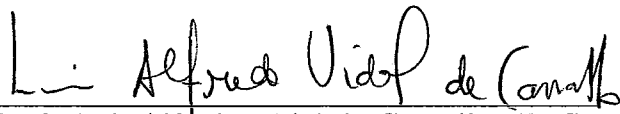


# Um Método de Aprendizado para Redes Neuronais Analógicas de Hopfield com Aplicação à Compactação de Imagens

*Jane Tavares Alvarez*

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

  
Prof. Luís Alfredo Vidal de Carvalho, D. Sc.  
(presidente)

  
Prof. Valmir Carneiro Barbosa, Ph. D.

  
Prof<sup>a</sup>. Doris Ferraz de Aragon, D. Sc.

RIO DE JANEIRO, RJ – BRASIL  
DEZEMBRO DE 1992

ALVAREZ, JANE TAVARES

Um Método de Aprendizado para Redes Neurais Analógicas de Hopfield com Aplicação à Compactação de Imagens [Rio de Janeiro] 1992 VI, 107p., 29.7 cm, (COPPE/UFRJ, M. Sc., ENGENHARIA DE SISTEMAS E COMPUTAÇÃO, 1992)

TESE – Universidade Federal do Rio de Janeiro, COPPE

1 – Redes Neurais 2 – Aprendizado 3 – Compactação

I. COPPE/UFRJ II. Título(Série).

*Ao Jobson*

# Agradecimentos

Agradeço aos professores Luís Alfredo e Valmir tanto pela confiança que depositaram em mim no desenvolvimento deste trabalho, quanto pelas orientações competentes.

À professora Doris Aragon e ao ILTC por terem sempre me incentivado e apoiado desde o meu ingresso na pós-graduação, colocando inclusive à disposição todo o material e equipamento necessários como as estações SUN.

Aos amigos da COPPE, em especial ao Eliseu e Alberto, sempre competentes e atenciosos, ao Guilherme, Zé e a Nahri, e ainda as meninas da secretaria, Cláudia e Denise, sempre prontas a atender.

Agradeço a todos os meus amigos, principalmente a Tei, ao Petrócio (sempre presente no início do Mestrado) e a Emília, sempre dispostos a ajudar, mas em especial eu agradeço aos meus amigos Maurício e Claudinha, por não medirem esforços em me apoiar, incentivar e pelas críticas sempre válidas.

Aos meus pais, obrigada !

Ao Jobson, pelo carinho, amizade, incentivo, pela ajuda no uso do Xview e do Guide, pelas "dicas" na implementação e pelas críticas sempre construtivas.

Enfim, agradeço ao Senhor, pela força maior.



Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

Um Método de Aprendizado para Redes Neurais Analógicas de Hopfield com  
Aplicação à Compactação de Imagens

Jane Tavares Alvarez

Dezembro de 1992

Orientadores: Luis Alfredo Vidal de Carvalho

Valmir Carneiro Barbosa.

Programa: Engenharia de Sistemas e Computação.

Neste trabalho são propostos três novos algoritmos de aprendizado para Redes Neurais Analógicas de Hopfield, que são avaliados, de forma empírica, utilizando-se padrões gerados aleatoriamente. Os resultados obtidos no aprendizado superaram os existentes até então. Determinado o algoritmo de melhor desempenho, iniciam-se os testes de aplicação, que consistem de duas etapas: o aprendizado de imagens de 400 pixels nas cores preto, cinza e branco e a compactação (realizada através de um método proposto bem intuitivo), seguida da descompactação (realizada via rede de Hopfield) dessas imagens. É interessante notar que, na primeira etapa dos testes de aplicação todos os grupos de imagens foram 100% aprendidos. A segunda etapa também produziu bons resultados, principalmente se levado em conta o aspecto visual das imagens. Todos os testes feitos neste trabalho foram realizados nas estações de trabalho SUN, utilizando a linguagem *C* como linguagem de programação.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

A Learning Method for Neural Analog Hopfield Networks with application in  
Image Compression

Jane Tavares Alvarez

December, 1992

Thesis Supervisors: Luis Alfredo Vidal de Carvalho

Valmir C. Barbosa.

Department: Programa de Engenharia de Sistemas e Computação.

In this work, three new algorithms for Analog Hopfield Neural Networks are presented, empirically evaluated with random patterns. The results obtained in learning overcame the ones which have been known. Once determined the algorithm of better performance, we began the application tests which are composed of two stages: the learning of images of 400 pixels in black, gray and white colors and the compression (accomplished by an intuitive proposed method) following by the decompression (accomplished by a Hopfield network) of these images. It is interesting to notice that in the first stage of the application tests, all the images groups have been 100% learned. The second stage also produced good results, mainly if we consider the visual aspect of the images. All the tests have been performed on a workstation (SUN), utilizing the C language as programming tool.

# Índice

<b>I</b>	<b>Introdução</b>	<b>1</b>
I.1	Objetivo da Tese . . . . .	3
I.2	Descrição dos capítulos . . . . .	3
<b>II</b>	<b>Redes Neurais</b>	<b>5</b>
II.1	Introdução às Redes Neurais . . . . .	5
II.1.1	Um pouco de Neurofisiologia . . . . .	5
II.1.2	Alguns Modelos Conexionistas Básicos . . . . .	7
II.2	O Modelo de Hopfield . . . . .	11
II.2.1	Introdução . . . . .	11
II.2.2	O Modelo Lógico de Hopfield . . . . .	12
II.2.3	O Modelo Analógico de Hopfield . . . . .	15
II.3	Aprendizado em Redes Neurais . . . . .	17
II.3.1	Uma idéia de Aprendizado . . . . .	17
II.3.2	Classificação do Aprendizado em Redes Neurais . . . . .	18
II.3.3	Regras de Aprendizado . . . . .	21
II.3.4	Métodos de Síntese . . . . .	28

<b>III Três Novos Algoritmos de Aprendizado</b>	<b>30</b>
III.1 Considerações iniciais . . . . .	30
III.2 Algumas considerações matemáticas . . . . .	32
III.3 Os algoritmos . . . . .	33
III.4 Os critérios de parada . . . . .	35
III.5 Implementação . . . . .	36
III.6 Resultados experimentais . . . . .	36
<b>IV Compactação de Imagens</b>	<b>43</b>
IV.1 Uma visão geral . . . . .	43
IV.2 Alguns meios para avaliar a fidelidade da imagem descompactada . . . . .	44
IV.3 Componentes Principais e Redes Neuronais . . . . .	45
IV.4 Compactação de imagem utilizando modelos de Redes Neuronais . . . . .	48
<b>V O método de compactação utilizado</b>	<b>51</b>
V.1 O algoritmo . . . . .	52
V.2 Implementação . . . . .	53
V.3 Os testes de compactação . . . . .	53
<b>VI Conclusão</b>	<b>64</b>

# Lista de Figuras

I.1	Esquema geral do sistema . . . . .	4
II.1	A estrutura do neurônio biológico . . . . .	6
II.2	Esquema de uma rede neuronal . . . . .	8
II.3	Regra de ativação do neurônio lógico . . . . .	11
II.4	Regra de ativação do neurônio semi-linear . . . . .	12
II.5	Perfil da superfície de energia com dois pontos de mínimo . . . . .	14
II.6	Neurônio representado pela figura de um recipiente . . . . .	16
II.7	Aplicação da regra de aprendizado a uma rede neuronal . . . . .	18
II.8	Aplicação do método de síntese . . . . .	19
II.9	Arquitetura da rede "Backpropagation" . . . . .	27
II.10	Classificador neuronal de padrões . . . . .	28
III.1	Dois perfis da função de energia $E$ com três pontos de mínimo no instante de tempo $t + \Delta t$ . . . . .	31
III.2	Esquema geral dos algoritmos . . . . .	34
III.3	Tabela 1 . . . . .	37
III.4	Tabela 2 . . . . .	38
III.5	Tabela 3 . . . . .	39

III.6 Tabela 4 . . . . .	39
III.7 Tabela 5 . . . . .	40
III.8 Tabela 6 . . . . .	41
III.9 Gráfico de desempenho dos algoritmos de aprendizado . . . . .	42
IV.1 Representação da imagem original $[R]$ através de <i>imagens bases</i> . . .	47
V.1 Gráfico 1 . . . . .	55
V.2 Gráfico 2 - número de imagens = 2 . . . . .	56
V.3 Gráfico 3 - número de imagens = 3 . . . . .	57
V.4 Gráfico 4 - número de imagens = 4 . . . . .	58
V.5 Gráfico 5 - número de imagens = 5 . . . . .	59
V.6 Gráfico 6 - número de imagens = 6 . . . . .	60
V.7 Gráfico 7 - número de imagens = 7 . . . . .	61
V.8 Gráfico 8 - número de imagens = 8 . . . . .	62
V.9 Gráfico 9 - número de imagens = 9 . . . . .	63

# Capítulo I

## Introdução

A Inteligência Artificial é uma linha de estudo dentro da Ciência da Computação que trata da solução de problemas, através da construção e do uso de sistemas inteligentes. Seu paradigma estabelece, fundamentalmente, três etapas básicas:

- i)* escolha de uma atividade cognitiva;
- ii)* desenvolvimento de uma teoria capaz de explicá-la; e
- iii)* confronto da teoria com a realidade, através de testes que tornem claras as potencialidades e as limitações da teoria.

Conforme a elaboração da teoria, a Inteligência Artificial pode seguir duas correntes com características bem definidas: *Simbolista* e a *Conexionista*.

O Paradigma Simbolista ou Cognitivista determina que a solução de problemas é um processo algorítmico, isto é, segue uma série de passos lógicos ou aritméticos bem definidos. Além disso, tendo como base a Lógica, os Simbolistas defendem a idéia de que a inteligência está armazenada em regras heurísticas, o que representa uma forma sequencial e centralizada de modelar o conhecimento. Através do uso de entidades chamadas símbolos e de regras que podem ser aplicadas sobre tais símbolos, temos que essa corrente se identifica com atividades cognitivas de *alto nível*, como linguagem, raciocínio e solução de problemas.

No entanto, para as atividades denominadas de *baixo nível*, o Paradigma Simbolista se mostra limitado, sendo o Conexionismo a escola mais adequada

no tratamento dos processos ligados à audição, visão, memória, etc.

Segundo os Conexionistas, a informação inteligente é processada de forma paralela (no tempo) e distribuída (no espaço) por uma população de elementos computacionais *simples*, os neurônios, que interagem uns com os outros de forma competitiva ou cooperativa. Neste caso, a inteligência está armazenada nas conexões entre os neurônios, bem como nas suas capacidades de excitar ou inibir, e emerge, não do neurônio isoladamente, mas sim da coletividade, isto é, de toda a *rede neuronal*.

Apesar de muitas das idéias que ajudaram a definir o Conexionismo existirem desde o século passado, tendo inclusive Freud como um de seus empreendedores, o seu estudo formal ainda é recente.

McCulloch e Pitts, em 1943, foram os primeiros a estabelecer, através do seu *neurônio lógico*, um modelo matemático capaz de definir o funcionamento de um neurônio. Já Rosenblatt, em 1958, utilizando o *neurônio lógico* e definindo uma forma de *treinamento*, criou uma teoria acerca da percepção visual — o *Perceptrons*. No entanto, o trabalho de Rosenblatt acabou sofrendo críticas um pouco duras por parte de Minsky e Papert que demonstraram a incapacidade de um perceptron de duas camadas computar funções que não fossem linearmente separáveis. Tal fato contribuiu tanto para o descrédito do trabalho de Rosenblatt quanto para uma estagnação no estudo das redes neuronais artificiais.

Entretanto, desde os anos 80 vem ocorrendo uma retomada dos estudos das Redes Neuronais Artificiais, o que se deve tanto ao desenvolvimento da tecnologia — que possibilitou a realização em hardware de modelos neuronais e de sua interconexão em redes, até o momento indispensáveis — quanto ao fato da escola simbolista, apesar de todos os seus avanços, não vir obtendo resultados significativos na realização de tarefas que o cérebro humano consegue executar facilmente.



## I.1 Objetivo da Tese

O objetivo deste trabalho é, em uma primeira fase, apresentar e avaliar três novos algoritmos de aprendizado para Redes Neurais Analógicas de Hopfield, utilizando padrões gerados aleatoriamente.

Os testes desta fase demonstraram que, dada uma Rede Neuronal Analógica de Hopfield com  $n$  neurônios, o número máximo de padrões estáveis pode superar os  $0.15n$  [2], principalmente se utilizado o terceiro algoritmo proposto, que obteve melhor desempenho.

