

**ALGORITMO HEBBIANO GENERALIZADO  
PARA EXTRAÇÃO DOS COMPONENTES PRINCIPAIS  
DE UM CONJUNTO DE DADOS NO DOMÍNIO COMPLEXO**

**DISSERTAÇÃO SUBMETIDA À  
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
PARA OBTENÇÃO DO GRAU DE  
MESTRE EM ENGENHARIA**

**MARIA CRISTINA FELIPPETTO DE CASTRO**

**PORTO ALEGRE, JUNHO DE 1996**

## Sumário

<b>Lista de Figuras</b> .....	V
<b>Simbologia</b> .....	X
<b>Resumo</b> .....	XIII
<b>Abstract</b> .....	XIV
<b>Introdução</b> .....	1
<b>Capítulo 1</b> Definições Gerais.....	5
<b>Capítulo 2</b> Análise dos Componentes Principais - A Transformação Karhunen-Loève.....	24
<b>Capítulo 3</b> A Transformada Hotelling na Compressão de Imagens.....	34
<b>Capítulo 4</b> O Aprendizado Hebbiano Generalizado para Análise dos Componentes Principais.....	39
4.1 O Aprendizado Hebbiano.....	40
4.2 O Algoritmo Hebbiano Generalizado.....	52

<b>Capítulo 5</b>	O Algoritmo Hebbiano Generalizado Complexo para Extração dos Componentes Principais – <i>CGHAPCA</i> .....	56
5.1	A Representação dos Dados.....	58
5.2	A Estrutura da Rede Neural Artificial.....	63
5.3	A Regra Complexa Generalizada para Treinamento da RNA sob Aprendizado Hebbiano.....	66
5.4	O Critério de Parada.....	72
5.5	O Critério para Atualização da Razão de Aprendizado.....	73
5.6	A Aceleração de Convergência do Algoritmo por Janelamento de Treino.....	74
5.7	O Treinamento da RNA Complexa.....	76
5.8	A Reconstrução dos Dados.....	77
5.9	A Compressão dos Dados.....	79
5.10	Sumário do <i>CGHAPCA</i> Aplicado à Compressão de Imagens Digitais.....	82
<b>Capítulo 6</b>	Resultados Experimentais e Conclusões.....	85
<b>Capítulo 7</b>	Sugestões para Futuros Trabalhos.....	118
	<b>Referências Bibliográficas</b> .....	120

## Lista de Figuras

1	O modelo de um neurônio artificial visto como um combinador linear .....	22
2	O modelo de uma rede neural artificial progressiva composta de uma única camada de neurônios lineares .....	23
3	RNA de uma única unidade, a ser treinada pelo aprendizado Hebbiano .....	43
4	RNA progressiva com uma única camada de neurônios lineares, a ser treinada pelo aprendizado Hebbiano generalizado .....	53
5	Modelo de rede neural artificial utilizado para treinamento via algoritmo <i>CGHAPCA</i> .....	63
6	Representação gráfica da RNA complexa com $p=64$ nós computacionais na camada de entrada e $k$ neurônios na camada de saída, apresentando os auto-vetores da matriz covariância dos dados de entrada, após a convergência do algoritmo <i>CGHAPCA</i> .....	65
7	RNA de uma única unidade, a ser treinada pelo aprendizado Hebbiano .....	66
8	RNA complexa a ser treinada pelo algoritmo, com $p$ nós na camada de entrada e $m$ neurônios na camada de saída .....	71
9	RNA Complexa com $p$ nós na camada de entrada e $m$ neurônios na camada de saída, apresentando os auto-vetores $Q$ da matriz covariância dos dados de entrada, após a convergência do algoritmo <i>CGHAPCA</i> .....	77

10	Modelo de RNA Complexa utilizado para compressão. A rede apresenta $p=64$ nós de entrada (para submatrizes com $p = l^2 = 64$ elementos) e $k$ neurônios na camada de saída (para extrair $k$ componentes principais) .....	80
11	Brasil : Imagem original .....	88
12	Histograma da Imagem da Fig.1 .....	88
13	Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.1.....	89
14	Imagem da Fig.1 reconstruída após compressão no domínio frequência (32 auto-vetores) .....	91
15	Imagem da Fig.1 reconstruída após compressão no domínio espaço (32 auto-vetores) .....	92
16	Histograma da Imagem da Fig. 4 .....	93
17	Imagem da Fig.1 reconstruída após compressão no domínio frequência (27 auto-vetores) .....	94
18	Imagem da Fig.1 reconstruída após compressão no domínio frequência (13 auto-vetores) .....	94
19	Imagem da Fig.1 reconstruída após compressão no domínio espaço (16 auto-vetores) .....	95
20	Lenna : Imagem original .....	96

21	Histograma da Imagem da Fig.20 .....	96
22	Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig. 20 .....	97
23	Imagem da Fig. 20 reconstruída após compressão no domínio frequência (32 auto-vetores) .....	98
24	Imagem da Fig. 20 reconstruída após compressão no domínio frequência (27 auto-vetores) .....	99
25	Imagem da Fig. 20 reconstruída após compressão no domínio espaço (32 auto-vetores) .....	100
26	Imagem da Fig. 20 reconstruída após compressão no domínio frequência (13 auto-vetores) .....	100
27	Imagem da Fig. 20 reconstruída após compressão no domínio espaço (16 auto-vetores) .....	101
28	Moon : Imagem original .....	102
29	Histograma da Imagem da Fig. 28 .....	102
30	Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig. 28 .....	103
31	Imagem da Fig. 28 reconstruída após compressão no domínio frequência (32 auto-vetores) .....	103
32	Imagem da Fig. 28 reconstruída após compressão no domínio frequência (27	

	auto-vetores) .....	104
33	Imagem da Fig. 28 reconstruída após compressão no domínio espaço (32 auto-vetores) .....	104
34	Imagem da Fig. 28 reconstruída após compressão no domínio frequência (13 auto-vetores) .....	105
35	Imagem da Fig.28 reconstruída após compressão no domínio espaço (16 auto-vetores) .....	105
36	Polígonos : Imagem original .....	107
37	Histograma da Imagem da Fig.36 .....	108
38	Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig. 36 .....	108
39	Imagem da Fig.36 reconstruída após compressão no domínio frequência (32 auto-vetores) .....	109
40	Imagem da Fig.36 reconstruída após compressão no domínio frequência (27 auto-vetores) .....	109
41	Imagem da Fig.36 reconstruída após compressão no domínio espaço (32 auto-vetores) .....	110
42	Imagem da Fig.36 reconstruída após compressão no domínio frequência (13 auto-vetores) .....	110
43	Imagem da Fig.36 reconstruída após compressão no domínio espaço (16 auto-vetores) .....	111

44	Eye : Imagem original .....	111
45	Histograma da Imagem da Fig.44 .....	112
46	Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig. 44 .....	112
47	Imagem da Fig.44 reconstruída após compressão no domínio freqüência (32 auto-vetores) .....	113
48	Imagem da Fig.44 reconstruída após compressão no domínio freqüência (27 auto-vetores) .....	113
49	Imagem da Fig.44 reconstruída após compressão no domínio espaço (32 auto-vetores) .....	114
50	Imagem da Fig.44 reconstruída após compressão no domínio freqüência (13 auto-vetores) .....	114
51	Imagem da Fig.44 reconstruída após compressão no domínio espaço (16 auto-vetores) .....	115
52	Imagem da Fig.1 comprimida e descomprimida utilizando a RNA complexa da imagem da Fig.36, PSNR= 19.35 dB .....	116

## Simbologia

$(.)^T$  - denota transposto

$(.)^*$  - denota conjugado

$\hat{(\cdot)}$  - denota estimado

$|(\cdot)|$  - denota módulo

$\|(\cdot)\|$  - denota norma

$E\{.\}$  - denota valor esperado

$[a_{ij}]$  - matriz com  $i$  linhas e  $j$  colunas, definida no campo dos números Reais

$[A_{ij}]$  - matriz com  $i$  linhas e  $j$  colunas, definida no campo dos números Complexos

$\bar{b}(x, y)$  - submatriz média da imagem  $f(x, y)$

$\bar{b}(u, v)$  - submatriz média do espectro  $F(u, v)$

$b_k(x, y)$  -  $k$ -ésima submatriz da imagem  $f(x, y)$

$b_k(u, v)$  -  $k$ -ésima submatriz do espectro  $F(u, v)$

$\bar{B}_{xy}$  - energia da submatriz média  $\bar{b}(x, y)$

$\bar{B}_{uv}$  - energia da submatriz média  $\bar{b}(u, v)$

*CGHAPCA* - Algoritmo Hebbiano Generalizado para Extração dos Componentes Principais de um Conjunto de Dados no Domínio Complexo

$C_x$  - matriz covariância

*DFT* - Transformada Discreta de Fourier

$E$  - energia de um sinal

$f(x, y)$  - função no domínio espaço (imagem)

$F(u, v)$  - espectro da imagem (Transformada de Fourier de uma imagem  $f(x, y)$ )

$\hat{F}(u,v)$  - espectro estimado da imagem

*FFT* - Transformada Rápida de Fourier

*GHA* - Algoritmo Hebbiano Generalizado

*GHAPCA* - Algoritmo Hebbiano Generalizado para Extração dos Componentes Principais de um Conjunto de Dados no Domínio Real

*H* - matriz Hotelling de transformação

*IDFT* - Transformada Discreta de Fourier Inversa

*IFFT* - Transformada Rápida de Fourier Inversa

*k* - número de componentes principais considerados

$k_{jt}$  - número de auto-vetores da janela de convergência

*l* - dimensão das submatrizes

*m* - número de neurônios da camada de saída da RNA

$n_b$  - número de submatrizes em que é particionado o espectro

*MSE* - Erro Médio Quadrático

*N* - dimensão da matriz imagem  $f(x,y)$

*p* - número de nós computacionais da camada de entrada da RNA

*PCA* - Análise dos Componentes Principais

*PSNR* - Relação Sinal-Ruído de Pico

$q_{ji}$  - auto-vetor “cristalizado” no peso sináptico  $w_{ji}$

$Q_{ji}$  - auto-vetor “cristalizado” no peso sináptico  $W_{ji}$

RNA - Rede Neural Artificial

$t_c$  - tempo de processamento

$t_{cjt}$  - tempo de processamento com aceleração por janelamento de treino

$w_{ji}$  - peso sináptico  $i$  do neurônio  $j$  (no contexto de RNAs)

$W_{ji}$  - peso sináptico  $i$  do neurônio  $j$  (no contexto de RNAs Complexas)

$x_i$  - nó computacional  $i$  pertencente à camada de entrada da RNA

$X_i$  - nó computacional  $i$  pertencente à camada de entrada da RNA Complexa

$y_j$  - neurônio  $j$  (no contexto de RNAs)

$Y_j$  - neurônio  $j$  (no contexto de RNAs Complexas)

$\alpha$  - constante de relação entre razão de aprendizado e auto-valor

$\lambda$  - auto-valor

$\mu_x$  - média de  $x$

$\eta$  - razão de aprendizado

$\rho$  - razão de compressão

$\sigma^2$  - variância

## **Resumo**

Este trabalho apresenta um algoritmo que extrai as componentes principais de um conjunto de dados de natureza complexa através de redes neurais artificiais complexas treinadas sob uma nova regra de aprendizado Hebbiano no domínio complexo. É estudada a aplicabilidade do algoritmo à compressão de imagens digitais. Os resultados indicam que o método no domínio complexo é mais eficiente em preservar as regiões de alto contraste da imagem do que o método no domínio real. A complexidade computacional do algoritmo é reduzida através da técnica de janelamento de convergência e são estabelecidas as relações analíticas entre os auto-valores das imagens (energias) nos domínios real e complexo.

## **Abstract**

In this work, Complex Artificial Neural Networks (CANN) are used to extract the principal components of a complex data set. The CANNs are trained by means of a new Hebbian learning rule in the complex domain. The proposed algorithm is applied to digital image compression. Results indicate that the method is more efficient in preserving high contrast regions than the real domain method. The computational complexity of the algorithm is reduced by the convergence windowing technique and the analytical relationship between the images eigen-values (energies) in the real and complex domain are stated.

## Introdução

Este trabalho propõe uma abordagem no domínio complexo das técnicas de Análise dos Componentes Principais (*PCA*) de conjunto de dados representativo de uma imagem através de Redes Neurais Artificiais (RNAs) progressivas de uma única camada neuronal sob treinamento por Algoritmo Hebbiano Generalizado (*GHA*).

Sugere-se aqui um novo algoritmo de aprendizado, no domínio complexo, desenvolvido de acordo com as leis do aprendizado Hebbiano e investiga-se a aplicabilidade de uma Rede Neural Artificial Complexa treinada sob este algoritmo para extração dos componentes principais de um conjunto de dados de natureza complexa.

A análise dos componentes principais permite identificar as características próprias de um conjunto de dados, aquelas características que constituem a essência deste conjunto, definindo-o completamente. É objeto deste estudo investigar como se apresentam estas características no domínio complexo, através da análise dos componentes principais de um conjunto de dados complexo.

O algoritmo objeto deste estudo é aplicado à compressão de imagens digitais e tem o desempenho avaliado quanto à fidelidade de reconstrução da imagem comprimida, comparativamente ao algoritmo no domínio real para extração dos Componentes Principais via Aprendizado Hebbiano Generalizado.

Análise dos Componentes Principais no domínio espaço é uma técnica básica para compressão de dados e extração de características. Esta técnica tem sido largamente utilizada em análise de dados estatísticos, teoria da comunicação e processamento de imagens.

O princípio básico da compressão de dados via Análise dos Componentes Principais é o mapeamento do conjunto de dados de um espaço de dados de maior dimensão para um espaço de dados de dimensão reduzida. A transformação é feita de tal forma que o conjunto original de dados venha a ser representado por um número reduzido de características efetivas retendo, no entanto, a maior parte da informação intrínseca contida nos dados.

A revisão bibliográfica realizada encontrou excelentes resultados na combinação de técnicas de domínio frequência a Redes Neurais Artificiais para tratamento de imagens, em extração de características [1][2] e reconhecimento de padrões [3]. Em muitos casos, os algoritmos de domínio complexo apresentam resultados mais abrangentes que os de domínio real, pois o domínio real nada mais é do que um sub-domínio do domínio complexo. Os resultados encontrados nestas aplicações validaram a investigação da combinação das duas técnicas em compressão de dados, encorajando a escolha deste tema.

Em um contexto histórico, Análise dos Componentes Principais foi introduzida por Pearson (1901) em um estudo biológico e desenvolvida por Hotelling (1933) para ser usada em psicometria [4].

No campo da teoria da probabilidade, Análise dos Componentes Principais foi primeiramente enfocada por Karhunen (1947) e, posteriormente, generalizada por Loève (1963) [4][5][6].

Redes Neurais Artificiais foram trazidas para a análise dos componentes principais por Oja (1982) [4][5], que demonstrou a estreita correspondência entre o comportamento de uma RNA de um único neurônio, treinada sob aprendizado Hebbiano

e a clássica transformação Karhunen-Loève. A posterior generalização deste algoritmo é devida a Sanger (1989) [4][5].

Decorrentes dos trabalhos de Oja e Sanger, outros algoritmos têm sido propostos, considerando novas regras de aprendizado e configurações de Redes Neurais Artificiais. Trabalhos relevantes encontrados na literatura incluem os de Földiak (1989) [4][5], Rubner e Tavan (1989) [4], Kung and Diamantaras (1990) [4], Brockett (1991) [4], Oja (1992)[4][5], Xu e Yuille (1992) [4][7], Chen e Liu (1992) [4], Bannour e Azimi-Sadjadi (1992) [4] e Chen e Chang (1993) [4].

O método apresentado a seguir é chamado Algoritmo Hebbiano Generalizado para Extração dos Componentes Principais no Domínio Complexo - *CGHAPCA*.

Esta dissertação versa sobre a fundamentação teórica, o desenvolvimento e a implementação do algoritmo proposto e estabelece as contribuições do novo método à compressão de dados de natureza complexa, investigando aplicabilidade e desempenho.

Para compreensão do *CGHAPCA* é necessário que se definam conceitos relativos a Redes Neurais Artificiais, Processamento de Imagens Digitais, Processos Estocásticos e conceitos matemáticos puros relativos à caracterização e representação de Vetores e Matrizes. Estas conceituações compõem o Capítulo 1.

O Capítulo 2 apresenta a Transformação Karhunen-Løeve, conhecida como Análise dos Componentes Principais. A razão para apresentar esta transformação estatística é que a compreensão de tal técnica é fundamental para entender como a Rede Neural Artificial treinada sob aprendizado Hebbiano extrai os componentes principais de um conjunto de dados.

A Transformada Hotelling, que aplica a técnica de Análise dos Componentes Principais à compressão de imagens é enfocada no Capítulo 3.

No Capítulo 4 é apresentado o Algoritmo Hebbiano Generalizado, descrito a partir dos conceitos básicos postos por Hebb. Após a conceituação básica, apresenta-se a aplicação, devida a Oja, dos postulados de Hebb a Redes Neurais Artificiais para o treinamento de um modelo composto de um único neurônio. O capítulo é finalizado com a generalização do algoritmo de Oja, desenvolvida por Sanger.

O Capítulo 5 discorre sobre o algoritmo *CGHAPCA* proposto nesta dissertação, caracterizando a representação dos dados, a estrutura da rede neural artificial complexa utilizada, a regra complexa generalizada para treinamento da RNA complexa sob aprendizado Hebbiano, o critério de parada (convergência do algoritmo), os critérios para aceleração do algoritmo através da atualização da razão de aprendizado e da aceleração de convergência do algoritmo por janelamento de treino, o treinamento da RNA Complexa, a reconstrução dos dados, a compressão dos dados e um sumário do *CGHAPCA* aplicado à compressão de imagens.

As simulações realizadas com imagens de diferentes estatísticas e as conclusões derivadas das simulações, recomendando o método para as aplicações adequadas, compõem o Capítulo 6, onde são apresentadas as imagens originais e aquelas comprimidas pelo *CGHAPCA* e gráficos representativos dos resultados.

As idéias que surgiram durante o processo de desenvolvimento do trabalho, sobre aplicabilidade e otimização do *CGHAPCA*, estão registradas no Capítulo 7 como sugestões para futuros trabalhos.

## Capítulo 1

### Definições Gerais

Tratando-se de um estudo multi-disciplinar, é importante que se definam as conceituações básicas pertinentes a cada uma das áreas do conhecimento envolvidas. Para tal propósito, este Capítulo apresenta definições concernentes a Vetores e Matrizes [8] [9] [10], Processos Estocásticos [12], Processamento de Imagens Digitais [13] [14] [15] [16] e Redes Neurais Artificiais [3] [4] [5].

#### **Definição 1: Vetores (em $\mathbf{R}$ e $\mathbf{C}$ )**

Um vetor  $x$  de  $N$  posições, sendo cada um de seus elementos representado por  $x_i$ , com  $i = 0, 1, 2, \dots, N-1$ , definido no campo dos números Reais (em  $\mathbf{R}$ ) será aqui representado por

$$x = [x_0, x_1, \dots, x_{N-1}]^T \quad (1)$$

Um vetor  $X$  de  $N$  posições, sendo cada um de seus elementos representado por  $X_i$ , com  $i = 0, 1, 2, \dots, N-1$ , definido no campo dos números Complexos (em  $\mathbf{C}$ ) será aqui representado por

$$X = [X_0, X_1, \dots, X_{N-1}]^T \quad (2)$$

onde

$$X_0 = x_{0R} + jx_{0I}, X_1 = x_{1R} + jx_{1I}, \dots, X_{N-1} = x_{(N-1)R} + jx_{(N-1)I} \quad (3)$$

Os símbolos  $i$  e  $j$  são usados para especificar índices inteiros de vetores.

O símbolo  $j$  é usado para representar  $\sqrt{-1}$ . O complexo conjugado de uma variável complexa  $Z$  é denotado por  $Z^*$ .

**Definição 2: Produto Interno (em  $\mathbf{R}$  )**

O produto interno de  $x$  e  $y$  é o escalar definido por

$$x^T y = x_0 y_0 + x_1 y_1 + \dots + x_{N-1} y_{N-1} \quad (4)$$

**Definição 3: Produto Interno (em  $\mathbf{C}$  )**

O produto interno de  $X$  e  $Y$  é o escalar definido por

$$X^T Y^* = X_0 Y_0^* + X_1 Y_1^* + \dots + X_{N-1} Y_{N-1}^* \quad (5)$$

**Definição 4: Produto Externo (em  $\mathbf{R}$  )**

O produto externo de  $x$  e  $y$  é a matriz definida por

$$x y^T = \begin{bmatrix} x_0 y_0 & x_0 y_1 & \dots & x_0 y_{N-1} \\ x_1 y_0 & x_1 y_1 & \dots & x_1 y_{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-1} y_0 & x_{N-1} y_1 & \dots & x_{N-1} y_{N-1} \end{bmatrix} \quad (6)$$

**Definição 5: Vetores Ortogonais (em  $\mathbf{R}$  e  $\mathbf{C}$  )**

Dois vetores são ortogonais se o produto interno entre estes dois vetores for zero.

Um conjunto de vetores é ortogonal se cada vetor do conjunto for ortogonal a todos os outros vetores do conjunto.

**Definição 6: Vetores Ortonormais (em  $\mathbf{R}$  e  $\mathbf{C}$  )**

Dois vetores são ortonormais se forem unitários e mutuamente ortogonais.

Um conjunto de vetores é ortonormal se for um conjunto ortogonal com a propriedade de cada vetor do conjunto ser um vetor unitário (vetor com norma unitária). Um conjunto ortonormal de vetores é linearmente independente.

**Definição 7: Norma de um Vetor (em  $\mathbb{R}$  )**

A norma de um vetor  $x$ , denotado por  $\|x\|$ , é definida pelo produto interno do vetor com ele próprio.

$$\|x\| = x_0^2 + x_1^2 + \dots + x_{N-1}^2 \tag{7}$$

Um vetor  $x$  cuja norma é  $\|x\| = 1$  é dito ser um vetor unitário.

**Definição 8: Norma de um Vetor (em  $\mathbb{C}$  )**

A norma de um vetor  $X$ , denotado por  $\|X\|$ , é definida pelo produto interno do vetor com ele próprio.

$$\|X\| = x_0 x_0^* + x_1 x_1^* + \dots + x_{N-1} x_{N-1}^* \tag{8}$$

Um vetor  $X$  cuja norma é  $\|X\| = 1$  é dito ser um vetor unitário.

**Definição 9: Matrizes (em  $\mathbb{R}$  e  $\mathbb{C}$  )**

Uma matriz  $A = [a_{ij}]$ , de dimensões  $N \times N$ , com  $i = 0, 1, \dots, N-1$  e  $j = 0, 1, \dots, N-1$ , definida no campo dos números Reais (em  $\mathbb{R}$  ) será aqui representada por

$$A = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0(N-1)} \\ a_{10} & a_{11} & \dots & a_{1(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{(N-1)0} & a_{(N-1)1} & \dots & a_{(N-1)(N-1)} \end{bmatrix} \tag{9}$$

Uma matriz  $A = [A_{ij}]$  de dimensões  $N \times N$ , com  $i = 0, 1, \dots, N-1$  e  $j = 0, 1, \dots, N-1$ , definida no campo dos números Complexos (em  $\mathbb{C}$ ) será aqui representada por

$$A = \begin{bmatrix} A_{00} & A_{01} & \dots & A_{0(N-1)} \\ A_{10} & A_{11} & \dots & A_{1(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)0} & A_{(N-1)1} & \dots & A_{(N-1)(N-1)} \end{bmatrix} \quad (10)$$

onde

$$\begin{aligned} A_{00} &= a_{00_R} + ja_{00_I}, \\ A_{01} &= a_{01_R} + ja_{01_I}, \\ &\dots \\ A_{10} &= a_{10_R} + ja_{10_I}, \\ A_{11} &= a_{11_R} + ja_{11_I}, \\ &\dots \\ A_{(N-1)(N-1)} &= a_{(N-1)(N-1)_R} + ja_{(N-1)(N-1)_I} \end{aligned} \quad (11)$$

**Definição 10: Matriz Transposta (em  $\mathbb{R}$  e  $\mathbb{C}$ )**

O transposto da matriz  $A$ , denotado por  $A^T$  é obtido intercambiando as linhas de  $A$  pelas colunas de  $A$ .

O transposto do produto de duas matrizes é o produto em ordem reversa dos transpostos das matrizes.

$$(AB)^T = B^T A^T \quad (12)$$

**Definição 11: Complexo Conjugado de uma Matriz (em  $\mathbb{C}$ )**

O complexo conjugado de uma matriz  $A$  é definido por

$$A^* = [A_{ij}^*] \quad (13)$$

O conjugado do produto de duas matrizes  $A$  e  $B$  é igual ao produto da matriz  $A$  conjugada pela matriz  $B$  conjugada.

$$[AB]^* = [A^* B^*] \quad (14)$$

**Definição 12: Conjugado Transposto de uma Matriz (em  $\mathbb{C}$ )**

O conjugado transposto de uma matriz  $A$  é definido por

$$A^{*T} = [A_{ji}^*] \quad (15)$$

**Definição 13: Matriz Diagonal (em  $\mathbb{R}$  e  $\mathbb{C}$ )**

Uma matriz quadrada é dita diagonal quando os elementos de ordem  $ij$  forem nulos para  $i \neq j$  e é denotada por

$$A = \text{diag}(a_{00}, a_{11}, \dots, a_{(N-1)(N-1)}) \quad (\text{em } \mathbb{R}) \quad (16)$$

$$A = \text{diag}(A_{00}, A_{11}, \dots, A_{(N-1)(N-1)}) \quad (\text{em } \mathbb{C}) \quad (17)$$

**Definição 14: Matriz Identidade (em  $\mathbb{R}$ )**

Uma matriz identidade é uma matriz diagonal cujos elementos são unitários. Aqui representada por

$$I = \text{diag}(1, 1, \dots, 1) \quad (18)$$

Sendo o inverso de uma matriz  $A$  denotado por  $A^{-1}$ ,

$$I = A^{-1} A = A A^{-1} \quad (19)$$

**Definição 15: Matriz Simétrica (em  $\mathbb{R}$ )**

Uma matriz  $A$  é dita simétrica quando  $a_{ij} = a_{ji}$  para todos os valores de  $i$  e  $j$ . Assim,

$$A = A^T \quad (20)$$

Os auto-valores de uma matriz real simétrica são reais. Sempre é possível determinar auto-vetores reais para matrizes reais e simétricas. Auto-vetores de uma matriz real e simétrica, correspondentes a distintos auto-valores, são ortogonais. Toda matriz real e simétrica possui um conjunto completo de auto-vetores ortonormais.

Se  $A$  é uma matriz simétrica,

$$x^T Ay = y^T Ax \quad (21)$$

onde  $x$  e  $y$  são vetores de dimensões compatíveis.

**Definição 16: Matriz Hermitiana (em  $\mathbb{C}$ )**

Uma matriz  $A$  é dita Hermitiana quando esta matriz for igual ao seu próprio complexo conjugado transposto. Portanto,

$$[A_{ij}] = [A_{ji}^*] \quad (22)$$

para todos os valores de  $i$  e  $j$ .

Matrizes Hermitianas são quadradas e possuem apenas elementos reais na diagonal principal. Os auto-valores de uma matriz Hermitiana são reais. A matriz é diagonalizável. Os auto-vetores correspondentes a auto-valores distintos são ortogonais e a matriz possui um conjunto ortonormal completo de auto-vetores.

Uma matriz simétrica de elementos reais é uma matriz Hermitiana para o caso particular em que as componentes imaginárias das entidades complexas são nulas.

Se  $A$  é uma matriz Hermitiana,

$$A^* = A^T \quad (23)$$

e

$$X^T AY = Y^T A^* X = Y^T A^T X \quad (24)$$

onde  $X$  e  $Y$  são vetores de dimensões compatíveis.

**Definição 17: Matriz Singular (em  $\mathbf{R}$  e  $\mathbf{C}$ )**

Uma matriz  $A$  é dita singular quando seu determinante, denotado por  $\det\{A\}$ , for nulo.

Esta situação ocorre quando a matriz apresentar linhas ou colunas proporcionais entre si.

$$\det\{A\} = 0 \quad (25)$$

**Definição 18: Matrizes Ortogonais (em  $\mathbf{R}$ )**

Uma matriz quadrada  $A$  é chamada ortogonal se

$$AA^T = A^T A = I \quad (26)$$

ou seja,

$$A^T = A^{-1} \quad (27)$$

Da Equação (26) conclui-se que os vetores linha (ou vetores coluna) de uma matriz ortogonal  $A$  são vetores unitários mutuamente ortogonais e formam, portanto, uma base ortonormal.

**Definição 19: Matriz Unitária (em  $\mathbf{C}$ )**

Uma matriz é chamada unitária se o seu inverso é igual ao seu conjugado transposto.

$$A^{-1} = A^{*T} \quad (28)$$

ou seja,

$$AA^{*T} = A^{*T} A = I \quad (29)$$

Uma matriz real, ortogonal é também unitária.

**Definição 20: Equação Característica de uma Matriz, Auto-valores e Auto-**

### vetores (em $\mathbf{R}$ e $\mathbf{C}$ )

Um vetor  $u$  ( $u \neq 0$ ) é um auto-vetor (ou vetor característico) de uma matriz quadrada  $A$  se existir um escalar  $\lambda$  tal que

$$Au = \lambda u \quad (30)$$

então  $\lambda$  é um auto-valor (ou valor característico) de  $A$ .

