$

Consideremos então uma soma de p matrizes aleatórias, dada por:

$$\mathbf{S} = \mathbf{A}_1\mathbf{E}(1) + \mathbf{A}_2\mathbf{E}(2) + \cdots + \mathbf{A}_p\mathbf{E}(p) \,. \tag{B2}$$

Ao recuperarmos qualquer valor armazenado, temos:

$$\mathbf{A}_i \cong \mathbf{S}\mathbf{E}^{\mathrm{T}}(i) = \mathbf{A}_i + \delta \,. \tag{B3}$$

A partir das equações (B1) (B2) e (B3), podemos demonstrar que a matriz de erro de saída (δ) tem média zero e desvio padrão dado por:

$$\sigma_\delta = \sqrt{\frac{p}{n}} \,. \tag{B4}$$

Consideremos agora a probabilidade (α) do módulo de qualquer um dos elementos da matriz de erro (δ_{ij}) ser maior ou igual a um dado limite ℓ. Assim,

$$\alpha = \mathrm{P}\{|\delta_{ij}| \geq \ell\}. \quad \text{(B5)}$$

Portanto, considerando uma distribuição de probabilidade normal com média zero e desvio padrão σ_δ, tem-se:

$$\alpha = \frac{2}{\sqrt{2\pi\sigma_\delta^2}} \int_\ell^\infty e^{-\frac{\delta^2}{2\sigma_\delta^2}} d\delta. \quad \text{(B6)}$$

Para valores de $(\sigma_\delta^2/\ell^2) << 1$, através de (B6), α pode ser aproximado por:

$$\alpha \approx \frac{2\sigma_\delta}{\ell\sqrt{2\pi}} e^{-\frac{\ell^2}{2\sigma_\delta^2}}. \quad \text{(B7)}$$

Aplicando a equação (B4) em (B7), temos:

$$\alpha \approx \frac{2\sqrt{p}}{\ell\sqrt{2\pi n}} e^{-\frac{\ell^2 n}{2p}}. \quad \text{(B8)}$$

Considerando que, para a restauração dos dados da MAH, utiliza-se $\mu = 0$, o máximo nível de ruído admissível (pior caso) é dado para $\ell \to 1$. Da expressão (B8), podemos então determinar, de forma aproximada, o valor de p. Assim, considerando-se admissível uma probabilidade de erro de recuperação α, o número máximo de vetores que pode ser armazenado em uma MAH de dimensão $(n \times n)$ é dado por:

$$p \cong \frac{n}{\ln\left[\frac{2}{\alpha^2\pi}\right] - 1}. \quad \text{(B9)}$$

Quando a dimensão do vetor de saída (*m*) for menor do que a dimensão de uma coluna da matriz de memória (*n*), teremos a informação do vetor repetida n/m vezes nos elementos da diagonal da matriz. Assim,

$$[\mathbf{a}_{ii}] = a_q \quad , i = kq \quad \text{para:} \begin{cases} k = 1, \ldots, n/m \\ q = 1, \ldots, m \end{cases} \quad \text{(B10)}$$

Portanto, para a recuperação do vetor original, n/m posições da matriz de saída deverão ser avaliadas (através da obtenção de um valor médio dessas posições) para a definição de cada elemento do vetor de saída. Dessa forma, o desvio padrão (σ_δ) a ser usado em (B7) é agora dado por:

$$\sigma_\delta = \sqrt{\frac{pm}{n^2}}. \quad \text{(B11)}$$

Sendo então, neste caso, o valor de p aproximado pela expressão (B12). Dessa forma, considerando-se admissível uma probabilidade de erro de recuperação α, o número máximo de vetores de dimensão (*m*) que pode ser armazenado em uma MAH de dimensão ($n \times n$) é dado por:

$$p \cong \frac{n^2/m}{\ln\left[\frac{2}{\alpha^2\pi}\right] - 1}. \quad \text{(B12)}$$

Referências

[1] J. B. Pollack. "Recursive Distributed Representations," Artificial Intelligence, no. 46, pp. 77-105, 1990.

[2] Haykin, S. *Neural Networks. A Comprehensive Foundation.* Macmillan College Publishing Company, Inc. 1994.

[3] B. Hunt, M. S. Nadar, and A. Goyal. "Synthesis of a Nonrecurrent Associative Memory Model Based on a Nonlinear Transformation in the Spectral Domain," IEEE Trans. on Neural Networks, vol. 4, no. 5, pp. 873-878. September 1993.

[4] M. J. Adamson and R. I. Damper. "A Recurrent Network which Learns to Pronounce English Text," Proc. of International Conference on Spoken Language Processing, vol. 3, pp. 1704-1707. Philadelphia, Pennsylvania, 1996.

[5] C. S. T. Kohonen. "Correlation Matrix Memories," IEEE. Trans. on Computer, vol. C-21, pp. 353, 1972.

[6] T. Poggio. "On Optimal Nonlinear Associative Recall," Biol. Cyb., vol. 19, pp. 201-209, 1975.

[7] J. B. Pollack. "Recursive Auto-Associative Memory: Devising Compositional Distributed Representations," Proc. of the 10th Annual Conference of the Cognitive Science Society, pp. 33-39. Montreal, Canada, 1988.

[8] T. Plate. "Holographic Reduced Representations," IEEE Trans. on Neural Networks, vol. 6, pp. 623-641, 1995.

[9] T. Plate. *Distributed Representations and Nested Compositional Structure.* Ph.D. Thesis, Department of Computer Science, University of Toronto, 1994.

[10] M. A. Minsky. *Framework for Representing Knowledge*. Morgan Kaufmann Publishers, 1988.

[11] K. G. Beauchamp, *Application of Walsh and Related Function*, Academic Press, 1984.