Definido o algoritmo de melhor desempenho, podemos observar pela figura I.1 que foram realizados o *treinamento* e a simulação de imagens de 400 pixels nas cores preto, cinza e branco. O produto gerado por esta etapa são imagens que foram compactadas e por fim descompactadas, isto é, restauradas com a ajuda do conhecimento previamente armazenado na rede neuronal. Finalmente, a imagem recuperada foi comparada com a imagem original inicialmente fornecida para *treinamento*.

A segunda fase deste trabalho também produziu bons resultados: primeiro, porque todos os grupos de imagens utilizados puderam ser 100% aprendidos; segundo, porque na etapa de compactação/descompactação as distorções produzidas, ao se utilizar uma taxa alta de compactação, não inviabilizaram a compreensão das imagens.

## I.2 Descrição dos capítulos

No capítulo II é feito um estudo dos conceitos e questões básicas referentes às redes neuronais. São apresentados os modelos de Hopfield (Lógico e Analógico), bem como uma visão global do aprendizado conexionista. Buscando exemplificar algumas formas de aprendizado, faz-se ainda um estudo breve das regras de Hebb, Delta, Delta Generalizada e de um método de síntese bastante simples.

O capítulo III, que é a parte central deste trabalho, se volta para a

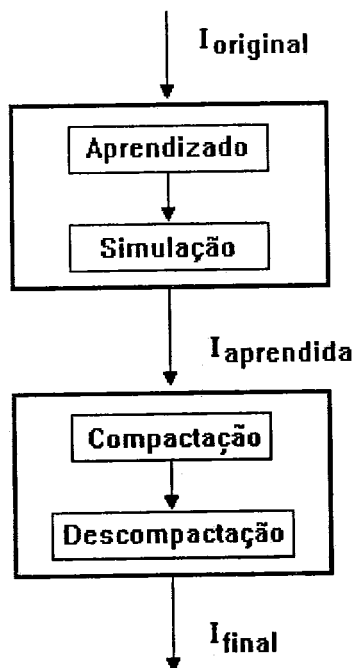


Figura I.1: Esquema geral do sistema

apresentação e estudo de três novos algoritmos de aprendizado para redes neuronais analógicas de Hopfield, que são avaliados utilizando-se uma série de dados aleatórios.

No capítulo IV faz-se um estudo básico e geral de alguns métodos de compactação de imagens, principalmente dos que utilizam redes neuronais.

Enfim, no capítulo V é feita uma aplicação de um dos métodos de aprendizado, através da compactação de vários grupos de imagens nas cores preto, cinza e branco. A qualidade da imagem obtida ao final do processo de compactação/descompactação é avaliada comparando-a à imagem original.

# Capítulo II

## Redes Neurais

### II.1 Introdução às Redes Neurais

Um dos objetivos centrais da pesquisa em neurobiologia é compreender como as propriedades biofísicas dos neurônios e a organização neuronal se combinam para prover velocidade e poder computacional satisfatórios. Portanto, será feito um breve estudo de Redes Neurais, correlacionando sempre que possível com o aspecto biológico.

#### II.1.1 Um pouco de Neurofisiologia

O sistema nervoso, mais precisamente o cérebro, é o grande responsável pelo processamento de informações no homem. Ele é formado por bilhões de células, chamadas neurônios, cujas funções básicas são integrar, armazenar e conduzir sinais elétricos. Assim como não existem duas células iguais, também não existem dois neurônios iguais, mas geralmente, um neurônio é formado (figura II.1) :

- por um *corpo celular* ou *soma*, dentro do qual se encontram os elementos vitais à célula;
- por uma *membrana celular*, que o envolve e separa do meio externo;
- por um único prolongamento do soma, chamado *axônio* que por sua vez se ramifica em *colaterais*, permitindo a relação do neurônio com várias outras células, mesmo as distantes;

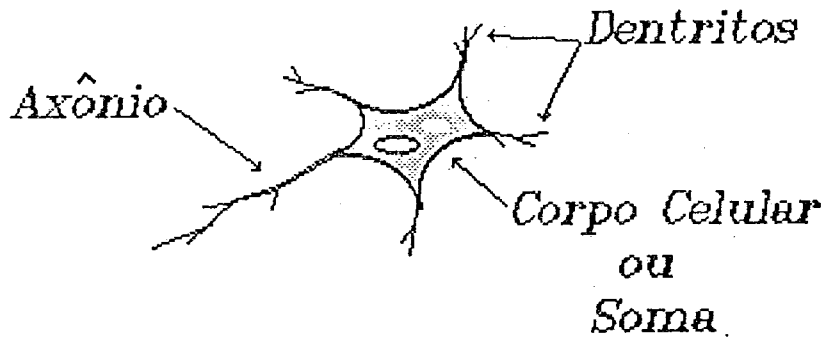


Figura II.1: A estrutura do neurônio biológico

- por vários prolongamentos do soma, denominados *dendritos*, onde em geral estão localizados pontos terminais dos axônios de outras células.

O ponto de conexão entre o axônio de um neurônio e o soma, dendrito ou até mesmo axônio de um outro neurônio é denominado *sinapse*, que pode ser respectivamente, classificada em axo-somática, axo-dendrítica ou axo-axônica.

Assim, os mais diversos neurônios se conectam através dos variados tipos de sinapses, formando redes neuronais, nas quais o fluxo de informação caminha sob a forma de impulsos nervosos. Tais impulsos são recebidos pelos dendritos da célula para serem integrados e só então transmitidos às outras células nervosas, através do axônio. Acredita-se [5] que o processo de aprendizado dos seres vivos resulte na alteração da capacidade de transmissão de informação através das sinapses, ou seja, na regulação das eficiências sinápticas.

Define-se *potencial de membrana* como a diferença de potencial elétrico entre o interior e o exterior da célula, que varia conforme a existência ou não de impulso nervoso no axônio da mesma. Quando não existe impulso nervoso, o potencial de membrana se encontra a um valor constante de  $-60mV$ , denominado *potencial de repouso*.

Ao ser a célula nervosa excitada, ocorre uma elevação no seu potencial de membrana (*despolarização*), que atinge um patamar, denominado *potencial limiar*. Neste ponto, deflagra-se um processo físico-químico que resulta numa

elevação rápida e breve do potencial de membrana a valores positivos bem acima do limiar, denominado *potencial de ação*. Após o término do potencial de ação, ocorre um processo de *hiperpolarização*, que representa o resgate das propriedades físico-químicas da membrana celular, resultando numa diminuição do potencial de membrana até sua chegada ao repouso. Durante essa fase de recuperação (*período refratário*), nenhum outro impulso nervoso pode ser provocado. Portanto, durante esse período, ao ser o neurônio continuamente estimulado, gera-se em seu axônio uma série de potenciais de ação numa frequência determinada pelo período refratário da célula. Essa sequência de potenciais de ação, conhecida como *trem de impulsos*, é a responsável pela transmissão de informação nas redes neuronais do sistema nervoso central.

Além do papel fundamental dos axônios que codificam informação em frequência, é importante comentar o papel dos dendritos, que são responsáveis pelo processo de decodificação da informação.

Suponhamos que um neurônio X tenha os seus dendritos conectados aos axônios dos neurônios Y e Z. Os potenciais de ação transmitidos através das sinapses pelos axônios de Y e Z, são somados e integrados pelos dendritos de X resultando na formação de um *potencial pós-sináptico*, que sofre pequenas variações (uma vez que na região das sinapses não existe potencial limiar) à medida que novos impulsos nervosos vão chegando à célula X.

## II.1.2 Alguns Modelos Conexionistas Básicos

Apesar da variedade dos modelos conexionistas, cada um com suas idéias e características próprias, existe uma estrutura básica [17] comum a todos eles que nos permite compreender, em linhas gerais, o paradigma Conexionista.

O primeiro passo, qualquer que seja o modelo de processamento paralelo, é definir um conjunto de unidades de processamento simples, os *neurônios*. Tais unidades, representadas por  $N_i$ , onde  $1 \leq i \leq n$ , podem estar ativas ou não, conforme o seu grau de excitação ou inibição. Assim, a cada neurônio  $N_i$ , pode-se associar um *estado de ativação*  $a_i(t)$  que representa o nível de excitação de  $N_i$  em

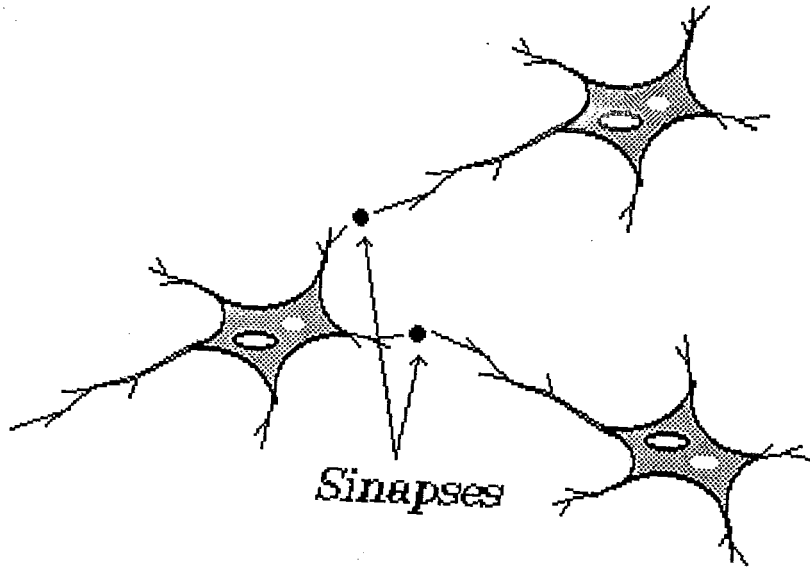


Figura II.2: Esquema de uma rede neuronal

um instante de tempo  $t$ .

Pode ser interessante saber o grau de excitação de todos os neurônios da rede em um instante de tempo. Assim, o *padrão de atividade*  $a(t) = (a_1(t), \dots, a_n(t))$  fornece, em algum instante de tempo  $t$  os estados de ativação de  $N_1, \dots, N_n$ .

Os neurônios emitem respostas  $o_i(t)$ , onde  $1 \leq i \leq n$ , que determinam os impulsos enviados pelo axônio do neurônio em função do atual estado de ativação  $a_i(t)$ . Dessa forma, podemos escrever a *regra de saída* como  $o_i(t) = f(a_i(t))$ .

Os neurônios se conectam através das sinapses (figura II.2), cujas eficiências são representadas pelos pesos das conexões  $w$  entre as células. Assim, se o axônio do neurônio  $N_i$  está ligado ao neurônio  $N_j$ , representamos por  $w_{j,i}$  o peso desta ligação, que pode ser positivo, negativo ou nulo, conforme a sinapse seja, respectivamente, excitatória, inibitória ou inexistente. É interessante notar ainda que, as sinapses excitatórias representam um sentido de cooperação entre as células, enquanto que as inibitórias representam competição entre elas.

Os pesos das conexões entre os neurônios que compõem uma rede neuronal podem ser armazenados numa matriz de conexões  $W$  denominada *padrão de conexão*. Tal matriz é mutável uma vez que as eficiências sinápticas estão relacionadas com a capacidade de aprendizado da rede, e representam o que os conexionistas chamam de inteligência. A obtenção de uma nova matriz de conexão provém da capacidade da rede se adequar a novas situações, ou seja, de aprender. Dessa forma, alguns modelos ainda dispõem de uma *regra de aprendizado*.

É necessário definir uma *regra de propagação* que combina todas as respostas que chegam à célula  $N_i$  num impulso total de entrada  $u_i(t)$ , sempre levando em conta a eficiência com que chega a informação. Assim,  $u_i(t) = g_i(o_i(t), W)$ . É interessante notar que  $u_i(t)$  corresponde ao *potencial pós-sináptico* do neurônio biológico.

Em um novo instante de tempo  $t + \Delta t$ , o estado de ativação do neurônio se altera em função do potencial pós-sináptico  $u_i(t)$  e do estado de ativação  $a_i$  no instante anterior  $t$ . Assim, a *regra de ativação* é dada por  $a_i(t + \Delta t) = h_i(a_i(t), u_i(t))$ .

Depois de estabelecida uma linguagem básica, podemos apresentar alguns modelos que revelam um pouco da evolução das redes neuronais.