A Equação (30) pode ser reescrita como

$$Au - \lambda u = 0 \quad (31)$$

ou

$$(A - \lambda I) u = 0 \quad (32)$$

Assumindo que  $u$  seja um auto-vetor da matriz  $A$ , deve existir um auto-valor  $\lambda$  tal que  $Au = \lambda u$ . A Equação (32) é um sistema de equações lineares homogêneo, para  $u$ . Se  $(A - \lambda I)$  tem um inverso, então a Equação (32) pode ser resolvida para  $u$ , obtendo

$$u = (A - \lambda I)^{-1} 0 \quad (33)$$

ou

$$u = 0 \quad (34)$$

Mas, como  $u$  é um auto-vetor, é não nulo. Portanto,  $u$  só pode ser um auto-vetor de  $A$  se  $(A - \lambda I)$  não possuir um inverso. Se uma matriz quadrada não possui inverso então o seu determinante é zero.

Assim,  $u$  será um auto-vetor de  $A$  se e somente se

$$\det(A - \lambda I) = 0 \quad (35)$$

A Equação (35) é chamada Equação Característica da matriz  $A$ . As raízes desta equação são os auto-valores de  $A$ .

Para cada auto-valor distinto da matriz  $A$  corresponderá pelo menos um auto-vetor, o qual pode ser determinado resolvendo o apropriado conjunto de Equações homogêneas.

Se um auto-valor  $\lambda_i$  é substituído na Equação (32), o correspondente auto-vetor  $u_i$  é a solução de

$$(A - \lambda_i I)u_i = 0 \quad (36)$$

**Definição 21: Média Estatística ou Valor Esperado de um Conjunto de Vetores Aleatórios (em  $\mathbf{R}$  e  $\mathbf{C}$ )**

Seja um conjunto de vetores aleatórios  $N$  dimensionais  $x$ ,  $x = [x_0, x_1, \dots, x_{(N-1)}]$ .

O vetor média ou valor esperado  $N$  dimensional, do conjunto de  $M$  vetores  $x$  é denotado por  $E\{x\}$  e definido como

$$\mu_x = E\{x\} \cong \frac{1}{M} \sum_{k=1}^M x_k \quad (37)$$

Se  $y$  é uma entidade determinística

$$E\{y\} = y \quad (38)$$

Um vetor do qual foi extraída a média é dito centralizado.

Os conceitos ora definidos se aplicam igualmente a vetores definidos no campo dos números complexos.

**Definição 22: Variância de um Conjunto de Vetores (em  $\mathbb{R}$  e  $\mathbb{C}$ )**

Seja o conjunto de vetores aleatórios  $N$  dimensionais  $x$ ,  
 $x = [x_0, x_1, \dots, x_{(N-1)}]$ .

O vetor Variância do conjunto de vetores  $x$  é definido como

$$\sigma^2 = E\{(x - \mu_x)(x - \mu_x)^*\} \quad (39)$$

onde  $\mu_x = E\{x\}$  é o vetor média ou valor esperado dos vetores  $x$ . Assim, tomando o valor esperado como em (37),

$$\sigma^2 \cong \frac{1}{M} \sum_{k=1}^M (x_k - \mu_x)(x_k - \mu_x)^* \quad (40)$$

Os conceitos ora definidos se aplicam igualmente a vetores definidos no campo dos números reais e complexos.

**Definição 23: Covariância de um Conjunto de Vetores (em  $\mathbb{R}$ )**

Seja o conjunto de vetores aleatórios  $N$  dimensionais  $x$ ,  
 $x = [x_0, x_1, \dots, x_{(N-1)}]$  definidos no campo dos Reais.

A matriz Covariância, de dimensões  $N \times N$ , dos vetores  $x$  é definida como

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\} \quad (41)$$

onde  $\mu_x = E\{x\}$  é o vetor média ou valor esperado dos vetores  $x$ . Assim, tomando o valor esperado como em (37),

$$C_x \cong \frac{1}{M} \sum_{k=1}^M (x_k - \mu_x)(x_k - \mu_x)^T \quad (42)$$

ou equivalentemente,

$$C_x = \begin{bmatrix} \sigma_{00}^2 & \sigma_{01} & \cdots & \sigma_{0(N-1)} \\ \sigma_{10} & \sigma_{11}^2 & \cdots & \sigma_{1(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{(N-1)0} & \sigma_{(N-1)1} & \cdots & \sigma_{(N-1)(N-1)}^2 \end{bmatrix} \quad (43)$$

onde os componentes  $\sigma_{ij}$ , para  $i, j = 0, 1, \dots, N-1$  são

$$\sigma_{ij} = E\{(x_i - \mu_i)(x_j - \mu_j)^T\} \quad (44)$$

Os termos diagonais  $\sigma_{ii}^2$  da matriz covariância (para  $i = j$ ) são as variâncias (energias excluindo o nível médio) individuais dos elementos de  $x$  e são sempre não-negativos. Os termos  $\sigma_{ij}$  que não pertencem à diagonal principal (para  $i \neq j$ ) são as covariâncias entre diferentes elementos do vetor  $x$ . Como  $\sigma_{ij} = \sigma_{ji}$ , a matriz covariância é simétrica.

#### **Definição 24: Covariância de um Conjunto de Vetores (em $\mathbb{C}$ )**

Seja o conjunto de vetores aleatórios  $N$  dimensionais  $X$ ,  $X = [X_0, X_1, \dots, X_{(N-1)}]$  definidos no campo dos complexos.

A matriz Covariância, de dimensões  $N \times N$ , dos vetores  $X$  é definida como

$$C_X = E\{(X - \mu_X)(X - \mu_X)^{*T}\} \quad (45)$$

onde  $\mu_x = E\{x\}$  é o vetor média ou valor esperado dos vetores  $x$ . Assim, tomando o valor esperado como em (37),

$$C_X \cong \frac{1}{M} \sum_{k=1}^M (X_k - \mu_X)(X_k - \mu_X)^{*T} \quad (46)$$

ou equivalentemente,

$$C_x = \begin{bmatrix} S_{00}^2 & S_{01} & \dots & S_{0(N-1)} \\ S_{10} & S_{11}^2 & \dots & S_{1(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ S_{(N-1)0} & S_{(N-1)1} & \dots & S_{(N-1)(N-1)}^2 \end{bmatrix} \quad (47)$$

onde os componentes  $S_{ij}$ , para  $i, j = 0, 1, \dots, N-1$  são

$$S_{ij} = E\{(X_i - \mu_x)(X_j - \mu_x)^{*T}\} \quad (48)$$

Os termos diagonais  $S_{ii}^2$  da matriz covariância (para  $i = j$ ) são as energias excluindo o nível médio (variâncias) dos elementos individuais e são sempre reais e não-negativos. Os termos  $S_{ij}$  que não pertencem à diagonal principal (para  $i \neq j$ ) são as covariâncias entre diferentes elementos do vetor  $x$ . Como  $S_{ij} = S_{ji}$ , a matriz covariância é simétrica.

**Definição 25: Ortogonalidade (em  $\mathbf{R}$  )**

Dois vetores aleatórios  $x$  e  $y$  são ortogonais se

$$E\{x^T y\} = 0 \quad (49)$$

**Definição 26: Ortogonalidade (em  $\mathbf{C}$  )**

Dois vetores aleatórios  $X$  e  $Y$  são ortogonais se

$$E\{X^T Y^*\} = 0 \quad (50)$$

**Definição 27: Descorrelação (em  $\mathbf{R}$  )**

Dois vetores aleatórios  $x$  e  $y$  são descorrelacionados se

$$\rho_{xy} = \frac{E\{(x - \mu_x)^T (y - \mu_y)\}}{\sigma_x^2 \sigma_y^2} = \frac{\sigma_{xy}^2}{\sigma_x^2 \sigma_y^2} = 0 \quad (51)$$

Se  $x$  e  $y$  são descorrelacionados,

$$E\{x^T y\} = E\{x^T\}E\{y\} \quad (52)$$

**Definição 28: Descorrelação (em C )**

Dois vetores aleatórios  $X$  e  $Y$  são descorrelacionados se

$$\rho_{XY} = \frac{E\{(X - \mu_X)^T (Y - \mu_Y)^*\}}{\sigma_X^2 \sigma_Y^2} = \frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2} = 0 \quad (53)$$

Se  $X$  e  $Y$  são descorrelacionados,

$$E\{X^T Y^*\} = E\{X^T\}E\{Y^*\} \quad (54)$$

**Definição 29: Imagens Digitais**

Imagens digitais são funções bidimensionais  $f(x,y)$  discretizadas tanto em coordenadas espaciais quanto em intensidade luminosa.

Este estudo considerará imagens monocromáticas. A intensidade luminosa é então chamada brilho, luminância ou nível de cinza.

Uma função representativa de uma imagem digital será considerada neste trabalho como uma matriz quadrada, cujas dimensões são potências inteiras de dois devido ao uso da Transformada Rápida de Fourier Bidimensional (*FFT* Bidimensional).

O valor de cada elemento da matriz corresponde ao nível de cinza associado à coordenada espacial daquele elemento. Denomina-se de pixel um elemento de tal matriz. Cada pixel é uma palavra binária de  $m$ -bits, correspondendo a cada um dos  $2^m$  possíveis valores de nível de cinza.

Neste estudo são considerados 256 níveis de cinza ( $m = 8$ ). Por convenção, branco é representado pelo valor numérico 255 e preto pelo valor numérico zero. Os demais níveis de cinza são representados pelos valores intermediários a este intervalo.

A Equação (55) apresenta uma imagem digital em sua representação matricial. Os valores de pixel  $f(x,y)$  associados às coordenadas espaciais  $x$  e  $y$  (com  $x$  e  $y = 0, 1, \dots, N-1$ ) são os níveis de cinza representativos da intensidade luminosa daquela coordenada.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix} \quad (55)$$

As imagens e suas matrizes representativas são igualmente referidas por  $f(x,y)$ .

**Definição 30: A Transformada Discreta de Fourier Bidimensional**

Uma imagem digital é mapeada do domínio espaço para o domínio frequência via Transformada Discreta de Fourier Bidimensional.

A Transformada Rápida de Fourier Bidimensional da função discreta  $f(x,y)$ , representativa de uma imagem bidimensional de  $N \times N$  pixels é definida por

$$F(u,v) = FFT\{f(x,y)\} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-j\frac{2\pi}{N}(ux+vy)} \quad (56)$$

onde  $F(u,v)$  é referido como o espectro de Fourier bidimensional da imagem  $f(x,y)$ .

A Transformada Rápida de Fourier Bidimensional Inversa da função discreta  $F(u,v)$ , representativa do espectro de uma imagem bidimensional de  $N \times N$  pixels é definida por

$$f(x,y) = IFFT\{F(u,v)\} = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) e^{j\frac{2\pi}{N}(ux+vy)} \quad (57)$$

**Definição 31: Energia de uma Imagem**

Seja  $f(x,y)$  uma imagem digital, a energia ( $E$ ) da imagem digital é definida por

$$E = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f^2(x, y) \quad (58)$$

A energia de uma imagem pode ser definida a partir de sua média e variância,

$$E = N^2(\sigma^2 + \mu^2) \quad (59)$$

onde

$$\mu = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad (60)$$

e

$$\sigma^2 = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [f(x, y) - \mu]^2 \quad (61)$$

### **Definição 32: Densidade Espectral de Energia de uma Imagem**

A Energia Total de uma Imagem pode ser obtida tanto no domínio espaço quanto no domínio frequência.

Sejam  $f(x, y)$  uma imagem digital e  $F(u, v)$  o espectro desta imagem.

A relação entre a Energia da imagem no domínio espaço e a Densidade de Energia do espectro da imagem digital é dada por

$$E = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f^2(x, y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F(u, v)|^2 \quad (62)$$

conhecida como o Teorema de Parseval em Duas Dimensões.

### **Definição 33: Relação Sinal-Ruído de Pico**

A diferença entre a imagem resultante de um processo e a respectiva imagem original pode ser considerada como ruído. A Relação Sinal-Ruído de Pico

(*PSNR*) é usada para determinar a fidelidade de reconstrução de uma imagem. A *PSNR* é dita de pico porque é normalizada para a energia do maior nível de cinza presente na imagem.

Assim, a *PSNR* é definida como

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (63)$$

onde o Erro Médio Quadrático (*MSE*) é dado por

$$MSE = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \quad (64)$$

com  $f(x, y)$  denotando a imagem original e  $\hat{f}(x, y)$  a imagem reconstruída.

**Definição 34: Auto-Valores da Matriz Covariância de um Sinal (em  $\mathbf{R}$  e  $\mathbf{C}$ )**

Se  $A$  é a matriz covariância de um sinal  $X$  complexo,  $A$  é Hermitiana, seus auto-valores são reais e representam, no espaço cuja base é o conjunto de auto-vetores, a energia do sinal, excluía sua média, na direção do auto-vetor associado. Portanto, um auto-valor representa a energia da respectiva componente principal na direção do auto-vetor associado, tendo sido excluído o valor médio do sinal  $X$ .

No caso real, usualmente o conceito de energia encontra-se substituído pelo conceito de variância estatística. No entanto, sendo o caso real um caso particular do complexo, é perfeitamente correto atribuir ao auto-valor o mesmo conceito de energia do domínio complexo, desde que o valor médio do vetor  $x$  seja feito nulo.

Note-se que a energia de um vetor  $X$  no domínio complexo é definida por  $X^T X^*$ . Para o caso real, em que inexistente a parte imaginária, a energia de um vetor  $x$  é definida por  $x^T x$ .

**Definição 35: Redes Neurais Artificiais**

Redes Neurais Artificiais (RNAs) são processadores paralelos distribuídos, constituídos por unidades básicas de processamento chamadas neurônios, que têm uma tendência natural para armazenar conhecimento obtido de dados experimentais e torná-lo disponível para uso. RNAs são freqüentemente utilizadas como ferramentas estatísticas.

Os pesos sinápticos são as transmitâncias das conexões inter-neurônios e são parâmetros livres de uma RNA, passíveis de atualização.

Uma RNA adquire conhecimento através de um processo de aprendizado. Um algoritmo de aprendizado é utilizado para efetuar o processo de aprendizado da RNA e tem por função alterar os pesos sinápticos. Razão de aprendizado é uma constante utilizada para controlar a velocidade de correção dos pesos sinápticos. A atualização dos pesos sinápticos é feita ao longo do processo de apresentar à rede o conjunto de exemplos a serem aprendidos. Este conjunto de exemplos denomina-se conjunto de treino.

Neurônios são as unidades básicas de processamento de uma RNA. A Figura (1) apresenta o modelo de um neurônio artificial visto como um combinador linear.

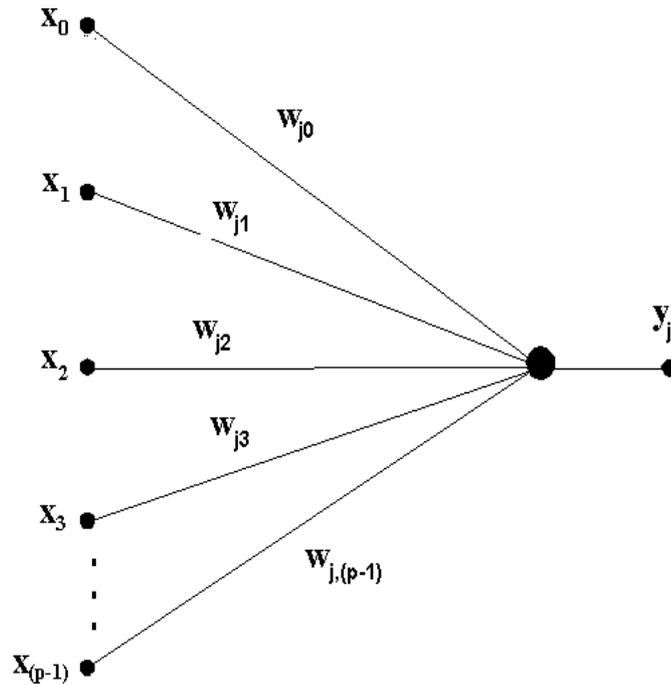


Figura 1 O modelo de um neurônio artificial visto como um combinador linear.

O modelo matemático utilizado para o neurônio linear pode ser expresso através da Equação (65),

$$y_j = \sum_{i=0}^{p-1} x_i w_{ji} \quad (65)$$

onde  $y_j$  são os sinais de saída dos neurônios,  $w_{ji}$  são os pesos sinápticos e  $x_i$  são os sinais de entrada fornecidos à RNA.

O modelo da RNA utilizado neste estudo é mostrado na Figura (2). A RNA é constituída de uma camada de entrada formada por um conjunto de nós de entrada e uma única camada de saída formada por neurônios lineares.

A RNA é dita progressiva porque as saídas dos neurônios em uma camada se conectam às entradas dos neurônios da camada seguinte.

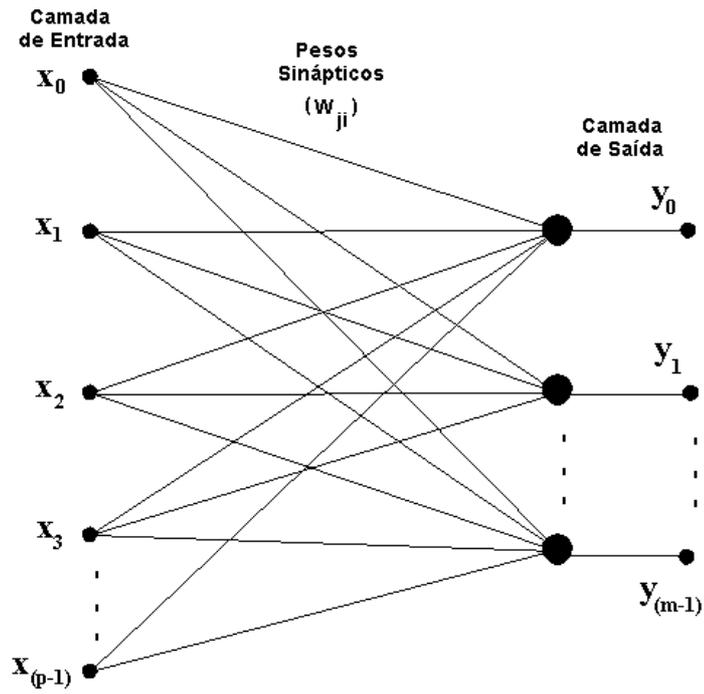


Figura 2 O modelo de uma Rede Neural Artificial Progressiva composta de uma única camada de neurônios lineares.

## Capítulo 2

### Análise dos Componentes Principais

#### (A Transformação Karhunen-Lòeve)

A Análise dos Componentes Principais (*PCA*) ou Transformação Karhunen-Lòeve é o conhecimento básico necessário para o desenvolvimento do estudo proposto nesta dissertação.

O intuito, ao apresentá-la neste capítulo, é que a compreensão da técnica sirva como introdução ao posterior estudo da Transformada Hotelling e do Algoritmo Hebbiano Generalizado para Análise dos Componentes Principais aplicados à compressão de imagens e apresentados nos Capítulos 3 e 4, respectivamente.

A análise das principais características de um conjunto de dados é uma tarefa de fundamental importância no processamento digital de sinais por permitir que informações significativas contidas em um sinal sejam representadas em um espaço de menores dimensões.

A extração destas características é possibilitada pela determinação das direções no espaço de dados de entrada nas quais as amostras exibem maior variância. Os dados de maior variância obtidos são, então, projetados em um subespaço de menor dimensão, produzindo a redução dimensional desejada.

O truncamento de um vetor de dados causa um erro médio quadrático igual à soma das variâncias dos elementos descartados [14]. Se o vetor de dados for

truncado de forma não criteriosa, os elementos eliminados poderão levar consigo características efetivas dos dados, caso sejam elementos de alta variância.

A solução para este problema reside em especificar uma transformação com a característica de que alguns componentes transformados possuam pequena variância para que, em um truncamento criterioso, representem pouca perda de informação.

A Transformação Karhunen-Lòeve é a solução indicada, pois maximiza a razão de decréscimo da variância, concentrando a maior parte da variância dos dados em alguns poucos elementos. Assim, a escolha dos elementos a serem descartados será ótima no sentido do Erro Médio Quadrático (*MSE*) [4][5].

O desenvolvimento do método para Análise dos Componentes Principais apresentado neste capítulo segue a proposta de Haykin [4].

Seja  $x$  um vetor aleatório, de dimensão  $p$ , representativo do conjunto de dados de interesse e de média zero. Caso o vetor  $x$  não possua média zero, a média deverá ser subtraída do vetor antes de iniciar a análise.

Seja  $u$  um vetor unitário, igualmente de dimensão  $p$ , sobre o qual será projetado o vetor representativo do conjunto de dados, vetor  $x$ . A projeção é dada pelo produto interno mostrado na Equação (66).

$$a = x^T u = u^T x \quad (66)$$

Seendo  $u$  um vetor unitário, a norma de  $u$  é dada pela Equação (67).

$$\|u\| = (u^T u) = 1 \quad (67)$$

A projeção  $a$  também é aleatória, com média e variância associadas à estatística do vetor de dados  $x$ .

A média da projeção  $a$  é dada por

$$E\{a\} = E\{u^T x\} \quad (68)$$

$$E\{a\} = u^T E\{x\} = 0 \quad (69)$$

A variância da projeção  $a$  é dada por

$$\sigma_a^2 = E\{a^2\} \quad (70)$$

$$\sigma_a^2 = E\{(u^T x)(x^T u)\} \quad (71)$$

$$\sigma_a^2 = u^T E\{xx^T\}u \quad (72)$$

onde o valor esperado do produto externo do vetor  $x$  por ele mesmo ( $E\{xx^T\}$ ) é a matriz covariância  $C_x$ , de dimensões  $p \times p$ , dos vetores de dados  $x$ .

Portanto, a Equação (72) pode ser escrita como

$$\sigma_a^2 = u^T C_x u \quad (73)$$

onde observa-se que a variância  $\sigma^2$  da projeção  $a$  é uma função do vetor unitário  $u$ .

$$\sigma_a^2 = f(u) \quad (74)$$

O intuito agora é determinar os vetores  $u$  ao longo dos quais a variância da projeção  $a$  é máxima, sujeitos à restrição dada pela Equação (67).

Se  $u$  é um vetor unitário tal que  $f(u)$  tenha um valor máximo, então, para qualquer pequena perturbação  $\delta u$  do vetor unitário  $u$ ,

$$f(u + \delta u) \approx f(u) \quad (75)$$

Aplicando-se a definição dada pela Equação (73) na Equação (75) e admitindo-se a igualdade devida ao valor de  $\delta u$  ser muito pequeno,

$$f(u + \delta u) = (u + \delta u)^T C_x (u + \delta u) \quad (76)$$

$$f(u + \delta u) = u^T C_x u + u^T C_x \delta u + \delta u^T C_x u + \delta u^T C_x \delta u \quad (77)$$

Como a matriz covariância do vetor de dados,  $C_x$ , é uma matriz simétrica, tem-se que

$$u^T C_x \delta u = \delta u^T C_x u \quad (78)$$

e a Equação (77) pode ser reescrita sob a forma

$$f(u + \delta u) = u^T C_x u + 2\delta u^T C_x u + \delta u^T C_x \delta u \quad (79)$$

Como  $\delta u$  é muito pequeno, o termo  $\delta u^T C_x \delta u$  no lado direito da Equação (79) pode ser desconsiderado. Além disto, considerando-se a definição das Equações (73) e (74), pode-se escrever que

$$f(u + \delta u) = f(u) + 2\delta u^T C_x u \quad (80)$$

Ao levar a Equação (75) à Equação (80), conclui-se que

$$\delta u^T C_x u = 0 \quad (81)$$

Como se pretende chegar ao conceito de auto-vetor (Capítulo 1, Definição 20) a outra condição a ser respeitada deriva de que apenas são admissíveis as perturbações  $\delta u$  para as quais a norma do vetor perturbado  $(u + \delta u)$  permaneça unitária, ou seja

$$\|u + \delta u\| = 1 \quad (82)$$

o que equivale a dizer que

$$(u + \delta u)^T (u + \delta u) = 1 \quad (83)$$

Expandindo o produto mostrado na Equação (83),

$$(u + \delta u)^T (u + \delta u) = u^T u + u^T \delta u + \delta u^T u + \delta u^T \delta u \quad (84)$$

onde

$$u^T \delta u = \delta u^T u \quad (85)$$

e onde o termo  $\delta u^T \delta u$  pode ser desconsiderado. Como  $(u^T u + 2 \delta u^T u)$  deve ser unitário (Equação 83), pode-se afirmar que

$$\delta u^T u = 0 \quad (86)$$

A análise da Equação (86) permite concluir que as perturbações  $\delta u$  devem ser ortogonais a  $u$ , sendo esta a única mudança permitida na direção de  $u$ , de maneira que a Equação (82) seja obedecida.

Nas Equações (81) e (86) residem as duas condições a serem respeitadas para a determinação dos vetores  $u$  para os quais  $f(u)$  terá valor extremo. Combinando convenientemente estas equações, pode-se escrever que

$$C_x (\delta u)^T u - \lambda (\delta u)^T u = 0 \quad (87)$$

onde o fator de escala  $\lambda$  foi introduzido para compatibilizar as dimensões dos elementos do vetor unitário  $u$  com a matriz covariância  $C_x$ . (Por exemplo, se a matriz covariância provém de um vetor cuja dimensão de seus elementos é [Volt],  $\lambda$  terá a dimensão de [Volt<sup>2</sup>].)

Reescrevendo a Equação (87),

$$(\delta u)^T (C_x u - \lambda u) = 0 \quad (88)$$

Para que a condição posta na Equação (88) seja satisfeita, é necessário e suficiente que

$$C_x u = \lambda u \quad (89)$$

que é a equação que define os vetores  $u$  para os quais  $f(u)$  tem valor máximo.

A Equação (89) é claramente reconhecida como a equação dos auto-valores, descrita na Definição 20, Capítulo 1, e tem soluções não-triviais para aqueles valores de  $\lambda$  que são os auto-valores da matriz  $C_x$ . Os vetores  $u$  associados a estes  $\lambda$  são os auto-vetores de  $C_x$ .

A matriz covariância é uma matriz Hermitiana e, portanto, possui auto-valores reais e não-negativos. Assumindo que os auto-valores da matriz covariância sejam distintos, os auto-vetores associados a estes auto-valores serão ortogonais (Capítulo 1, Definição 16).

Sejam  $\lambda_0, \lambda_1, \dots, \lambda_{p-1}$  os auto-valores da matriz  $C_x$ , de dimensões  $p \times p$ , e  $u_0, u_1, \dots, u_{p-1}$  os auto-vetores associados a estes auto-valores. Assim, da Equação (89) pode-se escrever

$$C_x u_j = \lambda_j u_j \quad \text{para } j = 0, 1, \dots, p-1 \quad (90)$$

Seja  $\Lambda$  uma matriz diagonal,  $p \times p$ , composta pelos auto-valores da matriz  $C_x$ , de acordo com a Equação (91)

$$\Lambda = \text{diag}[\lambda_0, \lambda_1, \dots, \lambda_j, \dots, \lambda_{p-1}] \quad (91)$$

onde os auto-valores estão ordenados de forma decrescente, de modo que  $\lambda_0$  seja o maior auto-valor.

Seja  $U$  uma matriz  $p \times p$  formada pelos auto-vetores correspondentes, conforme Equação (92).

$$U = [u_0, u_1, \dots, u_j, \dots, u_{p-1}] \quad (92)$$

De acordo com as Equações (91) e (92), a Equação (90) pode ser reescrita como

$$C_x U = U \Lambda \quad (93)$$

Como os auto-vetores de  $C_x$  (vetores-coluna de  $U$ ) são ortogonais, a matriz  $U$  é dita ortogonal. Os vetores de  $U$  satisfazem, ainda, a condição de ortonormalidade, pois

$$u_i^T u_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \quad (94)$$

A Equação (94) pode também ser escrita sob a forma

$$U^T U = I \quad (95)$$

de onde vem que

$$U^T = U^{-1} \quad (96)$$

A Equação (93) pode ser reescrita, considerando a Equação (96), sob a forma da transformação de similaridade ortogonal. Pré-multiplicando ambos os lados da Equação (93) por  $U^T$ ,

$$U^T C_x U = U^T U \Lambda \quad (97)$$

Considerando a Equação (96),

$$U^T C_x U = U^{-1} U \Lambda \quad (98)$$

De onde resulta a transformação ortogonal de similaridade, mostrada na Equação (99).

$$U^T C_x U = \Lambda \quad (99)$$

que, na forma expandida,

$$u_j^T C_x u_k = \begin{cases} \lambda_j, & k = j \\ 0, & k \neq j \end{cases} \quad (100)$$

Nas Equações (73) e (74) tem-se

$$\sigma_a^2 = f(u) = u^T C_x u \quad (101)$$

Comparando as Equações (100) e (101), tem-se que

$$f(u_j) = \lambda_j \quad \text{para } j = 0, 1, \dots, p-1 \quad (102)$$

Dois conclusões importantes seguem da interpretação da Equação (102).

Os auto-vetores da matriz covariância  $C_x$  dos vetores de dados de média zero representam as direções principais ao longo das quais a variância é máxima. Os auto-valores associados a estes auto-vetores definem os valores máximos de variância.

Em outras palavras, os auto-valores e auto-vetores representam os modos de energia do conjunto de dados (cujo nível médio é assumido zero). Cada modo de energia caracteriza-se por uma direção definida pelo respectivo auto-vetor no espaço de modos de variação dos dados, com nível de energia definido pelo auto-valor associado.