O *neurônio linear* [17] [18] [19] é o modelo mais simples, no qual o estado de ativação, impulso total de entrada e a resposta são escalares reais irrestitos e as regras de ativação, propagação e saída são funções lineares.

As respostas  $o_j(t)$  provenientes de todos os neurônios da rede são recebidas por algum neurônio  $N_i$  através de um impulso total de entrada e são computadas através de uma *regra de propagação* que se encarrega de combinar estas respostas ou impulsos  $o_j(t)$ . Assim, a regra de propagação é dada por

$$u_i(t) = \sum_{j=1}^n w_{ij} o_j(t),$$

onde  $w_{ij}$  são as eficiências sinápticas e  $n$  o número de neurônios da rede.

A regra de ativação é linear em relação a  $u_i(t)$ , não dependendo do

estado anterior do neurônio:

$$a_i(t) = \alpha u_i(t), \quad \alpha \in \mathfrak{R}_+.$$

A regra de saída é a função identidade, uma vez que a resposta  $o_i(t)$  relativa ao neurônio  $N_i$  se confunde com seu estado de ativação  $a_i(t)$ :

$$o_i(t) = a_i(t).$$

Se combinarmos as regras de propagação, ativação e saída, obtemos a equação que traduz o comportamento do neurônio linear:

$$o_i(t) = \sum_{j=1}^n \alpha w_{i,j} o_j(t), \quad \alpha \in \mathfrak{R}_+. \quad (\text{II.1})$$

No entanto, devido a sua simplicidade, este modelo neuronal possui uma capacidade de aprendizado bem limitada, sendo superado por modelos não-lineares, que são mais adequados na modelagem de processos neurocomputacionais.

O *neurônio lógico*, desenvolvido por McCulloch em 1943, demonstrou um comportamento mais plausível diante das necessidades computacionais. Segundo o modelo, o neurônio possui apenas dois estados de ativação, 1 ou 0, conforme a célula nervosa esteja, respectivamente, excitada ou inibida. Embora as regras de propagação e saída sejam iguais as do modelo anterior, a regra de ativação (figura II.3) decide se o estado de ativação  $a_i(t)$  de um neurônio  $N_i$  vai ser ativado, conforme o impulso total de entrada  $u_i(t)$  supere ou não o potencial limiar  $\theta_i \geq 0$ , cuja função é idêntica a do potencial limiar do neurônio biológico.

Unindo as regras de ativação, propagação e saída chegamos a equação que representa o comportamento do neurônio lógico:

$$o_i(t) = T\left(\sum_{j=1}^n w_{i,j} o_j(t) - \theta_i\right),$$

onde  $T(x) = \begin{cases} 1 & , x > 0 \\ 0 & , x \leq 0 \end{cases}$  é dita *função limiar*.

O neurônio lógico, embora superior ao neurônio linear, ainda não apresenta uma não-linearidade adequada, uma vez que a sua regra de ativação possui apenas dois estados de ativação.



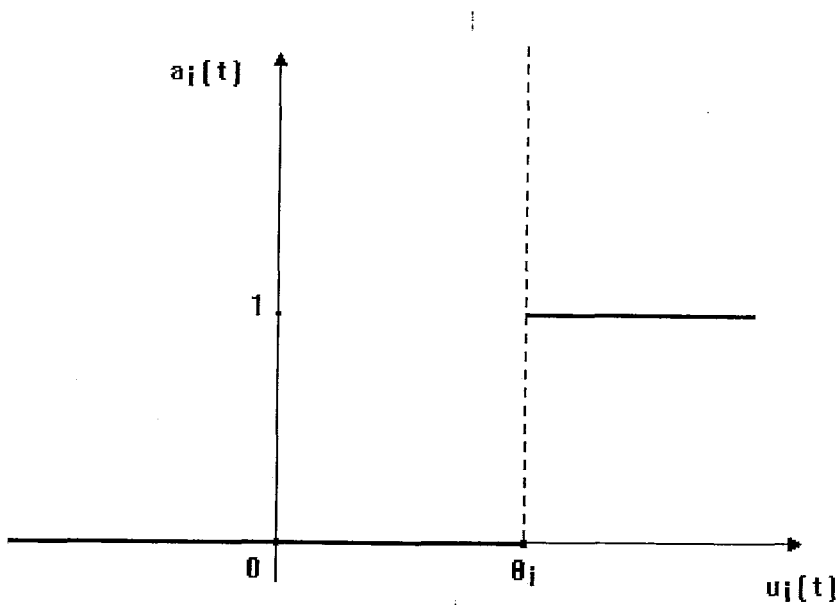


Figura II.3: Regra de ativação do neurônio lógico

O *neurônio semi-linear* é um modelo que superou o modelo lógico por possuir uma regra de ativação contínua, dita *sigmóide*:

$$a_i(t) = s(u_i(t)) = \frac{1}{1 + e^{-\gamma(u_i - \theta_i)}}, \quad (\text{II.2})$$

onde  $\gamma \in \mathfrak{R}_+$ ,  $\theta_i \geq 0$  e  $a_i(t) \in (0, 1)$ . Portanto, este neurônio possui infinitos estados de ativação contidos no intervalo real existente entre 0 e 1, o que representa uma identificação com o aspecto biológico da célula nervosa.

## II.2 O Modelo de Hopfield

### II.2.1 Introdução

Em 1982, Hopfield [2] propôs um modelo neuronal, de uma única camada, que se caracterizava, basicamente, pela estabilidade, ausente nos modelos até então, e pelas propriedades computacionais que podiam emergir como propriedades coletivas de sistemas com uma grande população de unidades simples de processamento, os neurônios. Tais propriedades coletivas — alguma capacidade de generalização, categorização, correção de erro e outras — na verdade, presentes também no segundo

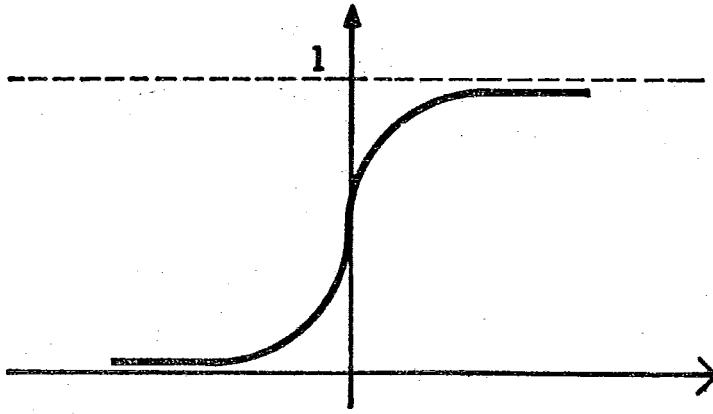


Figura II.4: Regra de ativação do neurônio semi-linear

modelo [3] proposto em 1984, acabam convergindo para uma propriedade fundamental na caracterização dos modelos em questão: a memória de acesso pelo conteúdo (CAM).

## II.2.2 O Modelo Lógico de Hopfield

Hopfield, utilizando os neurônios lógicos de McCullough e Pitts supôs que a comunicação entre os neurônios fôsse instantânea e recíproca, isto é, se o neurônio  $N_i$  estivesse fornecendo impulsos elétricos para o neurônio  $N_j$ , este também enviaria sinais elétricos para o neurônio  $N_i$ . Assim, as conexões entre  $N_i$  e  $N_j$  seriam simétricas, ou seja,  $w_{i,j} = w_{j,i}$ , para algum  $i$  e algum  $j$ .

Definido  $a_i(t)$  como o estado de ativação do neurônio  $N_i$  no instante de tempo  $t$ , temos que os neurônios apresentam apenas dois níveis de excitação (figura II.3), isto é,  $a_i(t) = 1$  (neurônio excitado) ou  $a_i(t) = 0$  (neurônio inibido).

A cada instante de tempo  $t$ , um e somente um neurônio  $N_i$  é convidado, aleatória e assincronicamente, a mudar o seu estado de ativação conforme o impulso total  $u_i(t) = \sum_{i,j} w_{i,j} a_j(t)$  propagado pelos demais neurônios da rede esteja acima ou não de um potencial limiar estipulado. Desta forma, o modelo segue o seguinte algoritmo estocástico:

se  $\sum_{i,j} w_{i,j} a_j(t) \geq \theta_i$  então

$$a_j(t) \leftarrow 1$$

caso contrário,  $a_j(t) \leftarrow 0$

onde  $w_{i,j}$  pode ser visto, em analogia com a biologia, como a sinapse do neurônio  $N_i$  para o neurônio  $N_j$  e  $\theta_i \geq 0$ , o potencial limiar referente ao neurônio  $N_i$ .

Partindo sempre da premissa que as conexões são simétricas, Hopfield desenvolveu uma função, denominada *energia computacional*<sup>1</sup>, fundamental para caracterizar a estabilidade do sistema. Segundo o físico e biólogo, a rede neuronal sempre evolui no sentido de minimizar esta função, e só cessa sua evolução dinâmica ao atingir um ponto de mínimo da referida função, onde enfim, estabiliza.

Sejam

$\Delta a_i(t) = (a_i(t + \Delta t) - a_i(t))$ , a variação do estado de ativação do neurônio  $N_i$  entre os instantes  $t$  e  $t + \Delta t$  e

$p_i = -\Delta a_i * (\sum_{i,j} w_{i,j} a_j(t) - \theta_i)$ , sendo  $w_{i,j} = w_{j,i}$ . É fácil notar que, qualquer que seja a variação  $\Delta a_i$ , o valor do produto  $p_i \leq 0$ . Portanto,  $p_i$  é um possível candidato à derivada da função de energia característica da rede neuronal que está sendo minimizada. Assim:

$$\Delta E_i(t) = -\Delta a_i * \left( \sum_{i,j} w_{i,j} a_j(t) - \theta_i \right)$$

$\Rightarrow$

$$E(t) = \int \sum_i \Delta E_i(t)$$

Integrando, teremos:

$$E(t) = \frac{-1}{2} \sum_i^n \sum_j^n w_{i,j} a_i(t) a_j(t) + \sum_i^n \theta_i a_i(t),$$

onde  $n$  é o número de neurônios da rede.

A partir do momento que cada rede neuronal está associada a uma superfície de energia representada num espaço vetorial n-dimensional, denominado

---

<sup>1</sup>Na verdade uma função de Liapunov [2].

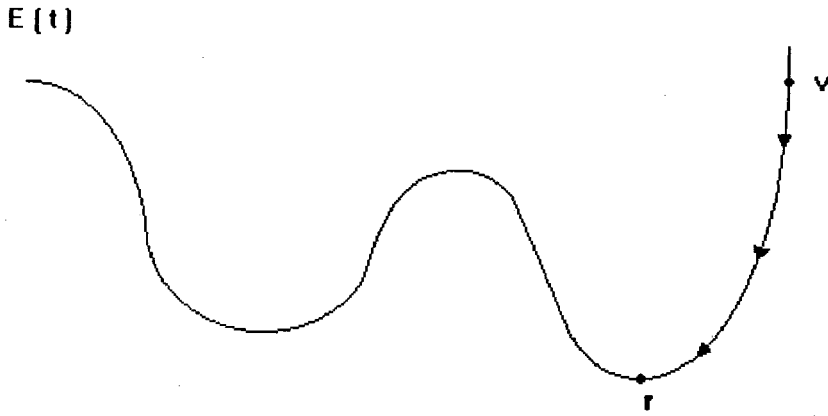


Figura II.5: Perfil da superfície de energia com dois pontos de mínimo

*espaço de estado*, temos que dado um ponto neste espaço de estado, a rede caminha sobre a superfície buscando os seus pontos mais baixos, cessando enfim sua evolução dinâmica ao atingir o ponto de mínimo mais próximo do início de sua trajetória (figura II.5). Assim, uma rede neuronal de Hopfield sempre evolui no sentido de minimizar a função de energia  $E$  característica até atingir um estado estável, isto é, um ponto de mínimo local da referida função.

Quando o fluxo gerado no espaço de estado produz um conjunto de pontos fixos estáveis, ou estados estáveis temos então, a configuração de uma memória associativa ou de uma memória de acesso pelo conteúdo (CAM). Se considerarmos um ponto estável como a informação relativa a uma certa memória então pontos próximos a este estado estável irão conter parte da informação a respeito desta memória. Assim, fornecida parte da informação, a rede evolui no espaço de estado descrevendo uma trajetória que diminui à medida que a energia computacional assume valores cada vez menores, e estabiliza enfim, quando atinge um ponto de mínimo local da superfície (estado estável). Assim, não importa a localização ou endereço da memória, mas sim o quanto se tem de informação, ou seja, o conteúdo.

### II.2.3 O Modelo Analógico de Hopfield

Insatisfeito com a limitação de seu primeiro modelo, Hopfield propôs em 1984 um novo modelo que trabalha não mais com o neurônio lógico, mas passa a considerar a continuidade das respostas dos neurônios da rede, presente naturalmente no neurônio biológico, além de levar em conta a característica capacitiva da membrana da celular.

Portanto, o segundo modelo de Hopfield utiliza o neurônio semi-linear, considera o fato de que os neurônios biológicos possuem relações contínuas de input-output e mantém a condição das sinapses simétricas, isto é,  $w_{i,j} = w_{j,i}$ . Assim, o aspecto degrau do estado de ativação do modelo lógico passa a ser, neste segundo modelo, mais contínuo, sendo caracterizado pela equação ( II.2).

A fim de se aproximar ainda mais do aspecto biológico da célula, Hopfield considerou o fato de que os neurônios biológicos possuem atrasos de tempo integrados devido a capacitância da membrana celular, e propôs uma nova regra de propagação. A regra de propagação pode ser definida pela equação de um capacitor que determina a taxa de variação de  $u_i$ . Assim:

$$C_i \frac{du_i(t)}{dt} = \sum_j w_{i,j} a_j(t) - \frac{u_i}{R_i} \quad (\text{II.3})$$

onde  $u_i(t) = s^{-1}(a_i(t)) = -\frac{1}{\gamma} \ln\left(\frac{1}{a_i} - 1\right) + \theta_i$ , isto é,  $u_i(t)$  é a inversa da sigmóide.

Intuitivamente, podemos encarar um neurônio qualquer  $N_i$  como um recipiente (figura II.6) com uma capacitância  $C_i$  de armazenar os sinais elétricos provenientes de outros neurônios, além de uma resistência  $R_i$  para o controle da capacidade máxima do recipiente.

Fixada uma capacitância  $C_i$  de armazenamento para  $N_i$ , tem-se que o volume do recipiente varia conforme uma taxa  $\frac{du_i(t)}{dt}$ , que determina uma elevação ou diminuição no conteúdo armazenado no recipiente. Neste caso então, o volume do recipiente pode ser dado por  $C_i \frac{du_i(t)}{dt}$ . Por outro lado, o volume real pode ser encarado como a soma das informações provenientes dos demais neurônios da rede, isto é,  $\sum_{i,j} w_{i,j} a_j(t)$  subtraindo-se a perda por fuga de corrente elétrica do neurônio  $N_i$  conforme a resistência  $R_i$ , ou seja,  $\sum_j w_{i,j} a_j(t) - \frac{u_i}{R_i}$ .

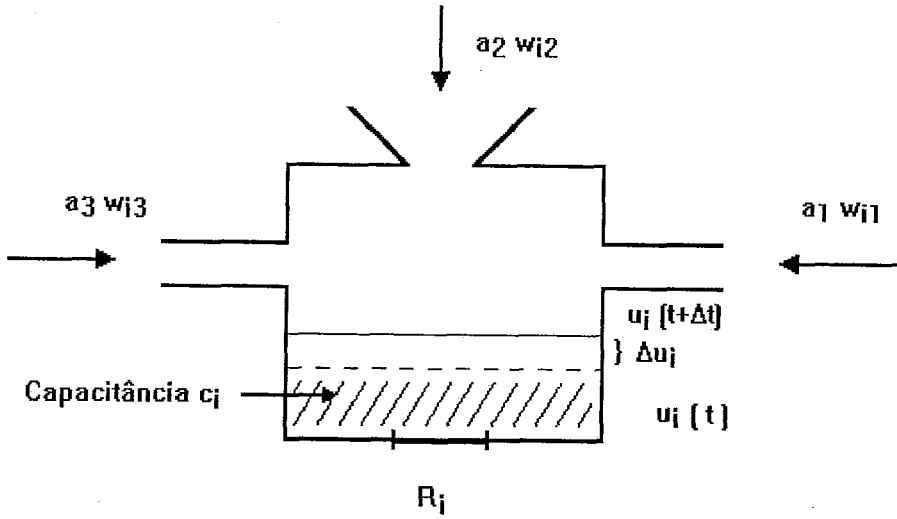


Figura II.6: Neurônio representado pela figura de um recipiente

Estabelecido o uso de novas regras de ativação (sigmoidal) e de propagação, é natural definirmos uma nova função de *energia computacional*. Considerando então que  $a_i = s_i(u_i)$ , temos pela figura ( II.4) que

$$\frac{du_i(t)}{dt} \frac{da_i(t)}{dt} > 0.$$

Daí e como  $C_i > 0$ , temos

$$-C_i \frac{du_i(t)}{dt} \frac{da_i(t)}{dt} < 0$$

Definindo  $p_i = -C_i \frac{du_i(t)}{dt} \frac{da_i(t)}{dt}$ , temos que  $p_i$  é um bom candidato a derivada da função de energia. Portanto,

$$\frac{dE_i(t)}{dt} = p_i = -C_i \frac{du_i(t)}{dt} \frac{da_i(t)}{dt}.$$

Daí e equação ( II.3), temos:

$$\frac{dE_i(t)}{dt} = \left( - \sum_j w_{i,j} a_j(t) + \frac{u_i}{R_i} \right) \frac{da_i(t)}{dt} < 0$$

$\Leftrightarrow$

$$dE_i = \left( - \sum_j w_{i,j} a_j(t) + \frac{u_i}{R_i} \right) da_i$$

$\Leftrightarrow$

$$\begin{aligned}
E_i &= \int -\sum_j w_{i,j} a_j da_i + \int \frac{u_i}{R_i} da_i \\
&\iff \\
E_i &= \frac{-1}{2} \sum_j w_{i,j} a_i a_j + \frac{1}{R_i} \int s^{-1}(a_i) da_i
\end{aligned}$$

Considerando todos os neurônios  $N_i$  tal que  $1 \leq i \leq n$ , temos

$$E = -\frac{1}{2} \sum_i^n \sum_j^n w_{i,j} a_i a_j + \sum_i^n \frac{1}{R_i} \int_0^{a_i(t)} s^{-1}(a_i) da_i, \quad (\text{II.4})$$

onde  $s^{-1}(a_i) = -\frac{1}{\gamma} \ln(\frac{1}{a_i} - 1) + \theta_i$ , como definido em ( II.3).

Portanto, o Modelo Contínuo de Hopfield além de preservar as propriedades emergentes já presentes no Modelo Lógico, se aproximou ainda mais do comportamento do neurônio biológico e satisfaz uma das características básicas do Conexionismo ausente em seu primeiro modelo, isto é, o paralelismo.

## II.3 Aprendizado em Redes Neurais

### II.3.1 Uma idéia de Aprendizado

Quando se usa o termo aprendizado nos vem logo a idéia de evolução, desenvolvimento ou aperfeiçoamento adquirido através de algum tipo de transformação ou mudança.

Conforme a linha de pesquisa a que nos ligamos, o significado de aprendizado acaba ganhando um enfoque mais específico. Em neurobiologia, por exemplo, Shepherd define aprendizado como uma mudança adaptativa no comportamento causada por experiência.

Em Inteligência Artificial, Carbonell define aprendizado como a habilidade de realizar novas tarefas que não podiam ser realizadas anteriormente, ou realizar melhor tarefas antigas, como resultado de mudanças produzidas no processo de aprendizado.

Tais idéias são compartilhadas pelo Conexionismo, apesar de ganharem uma visão mais própria. Segundo o Paradigma Conexionista, a inteligência está

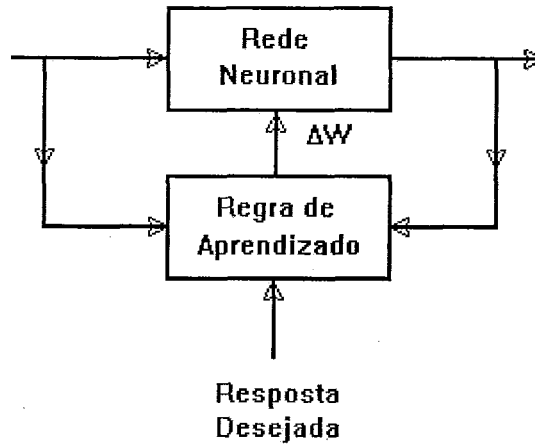


Figura II.7: Aplicação da regra de aprendizado a uma rede neuronal

armazenada nas sinapses, isto é, nas conexões entre os neurônios. Portanto, pode-se definir aprendizado como o tipo de mudança capaz de aumentar as eficiências sinápticas em alguma tarefa realizada pela rede neuronal. Ou seja, o aprendizado consiste basicamente, em determinar as eficiências sinápticas, isto é, um padrão de conexão  $W$  satisfatório que permita associar um estímulo fornecido (padrão de atividade inicial) a alguma resposta (padrão de atividade final) a que desejamos chegar.

A rede neuronal pode aprender um determinado padrão mediante a aplicação iterativa de alguma regra (regra de aprendizado) até a obtenção de um padrão de conexão satisfatório (figura II.7) ou através de métodos (métodos de síntese) que calculem o padrão de conexão de uma forma direta, sem necessidade de qualquer "sessão" de aprendizado (figura II.8).

## II.3.2 Classificação do Aprendizado em Redes Neurais

### Paradigmas de Aprendizado

Segundo Rumelhart e Zipser [19] [26] existem no mínimo quatro paradigmas que podem ser seguidos pelos modelos conexionistas:



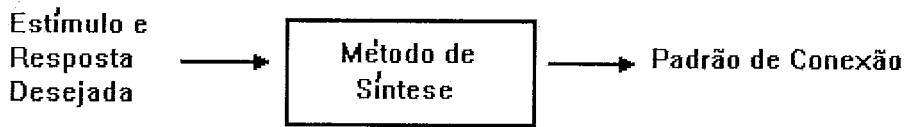


Figura II.8: Aplicação do método de síntese

### i) Auto-Associador

Conforme este paradigma, um ou mais padrões são supostamente "armazenados" no sistema. Assim, se parte de algum padrão inicialmente armazenado for fornecido ao sistema, é possível recuperarmos completamente o padrão original. Analogamente, se for fornecido um padrão que represente uma distorção de algum padrão original, este mais uma vez poderá ser totalmente recuperado. Isto caracteriza uma forma de *generalização*, uma vez que padrões semelhantes ao original podem ser mapeados no padrão original. Auto-associadores podem ser treinados por exemplo com a regra de Hebb e a regra Delta, que veremos mais adiante.

A arquitetura de um auto-associador consiste em um único conjunto de unidades completamente conectadas, de forma que qualquer unidade  $N_i$ , onde  $1 \leq i \leq n$ , está ligada a todas as outras unidades da rede.

### ii) Associador de Padrões

Neste paradigma apresenta-se um conjunto de pares de padrões ao sistema. O associador de padrões terá apenas que aprender as associações entre os padrões de entrada e saída, isto é, associar um padrão do par ao seu companheiro.

A arquitetura característica de um associador de padrões é um conjunto de unidades divididas em uma camada de entrada, uma ou mais camadas invisíveis e uma camada de saída.

### iii) Classificador

Segundo este paradigma existe um número fixo de categorias nas

quais os estímulos ou padrões de input deverão ser classificados. O objetivo aqui é que o sistema aprenda a classificar os estímulos fornecidos em categorias, a fim de que posteriormente, ao ser fornecido um estímulo particular ou um estímulo um pouco distorcido, o sistema seja capaz de classificá-lo corretamente.

#### *iv) Detector de Regularidades*

Ao sistema é apresentado um conjunto de padrões de entrada ou estímulos, sendo que a cada estímulo  $e_i$  está associado uma probabilidade  $p_i$ . O sistema deve então, "descobrir" estatisticamente, as características ou regularidades mais importantes dos padrões de entrada a fim de categorizá-los. É importante notar que, não existe um conjunto fixo de classes pré-estabelecido, portanto o próprio sistema é que deve determinar as categorias relevantes.

Apesar de existirem quatro paradigmas com características bem definidas, pode-se notar que todos se concentram basicamente na idéia do associador de padrões.

## **Níveis de Supervisionamento**

Existem basicamente três formas de aprendizado quanto ao supervisionamento:

#### *i) Aprendizado Supervisionado*

Neste tipo de aprendizado existe um professor que fornece os padrões de saída a rede e determina a medida de eficiência da rede neuronal na execução de alguma associação.