Entenda-se aqui por modo de energia a forma como os dados variam. O conjunto de dados varia de amostra para amostra. Certas variações deste conjunto são semelhantes entre si, de forma que podem ser incluídas em uma mesma classe. As variações pertencentes a uma mesma classe apresentam uma energia total associada à classe. Assim, cada classe de variação é um modo de energia. O próprio nível médio, previamente extraído do conjunto de dados, pode ser considerado como um modo energético cuja forma de variação é uma constante.

Como há  $p$  possíveis soluções para o vetor unitário  $u$ , também haverá  $p$  projeções  $a$  do vetor representativo do conjunto de dados  $x$  sobre o vetor unitário  $u$ . Assim, como em (66),

$$a_j = x^T u_j = u_j^T x \quad \text{para } j = 0, 1, \dots, p-1 \quad (103)$$

onde os  $a_j$  são chamados componentes principais do vetor de dados  $x$ , que são os modos principais de variação de  $x$ .

Para obter a reconstrução do vetor de dados  $x$  a partir das projeções  $a_j$  parte-se da Equação (104),

$$a = U^T x \quad (104)$$

onde

$$a = [a_0, a_1, \dots, a_{p-1}]^T \quad (105)$$

e

$$U^T x = [x^T u_0, x^T u_1, \dots, x^T u_{p-1}]^T \quad (106)$$

Assim, pré-multiplicando ambos os lados da Equação (104) por  $U$ , e considerando (96), o vetor de dados  $x$  será reconstruído por

$$x = Ua = \sum_{j=0}^{p-1} a_j u_j \quad (107)$$

Na Equação (107), os vetores unitários  $u_j$  representam uma base do espaço de dados. Ao considerar os  $p$  termos, a transformação reconstruirá completamente o vetor de dados.

A redução dimensional é obtida através da equação de reconstrução dos dados, descartando as combinações lineares que apresentarem menor variância (energia) e retendo os termos de maior variância (energia).

Sejam  $\lambda_0, \lambda_1, \dots, \lambda_m$  os  $m$  maiores auto-valores da matriz covariância do vetor de dados. O vetor de dados poderá, então, ser aproximado por

$$\hat{x} = \sum_{j=0}^{m-1} a_j u_j \quad \text{para } m < p \quad (108)$$

o que equivale a truncar a reconstrução expressa pela Equação (107).

O erro resultante da redução das combinações lineares relativas aos auto-valores menores é obtido subtraindo o vetor truncado do vetor original,

$$e = x - \hat{x} \quad (109)$$

ou, substituindo (107) e (108) em (109),

$$e = \sum_{j=m}^{p-1} a_j u_j \quad (110)$$

A variância total do vetor de dados  $x$ , de acordo com as Equações (73) e (100) é expressa por

$$\sum_{j=0}^{p-1} \sigma_j^2 = \sum_{j=0}^{p-1} \lambda_j \quad (111)$$

A variância total do vetor estimado  $\hat{x}$ , referente aos  $m$  auto-valores retidos é

$$\sum_{j=0}^{m-1} \sigma_j^2 = \sum_{j=0}^{m-1} \lambda_j \quad (112)$$

Quanto mais próximos de zero forem os auto-valores descartados, mais eficiente será a redução dimensional, no sentido de preservar a informação contida no conjunto de dados.

## Capítulo 3

### A Transformada Hotelling na Compressão de Imagens

A Transformada Hotelling é uma transformação bidimensional, baseada nas propriedades estatísticas de imagens. Esta transformada é uma aplicação do método Análise dos Componentes Principais (apresentado no Capítulo 2) à compressão de dados no domínio espaço e é também conhecida como Transformação Discreta Karhunen-Loève ou Transformada dos Auto-Vetores.

A transformação é obtida simplesmente multiplicando-se a matriz Hotelling de Transformação pelo vetor imagem centralizado (com a média extraída) para obter um novo vetor imagem. A compressão obtida é devida à redução dimensional realizada nesta matriz.

O desenvolvimento da Transformada Hotelling aqui descrito foi construído a partir da apresentação desenvolvida por Gonzales e Wintz [14].

Uma imagem  $f(x, y)$ , de dimensões  $N \times N$  é particionada em  $M$  sub-imagens  $f_m(x, y)$  de dimensões  $N_s \times N_s$ ,  $m = 0, \dots, M-1$ .

Cada uma destas  $M$  sub-imagens  $f_m(x, y)$  é representada por um vetor de dimensão  $N_s^2$  obtido de acordo com a seguinte lei de formação: os primeiros  $N_s$  elementos deste vetor serão os elementos da primeira linha da sub-imagem, lidos da esquerda para a direita, os seguintes  $N_s$  elementos deste vetor serão os elementos da segunda linha da sub-imagem, lidos da mesma forma, e assim sucessivamente, até que todas as linhas de  $f_m(x, y)$  tenham sido levadas ao vetor  $N_s^2$ -dimensional  $x_m$ ,

$$\mathbf{x}_m = \left[ x_{m_0}, x_{m_1}, \dots, x_{m_{N_s-1}}, \dots, x_{m_{N_s^2-1}} \right]^T \quad (113)$$

onde

$$x_{m_0} = f_m(0,0), x_{m_1} = f_m(0,1), \dots, x_{m_{N_s-1}} = f_m(0, N_s - 1), \dots, x_{m_{N_s^2-1}} = f_m(N_s - 1, N_s - 1).$$

O próximo passo é determinar a matriz covariância dos vetores  $\mathbf{x}$ , definida por

$$C_x = \frac{1}{M} \sum_{m=0}^{M-1} (\mathbf{x}_m - \boldsymbol{\mu}_x)(\mathbf{x}_m - \boldsymbol{\mu}_x)^T \quad (114)$$

onde

$$\boldsymbol{\mu}_x = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{x}_m \quad (115)$$

Para posterior compreensão da redução dimensional possibilitada por esta transformação é importante notar as dimensões dos vetores que estão sendo tratados. Nas Equações (114) e (115) constata-se que as dimensões do vetor média e da matriz covariância são, respectivamente,  $N_s^2$  e  $N_s^2 \times N_s^2$ .

Sejam  $q_i$  e  $\lambda_i$ , com  $i = 0, 1, \dots, N_s^2 - 1$  os auto-vetores e auto-valores de  $C_x$ . O conjunto dos auto-valores  $\lambda_i$  tem seus elementos ordenados de forma decrescente, portanto, o auto-vetor  $q_1$  é associado ao maior auto-valor,  $\lambda_1$ .

A matriz Hotelling de Transformação, denominada  $H$ , é a matriz cujas linhas são os auto-vetores da matriz covariância  $C_x$ .

$$H = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{N_s^2-1} \end{bmatrix} = \begin{bmatrix} q_{00} & q_{01} & \dots & q_{0(N_s^2-1)} \\ q_{10} & q_{11} & \dots & q_{1(N_s^2-1)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{(N_s^2-1)0} & q_{(N_s^2-1)1} & \dots & q_{(N_s^2-1)(N_s^2-1)} \end{bmatrix} \quad (116)$$

A transformação é obtida através da multiplicação da Matriz  $H$  pelos vetores  $x$  centralizados, conforme Equação (117).

$$y = H(x - \mu_x) \quad (117)$$

A análise do conjunto de vetores  $y$  transformados conduz a duas características importantes da Transformada Hotelling. A primeira delas referente à correlação entre os vetores transformados e a segunda referente à reconstrução dos vetores de dados originais  $x$  a partir dos vetores  $y$  transformados.

Para proceder à análise correlacional, determina-se a covariância dos vetores  $y$ . Estes vetores possuem média zero, como pode-se verificar ao tomar-lhes o valor esperado, expressões (118) a (121) .

$$\mu_y = E\{y\} \quad (118)$$

$$\mu_y = E\{H(x - \mu_x)\} \quad (119)$$

$$\mu_y = HE\{x\} - H\mu_x \quad (120)$$

$$\mu_y = 0 \quad (121)$$

Pela definição de covariância,

$$C_y = E\{(y - \mu_y)(y - \mu_y)^T\} \quad (122)$$

onde  $\mu_y = 0$  , como provado em (121). Isto leva a Expressão (122) a

$$C_y = E\{(y)(y)^T\} \quad (123)$$

Substituindo  $y$  dado pela Equação (117) na Equação (123) segue que,

$$C_y = E\{(H(x - \mu_x))(H(x - \mu_x))^T\} \quad (124)$$

Conforme visto no Capítulo 1, Definição 14, pode-se reescrever a Equação (124) sob a forma

$$C_y = E \{ H(x - \mu_x)(x - \mu_x)^T H^T \} \quad (125)$$

Aplicando a propriedade do valor esperado descrita na Definição 17, Equação (29),  $C_y$  resulta em

$$C_y = HE \{ (x - \mu_x)(x - \mu_x)^T \} H^T \quad (126)$$

onde o valor esperado tomado é a covariância dos vetores  $x$ . Pode-se, agora, escrever (126) como

$$C_y = H C_x H^T \quad (127)$$

Da teoria matricial e, conforme mostrado no Capítulo 2, Equação (99), sabe-se que  $C_y$  é uma matriz diagonal cujos elementos são os auto-valores de  $C_x$ . Portanto,  $C_y$  é da forma

$$C_y = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N_x^2-1}) \quad (128)$$

A observação da matriz covariância dos dados transformados (Expressão 128) mostra que todas as covariâncias  $\sigma_{ij}$  (para  $i \neq j$ ) são nulas, o que demonstra que os vetores  $y$  são descorrelacionados e ortogonais. Os termos da diagonal principal são as variâncias dos vetores  $y$  nas direções definidas pelos auto-vetores  $q$ .

A segunda característica de interesse apontada pela Equação (117), refere-se à reconstrução dos vetores de dados originais  $x$  a partir dos vetores  $y$  transformados, responsável pela aplicabilidade da Transformada Hotelling à compressão de imagens.

Partindo-se da Equação (117), com o intuito de isolar  $x$ , obtém-se

$$Hx = y - H\mu_x \quad (129)$$

Pré-multiplicando ambos os lados da igualdade por  $H^{-1}$ , chega-se a

$$H^{-1}Hx = H^{-1}y - H^{-1}H\mu_x \quad (130)$$

ou

$$x = H^{-1}y - \mu_x \quad (131)$$

e como  $H^{-1} = H^T$  ,

$$x = H^T y + \mu_x \quad (132)$$

Os vetores  $x$  são perfeitamente recuperados a partir de  $y$ , através da Equação (132), onde a matriz Hotelling de Transformação  $H$  é tomada integralmente.

Se, ao invés de tomar-se todos os auto-vetores  $q_i$  de  $C_x$  para a construção da matriz  $H$ , forem tomados apenas os auto-vetores associados aos auto-valores mais significativos (de maior variância), será obtida uma estimativa  $\hat{x}$  do conjunto de dados  $x$ .

Seja  $k$  o número de auto-vetores escolhido para a formação da matriz  $H_k$ .

Os vetores  $\hat{x}$  serão estimados por

$$\hat{x} = H_k^T y + \mu_x \quad (133)$$

A Expressão (133) pode ser vista como uma projeção dos vetores  $y$  sobre uma nova base formada pelos  $k$  auto-vetores escolhidos.

Quando  $k < i$  ocorrerá redução dimensional na representação  $\hat{x}$  dos vetores  $x$ , justificando a aplicabilidade da Transformada Hotelling em compressão de dados. O erro resultante na imagem reconstruída a partir de tal redução dimensional é conforme visto no Capítulo 2 , Equação (110).

## Capítulo 4

### **O Algoritmo Hebbiano Generalizado para Análise dos Componentes Principais**

Como visto nos Capítulos 2 e 3, para proceder à Análise dos Componentes Principais é necessário computar a matriz covariância do conjunto de dados de entrada para, então, aplicar um procedimento numérico que extraia os auto-valores e os correspondentes auto-vetores desta matriz. Os auto-vetores correspondentes aos auto-valores mais significativos são usados para extrair os componentes principais dos dados.

No entanto, para grandes conjuntos de dados, as dimensões da matriz covariância crescem significativamente, tornando inadequado este método de extração das componentes principais. Além disto, todos os auto-valores e auto-vetores precisam ser computados, mesmo que somente poucos auto-vetores (aqueles correspondentes aos maiores auto-valores) venham a ser utilizados no processo.

A solução eficiente para este problema deve encontrar os principais auto-vetores sem a necessidade de determinar a matriz covariância. Tal solução é alcançada ao abordar o problema utilizando Redes Neurais Artificiais (RNAs) [17].

Redes Neurais Artificiais têm a capacidade de resolver problemas desta natureza conectando múltiplas unidades de processamento. Através de um algoritmo de aprendizado, as RNAs podem aprender padrões diretamente dos padrões de entrada, tendo capacidade de generalizar as soluções obtidas [11].

A técnica que utiliza Redes Neurais Artificiais para produzir Análise dos Componentes Principais sobre um vetor de dados de interesse abordada neste estudo é o

Algoritmo Hebbiano Generalizado proposto por Sanger em 1989 [4]. Este algoritmo combina a ortogonalização de Gram-Schmidt [10] ao modelo de um único neurônio linear introduzido por Oja em 1982 [4].

O Algoritmo Hebbiano Generalizado é um dos algoritmos mais usados em aplicações práticas, porque extrai os componentes principais individualmente, um após o outro em ordem decrescente de variância, e também porque possibilita que, após treinada a rede, os resultados obtidos possam ser aplicados a um outro conjunto de dados [4].

O desenvolvimento apresentado neste capítulo segue a exposição de Haykin [4], Hassoun [5] e Hertz, Krogh e Palmer [18].

#### **4.1 - O Aprendizado Hebbiano**

Segundo Haykin [4], no contexto de RNAs, aprendizado é um processo pelo qual os parâmetros livres da rede são adaptados através da repetição de adequadas operações, por um número suficiente de vezes, pelo ambiente no qual a rede está contida.

Um algoritmo de aprendizado é um conjunto de regras bem definidas, prescritas para a solução do problema de aprendizado. O tipo de aprendizado é determinado pela maneira como a mudança nos parâmetros livres da RNA é feita.

O Aprendizado Hebbiano é uma classe de aprendizado não-supervisionado ou auto-organizado. Uma RNA treinada sob aprendizado não-supervisionado descobre padrões significativos ou característicos dos dados de entrada sem o auxílio de um tutor externo. Não há realimentação do ambiente para auxiliar a rede a determinar se a saída está ou não correta. A RNA deve descobrir por si padrões,

características, regularidades, correlações ou categorias nos dados de entrada. As unidades e conexões devem, portanto, apresentar algum grau de auto-organização.

Em alguns casos, os vetores de entrada devem ser mapeados em um conjunto dimensionalmente menor de padrões tal que preserve as relações existentes no conjunto original.

O processo de aprendizado auto-organizado consiste em modificar repetidamente todos os pesos sinápticos de uma RNA em resposta a padrões de entrada e de acordo com regras prescritas, até que atinjam uma pré-determinada condição.

O conhecimento que vai sendo “cristalizado” na RNA é obtido da observação repetida de parâmetros estatísticos (média, variância e matriz correlação) dos dados de entrada. Neste sentido, redundância de padrões provê conhecimento.

Hebb postulou, na obra “*The Organization of Behavior*” (1949), que sinapses biológicas eficazes mudam na proporção da correlação entre os sinais pré-sinápticos  $x$  e pós-sinápticos  $y$ .

O Postulado de Hebb diz que:

“Quando um neurônio da célula ‘A’ está próximo o suficiente para excitar a célula ‘B’, e, repetidamente ou persistentemente participa do processo de dispará-lo, algum processo de crescimento ou mudança metabólica ocorre em uma ou ambas as células, tal que a eficiência de ‘A’ como uma das células que dispara ‘B’ é aumentada.”

O Postulado de Hebb leva ao estabelecimento dos princípios que regem o Aprendizado:

1. “Se dois neurônios em cada lado de uma sinapse são ativados simultaneamente, então a força daquela sinapse é seletivamente aumentada.”

O primeiro princípio do Aprendizado Hebbiano afirma que as modificações nos pesos sinápticos, baseadas em sinais pré-sinápticos e pós-sinápticos, tendem a se auto-amplificar. Uma sinapse forte leva à coincidência entre os sinais pré-sinápticos e pós-sinápticos e isto leva ao acréscimo da força da sinapse.

2. “Se dois neurônios em cada lado de uma sinapse são ativados de forma assíncrona, então a sinapse é seletivamente enfraquecida ou eliminada.”

O segundo princípio refere-se à limitação de recursos, o que leva à competição entre sinapses e, portanto, à seleção das sinapses que crescem mais vigorosamente, às expensas de outras.

Para que haja estabilização do sistema, deve haver competição entre os pesos sinápticos por recursos limitados. Ou seja, um aumento na força de algumas sinapses na rede deve ser compensado por um decréscimo em outras. Assim, só as sinapses de sucesso podem crescer, enquanto as de menos sucesso tendem a enfraquecer, podendo, inclusive, desaparecer.

3. “Modificações nos pesos sinápticos tendem a cooperar.”

Uma única sinapse por si só não pode produzir eventos favoráveis, de forma eficiente. Portanto, é necessária cooperação entre um conjunto de sinapses

convergindo a um particular neurônio e levando sinais suficientemente fortes para ativar aquele neurônio.

Na rede de uma única unidade, mostrada na Figura (3), os sinais pré-sinápticos e pós-sinápticos, respectivamente  $x$  e  $y$ , podem ser equacionados por

$$w_0(n+1) = w_0(n) + \eta y_0(n)x(n) \quad (134)$$

$$y_0(n) = x^T(n) w_0(n) = w_0^T(n)x(n) \quad (135)$$

onde  $\eta$  é uma constante positiva que determina a razão de aprendizado,  $x(n)$  é a entrada da rede e  $y_0(n)$  é a saída da rede.

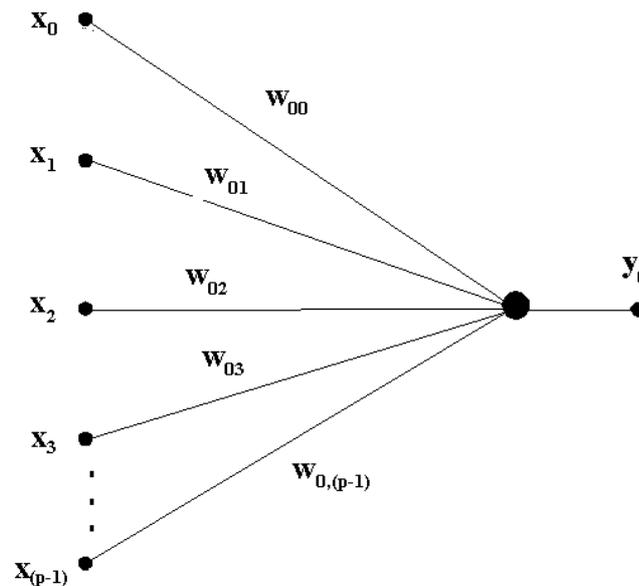


Figura 3 RNA de uma única unidade, a ser treinada pelo Aprendizado Hebbiano

A Equação (134) expressa a mudança no peso sináptico, que ocorre na proporção da correlação entre os sinais pré e pós-sinápticos. A Equação (135) expressa a saída da RNA.

O conjunto de vetores  $x$  de entrada possui distribuição de probabilidade arbitrária. A cada tempo  $n$  um vetor  $x$  é apresentado à rede, escolhido aleatoriamente do conjunto.

Segundo os princípios do Aprendizado Hebbiano, quanto mais provável for uma particular entrada  $x$ , maior será a correlação da saída  $y$  com esta entrada. Assim, quanto mais provável for  $x$ , maior será a saída  $y$ . Quanto maior for a saída  $y$ , mais a variação do peso sináptico que a encorajou aumentará. Desta forma os pesos sinápticos permaneceriam crescendo sem limite, impedindo a continuação do processo de aprendizado. Na realidade, o que ocorre é que, pelo aumento demasiado do peso sináptico, a rede é levada à saturação precoce e é incapaz de aprender os padrões apresentados.

Para validar esta afirmação, é necessário investigar a estabilidade do modelo. Suponha-se que, após um tempo de aprendizado suficientemente grande, os pesos sinápticos atinjam um ponto de equilíbrio.

Tomando-se o valor esperado da mudança dos pesos sinápticos (de acordo com a Equação (134)) para esta suposta condição de equilíbrio,

$$E\{\Delta w_0(n)\} = E\{\eta y_0(n)x(n)\} \quad (136)$$

Substituindo a Equação (135) na Equação (136),

$$E\{\Delta w_0(n)\} = E\{\eta x(n)x^T(n)w_0(n)\} \quad (137)$$

Considerando que o parâmetro razão de aprendizado  $\eta$  é uma entidade determinística e que os vetores de dados  $x$  e os vetores pesos sinápticos  $w$  são estatisticamente independentes,

$$E\{\Delta w_0(n)\} = \eta E\{x(n)x^T(n)\}E\{w_0(n)\} \quad (138)$$

E que, no equilíbrio,

$$E\{\Delta w_0(n)\}=0, \quad (139)$$

$$E\{x(n)x^T(n)\}=C_x \quad (140)$$

Para uma condição de equilíbrio o número de iterações  $n$  seria suficientemente grande para que o vetor peso sináptico tendesse ao equilíbrio. Ou seja, quando  $n \rightarrow \infty$ ,

$$E\{w_0(n)\}=q_0 \quad (141)$$

onde  $q_0$  é um valor constante.

Nestas condições, a Equação (138) se torna

$$C_x q_0 = 0 \quad (142)$$

Assim, no suposto ponto de equilíbrio, a solução da Equação (142), conforme a teoria de auto-valores e auto-vetores (Capítulo 1, Definição 20), é um auto-vetor  $q$  de  $C_x$  cujo correspondente auto-valor é zero.

Algumas considerações podem ser traçadas a partir da Equação (142). A matriz de correlação  $C_x$  é uma matriz Hermitiana que é sempre não-nula pois denota a correlação do sinal com ele próprio (mesmo um sinal de ruído tem auto-correlação não nula).

Os auto-valores de uma matriz Hermitiana são reais e poderão ser positivos (Capítulo 1, Definição 16). Qualquer flutuação de um peso sináptico que tenha uma componente ao longo da direção de um auto-vetor associado a um auto-valor positivo crescerá exponencialmente. A direção relativa ao maior auto-valor da matriz covariância poderá se tornar dominante, de forma que o peso sináptico  $w$  irá se

aproximar gradualmente de um auto-vetor  $q$  correspondendo a um auto-valor máximo, cuja norma crescerá demasiadamente.

O auto-vetor  $q$ , portanto, nunca se estabilizará e haverá somente soluções instáveis para o algoritmo de Aprendizado Hebbiano.

O problema da instabilidade do algoritmo, acima exposto, pode ser equacionado tomando a derivada da variação do peso sináptico  $\Delta w_0$  em relação ao próprio peso sináptico  $w_0$ , para investigar como a variação do peso sináptico se comporta ao longo do tempo  $n$ . Assim,

$$\frac{d}{d(w_0(n))}(\Delta w_0(n)) = \frac{d}{d(w_0(n))}(\eta y_0(n)x(n)) \quad (143)$$

Considerando as formas expandidas na Equação (143),

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) = \frac{d}{d(w_{0i}(n))}(\eta x_i(n) \sum_i x_i(n)w_{0i}(n)) \quad (144)$$

A derivada do somatório da expressão (144) em relação a  $w_{0i}(n)$  só existirá para o termo  $x_i(n)w_{0i}(n)$ , quando será igual a  $x_i(n)$ . A Equação (144) se tornará, portanto,

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) = \eta x_i^2(n) \quad (145)$$

De acordo com a Equação (145), a razão de variação da variação do peso sináptico é positiva para todo  $n$  e todo  $i$ . Isto implica que, ao longo de todo o processo de atualização dos pesos sinápticos, a variação será crescente, o que torna o auto-vetor  $q$  uma solução instável para o algoritmo.

Para evitar a divergência do algoritmo de aprendizado Hebbiano é preciso restringir o crescimento do vetor  $w$ . Oja propôs (1982) [4] como uma alternativa para a

estabilização do Aprendizado Hebbiano, uma modificação no algoritmo que permite ao vetor peso sináptico  $w$  atingir norma unitária, além de corresponder ao maior auto-valor de  $C_x$ .

A regra de Oja corresponde a adicionar um decaimento ao peso sináptico, proporcional a  $y_0^2(n)$ .

A Equação (146) expressa (novamente sob a forma vetorial) a regra de Aprendizado Hebbiano após deflacionada de um fator  $\eta y_0^2(n)w_0(n)$ .

$$w_0(n+1) = w_0(n) + \eta y_0(n)x(n) - \eta y_0^2(n)w_0(n) \quad (146)$$

A regra para o Aprendizado Hebbiano apresentada na Equação (146) pode ser reescrita como,

$$\Delta w_0(n) = \eta y_0(n) \{x(n) - y_0(n)w_0(n)\} \quad (147)$$

Para provar que  $\|w_0\|=1$  e que o peso sináptico  $w_0$  está “depositado” na direção do maior auto-vetor de  $C_x$ , considere-se a Equação (147). Tomando o valor esperado da mudança dos pesos sinápticos,

$$E\{\Delta w_0(n)\} = E\{\eta y_0(n)[x(n) - y_0(n)w_0(n)]\} \quad (148)$$

Reescrevendo a Equação (148),

$$E\{\Delta w_0(n)\} = E\{\eta [x(n)y_0(n) - y_0(n)y_0(n)w_0(n)]\} \quad (149)$$

Considerando que  $\eta$  é uma entidade determinística e levando a Equação (135) à Equação (149),

$$E\{\Delta w_0(n)\} = \eta E\{x(n)x^T(n)w_0(n) - w_0^T(n)x(n)x^T(n)w_0(n)w_0(n)\} \quad (150)$$

Considerando que os vetores  $x$  e  $w$  são estatisticamente independentes e que, no equilíbrio,

$$E\{\Delta w_0(n)\}=0 \quad (151)$$

$$E\{w_0(n)\}=q_0 \quad (152)$$

$$E\{x(n)x^T(n)\}=C_x \quad (153)$$

A Equação (150) se torna,

$$0=C_x q_0-(q_0^T C_x q_0)q_0 \quad (154)$$

Assim, um vetor no equilíbrio deve obedecer a

$$C_x q_0=(q_0^T C_x q_0)q_0 \quad (155)$$

Conforme visto no Capítulo 2, na descrição das propriedades da análise dos componentes principais, Equações (99) e (100),

$$q_0^T C_x q_0=\lambda_0 \quad (156)$$

e a Equação (155) pode ser reescrita como,

$$C_x q_0=\lambda_0 q_0 \quad (157)$$

A análise da Equação (157) mostra que, no equilíbrio,  $w_0(n)$  deve ser um auto-vetor  $q_0$  de  $C_x$ .

Pré-multiplicando ambos os lados da igualdade da Equação (157) por  $q_0^T$  pode-se escrever que,

$$q_0^T C_x q_0=q_0^T \lambda_0 q_0 \quad (158)$$

À esquerda do sinal de igualdade vê-se claramente a Equação (156).

Além disto, como à direita da igualdade  $\lambda_0$  é um número e

$$q_0^T q_0=\|q_0\| \quad (159)$$

onde  $\|q_0\|$  é a norma Euclidiana do vetor  $q_0$ , pode-se afirmar que

$$\lambda_0 = \lambda_0 q_0^T q_0 = \lambda_0 \|q_0\| \quad (160)$$

de onde conclui-se que o valor convergido  $q_0$  do vetor peso sináptico  $w_0$  tem norma unitária, conforme pretendíamos mostrar.

Há  $p$  possíveis auto-vetores normalizados da matriz  $C_x$  que satisfazem à condição da Equação (157), no entanto, apenas o auto-vetor  $q_0$ , correspondente ao maior auto-valor  $\lambda_0 = \lambda_{max}$  representa uma solução estável.

Para demonstrar que a solução representada pelo auto-vetor associado ao maior auto-valor é a solução estável, considere-se que, após um grande número de iterações  $n$  o peso sináptico  $w_0(n)$  esteja na vizinhança de um vetor  $q_j$  tal que,

$$w_0(n) = q_j(n) + \varepsilon, \quad n \rightarrow \infty \quad (161)$$

onde  $\varepsilon$  é um vetor pequeno que representa a distância do peso sináptico ao auto-vetor.

Partindo da Equação (150) onde tomou-se o valor esperado da variação dos pesos sinápticos e considerando as condições expressas pelas Equações (151), (152), (153) e (154), encontra-se

$$E\{\Delta w_0(n)\} = \eta \{C_x(q_j + \varepsilon) - (q_j + \varepsilon)^T C_x(q_j + \varepsilon)(q_j + \varepsilon)\}, \quad n \rightarrow \infty \quad (162)$$

Efetuada as operações à direita da igualdade, na Equação (162), considerando a condição de simetria da matriz Hermitiana  $C_x$  e desprezando os termos de segunda ordem e de ordens maiores em  $\varepsilon$ , é possível aproximar o valor esperado tomado em (162) para primeira ordem em  $\varepsilon$  por

$$E\{\Delta w_0(n)\} \cong \eta \{C_x q_j + C_x \varepsilon - q_j^T C_x q_j q_j - 2\varepsilon^T C_x q_j q_j - q_j^T C_x q_j \varepsilon\}, \quad n \rightarrow \infty \quad (163)$$

Uma vez que  $q_j$  é um auto-vetor normalizado da matriz de correlação  $C_x$ , pode-se escrever que

$$C_x q_j = \lambda_j q_j, \quad j = 1, 2, \dots, p-1 \quad (164)$$

e

$$q_i^T q_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (165)$$

a Equação (163) pode ser reescrita na forma simplificada

$$E\{\Delta w_0(n)\} \cong \eta \{C_x \varepsilon - 2 \lambda_j \varepsilon^T q_j q_j - \lambda_j \varepsilon\} \quad (166)$$

ou,

$$E\{\Delta w_0(n)\} \cong \eta \{C_x \varepsilon - 2 \lambda_j q_j q_j^T \varepsilon - \lambda_j \varepsilon\} \quad (167)$$

onde considerou-se que  $\varepsilon^T q_j$  é um escalar e que este produto interno também pode ser expresso como  $q_j^T \varepsilon$ .