#### *ii) Aprendizado por Reforço*

É uma forma intermediária de supervisionamento em que os pesos das eficiências sinápticas são alterados conforme um sinal externo fornecido ao sistema. Assim, o sinal pode ser positivo, representando uma recompensa ou ainda pode ser negativo, representando uma punição.

### iii) Aprendizado não-supervisionado

A rede neuronal estabelece suas próprias associações, sem que nenhum tipo de sinal ou informação seja fornecida ao sistema.

Em termos práticos fica difícil assegurar que esse ou aquele algoritmo de aprendizado é supervisionado ou não, pois o que acontece muitas vezes é que um mesmo algoritmo possui fases com níveis diferentes de supervisionamento.

### II.3.3 Regras de Aprendizado

O aprendizado em redes neuronais por meio de regras é feito aplicando-se a regra iterativamente ao longo de uma sessão de aprendizado. O projetista, elemento externo ao sistema, fornece o *estímulo*  $s$ , a *saída efetiva*  $o$  e a *saída desejada*  $d$ , que são integrados mediante a regra escolhida a fim de obter uma matriz de pesos  $W$  satisfatória, que varia ao longo do processo de aprendizado. Assim:

$$\Delta W = R(s, o, d).$$

Conforme o Paradigma Conexionista de paralelismo no tempo e distribuição no espaço, qualquer regra de aprendizado deve conter duas propriedades básicas: propriedade de instantaneidade e propriedade de localização. A instantaneidade significa que a regra não necessita de informações sobre os estados passados ou futuros da rede, enquanto que a localização defende que a regra de aprendizado deve ser calculada com dados espacialmente localizados, isto é, ao se alterar a ligação  $w_{i,j}$ , deve-se necessitar apenas de informações referentes aos neurônios  $i$  e  $j$ .

Escolhida a regra, o processo de aprendizado deve ser realizado durante um tempo adequado e por um número finito de iterações ao fim das quais o padrão de conexão deve estar bem definido. Mas isso nem sempre ocorre, pois existem regras que geram tanto oscilação quanto indefinição na geração da matriz de pesos. Além disso, as regras de aprendizado são limitadas se considerarmos que a rede neuronal reduz sua capacidade de aprendizado à medida que novos padrões de atividade são incorporados ao processo de aprendizado. Isso ocorre devido ao *fenômeno de interferência* [14] que provoca uma degradação na *performance* da

rede neuronal em relação ao que foi anteriormente aprendido. Isto é, a alteração dos pesos com vistas ao novo aprendizado pode destruir a capacidade da rede de se estabilizar em determinados padrões já considerados aprendidos.

## A Regra de Hebb

Em 1949, Donald Hebb [16] [17] [18] [19] lançou em sua obra *Organization of Behavior* uma teoria neurofisiológica, que apesar de desprezada na época, representou influência marcante para as teorias conexionistas, além de ser vista por alguns como uma descrição do aprendizado a nível celular. Dessa teoria, podemos destacar um importante paradigma do ponto de vista comportamental:

*”Quando o axônio de uma célula A está próximo o bastante de uma célula B, de forma a excitá-la, e o faz repetidamente, algum processo de crescimento sináptico, ou mudança metabólica, ocorre em uma ou ambas as células no sentido de aumentar a capacidade da célula A excitar a B.”*

Portanto, segundo Hebb, uma determinada conexão só é alterada se as células que se mantêm conectadas estiverem excitadas. Convém lembrar, no entanto, que a concepção de alteração sináptica como base para o aprendizado é mais antiga que o postulado de Hebb. Freud, por exemplo, no século passado já defendia tal idéia.

Apesar de Hebb ter formulado um princípio fundamental para a concepção do aprendizado conexionista, foi Eccles quem definiu pela primeira vez uma formulação matemática baseada em suas idéias. Segundo a regra de aprendizado definida por Eccles, a variação da eficiência sináptica  $\Delta w_{i,j}$  entre dois neurônios  $N_i$  e  $N_j$  é diretamente proporcional ao produto de seus estados de ativação em um instante qualquer de tempo  $t$ :

$$\Delta w_{i,j} = \eta a_i(t) a_j(t),$$

onde  $0 < \eta \ll 1$  é denominada *taxa de aprendizado*.

Existem outras extensões para a *regra de Hebb* que foram surgindo intuitiva e heurísticamente, conforme motivações práticas e biológicas.

## A Regra Delta

Na década de 60, Widrow e Hoff [20] formalizaram uma regra bastante conhecida hoje, a *regra Delta* ou *regra de Widrow-Hoff*.

A regra Delta trabalha com redes de apenas duas camadas<sup>2</sup> - uma camada de entrada e outra de saída - de neurônios lineares, onde o fluxo de informação caminha apenas da camada de entrada para a de saída.

Consideremos uma rede neuronal de duas camadas com  $n$  neurônios cada,  $o$  a *saída efetiva* da rede e  $t$ , o *target*, ou seja, a *saída desejada*, onde  $o \in \mathbb{R}^n$ ,  $t \in \mathbb{R}^n$ . A idéia consiste em comparar as saídas efetiva e desejada para cada unidade  $i$ , onde  $1 \leq i \leq n$ , e continuar gerando novas saídas  $o_i$ , até que a diferença entre  $o_i$  e  $t_i$  represente um mínimo satisfatório.

Seja então,

$$\epsilon = \frac{1}{2} \sum_{i=1,n} (t_i - o_i)^2, \quad (\text{II.5})$$

uma medida de erro da rede neuronal, onde  $t, o \in \mathbb{R}^n$ .

Assim, o objetivo da *Regra Delta* consiste em minimizar  $\epsilon$ , isto é, encontrar um conjunto de pesos que minimize esta função. Para tanto, ela recorre ao Método do Gradiente. Então, derivando  $\epsilon$  com respeito a  $w_{i,j}$ , onde  $1 \leq i, j \leq n$ , teremos:

$$\frac{\partial \epsilon}{\partial w_{i,j}} = \frac{\partial \epsilon}{\partial o_i} \frac{\partial o_i}{\partial w_{i,j}} \quad (\text{II.6})$$

Da equação ( II.5) temos:

$$\frac{\partial \epsilon}{\partial o_i} = -(t_i - o_i) \quad (\text{II.7})$$

Da equação ( II.1) que descreve o comportamento do neurônio linear , podemos obter:

$$\frac{\partial o_i}{\partial w_{i,j}} = \alpha s_j, \quad (\text{II.8})$$

---

<sup>2</sup>Qualquer rede neuronal com neurônios lineares pode ser substituída por uma equivalente com apenas duas camadas [20]

onde  $\alpha \in \mathfrak{R}_+$  e  $s \in \mathfrak{R}^n$  é o estímulo fornecido à camada de entrada.

Portanto, de ( II.6), ( II.7) e ( II.8), temos que o gradiente do erro  $\epsilon$  é dado por:

$$\frac{\partial \epsilon}{\partial w_{i,j}} = -\alpha(t_i - o_i)s_j, \quad \alpha \in \mathfrak{R}_+$$

Pode-se então estabelecer a *regra Delta* como

$$w_{i,j}(t + \Delta t) = w_{i,j}(t) + \eta(t_i - o_i)s_j,$$

onde  $0 < \eta \ll 1$ . Ou seja,

$$\Delta w_{i,j} = \eta(t_i - o_i)s_j,$$

onde  $0 < \eta \ll 1$  é a *taxa de aprendizado*.

É interessante notar que, quando  $t_i = o_i$ , para algum  $i$ , isto é, quando a resposta efetiva atinge a resposta desejada, a variação da eficiência sináptica é nula, o que significa que a regra delta fornece um padrão de conexão bem determinado e estável.

## Regra Delta Generalizada

A *Regra Delta* apresenta limitações, como por exemplo, o fato de só poder computar funções linearmente separáveis, não podendo computar funções simples como por exemplo, a função XOR (ou exclusivo). No entanto, se considerarmos uma rede neuronal não mais com neurônios lineares e com apenas duas camadas, uma de entrada e outra de saída (*representação externa*), mas sim com unidades invisíveis pertencentes a uma ou mais *camadas invisíveis*, isto é, uma rede com *representação interna*, este problema pode ser facilmente resolvido.

A princípio, a *Regra Delta Generalizada* foi desenvolvida independentemente, por Parker (1985), Le Cun (1985), e por Rumelhart, Hinton e Williams, do grupo PDP [20] e veio complementar a lacuna deixada pela Regra de Widrow-Hoff na solução de alguns problemas.

Segundo Rumelhart [20], a *Regra Delta Generalizada* ou *Aprendizado por Propagação de Erros* utiliza neurônios semi-lineares dispostos em múltiplas ca-

madras - uma de entrada, uma de saída e uma ou mais camadas invisíveis (intermediárias) - onde o estímulo caminha sempre da entrada para a saída da rede (figura II.9).

Como na seção anterior, definimos o erro do aprendizado relativo a um padrão  $p$  como

$$\epsilon_p = \frac{1}{2} \sum_p (t_i - o_i)^2, \quad t \in \mathfrak{R}^n, o \in \mathfrak{R}^n.$$

Novamente, nosso objetivo consiste em minimizar  $\epsilon_p$  em relação a  $w_{i,j}$ , ou seja, obter uma mudança de pesos da forma

$$\Delta_p w_{i,j} = -k \frac{\partial \epsilon_p}{\partial w_{i,j}},$$

onde  $k$  é a constante de proporcionalidade. Então:

$$\frac{\partial \epsilon_p}{\partial w_{i,j}} = \frac{\partial \epsilon_p}{\partial u_{p,i}} \frac{\partial u_{p,i}}{\partial w_{i,j}} \quad (\text{II.9})$$

Daí e como  $u_{p,i} = \sum_j w_{i,j} o_{p,j}$ , temos:

$$\frac{\partial u_{p,i}}{\partial w_{i,j}} = o_{i,p} \quad (\text{II.10})$$

Definindo

$$\delta_{p,i} = -\frac{\partial \epsilon_p}{\partial u_{p,i}}, \quad (\text{II.11})$$

podemos concluir de ( II.9), ( II.10) e ( II.11) que

$$-\frac{\partial \epsilon_p}{\partial w_{i,j}} = \delta_{p,i} o_{p,i},$$

ou melhor,

$$\Delta_p w_{i,j} = \eta \delta_{p,i} o_{p,j}, \quad (\text{II.12})$$

onde  $0 < \eta \ll 1$ .

O valor de  $\delta_{p,i}$  é facilmente visualizado quando só existem camadas externas, no entanto, como se deve calcular o valor de  $\delta_{p,i}$  quando existirem camadas invisíveis ? Pela equação ( II.11) podemos ter que,

$$\delta_{p,i} = -\frac{\partial \epsilon_p}{\partial u_{p,i}} = -\frac{\partial \epsilon_p}{\partial o_{p,i}} \frac{\partial o_{p,i}}{\partial u_{p,i}}, \quad (\text{II.13})$$

onde  $\frac{\partial o_{p,i}}{\partial u_{p,i}} = f'_i(u_{p,i})$ , uma vez que  $o_{p,i} = f_i(u_{p,i})$ .

Ao se analisar  $\frac{\partial \epsilon_p}{\partial o_{p,i}}$ , surgem dois casos:

*Caso 1.* O neurônio  $i$  é da camada de saída. Neste caso,

$$\frac{\partial \epsilon_p}{\partial o_{p,i}} = -(t_{p,i} - o_{p,i})$$

Assim,

$$\delta_{p,i} = (t_{p,i} - o_{p,i})f'_i(u_{p,i})$$

Portanto, daí e ( II.12):

$$\Delta w_{i,j} = \eta(t_{p,i} - o_{p,i})f'_i(u_{p,i}o_{p,j}),$$

onde  $0 < \eta \ll 1$ .

*Caso 2.* O neurônio  $i$  não é da camada de saída. Neste caso,

$$\frac{\partial \epsilon_p}{\partial o_{p,i}} = \mathcal{F}(o_i),$$

para algum neurônio  $i$  de alguma camada  $c$  intermediária. Assim:

$$\frac{\partial \epsilon_p}{\partial o_i} = \frac{\partial \epsilon_p}{\partial u_{p,1}} \frac{\partial u_{p,1}}{\partial o_i} + \dots + \frac{\partial \epsilon_p}{\partial u_{p,j}} \frac{\partial u_{p,j}}{\partial o_i} + \dots + \frac{\partial \epsilon_p}{\partial u_{p,k}} \frac{\partial u_{p,k}}{\partial o_i},$$

onde  $1 \leq j \leq k$  são neurônios da camada superior, isto é, da camada  $c + 1$ .