A mudança média no vetor peso sináptico projetada sobre a coordenada representada por outro auto-vetor  $q_i$  da matriz  $C_x$  é obtida pré-multiplicando  $E\{\Delta w_0(n)\}$  por  $q_i^T$ . Desta forma,

$$q_i^T E\{\Delta w_0(n)\} \cong \eta \{q_i^T C_x \varepsilon - 2 \lambda_j q_i^T q_j q_j^T \varepsilon - \lambda_j q_i^T \varepsilon\} \quad (168)$$

ou, considerando as Equações (164) e (165) e a propriedade de simetria de  $C_x$ ,

$$q_i^T E\{\Delta w_0(n)\} \cong \begin{cases} -2 \eta \lambda_i q_i^T \varepsilon, & i=j \\ \eta (\lambda_i - \lambda_j) q_i^T \varepsilon, & i \neq j \end{cases} \quad (169)$$

A análise da Equação (169) leva a concluir que,

- para  $i = j$ , A projeção do valor esperado da variação do peso sináptico na direção do próprio auto-vetor associado é sempre negativa. Portanto, como a variação do peso sináptico é sempre contrária à distância  $\varepsilon$  a solução é estável em todas as possíveis direções.

- para  $i \neq j$ , A projeção do valor esperado da variação do peso sináptico associado ao auto-vetor  $j$  na direção de um outro auto-vetor  $i$  qualquer somente é negativa se  $\lambda_j > \lambda_i$ . Portanto, a variação do peso sináptico será contrária à distância  $\varepsilon$ , indicando uma solução estável, se e somente se o auto-valor associado ao auto-vetor  $j$  for maior do que todos os demais auto-valores correspondentes a outras direções.

O problema da instabilidade do algoritmo pode ser equacionado tomando-se a derivada da variação do peso sináptico  $\Delta w_0$  em relação ao próprio peso sináptico  $w_0$ . Assim,

$$\frac{d}{d(w_0(n))}(\Delta w_0(n)) = \frac{d}{d(w_0(n))}(\eta y_0(n)(x(n) - y_0(n)w_0(n))) \quad (170)$$

Considerando as formas expandidas na Equação (170),

$$\begin{aligned} \frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) &= \\ &= \frac{d}{d(w_{0i}(n))} \left\{ \eta \left[ \sum_i x_i(n)w_{0i}(n) \right] \left[ x_i(n) - w_{0i}(n) \sum_i x_i(n)w_{0i}(n) \right] \right\} \end{aligned} \quad (171)$$

Tomando a derivada,

$$\begin{aligned} \frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) &= \\ &= \left[ \eta \sum_i x_i(n)w_{0i}(n) \right] \left[ -w_{0i}(n)x_i(n) - \sum_i x_i(n)w_{0i}(n) \right] + \\ &\quad + \left[ x_i(n) - w_{0i}(n) \sum_i x_i(n)w_{0i}(n) \right] \left[ \eta x_i(n) \right] \end{aligned} \quad (172)$$

Distribuindo os produtos mostrados em (172),

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) =$$

$$\begin{aligned}
&= [-\eta w_{0i}(n)x_i(n) \sum_i x_i(n)w_{0i}(n) - \eta \sum_i x_i(n)w_{0i}(n) \sum_i x_i(n)w_{0i}(n)] + \\
&\quad + [\eta x_i(n)x_i(n) - \eta x_i(n)w_{0i}(n) \sum_i x_i(n)w_{0i}(n)]
\end{aligned} \tag{173}$$

Assim, a derivada da variação do peso sináptico  $\Delta w_{0i}$  em relação ao próprio peso sináptico  $w_{0i}$  pode ser expressa por

$$\begin{aligned}
&\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) = \\
&= \eta x_i^2(n) - 2\eta x_i(n)w_{0i}(n) \sum_i x_i(n)w_{0i}(n) - \eta \left( \sum_i x_i(n)w_{0i}(n) \right)^2
\end{aligned} \tag{174}$$

A observação da Equação (174) não mais permite afirmar que a derivada será sempre positiva, como em (145). Portanto, a variação da variação do peso sináptico não é mais sempre positiva. A convergência do algoritmo ocorre se os valores de  $x(n)$  e  $w_0(n)$  são tais que a Equação (174) resulta em um valor negativo.

## 4.2 - O Algoritmo Hebbiano Generalizado

Na Seção anterior estabeleceu-se que um único neurônio linear submetido à regra Hebbiana de aprendizado auto-organizado tende a extrair a primeira componente principal de um vetor de dados de entrada estacionário [12], aquela componente correspondente ao maior auto-valor da matriz covariância dos dados de entrada.

A generalização do algoritmo ora apresentada segue a proposta de Sanger (1989) [4] e é obtida a partir do Algoritmo Hebbiano estabilizado pela Regra de Oja, discutido na Seção 4.1.

O Algoritmo Hebbiano Generalizado é usado para treinar uma Rede Neural Artificial progressiva, composta de uma camada de neurônios lineares. A RNA treinada por esta regra executará análise dos componentes principais de tamanho arbitrário sobre o vetor de entrada.

A Figura (4) apresenta o modelo da RNA a ser treinada pelo Algoritmo Hebbiano Generalizado.

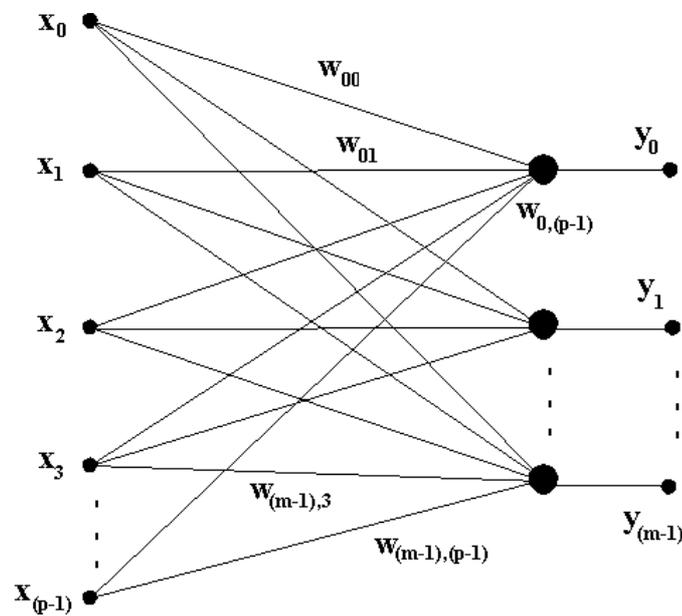


Figura 4 RNA progressiva com uma única camada de neurônios lineares, a ser treinada pelo Aprendizado Hebbiano Generalizado.

A RNA da Figura (4) tem  $p$  nós na camada de entrada e  $m$  neurônios na camada de saída, com  $m < p$ .

Os pesos sinápticos  $w_{ji}$  conectam os nós de entrada  $i$  aos neurônios da camada de saída  $j$ , com  $i = 0, 1, \dots, p-1$  e  $j = 0, 1, \dots, m-1$  (considerando novamente a notação expandida).

A saída  $y_j(n)$  produzida pelo neurônio  $j$  no tempo  $n$  em resposta ao conjunto de entradas  $x_i(n)$  é dada por

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji}(n)x_i(n), \quad j = 0, 1, \dots, m-1 \quad (175)$$

A adaptação do peso sináptico, devida ao Algoritmo Hebbiano Generalizado é expressa pela Equação (176),

$$\Delta w_{ji}(n) = \eta \{y_j(n)x_i(n) - y_j(n) \sum_{k=0}^j w_{ki}(n)y_k(n)\}, \quad \begin{array}{l} i = 0, 1, \dots, p-1 \\ j = 0, 1, \dots, m-1 \end{array} \quad (176)$$

onde  $\Delta w_{ji}(n)$  é a mudança aplicada ao peso sináptico  $w_{ji}(n)$  no tempo  $n$  e  $\eta$  é o parâmetro razão de aprendizado.

Ao observar a Equação (176) considerando  $j = 0$  percebe-se que o algoritmo para um simples neurônio, apresentado na Seção anterior, é um caso particular do Algoritmo Hebbiano Generalizado ora apresentado.

A equação que expressa a variação no peso sináptico imposta pelo Algoritmo Hebbiano Generalizado pode também ser escrita como

$$\Delta w_{ji}(n) = \eta y_j(n) x'_i(n) - \eta y_j^2(n) w_{ki}(n) \quad \begin{array}{l} i = 0, 1, \dots, p-1 \\ j = 0, 1, \dots, m-1 \end{array} \quad (177)$$

onde o vetor  $x'_i(n)$  representa a forma deflacionada do vetor de entrada  $x_i(n)$ , conforme Equação (178).

$$x'_i(n) = x_i(n) - \sum_{k=0}^{j-1} w_{ki}(n)y_k(n) \quad (178)$$

Cabe aqui examinar a operação do algoritmo passo a passo. Para o primeiro neurônio ( $j = 0$ ), a Equação (177) reduz-se ao caso de um único neurônio, examinado anteriormente, e a rede extrai o primeiro componente principal do vetor de dados de entrada  $x(n)$ .

Para o segundo neurônio ( $j = 1$ ), a Equação (178) torna-se

$$x'_i(n) = x_i(n) - w_{0i}(n)y_0(n) \quad (179)$$

Desde que o primeiro neurônio já tenha convergido para o primeiro componente principal, o segundo neurônio vê um vetor de entrada  $x'_i(n)$  do qual o primeiro auto-vetor da matriz  $C_x$  já foi extraído (deflacionado), conforme pode ser visto na Equação (176). O segundo neurônio extrairá, portanto, o primeiro componente principal de  $x'_i(n)$ , que é, na verdade, o segundo componente principal do vetor de entrada original,  $x_i(n)$  (segundo auto-valor e correspondente auto-vetor da matriz  $C_x$ ).

Repetindo este procedimento para os demais neurônios da rede, cada conjunto de pesos sinápticos convergido representa um auto-vetor da matriz de correlação do vetor de entrada. Os auto-vetores obtidos desta forma encontram-se ordenados em ordem decrescente do valor dos auto-valores associados, sendo o primeiro neurônio a convergir aquele associado ao maior auto-valor.

## Capítulo 5

### **O Algoritmo Hebbiano Generalizado Complexo para Extração dos Componentes Principais - *CGHAPCA* -**

Este capítulo descreve o Algoritmo Hebbiano Generalizado Complexo para Extração dos Componentes Principais - *CGHAPCA* - do conjunto de dados representativo do espectro de uma imagem.

Este novo método apresenta um algoritmo de aprendizado no domínio complexo, desenvolvido a partir do algoritmo Hebbiano generalizado, utilizado para treinar uma rede neural artificial complexa progressiva composta de uma única camada de neurônios complexos.

O algoritmo *CGHAPCA*, assim como os demais métodos existentes para extração dos componentes principais, aplica-se apropriadamente à compressão de dados pois concentra em um número reduzido de elementos a maior parte da energia (variância) dos dados, permitindo que a escolha dos componentes a serem descartados seja ótima no sentido do erro médio quadrático (*MSE*).

O *CGHAPCA* faz a extração dos componentes principais de uma imagem no domínio frequência, utilizando para isto uma RNA complexa e um algoritmo de aprendizado Hebbiano complexo.

A imagem digital é levada ao domínio complexo através da Transformada Rápida de Fourier Direta Bidimensional (*FFT Bidimensional*).

No domínio complexo, o espectro de Fourier da imagem é assimetrizado para ser transformado no conjunto de treino da RNA complexa. A assimetriação do

espectro de Fourier consiste na eliminação das componentes espectrais redundantes, presentes devido à propriedade de simetria espectral da Transformada de Fourier Bidimensional.

O espectro livre das informações redundantes é particionado em blocos (submatrizes). Estas submatrizes são transformadas em vetores linha. O conjunto de treino da RNA complexa é formado por estes vetores complexos resultantes do espectro de Fourier da imagem.

A rede neural complexa é treinada sob a regra complexa desenvolvida para o aprendizado Hebbiano generalizado, tendo por objetivo extrair os componentes principais do espectro da imagem.

A compressão obtida pelo método está relacionada ao número de componentes principais do espectro da imagem que são descartadas. Quanto mais componentes descartadas, maior será a compressão e menor a fidelidade da imagem reconstruída. Este compromisso é regido pelo tipo de aplicação a que se destina a compressão.

A RNA complexa terá tantos neurônios quantas componentes principais se desejar preservar. Após o treinamento, a rede complexa terá armazenado em seus pesos sinápticos os auto-vetores complexos correspondentes aos auto-valores reais da matriz covariância do conjunto de dados de entrada que se especificou preservar.

A convergência do algoritmo baseia-se em um critério de parada derivado das características esperadas dos vetores estimados obtidos após um número suficientemente grande de iterações.

Duas otimizações são propostas para o algoritmo, objetivando redução de tempo computacional. A primeira delas é o critério para atualização da razão de

aprendizado e a segunda, a aceleração de convergência do algoritmo por janelamento de treino.

A partir dos auto-vetores obtidos e das saídas dos neurônios após a convergência do algoritmo, para cada vetor pertencente ao conjunto de treino é obtida a reconstrução do espectro de Fourier, estimado a partir das componentes principais identificadas. Ao espectro de Fourier obtido são acrescentadas as componentes simétricas.

A Transformada de Fourier Inversa Bidimensional (*IFFT*) é aplicada para devolver os dados estimados pelo algoritmo ao domínio espaço.

A fidelidade da imagem reconstruída pelo algoritmo é determinada pelo parâmetro Relação Sinal-Ruído de Pico (*PSNR*).

## 5.1 A Representação dos Dados

O método se aplica a qualquer conjunto de dados de natureza complexa.

Quando aplicado à compressão de imagens digitais, o conjunto de dados a ser apresentado à RNA complexa é composto de vetores obtidos do espectro de Fourier da imagem. O espectro de Fourier é resultante da aplicação da Transformada de Fourier Bidimensional à imagem no domínio espaço. Neste trabalho o algoritmo utilizado para a implementação da Transformada de Fourier Bidimensional é o *Fast Fourier Transform* unidimensional desenvolvido por Cooley e Tukey [16], adaptado para duas dimensões. O algoritmo utilizado para a *FFT-2D* segue o trabalho de De Castro [3].

Seja uma imagem digital, bidimensional, representada por uma matriz  $f(x,y)$  de  $N \times N$  posições, sendo  $N$  uma potência inteira de dois. Neste trabalho é

assumido que as imagens no domínio espaço são quantizadas em 256 níveis de cinza. Antes de qualquer processamento os valores de pixel da imagem são normalizados para que o maior valor de pixel seja unitário. É então gerado o espectro de Fourier desta imagem,  $F(u, v)$ , que é uma matriz de  $N \times N$  elementos de natureza complexa.

Os dados são apresentados à RNA sob a forma de vetores unidimensionais de  $p$  elementos. Cada um destes vetores é obtido do particionamento do espectro de Fourier da imagem em submatrizes (ou blocos) não sobrepostas, de dimensões  $l \times l$ , onde  $l = 2^n$  elementos ( $n$  é uma potência inteira de dois e é determinado de acordo com as características estatísticas da imagem). O número de elementos dos vetores unidimensionais é, portanto,  $p = l^2$ . As imagens utilizadas neste trabalho possuem  $128 \times 128$  elementos.

Pixels vizinhos, em imagens no domínio espaço, são em geral correlacionados. À medida que as distâncias entre pixels aumentam, a correlação decai substancialmente. No domínio freqüência estas imagens são ainda mais correlacionadas, porque os respectivos espectros de Fourier são compostos de inúmeras funções “sinc” combinadas ( $sinc(x) = \sin(x)/x$ ) [13][14][15][16].

O tamanho escolhido para as submatrizes deve ser tal que permita captar elementos suficientemente correlacionados. Sendo originários de pixels altamente correlacionados, os vetores gerados das submatrizes, quando apresentados à RNA, possibilitam uma boa estimativa dos componentes principais do conjunto de dados representativo da imagem.

Uma submatriz de  $8 \times 8$  elementos foi determinada experimentalmente como sendo a que provê melhores resultados para imagens de  $128 \times 128$  pixels. Submatrizes maiores que  $8 \times 8$  implicam em uma maior diversidade de variações por

vetor de treino (menor correlação), tornando necessário aumentar o número de componentes principais utilizadas para que se obtenha a mesma *PSNR* (Capítulo 1, Definição 33). Submatrizes menores que  $8 \times 8$  pixels têm seus elementos altamente correlacionados, o que melhora a *PSNR*, no entanto, a taxa de compressão cai devido ao maior número de vetores utilizados para representar a imagem.

No particionamento do espectro de Fourier é considerada a propriedade da simetria espectral [16]. Devido à propriedade de simetria do espectro de uma imagem, as submatrizes são tomadas dos elementos não-redundantes do espectro e não de todos os elementos componentes do espectro. Esta operação reduz à metade o tamanho do conjunto de dados a ser tratado pela RNA, o que por si só já representa ganho em relação a trabalhar no domínio espaço.

Os elementos conjugados no espectro da imagem representam, para o treinamento da rede neural, um conjunto de valores redundantes. Estes valores redundantes podem ser descartados sem prejuízo para a estimação dos componentes principais, porque na reconstrução serão restituídos ao espectro.

Para uma imagem com espectro de Fourier composto por  $N \times N$  elementos, onde  $N = 2^n$ , com  $n$  podendo assumir qualquer valor inteiro, a simetria obedece à seguinte lei de formação:

- os elementos  $F(u,0)$  são conjugados dos elementos  $F(N-u,0)$ ;
- os elementos  $F(u,N/2)$  são conjugados dos elementos  $F(N-u,N/2)$ ;
- os elementos  $F(0,v)$  são conjugados dos elementos  $F(0,N-v)$ ;
- não possuem complexos conjugados os elementos  $F(0,0)$ ,  $F(N/2,0)$ ,  $F(0,N/2)$  e  $F(N/2,N/2)$ ;
- os demais elementos de  $F(u,v)$  são conjugados dos elementos  $F(N-u,N-v)$ .

Após descartar os elementos conjugados do espectro, a dimensão da matriz  $F(u,v)$  reduz-se a  $(N^2/2) + 2$  elementos. Estes 2 elementos são os elementos  $F(N/2,0)$  e  $F(N/2,N/2)$ .

Para que todos os blocos possuam a mesma dimensão para fins do particionamento da matriz, estes dois elementos são passados diretamente para o conjunto estimado. Os  $N/2$  primeiros elementos da coluna  $N/2$  são deslocados para as posições dos  $N/2$  últimos elementos da coluna zero.

Denomina-se a nova matriz assim formada de  $F_p(u,v)$ . A matriz  $F_p(u,v)$  possui  $N^2/2$  elementos e será particionada para a determinação das submatrizes que formarão o conjunto de treino da RNA complexa.

A operação de assimetria reduz à metade o número de vetores componentes de uma época, diminuindo o tempo computacional necessário para a convergência do algoritmo, além de representar uma redução no número de elementos a serem armazenados para reconstrução da imagem.

Antes de efetuar a Transformada de Fourier Inversa para a reconstrução dos dados no domínio espaço, o espectro é reconstruído (simetrizado) a partir de seus componentes conjugados.

As dimensões da matriz  $F_p(u,v)$  são múltiplas inteiras das dimensões das submatrizes consideradas ( de dimensões  $l \times l$ ). Do particionamento resultam  $n_b = (1/2)(N/l)^2$  submatrizes.

O número de submatrizes resultantes do particionamento da matriz é o número de vetores que são utilizados para treino da RNA complexa. A apresentação de

todo o conjunto de vetores à RNA uma vez é chamada uma época. Neste trabalho,  $N = 128$  e  $l = 8$ . Portanto, o conjunto de treino apresentado à RNA é composto de

$$\frac{1}{2} \left( \frac{N}{l} \right)^2 = 128 \text{ vetores de } l^2 = 64 \text{ elementos.} \quad (180)$$

As submatrizes são transformadas em vetores linha. Cada vetor linha é obtido de acordo com a seguinte lei de formação :

- Os  $l$  primeiros elementos deste vetor linha são os pertencentes à primeira linha da submatriz de  $l \times l$  elementos, lida da esquerda para a direita.
- Os próximos  $l$  elementos deste vetor linha são os pertencentes à segunda linha da submatriz, lida da mesma forma.
- Prossegue-se desta maneira até que sejam esgotados todos os  $l^2$  elementos da submatriz.

Uma exigência para a análise dos componentes principais é que os vetores envolvidos possuam média zero (Capítulo 2). Assim, obtém-se o vetor média espectral, resultante da média dos  $n_b = (1/2)(N/l)^2$  vetores e subtrai-se este vetor de cada vetor do conjunto, assim constituindo o conjunto de treino da RNA.

Cada vetor apresentado à RNA constitui uma amostra do conjunto de treino, composto de 128 vetores de 64 elementos. A cada época, a ordem de apresentação dos vetores do conjunto de treino é mudada aleatoriamente, de forma semelhante ao ato de embaralhar um conjunto de cartas.

Os vetores de dados são apresentados à RNA em ordem aleatória, para impedir que a apresentação ordenada destes vetores gere um ciclo de aumento e decréscimo de valores dos pesos sinápticos. Este ciclo, gerado pela ordem de apresentação dos vetores de dados, poderá interferir no aprendizado da RNA,

dificultando a convergência dos pesos sinápticos para os auto-vetores da matriz covariância dos dados.

O embaralhamento é obtido por um algoritmo que escolhe aleatoriamente dois vetores do conjunto, permutando-os entre si. Esta operação é executada ao final de cada época tantas vezes quantos forem o número de vetores de dados. A função densidade de probabilidade [12] do gerador de números aleatórios utilizado é uniforme.

## 5.2 A Estrutura da Rede Neural Artificial

A Figura (5) apresenta o modelo de RNA que é treinado pelo algoritmo ora proposto.

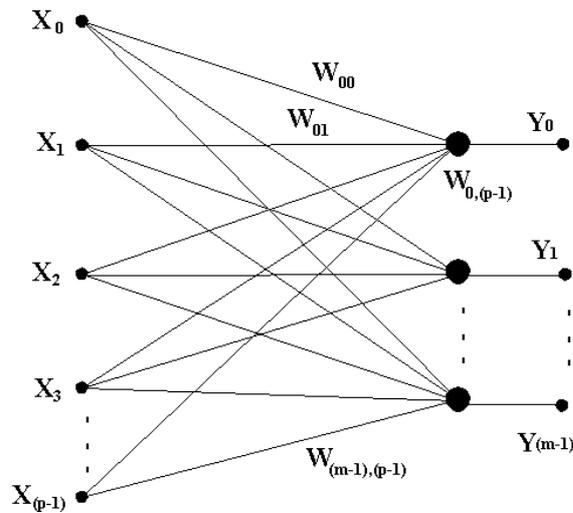


Figura 5 Modelo de rede neural artificial utilizado para treinamento via algoritmo *CGHAPCA*.

De acordo com a Figura (5), a camada de entrada da RNA possui  $p$  nós. Os  $p$  nós da camada de entrada da RNA recebem os respectivos elementos dos vetores de dados pertencentes ao conjunto de treino obtido do particionamento da matriz

representativa do espectro da imagem. Os vetores são compostos por  $p = l^2$  elementos, definindo o número de nós computacionais da camada de entrada da RNA.

Cada vetor representativo dos dados de entrada é da forma

$$\begin{aligned} X &= [X_0, X_1, \dots, X_{p-1}]^T, \\ X &= [x_{0_R} + jx_{0_I}, x_{1_R} + jx_{1_I}, \dots, x_{p-1_R} + jx_{p-1_I}]^T \end{aligned} \quad (181)$$

A camada de saída é formada por  $m$  neurônios, representados por

$$\begin{aligned} Y &= [Y_0, Y_1, \dots, Y_{m-1}]^T, \\ Y &= [y_{0_R} + jy_{0_I}, y_{1_R} + jy_{1_I}, \dots, y_{(m-1)_R} + jy_{(m-1)_I}]^T \end{aligned} \quad (182)$$

A cada neurônio de saída está associado, após a convergência do algoritmo, o conjunto de pesos sinápticos representativo do auto-vetor associado àquele componente principal. O número de neurônios da camada de saída é definido pelo número de componentes principais que se deseja extrair do conjunto de dados representativo do espectro da imagem original (associados aos auto-valores e auto-vetores da matriz covariância).

Quanto menor o número de componentes principais considerados, maior a taxa de compressão e menor a *PSNR* para uma mesma imagem.

Os pesos sinápticos, que conectam os nós fonte  $i$  aos neurônios  $j$  da camada de saída são representados por

$$W_{ji} = w_{ji_R} + jw_{ji_I}, \quad \begin{aligned} i &= 0, 1, \dots, p-1 \\ j &= 0, 1, \dots, m-1 \end{aligned} \quad (183)$$

Após a convergência do algoritmo, a RNA terá “cristalizado” em seus pesos sinápticos ( $W_{ji}$ ) os auto-vetores ( $Q_{ji}$ ) associados aos auto-valores ( $\lambda_j$ ) da matriz covariância dos dados de entrada.

O auto-vetor  $Q_{0i}$  associado ao maior auto-valor terá seus elementos “cristalizados” nos pesos sinápticos  $W_{0i}$ ,  $i = 0, 1, \dots, p-1$ , associados ao neurônio  $Y_0$  (primeiro neurônio da camada de saída). O neurônio  $Y_1$  (segundo neurônio da camada de saída) conterá, em seus pesos sinápticos  $W_{1i}$ , os elementos do segundo auto-vetor  $Q_{1i}$ ,  $i = 0, 1, \dots, p-1$ , associado ao segundo auto-valor.

A título de exemplo, a Figura (6) apresenta a representação gráfica dos pesos sinápticos (auto-vetores) “cristalizados” nos neurônios da camada de saída da RNA, após a convergência do algoritmo, para o modelo particular de RNA utilizado neste estudo. A rede da figura possui  $p = l^2 = 64$  nós computacionais na camada de entrada e são considerados  $k$  auto-vetores ( $k < m$ ) para a reconstrução dos dados.

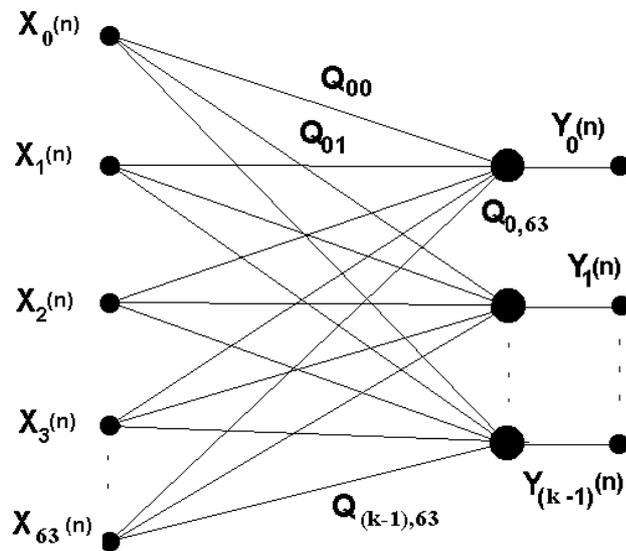


Figura 6 Representação gráfica da RNA complexa com  $p=64$  nós computacionais na camada de entrada e  $k$  neurônios na camada de saída, apresentando os auto-vetores da matriz covariância dos dados de entrada, após a convergência do algoritmo *CGHAPCA*.

### **5.3 A Regra Complexa Generalizada para Treinamento da RNA sob Aprendizado Hebbiano**

A regra complexa de aprendizado foi construída com base na proposta de Oja para um modelo composto de um único neurônio e na posterior generalização do modelo, desenvolvida por Sanger, para uma rede constituída de  $p$  nós de entrada e  $m$  neurônios de saída.

A matriz covariância de um conjunto de dados complexos é uma matriz Hermitiana, cujos auto-vetores são compostos de elementos complexos, conforme Definições 16 e 24, Capítulo 1. Portanto, os pesos sinápticos convergidos serão os auto-vetores complexos de uma matriz Hermitiana representativa da covariância dos vetores complexos de entrada.

Com o objetivo de desenvolver uma formulação complexa para o aprendizado Hebbiano, considerações foram feitas no sentido de preservar os princípios do aprendizado de Hebb para um modelo onde os sinais de entrada e de saída da RNA são complexos, assim como serão complexos os pesos sinápticos estabilizados, após a convergência do algoritmo.

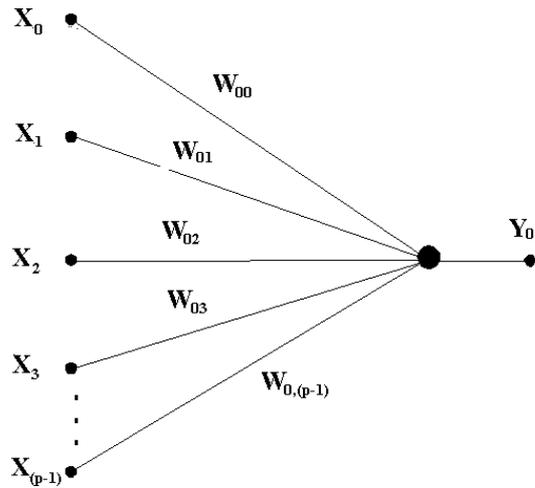


Figura 7 RNA de uma única unidade, a ser treinada pelo Aprendizado Hebbiano

Considere-se a rede de um único neurônio, mostrada na Figura (7), onde os sinais pré-sinápticos  $X(n)$ , pós-sinápticos  $Y_0(n)$  e os pesos sinápticos  $W_0(n)$  são complexos.