Portanto:

$$\frac{\partial \epsilon_p}{\partial o_i} = \sum_k \frac{\partial \epsilon_p}{\partial u_{p,k}} \frac{\partial u_{p,k}}{\partial o_i} \quad (\text{II.14})$$

Segundo a regra de propagação do neurônio semi-linear, temos que

$\frac{\partial u_{p,k}}{\partial o_i} = w_{k,i}$ . Daí e equação ( II.14), temos:

$$\frac{\partial \epsilon_p}{\partial o_i} = - \sum_k \delta_{p,k} w_{k,i} \quad (\text{II.15})$$

Das equações ( II.12), ( II.13) e ( II.15), concluímos que

$$\Delta_p w_{i,j} = \eta \sum_k w_{k,i} f'_i(u_{p,i}) o_{p,j},$$



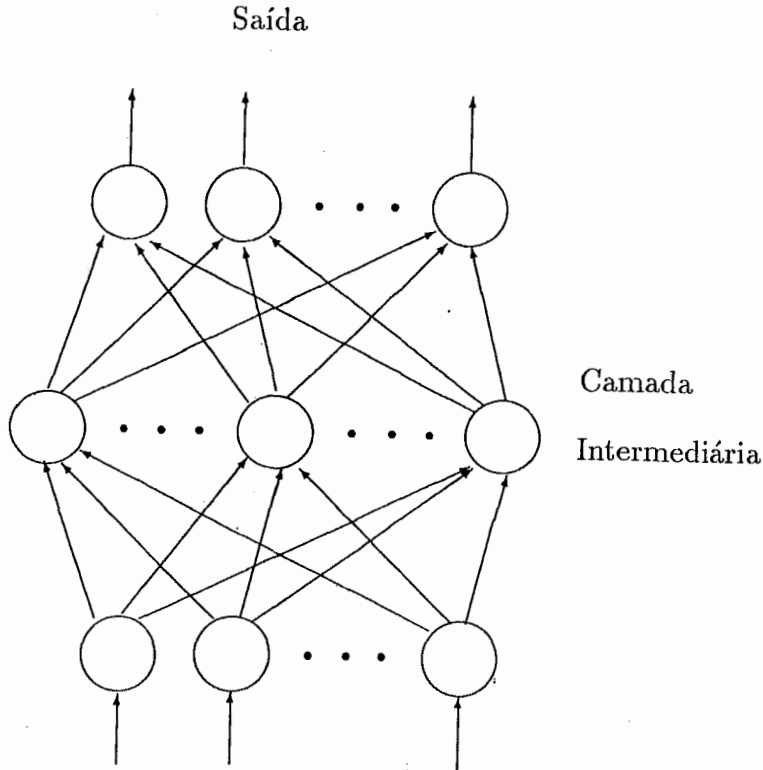


Figura II.9: Arquitetura da rede "Backpropagation"

ou seja,

$$\Delta_p w_{i,j} = \eta f'_i(u_{p,i}) \sum_k \delta_{p,k} w_{k,i} o_{p,j},$$

onde  $0 < \eta \ll 1$  e  $f'$  é a derivada da função  $f$  de ativação.

A aplicação do *Aprendizado por Propagação de Erros* envolve duas fases:

- i*) Na primeira fase (*FORWARD*), a entrada é fornecida à rede e propagada sempre no sentido *camada inferior* → *camada superior*, até o cálculo da saída  $o_{p,i}$  para cada unidade  $i$ ;
- ii*) A segunda fase (*BACKWARD*), envolve a propagação dos sinais de erro para as camadas anteriores, realizando-se enfim, a mudança dos pesos. Essa segunda fase é responsável por um outro nome pelo qual o algoritmo também é conhecido: *backpropagation*.

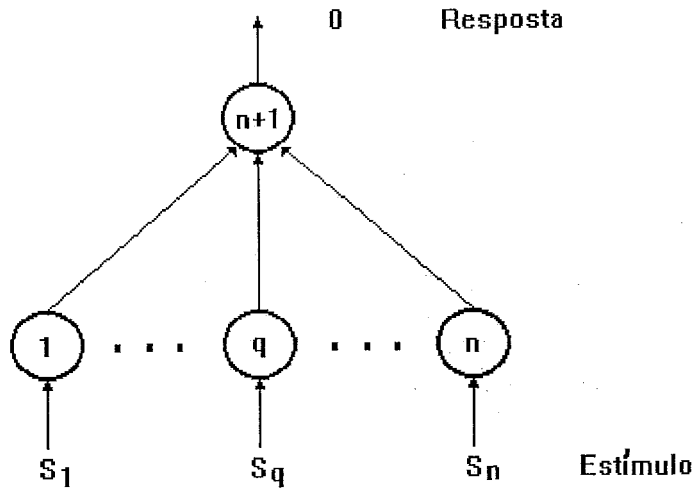


Figura II.10: Classificador neuronal de padrões

### II.3.4 Métodos de Síntese

Um outro meio de se obter as eficiências sinápticas é gerando de uma única vez o padrão de conexão, através de um procedimento matemático direto e preciso - os Métodos de Síntese. Mas, apenas modelos conexionistas simplificados podem ser utilizados por estes métodos, como por exemplo, redes neuronais com neurônios lineares.

Abordaremos um exemplo bem simples apenas para esclarecer um pouco mais a idéia de um método de síntese (figura II.10). Vejamos então o caso de um classificador de padrões, cuja função é fornecer resposta  $o \in \mathfrak{R}$ , representante de uma classe, quando estimulado por uma entrada  $s \in \mathfrak{R}^n$ , integrante desta classe.

A equação do neurônio  $N_n + 1$  que emite a resposta  $o$  é dada por

$$o = \sum_{j=1,n} w_j s_j, \quad (\text{II.16})$$

onde  $w_j$  é a eficiência sináptica entre os neurônios  $N_j$  e  $N_n + 1$ . Em termos vetoriais temos o produto escalar

$$O = W^T S,$$

onde  $W, S \in \mathfrak{R}^n$ . Considerando  $\|S\|^2 = S^T S = 1$ , podemos encontrar o padrão de conexão  $W$  que resolve o classificador de padrões, ou seja,  $W = OS$ . Portanto, o método de síntese para o classificador de padrões é

$$W = OS$$

onde  $\|S\| = 1$ .

# Capítulo III

## Três Novos Algoritmos de Aprendizizado

### III.1 Considerações iniciais

Em geral, o aprendizado em redes neurais consiste apenas na obtenção de um padrão de conexão  $W$  que permita associar padrões eficientemente. No entanto, serão apresentados três novos algoritmos de aprendizado para redes neurais analógicas de Hopfield, dos quais dois não se restringem tão-somente a obter um matriz de pesos satisfatória. Assim, buscando maior liberdade, que em outros tipos de redes pode significar apenas um aumento do número de conexões, trabalharemos também variando o potencial limiar  $\theta$  e o ganho  $\gamma$ , dos estados de ativação dos neurônios da rede, a fim de contribuir para a realização de um aprendizado mais eficiente.

Estes três novos algoritmos trabalham, respectivamente, variando  $W$  [1],  $W$  e  $\theta$  e por fim,  $W$ ,  $\theta$  e  $\gamma$ . No entanto, apenas este último algoritmo será abordado de forma mais completa, pois além de constituir uma *extensão* dos algoritmos anteriores, ele demonstrou, através dos testes realizados, uma *performance* superior.

Intuitivamente, o que se propõe é encontrar uma matriz de pesos  $W$ , um potencial limiar  $\theta$  e um ganho  $\gamma$  tal que dados  $m$  padrões de atividade  $v^1, v^2, \dots, v^m$  se consiga ajustar a função de energia  $E$ , definida pela equação ( III.1), a todos estes pontos (figura III.1), de modo que eles constituam mínimos locais da referida função.

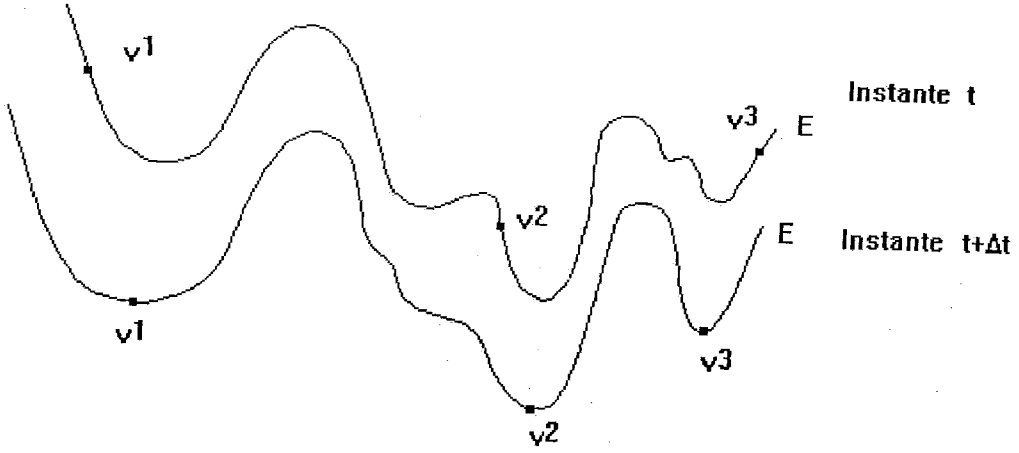


Figura III.1: Dois perfis da função de energia  $E$  com três pontos de mínimo no instante de tempo  $t + \Delta t$

Consideremos uma rede neuronal analógica de Hopfield com  $n$  neurônios  $N_1, \dots, N_n$ ,  $m$  padrões de atividade, capacitância  $C$ , resistência  $R$  e impulso total de entrada  $u_i$ , relativo ao neurônio  $N_i$ , dado pela equação:

$$C \frac{du_i(t)}{dt} = \sum_j w_{i,j} a_j(t) - \frac{u_i(t)}{R},$$

onde  $W = [w_{i,j}]$ . Foi visto no capítulo anterior que  $v_i \in (0, 1)$  representa o valor de ativação do neurônio  $N_i$  e  $w_{i,j} = w_{j,i}$ , os pesos sinápticos, significam que o padrão de conexão  $W$  é uma matriz simétrica.

Assim, para uma matriz  $W$ , limiar  $\theta$ , ganho  $\gamma$  e padrão de atividade  $v \in (0, 1)^n$  a rede possui energia

$$E(W, \theta, \gamma, v) = -1/2 v^T W v + 1/R \sum_{i=1}^n \int_{y=0}^{y=v_i} s^{-1}(y) dy, \quad (\text{III.1})$$

onde  $s^{-1}(y) = -\frac{1}{\gamma} \ln\left(\frac{1}{y} - 1\right) + \theta$ .

Um padrão  $v$  é *mínimo local* de  $E(W, \theta, \gamma, v)$ , se e somente se,

1.  $\nabla_v E(W, \theta, \gamma, v)^1 = 0$  e
2. se a matriz hessiana<sup>2</sup>  $H(W, \theta, \gamma, v)$  de  $E(W, \theta, \gamma, v)$  com respeito a  $v$  é do tipo *definida semi-positiva*, isto é, se  $x^T H(W, \theta, \gamma, v) x \geq 0$ , para qualquer  $x \in \mathbb{R}^n$ .

Em termos práticos, é difícil satisfazer a condição 2 para qualquer

<sup>1</sup> $\nabla_v$  significa o gradiente de  $E$  com respeito a  $v$

<sup>2</sup>A matriz hessiana é formada pelas derivadas segundas

$x \in \mathfrak{R}^n$ , principalmente pelo fato de  $\mathfrak{R}^n$  ser incontável. Portanto, nos deteremos apenas a primeira condição, apesar de não ser possível assegurar a minimalidade da função de energia, isto é, que um dado padrão de atividade  $v$  é de fato um ponto de mínimo da equação (III.1).

O problema então consiste em, dados  $m$  padrões de atividade  $v^1, \dots, v^m \in (0,1)^n$ , determinar uma matriz de pesos  $W$  simétrica, um novo potencial limiar  $\theta$  e um novo ganho  $\gamma$ , de forma que  $v^1, \dots, v^m$  representem pontos de mínimo (ou pontos próximos aos pontos de mínimo) da superfície definida pela função  $E$  de energia (equação (III.1)).

## III.2 Algumas considerações matemáticas

Seja  $S^{-1}(v)$  um vetor cuja  $i$ -ésima componente é dada por  $s^{-1}(v_i)$ , onde  $v_i \in (0,1)$ . Levando em conta apenas a primeira condição da definição dada anteriormente, pode-se ter que se  $v$  é um padrão estável de  $E(W, \theta, \gamma, v)$  então  $v$  é um padrão estável sobre uma matriz de conexões  $W$ , um potencial limiar  $\theta$ , e um ganho  $\gamma$ . Portanto, se  $v$  é um padrão estável de  $E(W, \theta, \gamma, v)$  então  $\nabla_v E(W, \theta, \gamma, v) = 0$ , ou seja,  $Wv - \frac{S^{-1}(v)}{R} = 0$ .

Mas, temos que

$$\begin{aligned} Wv - \frac{S^{-1}(v)}{R} &= 0 \\ \iff \\ \langle Wv - \frac{S^{-1}(v)}{R}, Wv - \frac{S^{-1}(v)}{R} \rangle &= 0 \\ \iff \\ \|Wv - \frac{S^{-1}(v)}{R}\|^2 &= 0 \end{aligned}$$

Logo, dados  $m$  padrões de atividade, eles são todos simultaneamente estáveis se,

$$\sum_{p=1}^m \|Wv^p - \frac{S^{-1}(v^p)}{R}\|^2 = 0$$

Ou seja,

$$\epsilon = \sum_{p=1}^m (Wv^p - \frac{S^{-1}(v^p)}{R})^T (Wv^p - \frac{S^{-1}(v^p)}{R}),$$

sendo  $\epsilon = 0$ . Para facilitar, seja  $\delta^p = Wv^p - \frac{S^{-1}(v^p)}{R}$ . Então, da equação anterior podemos ter que

$$\epsilon = \sum_{p=1, m} (\delta^p)^T \delta^p, \quad e \quad \epsilon = 0.$$

Nosso problema agora se reduz a obter um  $\epsilon$  mínimo a fim de que  $v^1, v^2, \dots, v^m$  sejam enfim, estáveis (ou estejam próximos de pontos estáveis). Dessa forma, é preciso minimizar  $\epsilon = f(W, \theta, \gamma, v^1, v^2, \dots, v^m)$  com respeito a  $W$ ,  $\theta$  e  $\gamma$ , ou seja, obter  $\min_W \epsilon$ ,  $\min_{\theta} \epsilon$  e  $\min_{\gamma} \epsilon$ .

Minimizar  $\epsilon$  com respeito à  $W$  significa calcular  $\nabla_W \epsilon$ , isto é,  $\nabla_W f^3$ .

Desta forma,

$$\nabla_W f = \sum_{p=1, m} \delta^p (v^p)^T + v^p (\delta^p)^T.$$

Analogamente,  $\min_{\theta} \epsilon$  significa obter o valor de  $\nabla_{\theta} \epsilon$ . Assim,

$$\nabla_{\theta} f = \sum_{p=1}^m \frac{-2}{R} \delta^p.$$

E por fim, para minimizar  $\epsilon$  com respeito a  $\gamma$ , basta calcular

$$\nabla_{\gamma} f = \sum_{p=1}^m -\frac{2}{R\gamma^2} \delta^p \ln\left(\frac{1-v^p}{v^p}\right), \quad \gamma \in \mathfrak{R}_+^n.$$

### III.3 Os algoritmos

‘A medida que se aumentam os graus de liberdade da rede neuronal, um novo algoritmo surge. Portanto, o primeiro algoritmo criado foi estendido e deu origem ao aprendizado em  $W$  e  $\theta$ . Este por sua vez foi expandido, gerando o algoritmo de aprendizado em  $W$ ,  $\theta$  e  $\gamma$  (figura III.2).

<sup>3</sup>Na verdade, o  $\nabla_W f$  é uma matriz  $n \times n$ , onde o elemento  $i, j$  é dado por  $\frac{\partial f}{\partial w_{ij}}$

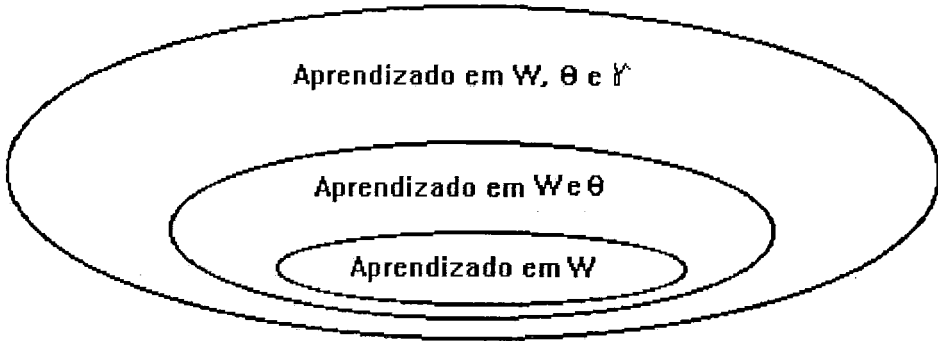


Figura III.2: Esquema geral dos algoritmos

**Algoritmo 1: Algoritmo de Aprendizado em  $W$**

$W \leftarrow W_{inicial}$

enquanto  $f$  puder ser localmente minimizada faça

$W \leftarrow W - \eta_W \nabla_W f$ , para algum  $\eta_W > 0$

fim-do-enquanto

**Algoritmo 2: Algoritmo de Aprendizado em  $W$  e  $\theta$**

$W \leftarrow W_{inicial}$

$\theta \leftarrow \theta_{inicial}$

enquanto  $f$  puder ser localmente minimizada faça

$W \leftarrow W - \eta_W \nabla_W f$ , para algum  $\eta_W > 0$

$\theta \leftarrow \theta - \eta_\theta \nabla_\theta f$ , para algum  $\eta_\theta > 0$

fim-do-enquanto



**Algoritmo 3: Algoritmo de Aprendizado em  $W$ ,  $\theta$  e  $\gamma$**  $W \leftarrow W_{inicial}$  $\theta \leftarrow \theta_{inicial}$  $\gamma \leftarrow \gamma_{inicial}$ enquanto  $f$  puder ser localmente minimizada faça $W \leftarrow W - \eta_W \nabla_W f$ , para algum  $\eta_W > 0$  $\theta \leftarrow \theta - \eta_\theta \nabla_\theta f$ , para algum  $\eta_\theta > 0$  $\gamma \leftarrow \gamma - \eta_\gamma \nabla_\gamma f$ , para algum  $\eta_\gamma > 0$ 

fim-do-enquanto

Uma vez que buscamos os pontos de mínimo da função e o  $\nabla f$  aponta no sentido do crescimento máximo de  $f$ , estamos considerando  $-\nabla f$ . Assim,  $W$ ,  $\theta$  e  $\gamma$  variam proporcional e respectivamente, às taxas de aprendizado  $\eta_W$ ,  $\eta_\theta$  e  $\eta_\gamma$ , conforme  $-\nabla f$ .

### III.4 Os critérios de parada

A condição para que os algoritmos parem é que  $f$  não possa ser localmente minimizada, ou seja, é preciso que seu gradiente seja nulo. Mas, se  $\nabla_* f = 0$ , para  $*$  =  $W$ ,  $\theta$ , ou  $\gamma$ , então temos que  $\|\nabla_* f\| = 0$ . No entanto, em termos computacionais não é fácil garantir que  $\nabla_* f = 0$ . Assim, foi utilizado um erro  $ERRO_*$  considerado aceitável em relação aos mínimos procurados, e que possibilitou um tempo razoável de processamento para os algoritmos. Dessa forma então, os algoritmos 1, 2 e 3 param quando, respectivamente,

$$i) \frac{\|\nabla_W f\|}{m} < ERRO_W;$$

$$ii) \frac{\|\nabla_W f\|}{m} < ERRO_W \text{ e } \frac{\|\nabla_\theta f\|}{m} < ERRO_\theta;$$

$$iii) \frac{\|\nabla_W f\|}{m} < ERRO_W, \frac{\|\nabla_\theta f\|}{m} < ERRO_\theta \text{ e } \frac{\|\nabla_\gamma f\|}{m} < ERRO_\gamma, \text{ isto é, quando } \|\nabla_* f\| \text{ atinge valores bem próximos de zero.}$$

É bom notar que, os valores de  $ERRO_W$ ,  $ERRO_\theta$  e  $ERRO_\gamma$  foram

tomados iguais a  $10^{-1}$ , sendo definidos empiricamente no decorrer dos testes de avaliação dos algoritmos.

### III.5 Implementação

O modelo analógico de Hopfield, bem como os três algoritmos de aprendizado propostos, foram implementados na linguagem C. Foram utilizadas as workstations SUN, tanto pela alta velocidade de processamento quanto pela capacidade em trabalhar com grande quantidades de dados, o que é muitas vezes praticamente inviável em microcomputadores do tipo PC AT.

### III.6 Resultados experimentais

Hopfield[2] concluiu experimentalmente, que dada uma rede com  $n$  neurônios o número máximo  $p$  de padrões estáveis que se pode obter é  $p \approx 0.15n$ . Já McEliece [25] mostrou um limite estatístico de  $p \leq \frac{n}{2} \log n$ . No entanto, será levado em conta apenas a conclusão do experimento feito por Hopfield, a fim de se poder ter uma idéia do desempenho dos algoritmos propostos.

Cada algoritmo de aprendizado foi testado utilizando-se 50 neurônios e  $m$  padrões de atividade gerados aleatoriamente. Para cada algoritmo foi feito um levantamento em função do número de padrões simultaneamente aprendidos. Assim, dados  $m$  padrões calcula-se uma fração ou porcentagem conforme o número  $m^*$  de padrões que tenham sido simultaneamente aprendidos.

Concluído o aprendizado, ocorre a fase de simulação em que a rede evolui dinamicamente até que o valor da energia computacional, definida pela equação ( II.4), atinja um mínimo satisfatório, onde enfim estabiliza. No entanto, a fim de que a simulação tivesse um caráter ainda mais preciso, resolvemos considerar a norma quadrática média da diferença entre dois padrões de atividade em tempos consecutivos, ou seja,

$$\frac{\|v_{t+\Delta t} - v_t\|^2}{n}$$

$m^*$	$m$										
	1	2	3	4	5	6	7	8	9	10	15
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	-	1.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	-	-	0.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	-	-	-	0.9	0.3	0.0	0.1	0.0	0.0	0.0	0.0
5	-	-	-	-	0.7	0.4	0.1	0.0	0.0	0.0	0.0
6	-	-	-	-	-	0.6	0.1	0.2	0.1	0.0	0.0
7	-	-	-	-	-	-	0.7	0.4	0.2	0.0	0.0
8	-	-	-	-	-	-	-	0.4	0.3	0.2	0.0
9	-	-	-	-	-	-	-	-	0.4	0.3	0.0
10	-	-	-	-	-	-	-	-	-	0.5	0.0
11	-	-	-	-	-	-	-	-	-	-	0.2
12	-	-	-	-	-	-	-	-	-	-	0.1
13	-	-	-	-	-	-	-	-	-	-	0.5
14	-	-	-	-	-	-	-	-	-	-	0.2
15	-	-	-	-	-	-	-	-	-	-	0.0

Figura III.3: Tabela 1

Mesmo assim, poderia haver o risco de que a simulação fosse processada em um tempo muito aquém do necessário. Dessa forma, considerou-se que esta etapa levaria um tempo mínimo de simulação igual a 0.1, ou seja, a simulação seria processada com pelo menos 100 iterações.

Para o primeiro algoritmo, foram considerados nulos a matriz  $W$  inicial e o potencial limiar  $\theta$ , enquanto que  $R = C = \gamma = 1.0$ . Segundo os resultados dos testes que foram realizados para até  $m = 15$  padrões, a Tabela 1 nos mostra que, a partir de  $m = 3$  começa a haver uma interferência gradual entre os padrões a serem ensinados. No entanto, o desempenho do algoritmo supera os 15% dos 50 neurônios utilizados, pois ao serem fornecidos a rede  $m = 10$  padrões, é possível ensinar todos os 10 em 50% dos casos, além de ser possível, no pior caso, ensinar  $m^* = 8$  em 20% dos casos. Mesmo ao serem dados  $m = 15$  padrões, é possível ensinar, simultaneamente,  $m^* = 13$  padrões em 50% dos casos e  $m^* = 11$  padrões em 20% dos casos, apesar de não serem simultaneamente aprendidos todos os 15 padrões.