A saída  $Y_0(n)$  do neurônio da camada de saída da RNA para a iteração  $n$ , produzida em resposta ao conjunto de entradas  $X(n)$  é dada por

$$Y_0(n) = X(n)^T W_0(n)^* = W_0(n)^{*T} X(n) \quad (184)$$

De acordo com o postuldo de Hebb (Capítulo 4, Secção 4.1) quanto mais eficiente for um neurônio em disparar um outro neurônio, mais a intensidade da conexão entre eles (sinapse) será reforçada. Decorrente deste postuldo de aprendizado pode-se afirmar que o peso sináptico  $W_0(n)$  é aumentado quando os valores de  $X(n)$  e  $Y_0(n)$  são grandes. De uma forma mais genérica, pode-se dizer que o peso sináptico  $W_0(n)$  é modificado de acordo com a correlação entre os sinais pré e pós-sinápticos  $X(n)$  e  $Y_0(n)$ . Assim, a atualização do peso sináptico

$$\Delta W_0(n) = W_0(n+1) - W_0(n) \quad (185)$$

pode ser escrita como

$$\Delta W_0(n) = \eta Y_0(n)^* X(n) \quad (186)$$

onde  $\eta$  é uma constante positiva que determina a razão de aprendizado.

Fazendo uso da Equação (184), a Equação (186) pode ser expressa em função das entradas  $X(n)$  e dos pesos sinápticos  $W_0(n)$ .

$$\Delta W_0(n) = \eta X(n) X(n)^T W_0(n) \quad (187)$$

De acordo com a regra de Oja (Capítulo 4, Secção 4.1) o vetor dos pesos sinápticos irá convergir para o primeiro componente principal dos dados de entrada, após um número suficientemente grande de iterações (ou, quando  $n \rightarrow \infty$ ). No entanto, assim como no modelo de um único neurônio de Oja, para o modelo complexo a solução também não é uma solução estável.

Para investigar a estabilidade, conforme visto no Capítulo 4, considere-se que, após a convergência do algoritmo, o valor esperado da mudança dos pesos sinápticos será

$$E\{\Delta W_0(n)\} = E\{\eta X(n) X(n)^T W_0(n)\} \quad (188)$$

Sendo  $\eta$  um valor determinístico e os vetores  $X(n)$  e  $W_0(n)$  estatisticamente independentes,

$$E\{\Delta W_0(n)\} = \eta E\{X(n) X(n)^T\} E\{W_0(n)\} \quad (189)$$

No equilíbrio, a matriz covariância  $C_x$  é expressa pelo valor esperado do produto do vetor de entrada  $X(n)$  pelo seu conjugado transposto  $X(n)^{*T}$ , pois representa energia no domínio complexo (Capítulo 1, Definição 34),

$$E\{X(n) X(n)^{*T}\} = C_x, \quad (190)$$

Quando o número de iterações  $n$  se aproxima do infinito, o peso sináptico  $\Delta W_0(n)$  não deve mais variar e o vetor peso sináptico deve estabilizar em um vetor  $Q_0$  constante ( $W_0(n) \rightarrow Q_0$ ). Assim, quando  $n \rightarrow \infty$ ,

$$E \{ \Delta W_0(n) \} = 0 \quad (191)$$

e

$$E \{ W_0(n) \} = Q_0, \quad (192)$$

Nestas condições, a Equação (189) se torna

$$C_x Q_0 = 0 \quad (193)$$

E, da mesma forma que na regra de aprendizado de domínio real, no suposto ponto de equilíbrio, a solução da Equação (193) é um auto-vetor  $Q$  de  $C_x$ , cujo correspondente auto-valor é zero, o que novamente representa uma solução instável (conforme Capítulo 4, Seção 4.1).

Para evitar a divergência do algoritmo complexo de aprendizado Hebbiano é preciso restringir o crescimento do vetor peso sináptico, assim como no caso real.

A proposta de Oja (que corresponde a adicionar um decaimento ao peso sináptico proporcional a  $Y_0^2(n)$ ) é adotada como um dos elementos estabilizadores que permite a convergência do algoritmo.

A Equação (194) expressa a regra de Aprendizado Hebbiano após deflacionada de um fator  $\eta Y_0^2(n)W_0(n)$ .

$$\Delta W_0(n) = \eta Y_0(n)^* X(n) - \eta Y_0^2(n)W_0(n) \quad (194)$$

A regra complexa deflacionada pode ser reescrita como

$$\Delta W_0(n) = \eta \{ X(n)Y_0(n)^* - Y_0(n)Y_0(n)^* W_0(n) \} \quad (195)$$

Substituindo a Equação (184) na Equação (195),

$$\Delta W_0(n) = \eta \{X(n)X(n)^{T*} W_0(n) - W_0(n)^{T*} X(n)X(n)^{T*} W_0(n)\} \quad (196)$$

Tomando o valor esperado da mudança dos pesos sinápticos,

$$E\{\Delta W_0(n)\} = E\{\eta [X(n)X(n)^{T*} W_0(n) - W_0(n)^{T*} X(n)X(n)^{T*} W_0(n)]\} \quad (197)$$

Considerando que  $\eta$  é uma entidade determinística,

$$E\{\Delta W_0(n)\} = \eta E\{[X(n)X(n)^{T*} W_0(n) - W_0(n)^{T*} X(n)X(n)^{T*} W_0(n)]\} \quad (198)$$

Considerando que os vetores  $X(n)$  e  $W_0(n)$  são estatisticamente independentes e que, no equilíbrio,

$$E\{\Delta W_0(n)\} = 0, \quad (199)$$

$$E\{W_0(n)\} = Q_0, \quad (200)$$

e

$$E\{X(n)X(n)^{T*}\} = C_x \quad (201)$$

A Equação (199) se torna,

$$0 = C_x Q_0 - (Q_0^{*T} C_x Q_0) Q_0 \quad (202)$$

Assim, um vetor no equilíbrio deve obedecer a

$$C_x Q_0 = (Q_0^{*T} C_x Q_0) Q_0 \quad (203)$$

Conforme visto no Capítulo 2, na descrição das propriedades da análise dos componentes principais, Equações (99) e (100),

$$Q_0^{*T} C_x Q_0 = \lambda_0 \quad (204)$$

e a Equação (204) pode ser reescrita como,

$$C_x Q_0 = \lambda_0 Q_0 \quad (205)$$

A análise da Equação (205) mostra que, no equilíbrio,  $W_0(n)$  deve ser um auto-vetor  $Q_0$  de  $C_x$ , que é o resultado esperado para a regra complexa de aprendizado proposta, condizente ao obtido no Capítulo 4, Secção 4.1.

Como cada elemento de um auto-vetor complexo possui um grau de liberdade a mais que um auto-vetor real, constatou-se experimentalmente que a regra de estabilização de Oja garante que o conjunto de auto-vetores resultante seja estável, mas apenas ortogonal e não ortonormal.

Para garantir a ortonormalidade dos auto-vetores, este trabalho propõe uma condição adicional à regra de aprendizado complexa. Esta condição adicional normaliza para 1 o módulo do peso-sináptico associado ao maior auto-valor, a cada vetor de treino apresentado à RNA complexa. A normalização, executada somente nos pesos sinápticos do primeiro neurônio, a cada iteração de uma época, ao longo de todo o processo de treinamento é suficiente para tornar ortonormal o conjunto de auto-vetores resultante.

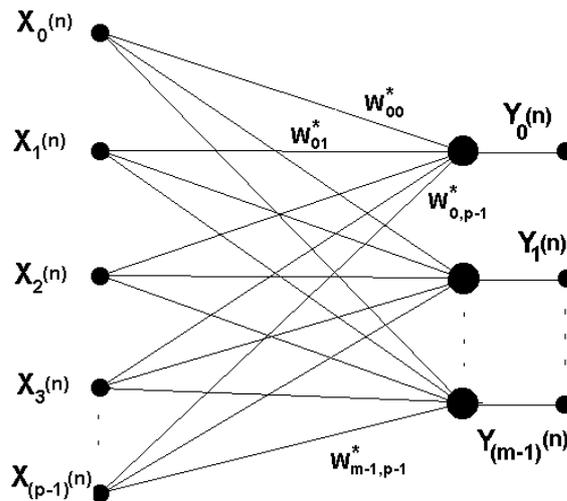


Figura 8 RNA complexa a ser treinada pelo algoritmo, com  $p$  nós na camada de entrada e  $m$  neurônios na camada de saída.

A Figura (8) apresenta o modelo de RNA complexa a ser treinado pelo algoritmo.

Partindo das Equações (184) e (194) e implementando a condição adicional necessária à ortonormalização, a regra complexa generalizada fica definida pelas Equações (206), (207) e (208), aqui apresentadas na forma expandida.

$$Y_j(n) = \sum_{i=0}^{p-1} W_{ji}(n)^* X_i(n), \quad j = 0, 1, \dots, m-1 \quad (206)$$

A atualização dos pesos sinápticos do primeiro neurônio, incluindo a normalização é dada por

$$W_{0i}(n+1) = \frac{W_{0i}(n) + \eta Y_0(n)^* [X_i(n) - W_{0i}(n) Y_0(n)]}{\|W_{0i}(n) + \eta Y_0(n)^* [X_i(n) - W_{0i}(n) Y_0(n)]\|}, \quad i = 0, 1, \dots, p-1 \quad (207)$$

Para os demais neurônios, a atualização dos pesos sinápticos segue

$$W_{ji}(n+1) = W_{ji}(n) + \eta Y_j(n)^* \{X_i(n) - \sum_{k=0}^j W_{ki}(n) Y_k(n)\}, \quad \begin{array}{l} i = 0, 1, \dots, p-1 \\ j = 1, 2, \dots, m-1 \end{array} \quad (208)$$

Sob esta regra de aprendizado, os  $m$  vetores peso sinápticos convergem para os  $m$  auto-vetores correspondentes aos  $m$  maiores auto-valores dos dados de entrada, quando o número de iterações  $n$  se aproxima do infinito.

#### 5.4 - O Critério de Parada

O critério de parada adotado para a convergência do algoritmo é derivado das características esperadas dos vetores estimados obtidos.

A convergência do algoritmo ocorre quando os pesos sinápticos armazenados na rede neural artificial complexa estabilizam em valores que correspondem aos auto-vetores associados aos auto-valores da matriz Covariância dos dados de entrada. Então, no equilíbrio, quando  $n \rightarrow \infty$ , é correto que

$$E \{ \Delta W_{ji}(n) \} = 0 \quad (209)$$

Baseado nesta propriedade, pode-se considerar que os pesos sinápticos do neurônio  $j$  tenham convergido para o auto-vetor associado quando as razões entre todos os seus  $p$  pesos sinápticos, de uma iteração para a próxima, sejam unitárias, isto é,  $(W_{ji}(n+1) / W_{ji}(n)) \approx 1, i = 0, 1, \dots, p-1$ .

Para evitar efeitos de variação transiente em  $W_{ji}(n)$  a cada iteração (devida ao uso eventual de uma alta razão de aprendizado) e no intuito de simplificar o critério de convergência, utilizou-se neste trabalho a média em três iterações da razão  $\|W_j(n+1)\| / \|W_j(n)\|$ , onde  $W_j$  representa o vetor de  $p$  pesos sinápticos do neurônio  $j$ . Especificamente, se

$$\frac{1}{3} \left\{ \frac{\|W_j(n+1)\|}{\|W_j(n)\|} + \frac{\|W_j(n+2)\|}{\|W_j(n+1)\|} + \frac{\|W_j(n+3)\|}{\|W_j(n+2)\|} \right\} - 1 < 1 \times 10^{-4} \quad (210)$$

considera-se que o vetor peso sináptico do neurônio  $j$  tenha convergido para o auto-vetor associado. O limiar de  $1 \times 10^{-4}$  foi determinado experimentalmente.

## 5.5 - O Critério para Atualização da Razão de Aprendizado

É conveniente, para a convergência do algoritmo, que a razão de aprendizado seja inicializada com um valor baixo ( da ordem de  $1 \times 10^{-9}$  por exemplo).

Após a convergência do primeiro auto-vetor, no entanto, a razão de aprendizado deve ser aumentada, pois, como já ocorreu a deflação da primeira componente principal do vetor de treino, o processo de aprendizado pode ser acelerado sem detrimento da convergência.

O critério adotado baseia-se na proposta de Chen e Chang [11]. A razão de aprendizado é feita, a cada época, igual ao inverso do auto-valor instantâneo dividido por uma constante arbitrária  $\alpha$  que varia com o tipo de imagem.

$$\eta = \frac{1/\lambda}{\alpha} \quad (211)$$

Experimentalmente, determinou-se que a constante  $\alpha$  deve estar contida no intervalo [500, 2000]. Um valor baixo para  $\alpha$  acelera a convergência do algoritmo. No entanto, valores inferiores a 500 podem conduzir a *overflow* das variáveis de ponto flutuante de 8 bytes utilizadas neste trabalho. Um valor muito alto para  $\alpha$  torna a razão de aprendizado demasiadamente pequena, retardando a convergência.

## 5.6 - A Aceleração de Convergência do Algoritmo por Janelamento de Treino

À medida que um neurônio converge para um auto-vetor, todos os subseqüentes ajustam-se correspondentemente, obedecendo ao princípio da deflação, conforme Equação (208).

No entanto, é característica do aprendizado Hebbiano (Capítulo 4, Secção 4.2) que cada um dos  $m$  auto-vetores inicie o próprio processo de convergência após a

convergência do auto-vetor anterior, não sendo alterados os pesos sinápticos dos neurônios já convergidos.

Observou-se experimentalmente que somente estão realmente próximos ao ponto de convergência alguns poucos neurônios imediatamente subseqüentes ao que já convergiu.

O processo de acomodação continuada de todos os neurônios subseqüentes a um neurônio que já convergiu implica em um custo computacional desnecessário, já que somente estarão realmente no caminho da convergência aqueles poucos neurônios imediatamente subseqüentes ao que já convergiu.

Para evitar que todos os  $m$  auto-vetores sejam ajustados durante a deflação de um particular auto-vetor, o *CGHAPCA* propõe o janelamento de convergência. O janelamento de convergência é um novo método que considera a convergência de  $k_{jt}$  ( $k_{jt} < m$ ) auto-vetores de cada vez. Havendo convergido o primeiro auto-vetor da janela de convergência, a janela se desloca uma posição para a frente.

Este processo de janelamento é aplicado ao algoritmo sem detrimento algum para a precisão do processo de convergência.

Determinou-se experimentalmente a conveniência de uso de uma janela de convergência que considera a convergência de três auto-vetores.

O ganho de tempo de processamento devido à aceleração de convergência do algoritmo por janelamento de treino, nos experimentos realizados neste trabalho (em que é considerada uma janela de convergência de três auto-vetores), é da ordem de

$$t_{cjt} \approx \frac{t_c}{10} \quad (212)$$

onde  $t_{cjt}$  denota o tempo de processamento com aceleração do algoritmo por janelamento de treino e  $t_c$  o tempo de processamento que considera o ajuste de todos os auto-vetores durante a convergência de um auto-vetor.

A relação entre os parâmetros  $t_{cjt}$  e  $t_c$  foi experimentalmente investigada neste estudo. Esta relação é evidentemente dependente do número total de auto-vetores que estão sendo considerados ( $m$ ), o que equivale a dizer que é dependente do número total de componentes principais que estão sendo considerados. Quanto maior for  $m$ , maior será o ganho de tempo computacional devido à aplicação do janelamento de convergência.

## 5.7 - Treinamento da RNA Complexa

A Figura (8), Secção (5.3), apresenta o modelo de RNA Complexa a ser treinado pelo algoritmo.

A cada época de treino são apresentados à RNA as  $n_b$  submatrizes (ou blocos) do espectro da imagem sob a forma de vetores de treino. Portanto, uma época é constituída de  $n_b$  iterações. A cada iteração  $n$ ,  $n = 0, 1, \dots, n_b - 1$ , é apresentado o  $n$ -ésimo vetor do conjunto de treino à RNA Complexa, produzindo a saída  $y(n)$ .

Após a convergência, a RNA apresenta um conjunto de pesos sinápticos constituído por  $m$  vetores de  $p$  elementos, equivalente aos auto-vetores associados aos auto-valores da matriz covariância dos dados de entrada, denotado por  $Q_{ji}$  ( $W_{ji}(\text{equilíbrio}) = Q_{ji}$ ).

A ordem de convergência dos pesos sinápticos é tal que, nos pesos sinápticos associados ao primeiro neurônio da camada de saída da RNA Complexa ( $W_{0i}(n)$ ) estão as componentes do auto-vetor ( $Q_{0i}$ ) associado ao maior auto-valor ( $\lambda_0$ ) da matriz covariância dos dados de entrada. Nos pesos sinápticos associados ao segundo neurônio da camada de saída da RNA Complexa ( $W_{1i}(n)$ ) estão as componentes do auto-vetor ( $Q_{1i}$ ) associado ao segundo maior auto-valor ( $\lambda_1$ ) da matriz covariância dos dados de entrada e, assim, em ordem decrescente até o último peso sináptico ( $\lambda_0 \Leftrightarrow Q_{0i}, \lambda_1 \Leftrightarrow Q_{1i}, \lambda_2 \Leftrightarrow Q_{2i}, \dots, \lambda_j \Leftrightarrow Q_{ji}$ ).

## 5.8 - A Reconstrução dos Dados

A Figura (9) apresenta uma RNA Complexa, já convergida para os auto-vetores  $Q$ , com  $p$  nós na camada de entrada e  $m$  neurônios na camada de saída. Está sendo apresentado à rede a  $n$ -ésima submatriz que compõe o espectro da imagem.

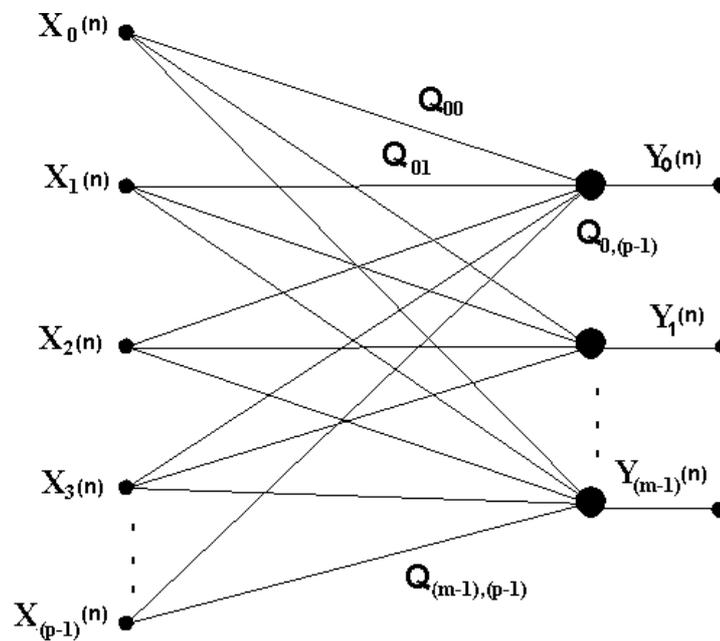


Figura 9 RNA Complexa com  $p$  nós na camada de entrada e  $m$  neurônios na camada de saída, apresentando os auto-vetores  $Q$  da matriz covariância dos dados de entrada, após a convergência do algoritmo *CGHAPCA*.

A reconstrução dos dados é obtida pela Equação (213).

$$\hat{X}_i(n) = \sum_{j=0}^{m-1} Y_j(n) Q_{ji}^* \quad (213)$$

onde, cada  $\hat{X}_i(n)$  representa a reconstrução do vetor  $n$  associado ao bloco ou submatriz de ordem  $n$  do espectro estimado de Fourier da imagem.

Por exemplo, para reconstrução da submatriz de ordem zero, as componentes do vetor  $\hat{X}_i(0)$  com  $i = 0, 1, \dots, p-1$  são reconstruídas por

$$\begin{aligned}\hat{X}_0(0) &= Y_0(0) Q_{00}^* + Y_1(0) Q_{10}^* + \dots + Y_{m-1}(0) Q_{(m-1)0}^* \\ \hat{X}_1(0) &= Y_0(0) Q_{01}^* + Y_1(0) Q_{11}^* + \dots + Y_{m-1}(0) Q_{(m-1)1}^* \\ &\dots \\ \hat{X}_{p-1}(0) &= Y_0(0) Q_{0(p-1)}^* + Y_1(0) Q_{1(p-1)}^* + \dots + Y_{m-1}(0) Q_{(m-1)(p-1)}^*\end{aligned}\quad (214)$$

Ao  $n$ -ésimo vetor reconstruído,  $\hat{X}_i(n)$ , com  $n = 0, 1, \dots, n_b-1$ , é acrescentado o vetor média espectral previamente extraído. O vetor resultante é transformado no bloco (ou submatriz) de ordem  $n$ , componente do espectro de Fourier estimado  $\hat{F}_p(u, v)$ .

O vetor linha  $\hat{X}_i(n)$  é transformado na  $n$ -ésima submatriz, de acordo com o seguinte algoritmo:

- A primeira linha da  $n$ -ésima submatriz (vista aqui como uma matriz de  $l \times l$  elementos) é formada pelos  $l$  primeiros elementos do  $n$ -ésimo vetor.
- A segunda linha da submatriz é formada pelos segundos  $l$  elementos do  $n$ -ésimo vetor.
- Procede-se desta forma até que todos os  $p = l^2$  elementos do  $n$ -ésimo vetor sejam transferidos para a  $n$ -ésima submatriz.

O conjunto de  $n_b$  submatrizes irá compor o espectro estimado parcial  $\hat{F}_p(u, v)$  da imagem (sem as componentes conjugadas), de tamanho  $N \times N/2$  elementos, sendo  $N \times N$  o tamanho da imagem original. Seja  $b_n(u, v)$  a  $n$ -ésima submatriz,  $n = 0, 1, \dots, n_b - 1$ . A Equação (215) apresenta a composição de  $\hat{F}_p(u, v)$  a partir dos  $n_b$  blocos (ou submatrizes).

$$\hat{F}_p(u, v) = \begin{bmatrix} b_0(u, v) & b_1(u, v) & \cdots & b_{\frac{N}{2l}-1}(u, v) \\ b_{\frac{N}{2l}}(u, v) & b_{\frac{N}{2l}+1}(u, v) & \cdots & b_{\frac{N}{l}-1}(u, v) \\ \vdots & & & \vdots \\ b_{n_b-\frac{N}{2l}}(u, v) & \cdots & \cdots & b_{n_b-1}(u, v) \end{bmatrix} \quad (215)$$

A matriz  $\hat{F}_p(u, v)$ , no entanto, contém apenas os elementos não-conjugados do espectro, devendo ser restituída sua simetria para que possa ser retornada ao domínio espaço pela Transformada Inversa de Fourier Bidimensional.

A matriz espectro de Fourier estimada total  $\hat{F}(u, v)$  é obtida restituindo as componentes conjugadas de  $\hat{F}_p(u, v)$  de acordo com as regras de simetria apresentadas na Secção 5.1. Retorna-se a imagem, então, ao domínio espaço através da Transformada Inversa de Fourier Bidimensional (*IFFT-2D*).

## 5.9 - A Compressão dos Dados

A compressão decorrente do uso do método é função do número de componentes principais do espectro da imagem que vierem a ser descartadas. Quanto maior o número de componentes principais descartadas, maior a compressão e menor a fidelidade da imagem reconstruída com relação à imagem original.

A compressão é definida a partir da equação de reconstrução dos dados (Equação 5.34). Para reconstrução completa dos dados, todos os  $m$  auto-vetores (associados aos  $m$  componentes principais) são considerados.

A compressão será obtida se, ao invés de  $m$  auto-vetores, forem considerados  $k$  auto-vetores, com  $k < m$ .

Para exemplificar, considere-se a Figura (10). Na rede mostrada na figura, as componentes do vetor  $\hat{X}_i(0)$  com  $i = 0, 1, \dots, p-1$  são reconstruídas por

$$\begin{aligned} \hat{X}_0(0) &= Y_0(0) Q_{00}^* + Y_1(0) Q_{10}^* + \dots + Y_{k-1}(0) Q_{(k-1)0}^* \\ \hat{X}_1(0) &= Y_0(0) Q_{01}^* + Y_1(0) Q_{11}^* + \dots + Y_{k-1}(0) Q_{(k-1)1}^* \\ &\dots \\ \hat{X}_{p-1}(0) &= Y_0(0) Q_{0(p-1)}^* + Y_1(0) Q_{1(p-1)}^* + \dots + Y_{k-1}(0) Q_{(k-1)(p-1)}^* \end{aligned} \quad (216)$$

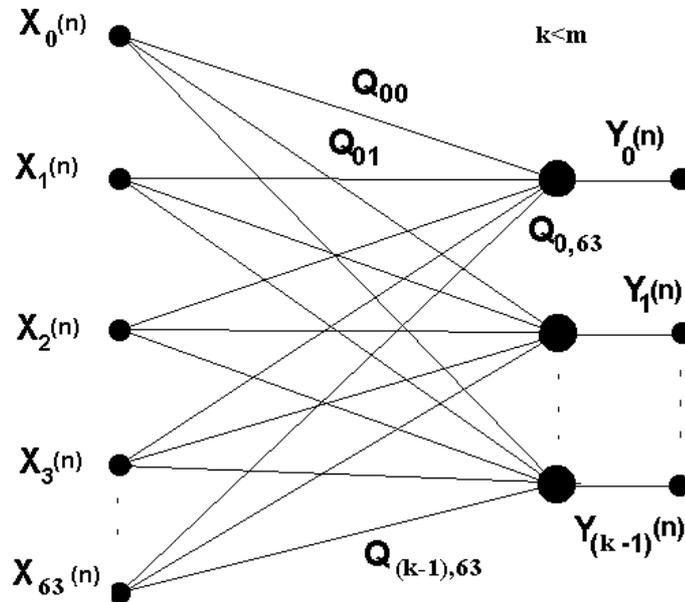


Figura 10 Modelo de RNA Complexa utilizada para compressão. A rede apresenta  $p=64$  nós de entrada (para submatrizes com  $p = l^2 = 64$  elementos) e  $k$  neurônios na camada de saída (para extrair  $k$  componentes principais).

Como  $k < m$ , as componentes do vetor  $\hat{X}_i(0)$ , com  $i = 0, 1, \dots, p-1$ , sofrem efetiva redução dimensional.

Os auto-vetores  $Q_{j i}$ , com  $j = k, k+1, \dots, m-1$  não considerados na reconstrução de  $\hat{X}_i(0)$  e responsáveis pela redução dimensional, não representam perda considerável de informação por estarem associados aos auto-valores de menor energia da matriz covariância do conjunto de dados.

Se a arquitetura da rede for tal que possua  $k$  neurônios na camada de saída, cada um com  $p = l^2$  sinapses, após a convergência do algoritmo a rede apresentará  $k$  vetores de  $p$  elementos (referentes aos auto-vetores convergidos) e  $n_b$  vetores de  $k$  elementos (referentes aos  $n_b = (1/2)(N/l)^2$  conjuntos de valores de saída dos  $k$  neurônios da camada de saída da rede). A partir destes elementos e do vetor média espectral previamente extraído (composto de  $p$  elementos complexos) será reconstruída a imagem comprimida.

A razão entre o número de unidades de armazenamento necessárias para representar a imagem original e o número de unidades de armazenamento necessárias para representar a imagem comprimida é chamada de razão de compressão e é denotada por  $\rho$ . A razão de compressão<sup>1</sup> pode, então, ser determinada por

$$\rho = \frac{2k(p + n_b) + 2p}{N \times N} \quad (217)$$

---

<sup>1</sup> Para efeito de avaliação do desempenho e da aplicabilidade do método de domínio complexo, foi desenvolvido um algoritmo para extração das componentes principais de um conjunto de dados no domínio real. O algoritmo de domínio real é chamado GHAPCA. Para a compressão no domínio real, via GHAPCA, a razão de compressão é  $\rho = \frac{k(p + n_b) + p}{N \times N}$ , onde  $n_b = (N/l)^2$ .

O método permite atingir uma compressão efetiva maior do que a compressão expressa pela razão de compressão  $\rho$ , através da adoção de um critério para alocação dos bits necessários para o armazenamento da imagem comprimida.

Neste trabalho a compressão é avaliada através da razão de compressão  $\rho$ . No entanto, são sugeridos dois critérios para otimizar a compressão baseados na alocação de diferentes números de bits por unidade de armazenamento da imagem comprimida. No primeiro critério a otimização é alcançada através da atribuição de um maior número de bits para o armazenamento dos auto-vetores mais significativos da imagem (associados aos maiores auto-valores) e de um menor número de bits para os auto-vetores menos significativos (associados aos menores auto-valores) [4]. A segunda sugestão é a utilização do Código Huffman que é baseado na entropia da imagem comprimida [13] [14].

Avalia-se a fidelidade dos resultados obtidos pelo *CGHAPCA*, através da determinação da Relação Sinal-Ruído de Pico (*PSNR*), apresentada no Capítulo 1, Definição 33, onde as diferenças entre a imagem comprimida e a imagem original são encaradas como se fossem ruído.

### **5.10 - Sumário do *CGHAPCA* Aplicado à Compressão de Imagens Digitais**

Os passos executados pelo algoritmo são:

1. A imagem no domínio espaço é normalizada para que o maior valor de pixel seja 1 e o menor valor de pixel seja zero.
2. A imagem é levada ao domínio frequência, determinando-se o respectivo espectro de Fourier via *FFT* Bidimensional Direta (*FFT*);
3. As regiões de elementos conjugados da matriz espectro de Fourier são

- eliminadas, considerando-se a simetria espectral;
4. Transforma-se o espectro de Fourier resultante da operação de eliminação de redundância em vetores unidimensionais;
  5. Determina-se o vetor média espectral do conjunto de vetores unidimensionais. Subtrai-se este vetor do conjunto de vetores, formando o conjunto de treino a ser apresentado à RNA Complexa;
  6. Inicializa-se os pesos sinápticos de maneira aleatória (distribuição de probabilidade uniforme) com valores pertencentes ao intervalo  $[0, a + jb]$ , tal que  $(a^2 + b^2) \approx (1/p)$ , sendo  $p$  o número de pesos sinápticos por neurônio (com isto obtém-se norma inicial unitária). Arbitra-se uma razão de aprendizado inicial (usualmente no intervalo  $[1 \times 10^{-9}, 5 \times 10^{-9}]$ ).
  7. Apresenta-se o conjunto de treino à RNA Complexa. A cada vetor apresentado, atualiza-se os pesos sinápticos de acordo com a regra de aprendizado do *CGHAPCA*. Procede-se desta maneira até que todos os vetores de treino sejam apresentados à RNA, o que constitui uma época de treino.
  8. Ao final de cada época embaralha-se o conjunto de vetores de treino de maneira aleatória (distribuição de probabilidade uniforme).
  9. Determina-se o auto-valor do neurônio que está convergindo associado ao maior auto-valor na janela de treino. Determina-se a nova razão de aprendizado proporcionalmente ao inverso do auto-valor (Equação 211). Atualiza-se a razão de aprendizado deste neurônio e dos próximos incluídos na janela de treino com a nova razão de aprendizado.

10. Os passos (7) a (9) são repetidos até a convergência do neurônio associado ao maior auto-valor na janela de treino, após o que a janela de treino é deslocada um neurônio para frente.
11. Repete-se os passos (7) a (10) até que todos os neurônios convirjam.
12. Reconstroi-se o espectro de Fourier da imagem a partir dos componentes estimados pela RNA Complexa (auto-vetores e saídas dos neurônios da RNA Complexa).
13. Soma-se a todos os vetores reconstruídos o vetor média espectral previamente subtraído.
14. Utiliza-se a propriedade da simetria espectral para recompor o espectro de Fourier a partir das componentes espectrais obtidas.
15. O espectro de Fourier obtido pelo algoritmo é devolvido ao domínio espaço, através da aplicação da *FFT* Bidimensional Inversa (*IFFT*).
16. A imagem no domínio espaço é desnormalizada para que o maior valor de pixel seja 255 e o menor valor de pixel seja zero.

## Capítulo 6

### Resultados Experimentais e Conclusões

Neste capítulo são apresentados e interpretados os resultados da aplicação do algoritmo proposto - *CGHAPCA* - à compressão de imagens digitais.

Para avaliação do desempenho e da aplicabilidade do método de domínio complexo é feita a comparação de imagens comprimidas no domínio real, via algoritmo *GHAPCA* (referido no Capítulo 5, Secção 5.9 e desenvolvido para este fim), com imagens comprimidas pelo algoritmo de domínio complexo, *CGHAPCA*.

As imagens são avaliadas quanto à fidelidade de reconstrução, relativa à imagem original. Os parâmetros de avaliação são a Relação Sinal-Ruído de Pico (*PSNR*, Capítulo 1, Definição 33) e a impressão subjetiva do sistema visual humano [13][14].

Todas as imagens possuem 256 níveis de cinza e dimensões  $128 \times 128$  pixels. Foram escolhidas imagens com diferentes estatísticas [6][13][14].

Constatou-se experimentalmente que as vantagens da análise dos componentes principais no domínio complexo estão estreitamente relacionadas com a maior razão de decréscimo dos auto-valores da imagem no domínio complexo, com relação à razão de decréscimo dos auto-valores da imagem no domínio real. Para

expressar a relação entre os auto-valores em ambos os domínios, foram estabelecidas relações analíticas entre os auto-valores das imagens (energias) nos domínios real e complexo.

A Energia de uma imagem conserva-se tanto no domínio espaço quanto no domínio freqüência. Segundo o Teorema de Parseval em Duas Dimensões (Capítulo 1, Definição 32),

$$E_{xy} = E_{uv} = E \quad (218)$$

onde  $E_{xy}$  denota a energia no domínio espaço e  $E_{uv}$  denota a energia no domínio freqüência .

Para uma imagem de tamanho  $N \times N$  pixels, constata-se que a relação de energia entre os auto-valores da imagem no domínio espaço e os auto-valores de seu espectro (aqui considerado ao todo, sem eliminação dos elementos simétricos conjugados), pode ser expressa por

$$\frac{E}{n_b \left( \sum \lambda_{xy} + \overline{B}_{xy} \right)} = \frac{N^2 E}{n_b \left( \sum \lambda_{uv} + \overline{B}_{uv} \right)} = 1 \quad (219)$$

onde

- $n_b$  é o número de submatrizes (blocos) de  $l \times l$  elementos em que são divididas tanto as matrizes representativas da imagem no domínio espaço como no domínio freqüência,
- $\sum \lambda_{xy}$  denota o somatório dos auto-valores obtidos no domínio espaço,
- $\sum \lambda_{uv}$  denota o somatório dos auto-valores obtidos no domínio freqüência,
- $\overline{B}_{xy}$  é a energia da submatriz média  $\overline{b}(x, y)$  no domínio espaço.

$\bar{B}_{xy}$  é definido por

$$\bar{B}_{xy} = \sum_{x=0}^{l-1} \sum_{y=0}^{l-1} (\bar{b}(x,y))^2 \quad (220)$$

onde  $\bar{b}(x,y)$  é expresso por

$$\bar{b}(x,y) = \frac{1}{n_b} \sum_{k=0}^{n_b-1} b_k(x,y) \quad (221)$$

e  $b_k(x,y)$  denota a  $k$ -ésima submatriz com  $l \times l$  elementos da imagem  $f(x,y)$ .

- $\bar{B}_{uv}$  é a energia da submatriz média  $\bar{b}(u,v)$  no domínio freqüência.

$\bar{B}_{uv}$  é definido por

$$\bar{B}_{uv} = \sum_{u=0}^{l-1} \sum_{v=0}^{l-1} (\bar{b}(u,v))^2 \quad (222)$$

onde  $\bar{b}(u,v)$  é expresso por

$$\bar{b}(u,v) = \frac{1}{n_b} \sum_{k=0}^{n_b-1} b_k(u,v) \quad (223)$$

e  $b_k(u,v)$  denota a  $k$ -ésima submatriz com  $l \times l$  elementos do espectro  $F(u,v)$  da imagem  $f(x,y)$ .

Verificou-se que o somatório dos auto-valores, seja no domínio espaço, seja no domínio freqüência, é igual à energia média das submatrizes  $b_k$  tendo sido subtraída a matriz média  $\bar{b}$ . Ou seja,

$$\sum \lambda_{xy} = \frac{1}{n_b} \sum_{k=0}^{n_b-1} \sum_{x=0}^{l-1} \sum_{y=0}^{l-1} (b_k(x,y) - \bar{b}(x,y))^2 \quad (224)$$

$$\sum \lambda_{uv} = \frac{1}{n_b} \sum_{k=0}^{n_b-1} \sum_{u=0}^{l-1} \sum_{v=0}^{l-1} (b_k(u,v) - \bar{b}(u,v))^2 \quad (225)$$

O primeiro grupo de figuras apresentado neste capítulo (Figuras (11) a (18)) é constituído dos resultados experimentais referentes à imagem original Brasil (Figura (11)), cujo histograma é mostrado na Figura (12).

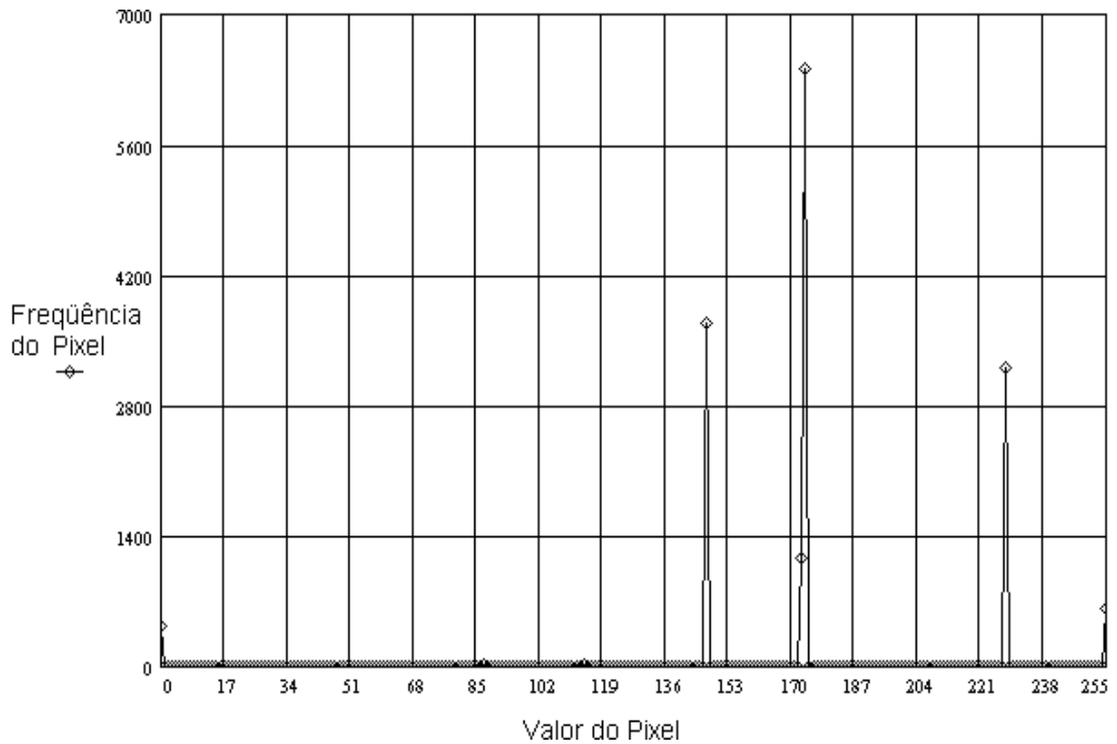
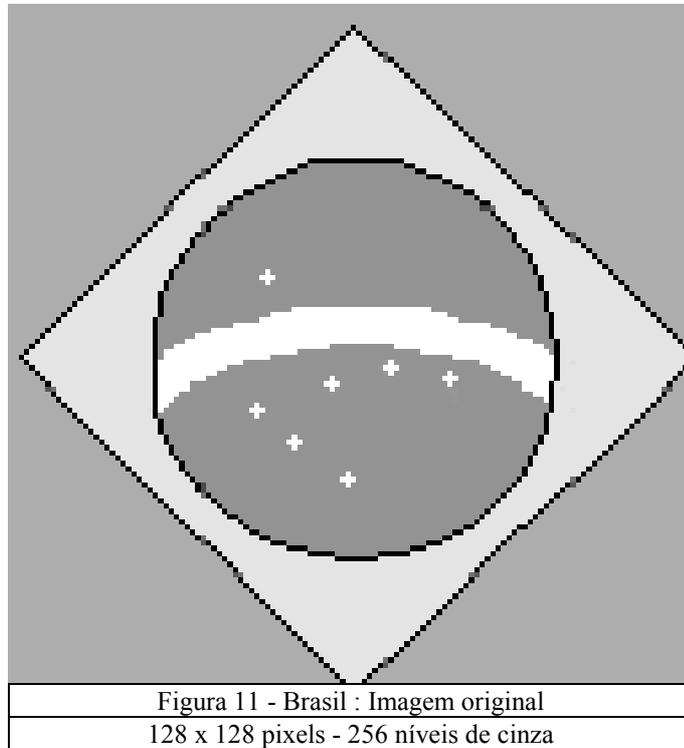


Figura 12 - Histograma da Imagem da Fig.11.

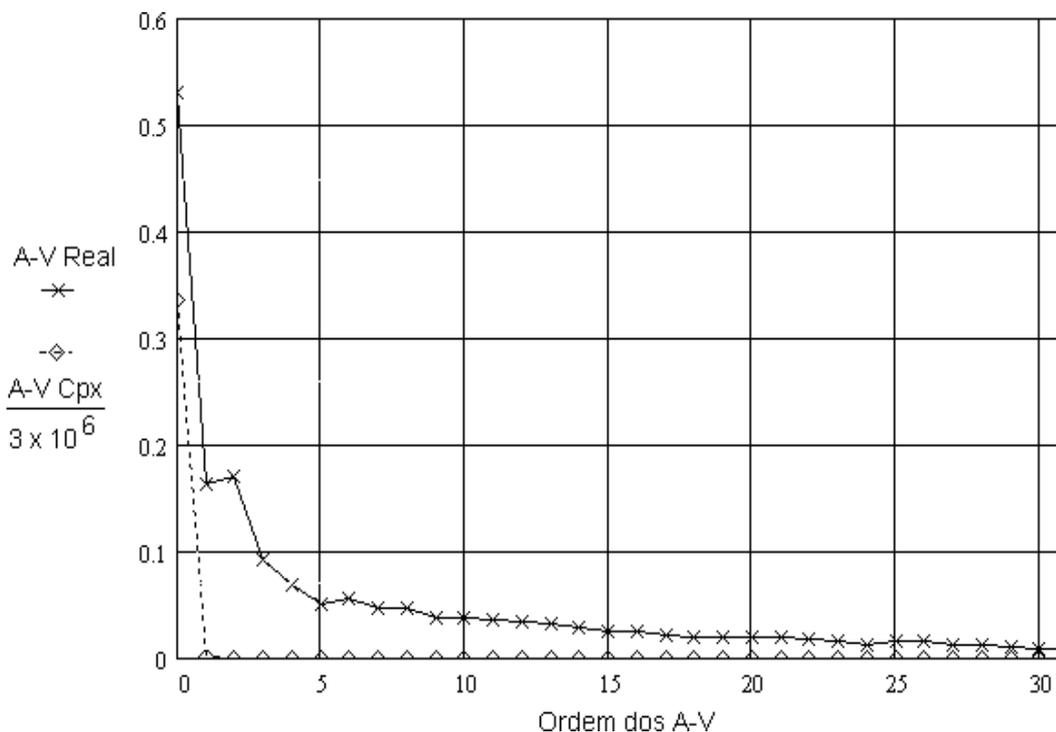


Figura 13 - Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.11

A Figura (13) traz o gráfico da distribuição de auto-valores para esta imagem, nos domínios real e complexo. A observação do comportamento das curvas de distribuição dos auto-valores para esta imagem permite constatar que a razão de decréscimo dos auto-valores do domínio complexo é maior que a razão de decréscimo dos auto-valores do domínio real. A observação das curvas de distribuição de auto-valores plotadas para as demais imagens que constituem o conjunto de exemplos apresentado neste capítulo permite a mesma conclusão (ver Figuras (22), (30), (38) e (46) constantes dos demais blocos de figuras deste capítulo).

Tanto maior será a razão de decréscimo dos auto-valores do domínio complexo em relação à razão de decréscimo dos auto-valores do domínio real, quanto

maior for a descontinuidade entre os valores de frequência absoluta de pixels no histograma da imagem. Tal tipo de histograma sugere variação abrupta de contraste dos elementos da imagem, no domínio espaço. O histograma da Figura (12) demonstra esta característica, presente na imagem original Brasil (Figura (11)).

Para imagens com esta característica de variação abrupta de contraste no domínio espaço, a *PSNR* da imagem comprimida no domínio complexo é maior que a *PSNR* da imagem comprimida no domínio real. Isto é explicado através da propriedade espaço-frequência que afirma que uma variação abrupta no domínio espaço implica em uma variação suave no domínio frequência [14]. Desta forma, a correlação entre os elementos das submatrizes no domínio frequência é maior do que a correlação entre os elementos das submatrizes no domínio real. A compressão no domínio complexo tende, em função disto, a preservar as características de alto contraste da imagem e apresentar maior erro relativo em regiões constantes no domínio espaço. A compressão no domínio real, por outro lado, tende a preservar as regiões constantes da imagem e apresentar maior erro relativo em regiões de alto contraste no domínio espaço.

A Figura (14) apresenta a reconstrução da imagem original da Figura (11) após compressão no domínio frequência, utilizando submatrizes de  $8 \times 8$  elementos e considerando 32 auto-vetores. A razão de compressão ( $\rho$ ) obtida apenas devido ao fato de preservar 32 auto-vetores e descartar os demais é da ordem de 76% e a Relação Sinal-Ruído de Pico da imagem comprimida foi determinada como sendo 34,13 dB. Nesta reconstrução, as características de alto contraste da imagem foram preservadas, como pode ser notado nos pequenos detalhes (estrelinhas brancas). Já as regiões sem

grandes variações de contraste apresentam maior erro de reconstituição, o que pode ser observado nas regiões constantes (retângulo, losango e círculo).

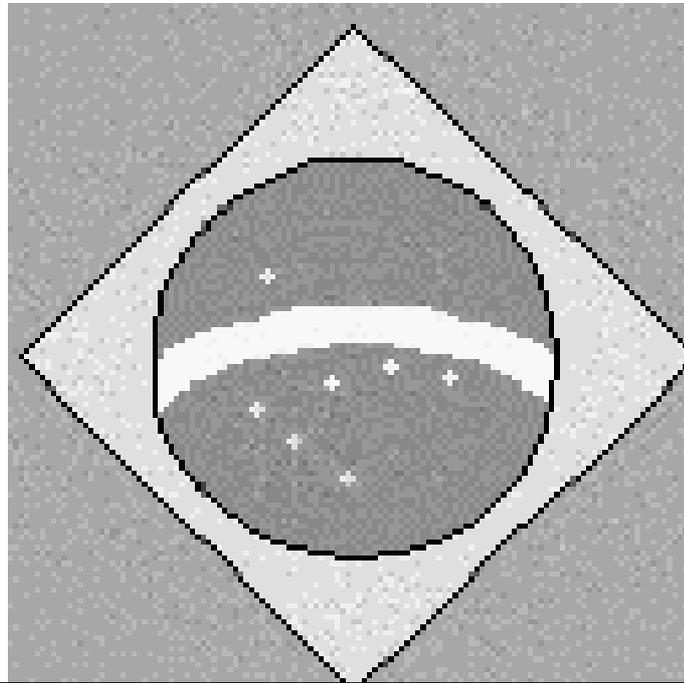


Figura 14 - Imagem da Fig.11 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.758	34.13 dB	32	8 x 8

Para enfatizar esta característica, observe-se também a Figura (15). Esta figura apresenta a reconstrução da imagem original da Figura (11) após compressão no domínio espaço, utilizando as mesmas dimensões para as submatrizes e considerando o mesmo número de auto-vetores utilizado na compressão realizada no domínio freqüência (ilustrada pela Figura (14)). A razão de compressão obtida é da ordem de 63% e a Relação Sinal-Ruído de Pico da imagem comprimida foi determinada como sendo 28.63 dB. Conforme comentado nos parágrafos anteriores, a imagem da Figura (15) apresenta maior erro de reconstituição nas regiões de alto contraste, o que pode ser observado na pouca definição obtida nas estrelinhas reconstruídas. Já as regiões

constantes da imagem (retângulo, losango e círculo) apresentam menor erro de reconstrução.

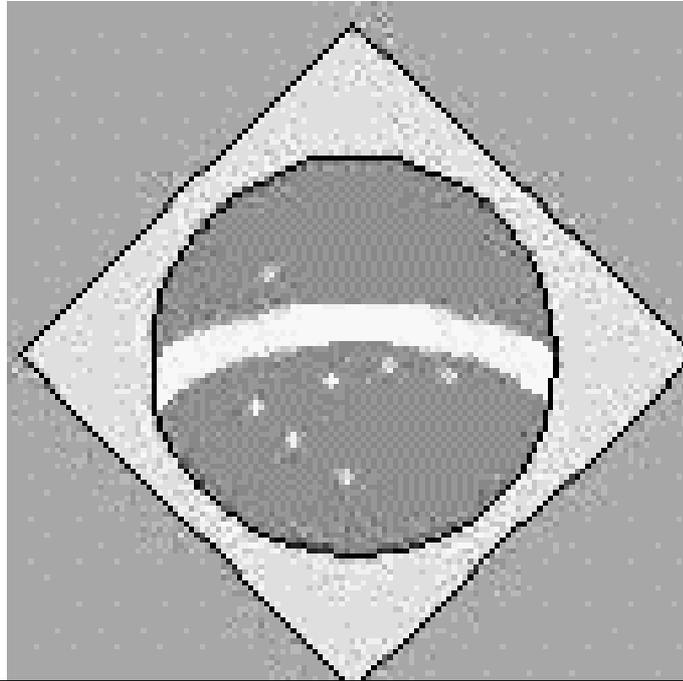


Figura 15 - Imagem da Fig.11 reconstruída após compressão no domínio espaço.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.629	28.63 dB	32	8 x 8

Considerando que o sistema visual humano apresenta sensibilidade ao contraste (*Mach Band Effect* [13]), a compressão no domínio complexo é superior à compressão no domínio real. É importante salientar que a *PSNR* não tem condições de avaliar o efeito que esta característica causa ao sistema visual humano. Há casos, entre as simulações realizadas, em que as *PSNR* são aproximadamente iguais nos domínios real e complexo e, no entanto, o efeito visual no domínio complexo é superior.

A Figura (16) mostra o histograma da imagem da Figura (14). Nota-se, da observação da figura, que houve um alargamento de freqüências absolutas em torno dos valores de pixel existentes na imagem original. Quanto maior for a fidelidade da

imagem obtida com relação à imagem original, mais aguda se tornará a curva, tendendo ao valor de frequência central que é o nível de cinza presente na imagem original.

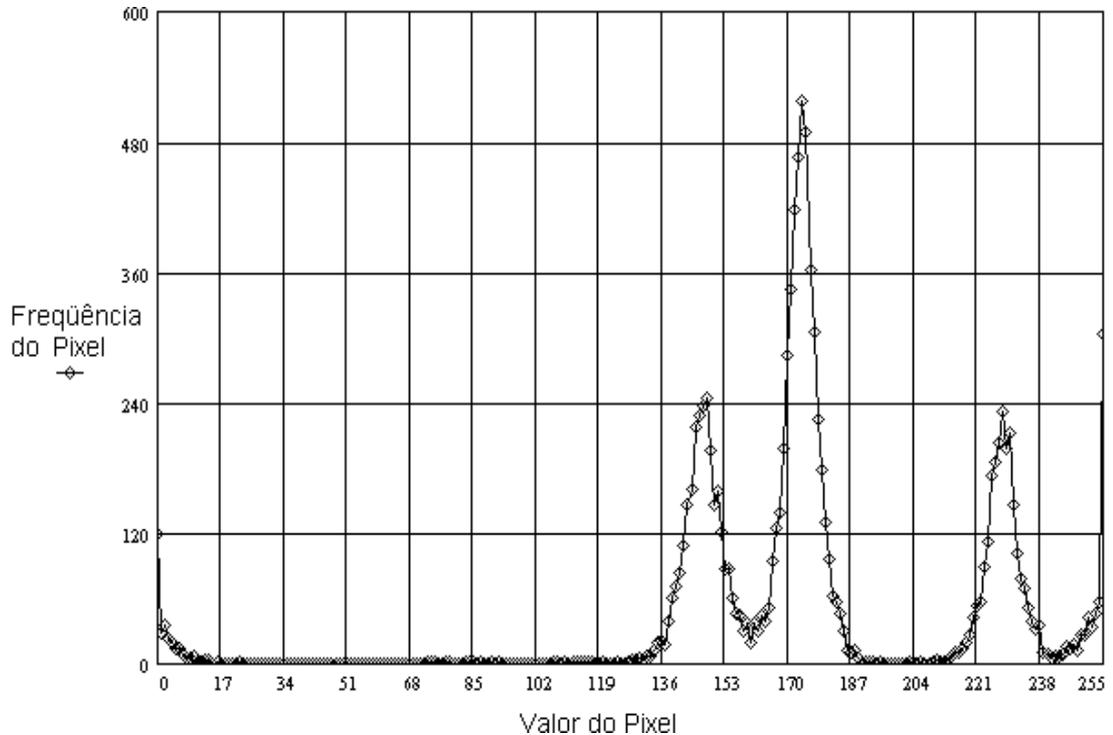


Figura 16 - Histograma da Imagem da Fig. 14  
(Imagem da Figura 11 reconstruída após compressão no domínio frequência).

As imagens das Figuras (15) e (17) apresentam entre si a mesma razão de compressão, assim como as imagens das Figuras (18) e (19). Para que a igualdade entre as razões de compressão fosse obtida, consideraram-se respectivamente, 32 e 27 auto-vetores para as imagens mostradas nas Figuras (15) e (17) e 13 e 16 auto-vetores para as imagens mostradas nas Figuras (18) e (19). Estas imagens foram colocadas no conjunto de resultados para que possam ser comparadas em termos de *PSNR* e qualidade visual subjetiva.

A Figura (17) apresenta a imagem da Figura (11) reconstruída após compressão no domínio frequência, utilizando submatrizes de  $8 \times 8$  elementos e tendo sido preservados 27 auto-vetores. Esta imagem é superior, tanto em *PSNR* quanto em

qualidade visual subjetiva à imagem da Figura (15), que apresenta a imagem da Figura (11) reconstruída após compressão no domínio real, utilizando a mesma dimensão para as submatrizes e tendo sido preservados 32 auto-vetores. O mesmo podendo ser observado com relação às imagens das Figuras (18) e (19).

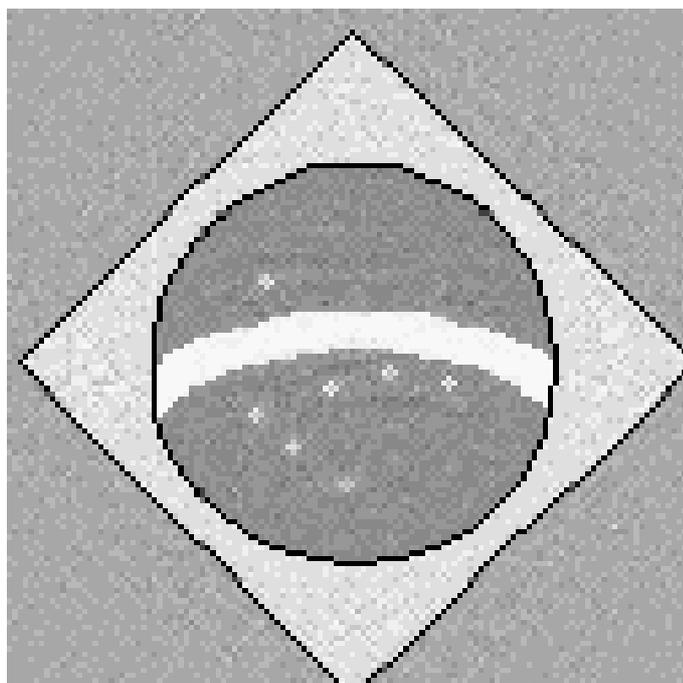


Figura 17 - Imagem da Fig.11 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.641	30.80 dB	27	8 x 8

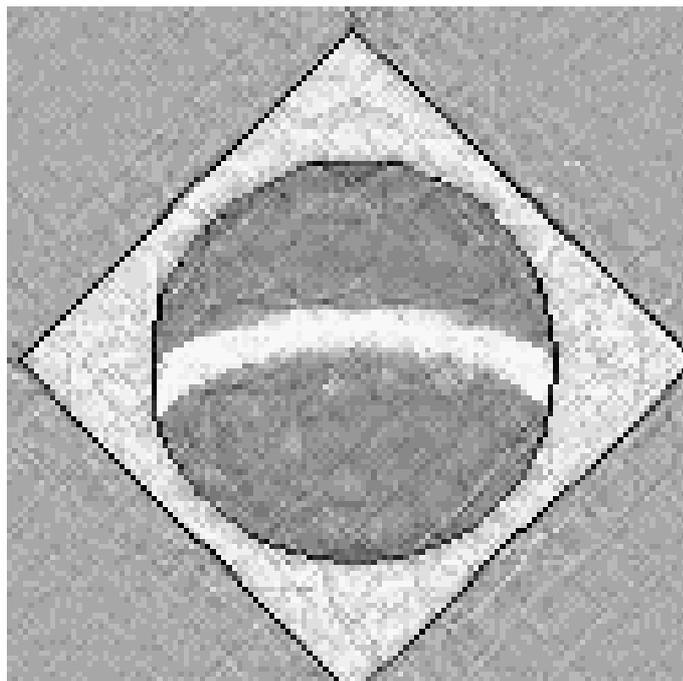


Figura 18 - Imagem da Fig.11 reconstruída após compressão no domínio freqüência.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.313	23.20 dB	13	8 x 8

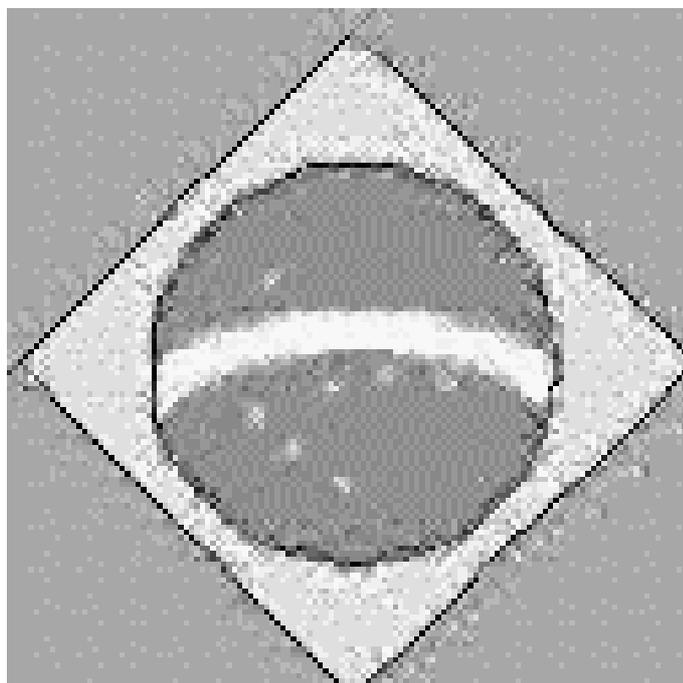


Figura 19 - Imagem da Fig.11 reconstruída após compressão no domínio espaço.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.316	22.71 dB	16	8 x 8

Os dois conjuntos de figuras apresentados a seguir são referentes a imagens cujos histogramas apresentam continuidade entre as frequências absolutas.

O primeiro conjunto refere-se à imagem original Lenna, apresentada na Figura (20) e cujo histograma é mostrado na Figura (21). O histograma da Figura (21) apresenta uma descontinuidade não tão expressiva entre os valores de frequência absoluta de pixels, o que sugere uma variação menor de contraste dos elementos da imagem no domínio espaço, se comparada à variação revelada pelo histograma da imagem Brasil (Figura (12)). Neste caso, a razão de decréscimo dos auto-valores do domínio complexo em relação à razão de decréscimo dos auto-valores do domínio real também será menos expressiva, como pode ser visto no gráfico de distribuição de auto-valores mostrado na Figura (22).



Figura 20 - Lenna : Imagem original
128 x 128 pixels - 256 níveis de cinza

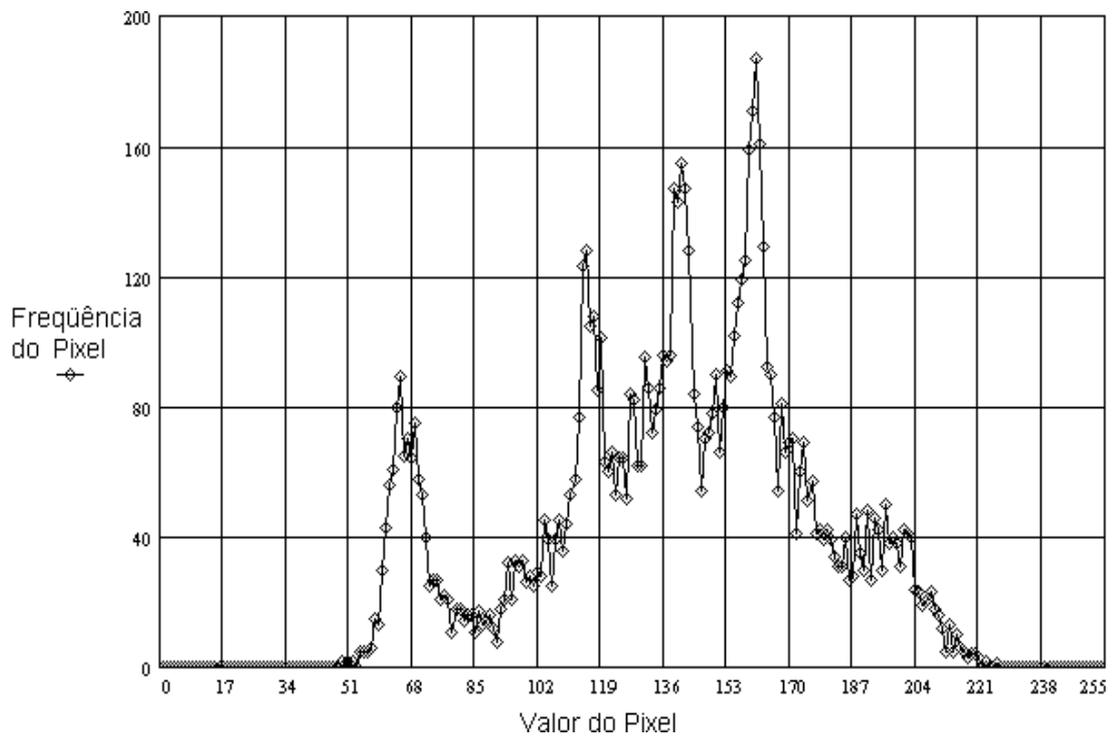


Figura 21 - Histograma da Imagem da Fig.20.

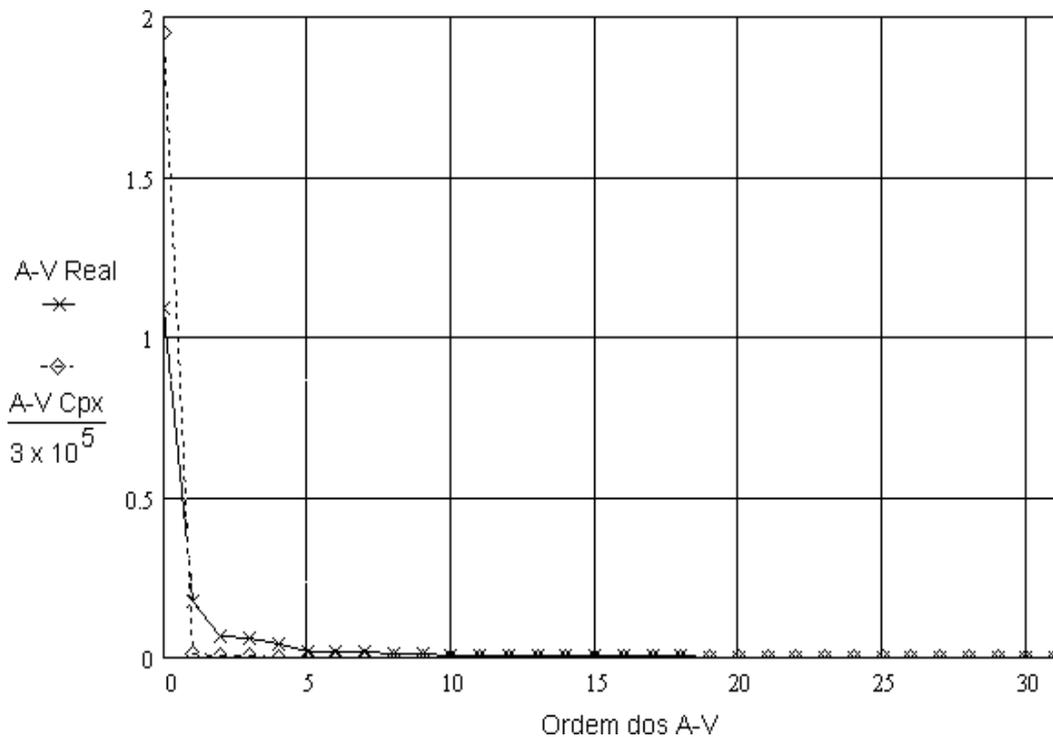


Figura 22 - Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.20

Para imagens com uma característica de variação de contraste menos abrupta no domínio espaço, a *PSNR* da imagem comprimida no domínio complexo é igual ou ligeiramente menor que a *PSNR* da imagem comprimida no domínio real, fato também explicado pela relação espaço-freqüência de que uma variação suave no domínio espaço implica em uma variação abrupta no domínio freqüência [14]. Desta forma, a correlação entre os elementos das submatrizes no domínio freqüência é menor do que a correlação entre os elementos das submatrizes no domínio real, dificultando a extração das componentes principais.

No entanto, mesmo nos casos em que a *PSNR* das imagens reconstruídas após compressão no domínio complexo é ligeiramente menor do que a *PSNR* da

imagem reconstruída após compressão no domínio real, a impressão causada ao sistema visual humano é igual ou superior à impressão causada pela imagem reconstruída após compressão no domínio real. Conforme já mencionado, isto é devido ao efeito da melhor recuperação dos contrastes presentes na imagem original, após a compressão no domínio complexo.

A Figura (23) apresenta a reconstrução da imagem original da Figura (20) após compressão no domínio frequência, utilizando submatrizes de  $8 \times 8$  elementos e considerando 32 auto-vetores. A razão de compressão obtida apenas devido ao fato de preservar 32 auto-vetores e descartar os demais é da ordem de 76% e a Relação Sinal-Ruído de Pico da imagem comprimida foi determinada como sendo 37,81 dB.



Figura 23 - Imagem da Fig.20 reconstruída após compressão no domínio frequência.			
$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.758	37.81 dB	32	8 x 8

As imagens das Figuras (24) e (25) apresentam entre si a mesma razão de compressão, assim como as imagens das Figuras (26) e (27). Estas imagens foram

colocadas no conjunto de resultados para que possam ser comparadas em termos de *PSNR* e qualidade visual subjetiva.

A Figura (24) apresenta a imagem da Figura (20) reconstruída após compressão no domínio freqüência, utilizando submatrizes de  $8 \times 8$  elementos e tendo sido preservados 27 auto-vetores. Esta imagem é inferior quanto à *PSNR* com relação à imagem da Figura (25) que apresenta a imagem da Figura (20) reconstruída após compressão no domínio real, utilizando a mesma dimensão para as submatrizes e tendo sido preservados 32 auto-vetores. No entanto, quanto à qualidade visual subjetiva, a imagem da Figura (24) em nada perde para a imagem da Figura (25). O mesmo sendo válido com relação às imagens das Figuras (26) e (27).



Figura 24 - Imagem da Fig.20 reconstruída após compressão no domínio freqüência.			
$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.641	35.65 dB	27	8 x 8



Figura 25 - Imagem da Fig.20 reconstruída após compressão no domínio espaço.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.629	37.28 dB	32	8 x 8



Figura 26 - Imagem da Fig.20 reconstruída após compressão no domínio frequência.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.313	29.39 dB	13	8 x 8



Figura 27 - Imagem da Fig.20 reconstruída após compressão no domínio espaço.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.316	31.44 dB	16	8 x 8

O próximo conjunto de figuras apresentado refere-se à imagem original Moon, apresentada na Figura (28). O histograma da imagem original Moon mostrado na Figura (29) apresenta, assim como o histograma da imagem original Lenna (Figura (21)), continuidade entre as frequências absolutas e a razão de decréscimo dos auto-valores do domínio complexo não é muito maior que a razão de decréscimo dos auto-valores do domínio real, como pode ser visto no gráfico de distribuição de auto-valores mostrado na Figura (30).

A exemplo das imagens apresentadas nos conjuntos de imagens anteriores, as imagens mostradas nas Figuras (31) a (35) estão presentes para que sejam comparadas quanto à *PSNR* e à qualidade subjetiva devida ao sistema visual humano.

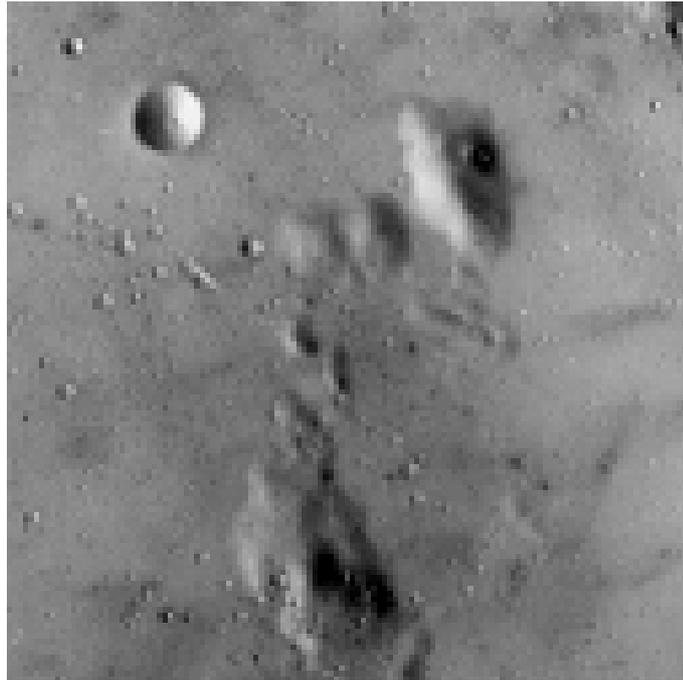


Figura 28 - Moon : Imagem original  
128 x 128 pixels - 256 níveis de cinza

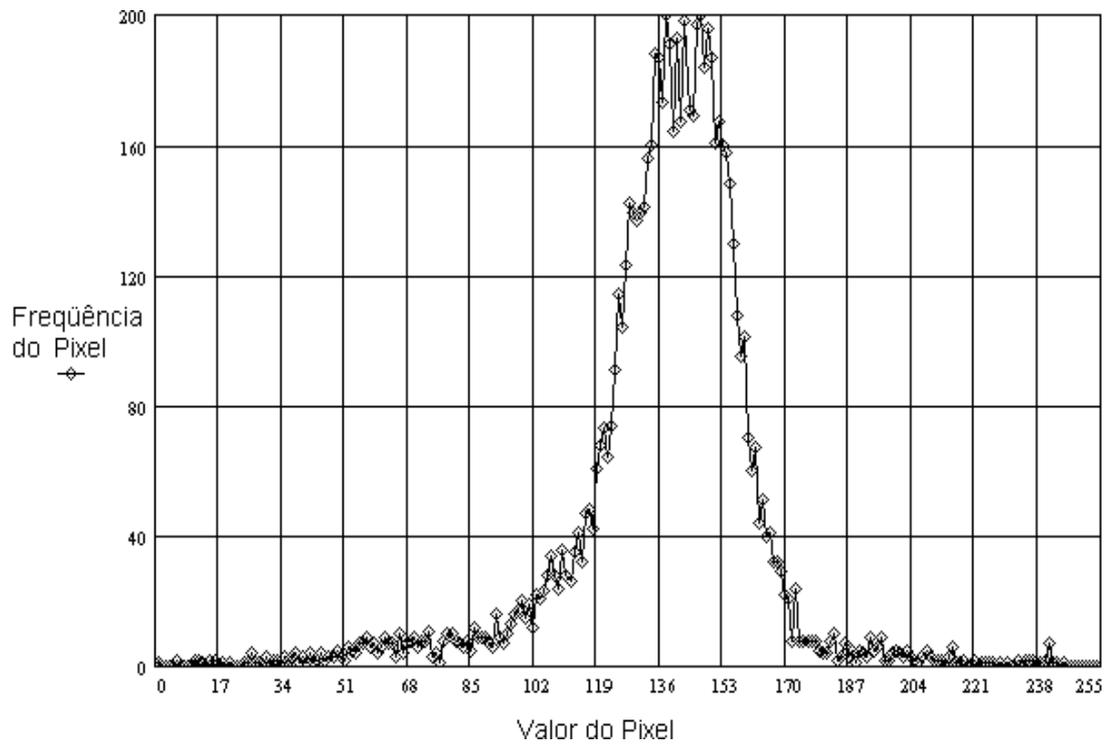


Figura 29 - Histograma da Imagem da Fig.28.

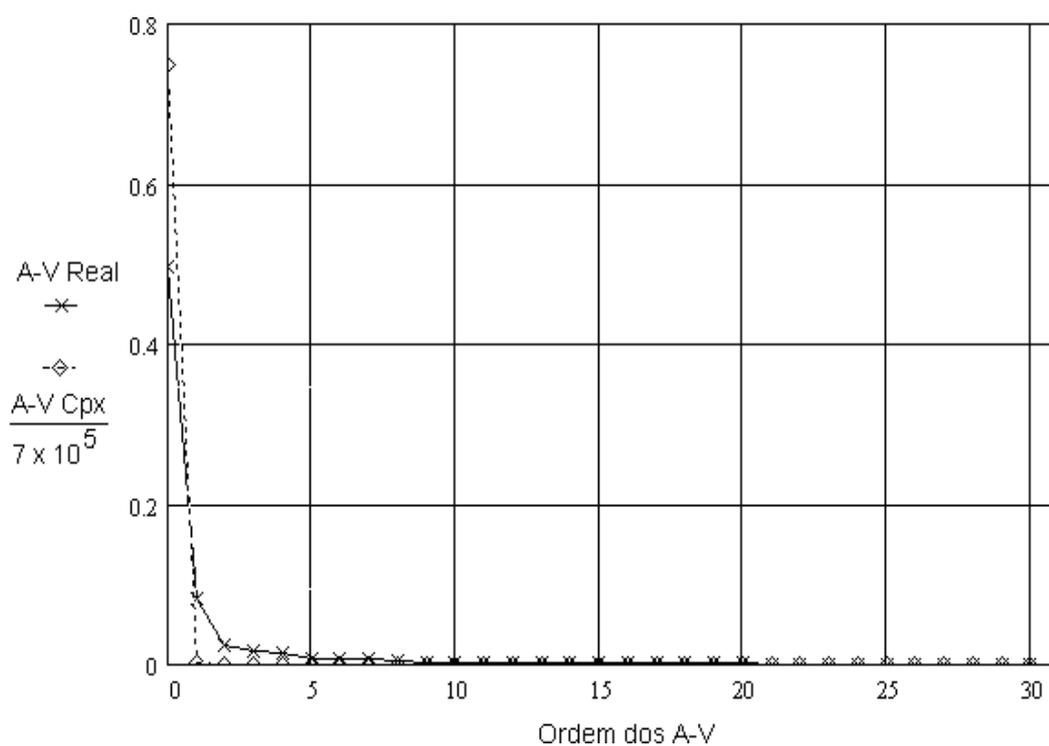


Figura 30 - Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.28



Figura 31 - Imagem da Fig.28 reconstruída após compressão no domínio frequência.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.758	38.73 dB	32	8 x 8



Figura 32 - Imagem da Fig.28 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.641	37.13 dB	27	8 x 8



Figura 33 - Imagem da Fig.28 reconstruída após compressão no domínio espaço.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.629	38.25 dB	32	8 x 8



Figura 34 - Imagem da Fig.28 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.313	32.69 dB	13	8 x 8



Figura 35 - Imagem da Fig.28 reconstruída após compressão no domínio espaço.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.316	33.91 dB	16	8 x 8

As figuras a seguir apresentadas, referentes às imagens originais Polígonos (Figura (36)) e Eye (Figura (44)), constituem novos exemplos de imagens em que o histograma sugere variação abrupta de contraste dos elementos da imagem, no domínio espaço.

Conforme já observado, quanto maior for a descontinuidade entre os valores de frequência absoluta de pixels no histograma, maior será a razão de decréscimo dos auto-valores do domínio complexo em relação à razão de decréscimo dos auto-valores do domínio real, o que é demonstrado pelos gráficos de distribuição dos auto-valores nos domínios real e complexo.

Para imagens com esta característica de variação abrupta de contraste no domínio espaço, a correlação entre os elementos das submatrizes no domínio frequência é maior do que a correlação entre os elementos das submatrizes no domínio real. Da mesma forma que o observado para a imagem Brasil (Figuras (11) a (19)), a compressão no domínio complexo tende, em função disto, a preservar as características de alto contraste da imagem e apresentar maior erro relativo em regiões constantes no domínio espaço. A compressão no domínio real, por outro lado, tende a preservar as regiões constantes da imagem e apresentar maior erro relativo em regiões de alto contraste no domínio espaço.

As imagens que constituem o conjunto relativo à imagem original Polígonos (Figuras (36) a (43)) e as que constituem o conjunto relativo à imagem original Eye (Figuras (44) a (51)) são apresentadas também para exemplificar uma outra

característica em que o método para análise dos componentes principais no domínio complexo é superior ao método para análise dos componentes principais no domínio real.

As imagens reconstruídas após a compressão no domínio real apresentam o efeito de “*blocking*” ou efeito xadrez, intrínseco à divisão da imagem em submatrizes, como pode ser percebido nas imagens das Figuras (41), (43), (49) e (51).

No método de domínio espectral este efeito é menos sentido porque as imagens são montadas em blocos no domínio frequência e, através da Transformada Inversa de Fourier Bidimensional (*IFFT*), o efeito xadrez em frequência é retornado ao domínio espaço sob a forma de funções *sinc* [6][13][14]. Como as funções *sinc* não apresentam variações abruptas, o efeito percebido pelo sistema humano de visão é um efeito de suavização.

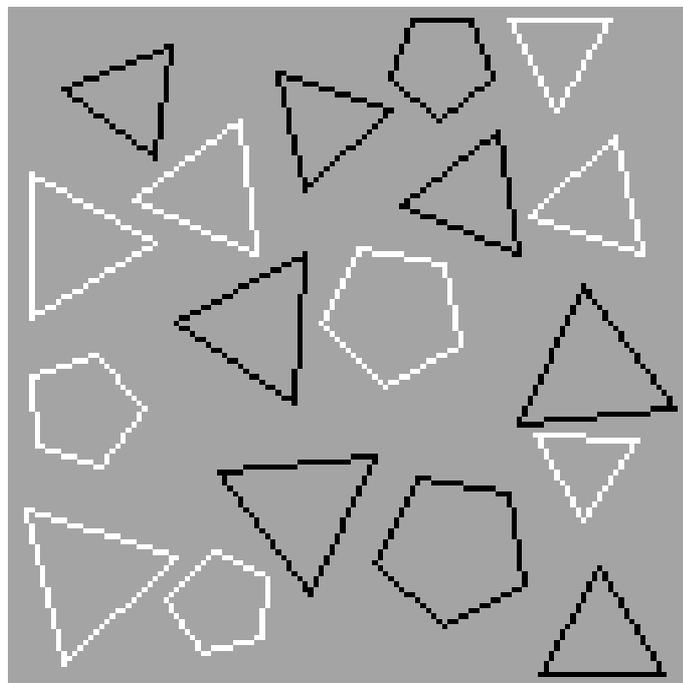


Figura 36 - Polígonos : Imagem original
Dimensões : 128 x 128 pixels
Polígonos brancos(255) e pretos(0) sobre fundo cinza(166)

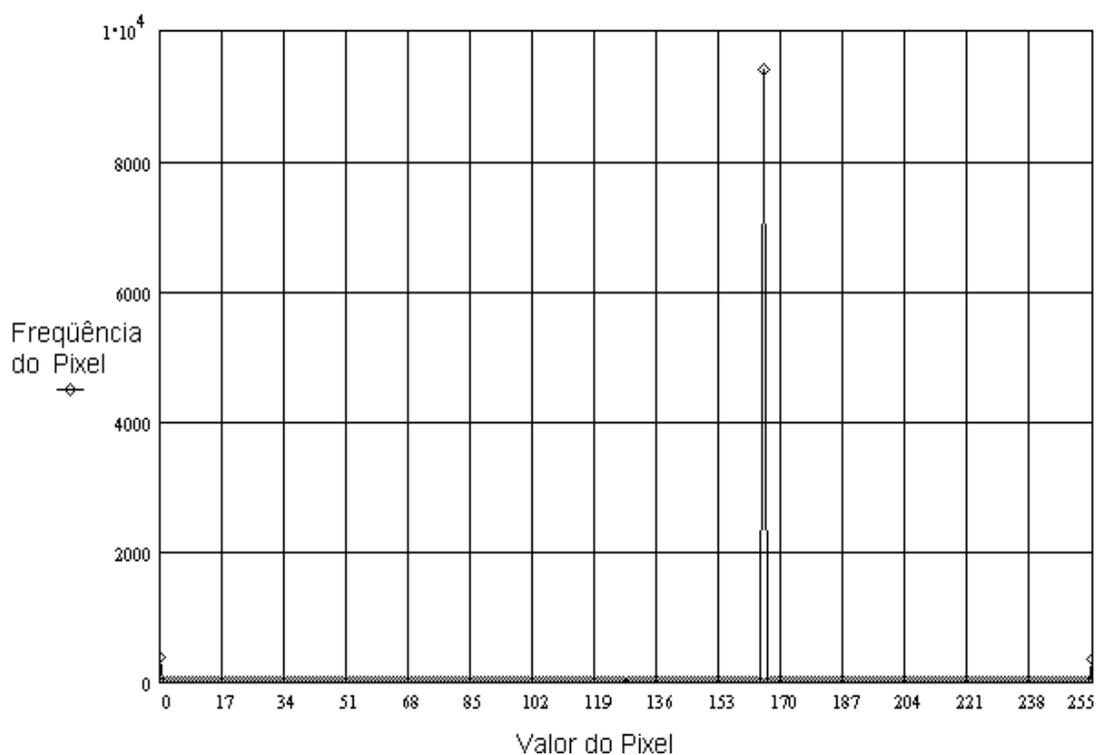


Figura 37 - Histograma da Imagem da Fig.36.

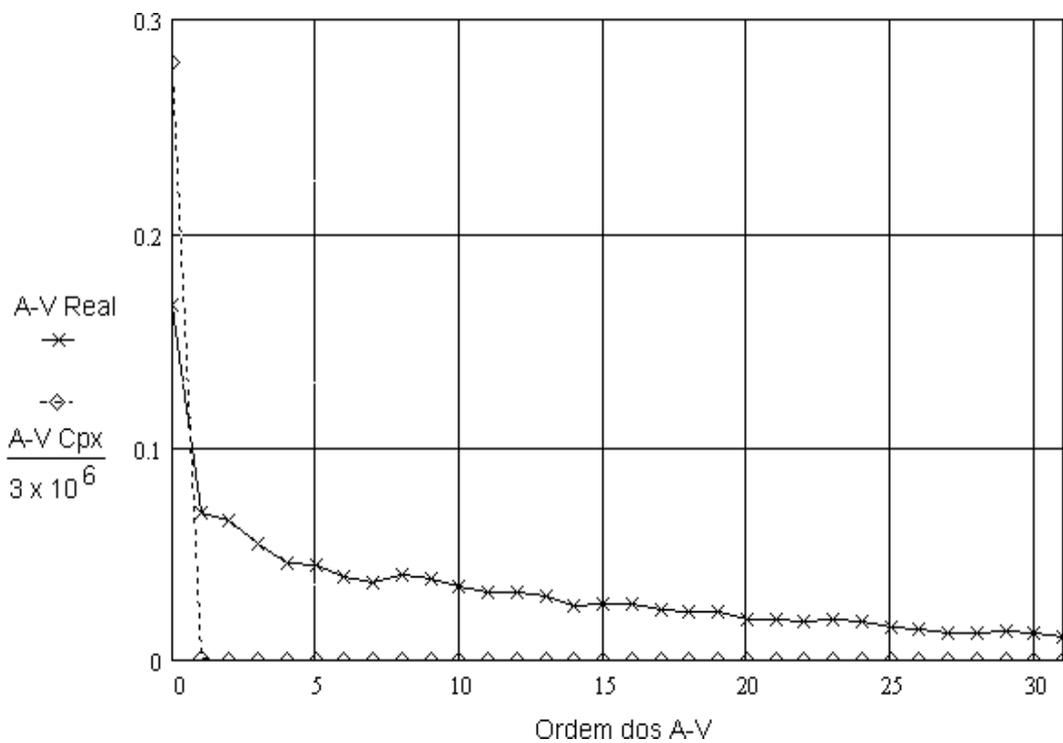


Figura 38 - Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.36

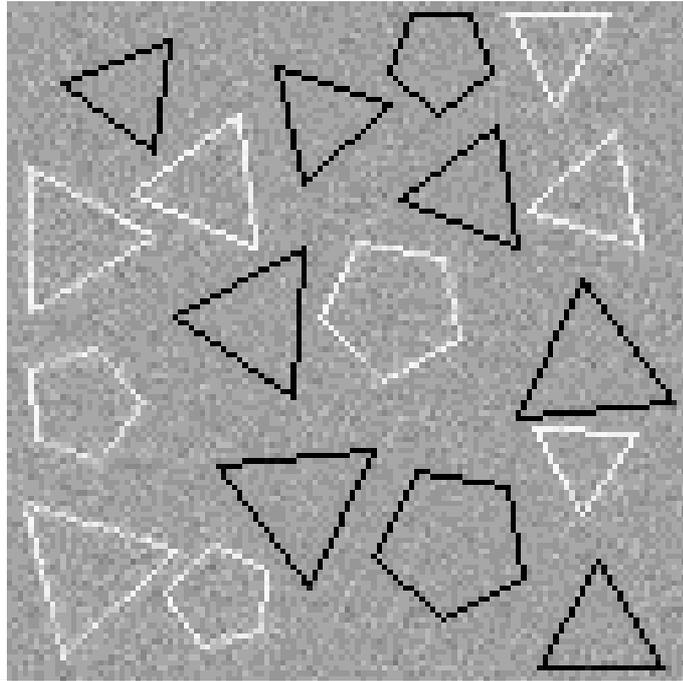


Figura 39 - Imagem da Fig.36 reconstruída após compressão no domínio frequência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.758	26.85 dB	32	8 x 8

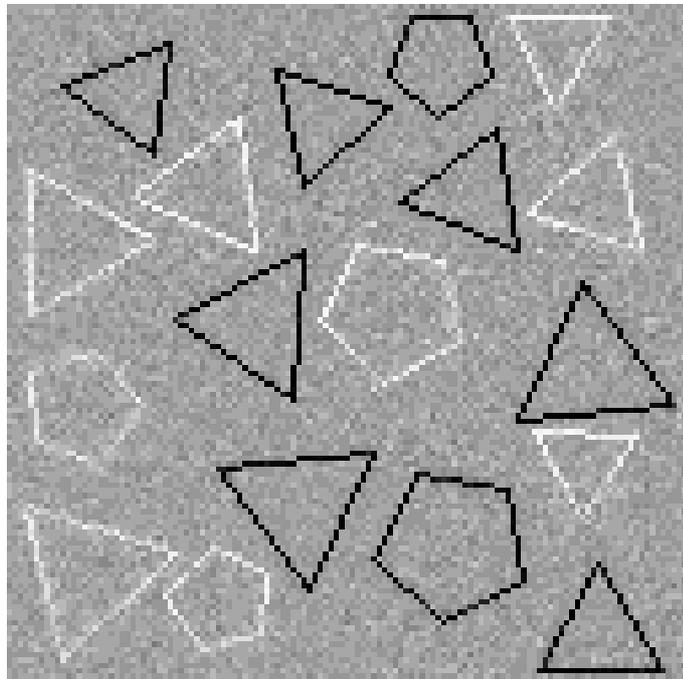


Figura 40 - Imagem da Fig.36 reconstruída após compressão no domínio frequência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.641	25.16 dB	27	8 x 8

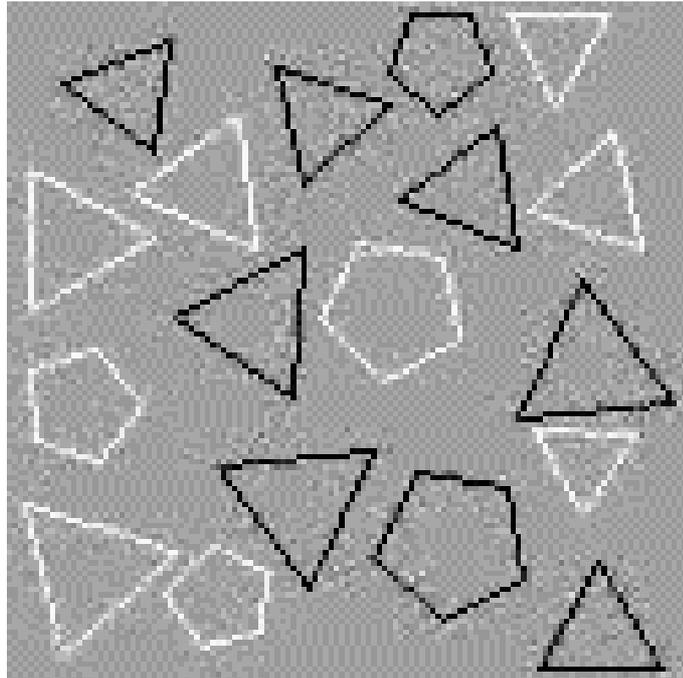


Figura 41 - Imagem da Fig.36 reconstruída após compressão no domínio espaço.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.629	25.07 dB	32	8 x 8

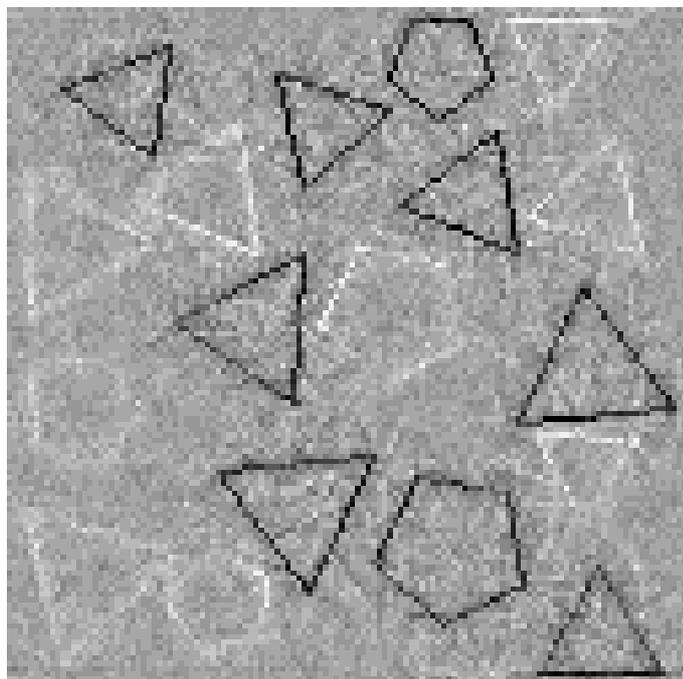


Figura 42 - Imagem da Fig.36 reconstruída após compressão no domínio freqüência.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.313	20.92 dB	13	8 x 8

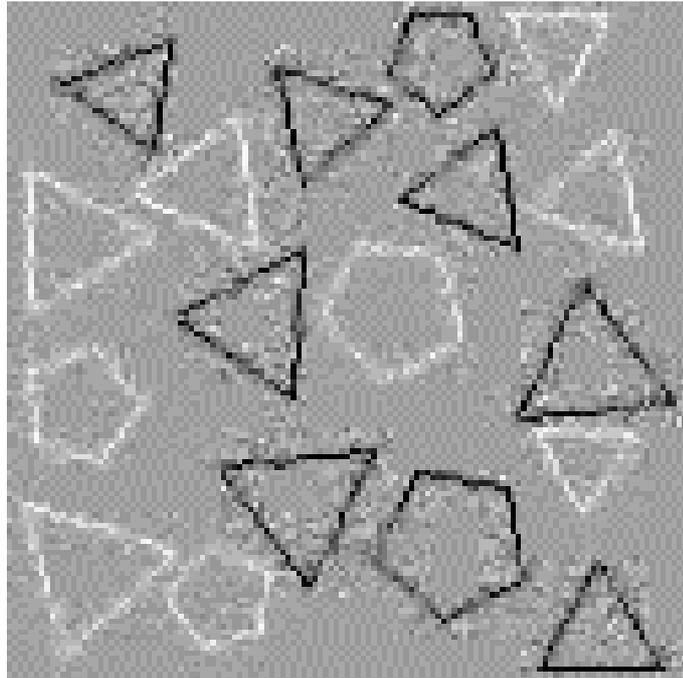


Figura 43 - Imagem da Fig.36 reconstruída após compressão no domínio espaço.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.316	21.27 dB	16	8 x 8

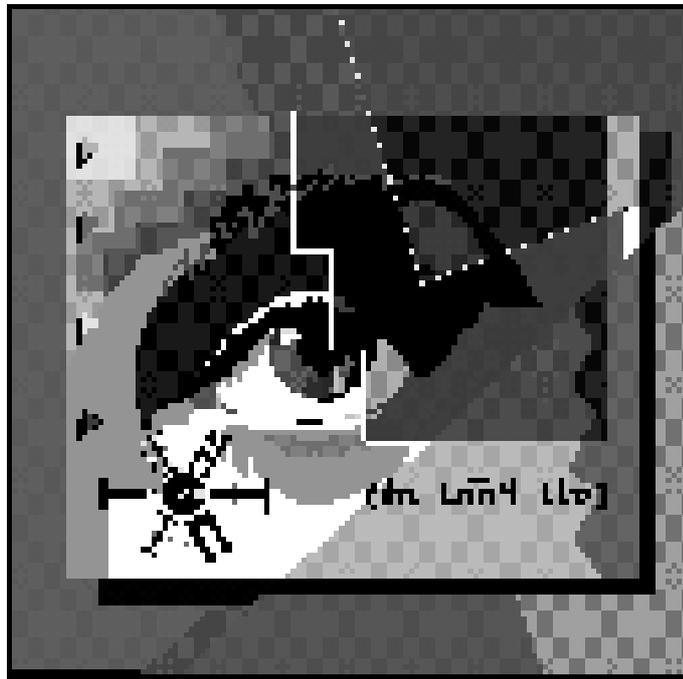


Figura 44 - Eye : Imagem original  
128 x 128 pixels - 256 níveis de cinza

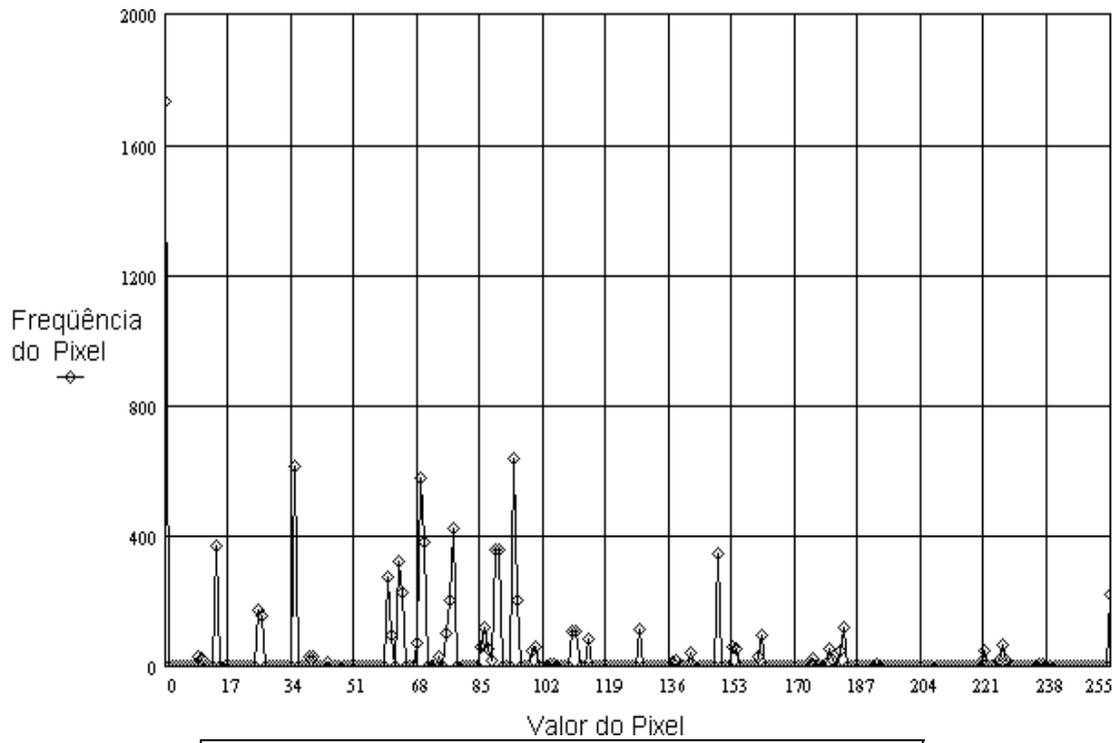


Figura 45 - Histograma da Imagem da Fig.44.

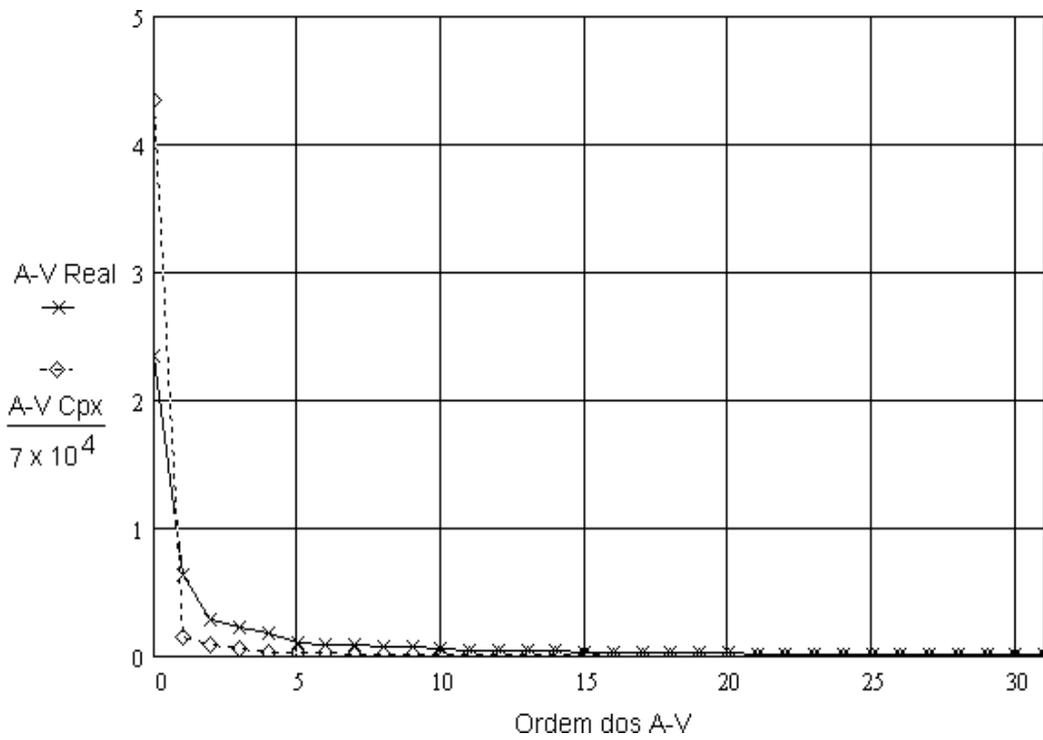


Figura 46- Distribuição de auto-valores nos domínios real e complexo da Imagem da Fig.44

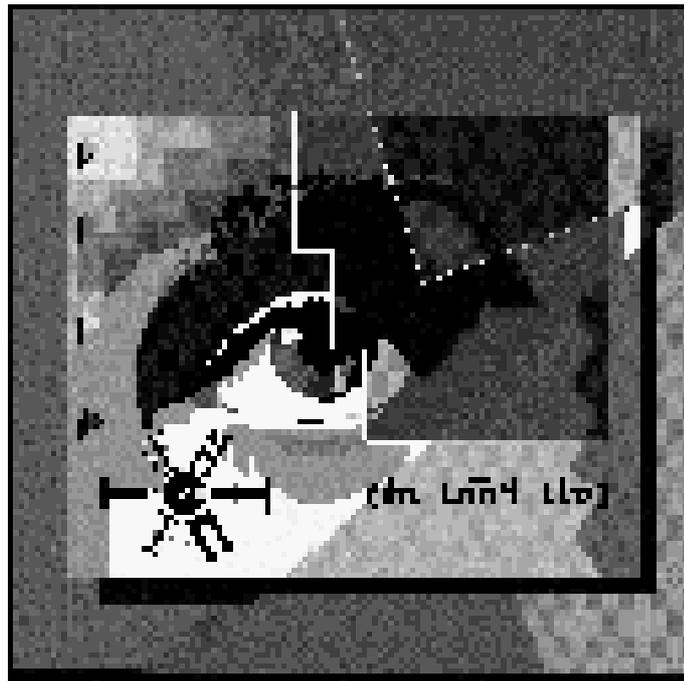


Figura 47 - Imagem da Fig.44 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.758	29.97 dB	32	8 x 8

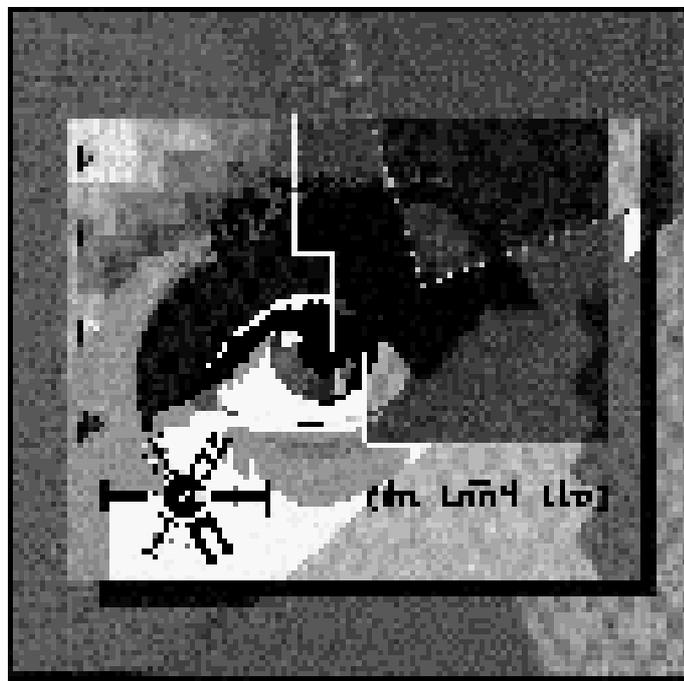


Figura 48 - Imagem da Fig.44 reconstruída após compressão no domínio freqüência.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.641	27.69 dB	27	8 x 8

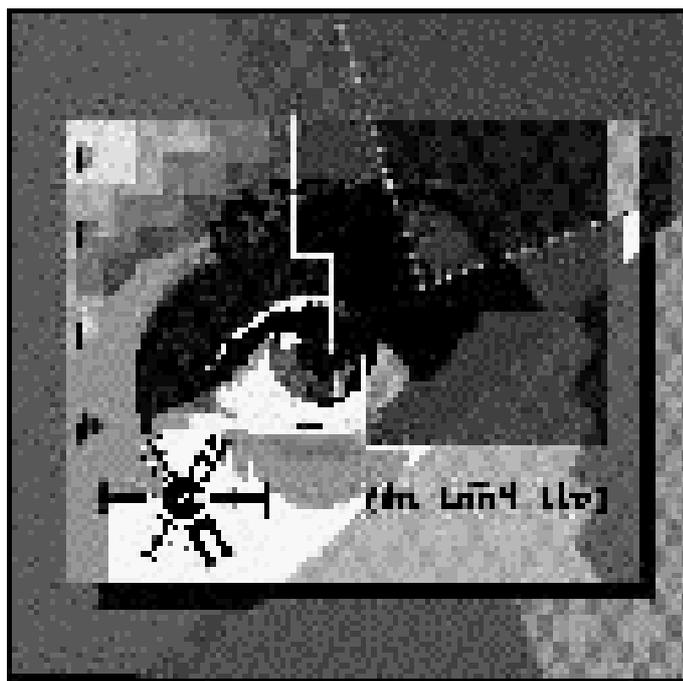


Figura 49 - Imagem da Fig.44 reconstruída após compressão no domínio espaço.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.629	27.78 dB	32	8 x 8

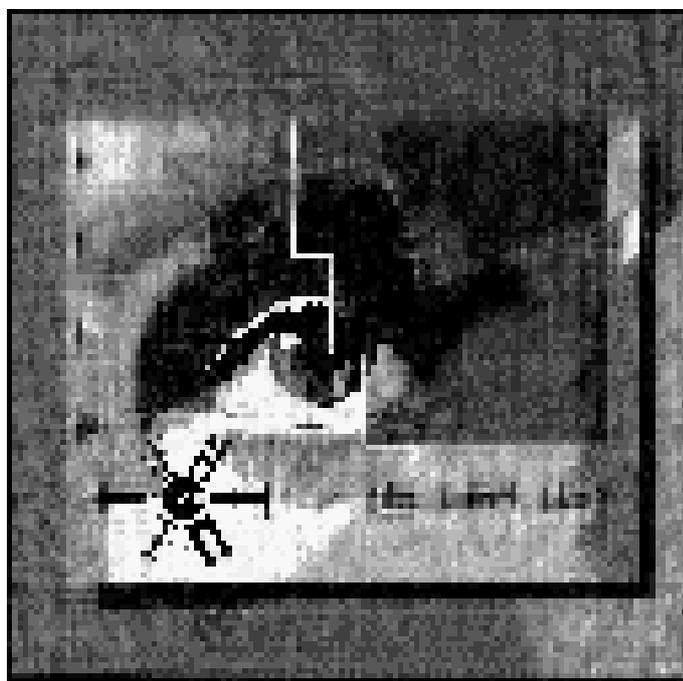


Figura 50 - Imagem da Fig.44 reconstruída após compressão no domínio frequência.

$\rho$	PSNR	Auto-Vetores	Sub-matriz
0.313	21.77 dB	13	8 x 8

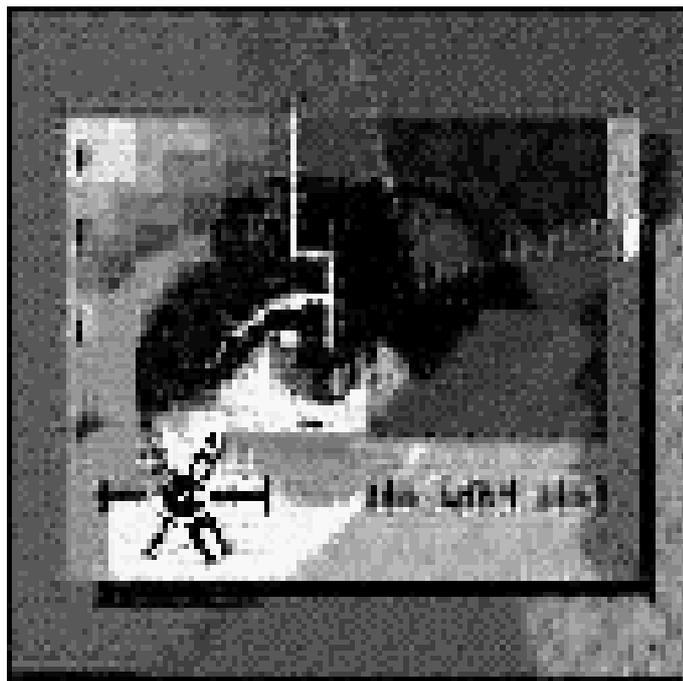


Figura 51 - Imagem da Fig.44 reconstruída após compressão no domínio espaço.

$\rho$	<i>PSNR</i>	Auto-Vetores	Sub-matriz
0.316	22.16 dB	16	8 x 8

Outra característica do algoritmo *CGHAPCA* a salientar é a capacidade de generalização do algoritmo. A capacidade de generalização permite que os pesos sinápticos (auto-vetores) convergidos para um conjunto de dados possam ser utilizados para comprimir um outro conjunto de dados, cuja estatística se assemelhe.

A Figura (52) apresenta a imagem da Figura (11) (Brasil original) comprimida e descomprimida utilizando a rede neural artificial complexa da imagem da Figura (36) (Polígonos original). A operação consiste em inicializar os pesos sinápticos da rede neural artificial da imagem a ser comprimida (imagem “A”), com os pesos sinápticos (auto-vetores) obtidos após a convergência do algoritmo para uma rede neural treinada para uma outra imagem (imagem “B”). Apresentando-se à RNA da imagem “B” vetores de treino da imagem “A”, ao final de uma época, com razão de

aprendizado nula, são obtidos os valores associados aos neurônios de saída e a imagem pode ser reconstruída a partir das componentes principais consideradas.

Esta é uma característica interessante, se considerarmos que os quadros de uma imagem em movimento, ao longo de uma seqüência de exposições, em muito se assemelham. Neste caso seria possível reconstruir quadro após quadro da imagem em movimento com os parâmetros da rede treinados para o primeiro quadro da janela.

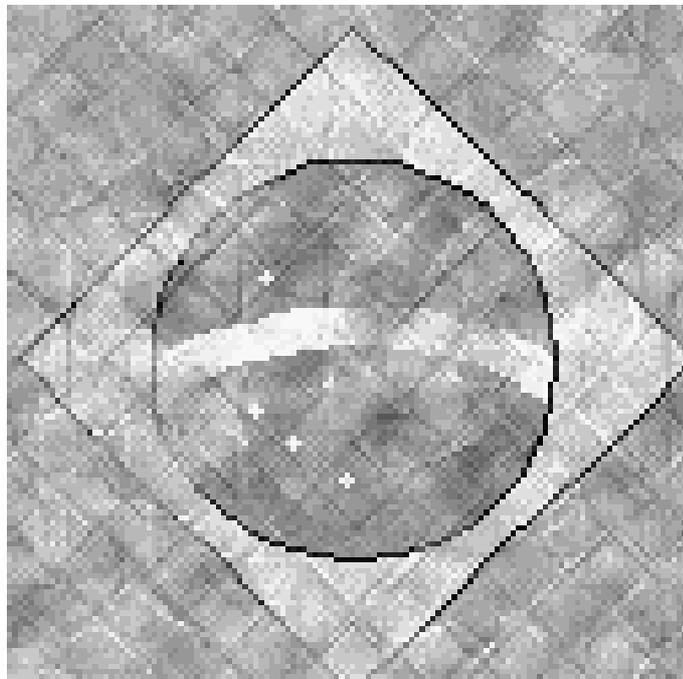


Figura 52- Imagem da Fig. 11 comprimida e descomprimida utilizando a RNA Complexa da imagem da Fig. 36  
 $PSNR= 19.35$  dB

Outra aplicação de extrema valia desta característica seria para o caso da necessidade de armazenamento e posterior recuperação de uma grande quantidade de imagens estatisticamente semelhantes, diferentes apenas por pequenos detalhes, como as imagens obtidas de impressões digitais. Além da vantagem de poder-se usar os

parâmetros de redes treinadas para um ou alguns poucos casos, soma-se, nesta aplicação, a grande vantagem que o método apresenta no domínio espectral, de salientar os contrastes.

## Capítulo 7

### Sugestões para Futuros Trabalhos

Neste capítulo são apresentadas as sugestões para futuros trabalhos que ocorreram durante o desenvolvimento desta dissertação. As sugestões dizem respeito à aplicabilidade e otimização deste novo algoritmo.

Uma otimização para o método, que poderá acarretar uma diminuição do número de épocas de treino e uma alteração na razão de decréscimo dos auto-valores, pode ser obtida através da adoção de um outro critério para a transformação das submatrizes utilizadas para o treino da RNA em vetores linha. Neste trabalho todas as linhas das submatrizes são lidas da esquerda para a direita. Uma sugestão para alterar a correlação entre os elementos do vetor é que a leitura das linhas pares da submatriz seja feita da esquerda para a direita e a leitura das linhas ímpares, da direita para a esquerda.

Como o janelamento de convergência proposto neste trabalho é uma técnica nova, cabe uma análise detalhada dos ganhos relativos à complexidade computacional, decorrentes de sua aplicação ao algoritmo.

Ainda dentro do escopo do estudo da complexidade do algoritmo, seria válida uma avaliação em termos da diminuição da complexidade computacional devida à redução do conjunto de treino (possibilitada pelo método de domínio complexo e devida à propriedade de simetria espectral) e do aumento de complexidade relativo a transformar os dados do domínio real para o domínio complexo.

Conforme apresentado no Capítulo 6, uma importante aplicação do método para análise dos componentes principais no domínio espectral vem da

capacidade de generalização do *CGHAPCA*. É sugerido que, em futuros estudos, seja avaliada a aplicabilidade do método para imagens em movimento, ou outras aplicações em que os quadros sejam estatisticamente semelhantes.

Sugere-se também que seja utilizada uma RNA supervisionada do tipo *Multi Layer Perceptron* no domínio complexo [1], cujo conjunto de treino sejam as componentes principais resultantes de várias imagens, no sentido de aumentar a capacidade de generalização do método.

Para que o método possa ser estendido a imagens coloridas (sistema *RGB*), sugere-se que sejam treinadas 3 redes neurais complexas, uma para a componente *R*, outra para a componente *G* e outra para a componente *B*.

Neste trabalho, a análise dos componentes principais através do aprendizado Hebbiano é feita no domínio frequência. Uma investigação interessante seria estudar esta análise através da Transformada Wavelet.

Também seria relevante a aplicação do algoritmo à análise dos componentes principais de outros conjuntos de dados (não representativos de imagens) como, por exemplo, os dados gerados por reflexão de ondas eletromagnéticas em uma superfície, onde o módulo da onda refletida dá a refração da superfície e a fase da onda refletida dá a distância ao transmissor, constituindo, portanto, um conjunto de dados complexo.

## Referências Bibliográficas

- [1] D. L. Bix e S. J. Pipenberg, “A Complex Mapping Network for Phase Sensitive Classification”, *IEEE Transactions on Neural Networks*, vol.4, n° 1, 1993.
- [2] A. H. Tewfik e M. Deriche, “ An Eigenstructure Approach to Edge Detection”, *IEEE Transactions on Image Processing*, vol.2, n° 3, 1993.
- [3] F. C. C. de Castro, “Reconhecimento e Localização de Padrões em Imagens Utilizando Redes Neurais Artificiais como Estimador de Correlação Espectral”, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, 1995.
- [4] S. Haykin, *Neural Networks*, Macmillan College, New York, NY, 1994.
- [5] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, 1995.
- [6] E. L. Hall, *Computer Image Processing and Recognition*, Computer Science and Applied Mathematics, 1979.
- [7] L. Xu e A. L. Yuille, “Robust Principal Component Analysis by Self-Organizing Rules Based on Statistical Physics Approach”, *IEEE Transactions on Neural Networks*, vol.6, n° 6, 1995.
- [8] R. Bronson, *Matrix Methods: An Introduction*, Academic Press Inc, San Diego, CA, 1991.
- [9] F. R. Gantmacher, *The Theory of Matrices*, vol.1, Chelsea Publishing Company, New York, NY, 1977.
- [10] F. Ayres Jr., *Theory and Problems of Matrices*, Schaum’s Outline Series in Mathematics, McGraw-Hill Book Company, 1962.
- [11] L. H. Chen e S. Chang, “An Adaptive Learning Algorithm for Principal

- Component Analysis”, *IEEE Transactions on Neural Networks*, vol.6, n° 5, 1995.
- [12] K. S. Shanmugan e A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, 1988.
- [13] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [14] R. Gonzales e P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.
- [15] A. V. Oppenheim e R. W. Schaffer, *Digital Signal Processing*, Prentice-Hall, 1975.
- [16] E. O. Brigham, *The Fast Fourier Transform and its Applications*, AvanteK, 1988.
- [17] S. Bannour e M. R. Azimi-Sadjadi, “Principal Component Extraction Using Recursive Least Squares Learning”, *IEEE Transactions on Neural Networks*, vol.6, n° 2, 1995.
- [18] J. Hertz, A. Krogh e R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.