$m$	$\eta_W$
1	0.05
2, 3	0.02
4, 5, 6, 7	0.01
8, 9, 10	0.005
15	0.02

Figura III.4: Tabela 2

Os valores da taxa de aprendizado  $\eta_W$  (figura III.4) são completamente empíricos, variando à medida que se aumenta a quantidade  $m$  de padrões a serem treinados e se deseja manter um bom nível de treinamento.

No algoritmo 2, tanto o padrão de conexão  $W$  inicial quanto o potencial limiar inicial foram gerados aleatoriamente. No entanto,  $\gamma$ ,  $R$ , e  $C$  assumiram o mesmo valor constante igual a 1.0, como no algoritmo 1.

Os testes foram realizados para até  $m = 25$  padrões e apresentaram um desempenho superior ao do algoritmo 1, o que se atribui essencialmente, a um maior grau de liberdade no aprendizado. Por exemplo, pela Tabela 3 temos que, dados  $m = 9$  padrões, foram simultaneamente aprendidos todos os 9 em 60% dos casos, enquanto que pelo algoritmo 1 isso ocorreu apenas em 40% dos casos. Já ao se considerar  $m = 15$  padrões é possível aprender todos os 15 em 30% dos casos, sendo ainda possível dos  $m = 15$  padrões serem aprendidos, simultaneamente,  $m^* = 9$  padrões em 10% dos casos.

Considerando  $m = 20$  ou  $m = 25$ , foi possível ensinar tanto  $m^* = 20$  quanto  $m^* = 25$  em 20% dos casos testados.

Os valores das taxas de aprendizado  $\eta_W$  e  $\eta_\theta$  utilizados podem ser vistos na Tabela 4 conforme o número  $m$  de padrões utilizados.

Para o último dos algoritmos a ser testado, utilizou-se  $W_{inicial}$  e  $\theta_{inicial}$  aleatórios,  $\gamma_i$  inicial 0.5, para  $1 \leq i \leq 50$ , sendo preservados os valores de  $R = C = 1.0$ . A variação de  $\gamma$  permitiu um aprendizado melhor em relação ao obtido pelos demais algoritmos. Por exemplo, a Tabela 5 nos mostra que, dados

$m^*$	$m$												
	1	2	3	4	5	6	7	8	9	10	15	20	25
0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	-	0.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	-	-	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	-	-	-	1.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	-	-	-	-	0.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	-	-	-	-	-	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	-	-	-	-	-	-	1.0	0.3	0.0	0.0	0.0	0.0	0.0
8	-	-	-	-	-	-	-	0.7	0.4	0.1	0.0	0.0	0.0
9	-	-	-	-	-	-	-	-	0.6	0.4	0.1	0.0	0.0
10	-	-	-	-	-	-	-	-	-	0.5	0.0	0.0	0.0
11	-	-	-	-	-	-	-	-	-	-	0.0	0.0	0.0
12	-	-	-	-	-	-	-	-	-	-	0.1	0.0	0.0
13	-	-	-	-	-	-	-	-	-	-	0.2	0.0	0.0
14	-	-	-	-	-	-	-	-	-	-	0.3	0.1	0.0
15	-	-	-	-	-	-	-	-	-	-	0.3	0.1	0.0
16	-	-	-	-	-	-	-	-	-	-	-	0.1	0.0
17	-	-	-	-	-	-	-	-	-	-	-	0.2	0.1
18	-	-	-	-	-	-	-	-	-	-	-	0.2	0.0
19	-	-	-	-	-	-	-	-	-	-	-	0.1	0.1
20	-	-	-	-	-	-	-	-	-	-	-	0.2	0.2
21	-	-	-	-	-	-	-	-	-	-	-	-	0.1
22	-	-	-	-	-	-	-	-	-	-	-	-	0.2
23	-	-	-	-	-	-	-	-	-	-	-	-	0.1
24	-	-	-	-	-	-	-	-	-	-	-	-	0.0
25	-	-	-	-	-	-	-	-	-	-	-	-	0.2

Figura III.5: Tabela 3

$m$	$\eta_W$	$\eta_\theta$
1, 2, 3, 4	0.002	$2e^{-06}$
5, 6, 7, 8, 9, 10, 15, 20, 25	0.001	$5e^{-07}$

Figura III.6: Tabela 4

$m^*$	$m$												
	1	2	3	4	5	6	7	8	9	10	15	20	25
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	-	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	-	-	1.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	-	-	-	0.9	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
5	-	-	-	-	0.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	-	-	-	-	-	0.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0
7	-	-	-	-	-	-	0.9	0.2	0.0	0.0	0.0	0.0	0.0
8	-	-	-	-	-	-	-	0.8	0.1	0.0	0.0	0.0	0.0
9	-	-	-	-	-	-	-	-	0.8	0.1	0.0	0.0	0.0
10	-	-	-	-	-	-	-	-	-	0.9	0.0	0.0	0.0
11	-	-	-	-	-	-	-	-	-	-	0.1	0.0	0.0
12	-	-	-	-	-	-	-	-	-	-	0.2	0.0	0.0
13	-	-	-	-	-	-	-	-	-	-	0.1	0.0	0.0
14	-	-	-	-	-	-	-	-	-	-	0.1	0.0	0.0
15	-	-	-	-	-	-	-	-	-	-	0.5	0.2	0.0
16	-	-	-	-	-	-	-	-	-	-	-	0.1	0.0
17	-	-	-	-	-	-	-	-	-	-	-	0.3	0.0
18	-	-	-	-	-	-	-	-	-	-	-	0.1	0.0
19	-	-	-	-	-	-	-	-	-	-	-	0.0	0.1
20	-	-	-	-	-	-	-	-	-	-	-	0.3	0.0
21	-	-	-	-	-	-	-	-	-	-	-	-	0.2
22	-	-	-	-	-	-	-	-	-	-	-	-	0.1
23	-	-	-	-	-	-	-	-	-	-	-	-	0.0
24	-	-	-	-	-	-	-	-	-	-	-	-	0.2
25	-	-	-	-	-	-	-	-	-	-	-	-	0.2

Figura III.7: Tabela 5

$m = 15$  padrões foi possível ensinar todos os 15 padrões em 50% dos casos, contra 30% e 0% obtidos respectivamente, pelos algoritmos 2 e 1. Ainda, ao serem utilizados  $m = 20$  padrões foi possível ensiná-los todos em 30% dos casos, sendo também possível (no pior caso) ensinar  $m^* = 15$  padrões em 20% dos testes realizados.

Os valores das taxas de aprendizado  $\eta_W$ ,  $\eta_\theta$  e  $\eta_\gamma$  referentes ao algoritmo 3 são apresentados na Tabela 6.

É interessante ressaltar o papel das taxas  $\eta_W$ ,  $\eta_\theta$  e  $\eta_\gamma$ , que tão-somente medem a velocidade do aprendizado. Dessa forma, em geral, enquanto que taxas

$m$	$\eta_W$	$\eta_\theta$	$\eta_\gamma$
1, 2, 3, 4, 5	0.001	$5e^{-07}$	$1e^{-07}$
5, 6, 7, 8	0.001	$1e^{-07}$	$5e^{-09}$
9	0.001	$5e^{-08}$	$2e^{-09}$
9	0.001	$5e^{-08}$	$2e^{-09}$
10	0.001	$5e^{-08}$	$2e^{-10}$
15	0.0005	$2e^{-08}$	$1e^{-11}$
20	0.00025	$1e^{-08}$	$5e^{-12}$

Figura III.8: Tabela 6

maiores garantem uma rapidez no processo de aprendizado, podendo levar a oscilações à medida que se aproximam dos pontos de mínimo, taxas menores, apesar de causarem mais lentidão no processo, contribuem para reduzir ou até eliminar estas oscilações.

Pela análise dos testes realizados, concluímos empiricamente que, o algoritmo de aprendizado em  $W$ ,  $\theta$  e  $\gamma$  apresenta um desempenho superior em relação aos demais (figura III.9). Dessa forma, definido o algoritmo de trabalho, iniciamos os testes de aplicação da tese, que consistiram em utilizar grupos de uma até nove imagens com 400 pixels, ensiná-las a rede neuronal para só então, compactá-las e finalmente, restaurá-las. Entretanto, é importante notar que, cada pixel de uma imagem qualquer  $I$  representa na verdade, o estado de ativação de algum neurônio que a constitui.

No que diz respeito ao aprendizado destas imagens, que serão melhor detalhadas na seção V.3, temos que não houve o mínimo possível de degradação, isto é, o aprendizado foi 100% inclusive nos casos em que se utilizou um número maior de imagens para serem simultaneamente aprendidas.

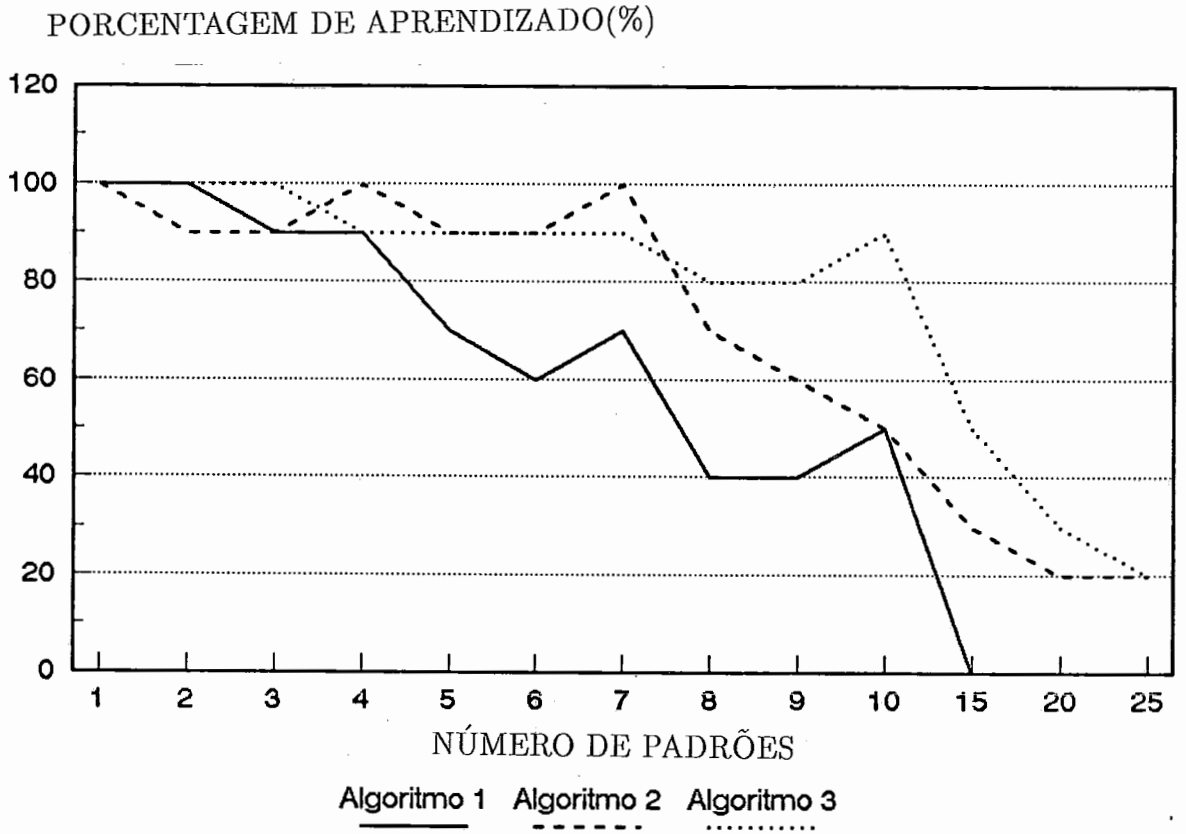


Figura III.9: Gráfico de desempenho dos algoritmos de aprendizado



# Capítulo IV

## Compactação de Imagens

### IV.1 Uma visão geral

Com o contínuo crescimento da tecnologia das comunicações, através de transmissões de fax, transmissões de satélite para realizar reconhecimento e fotografias do tempo, entre outras, a demanda na transmissão e armazenamento de imagens vem crescendo progressivamente. Como resultado desta demanda, tanto a largura de banda do canal de transmissão quanto o espaço em disco vêm se tornando limitados, fazendo surgir portanto, a necessidade de algoritmos cada vez mais eficientes de compactação de imagens.

Uma imagem compactada quando reconstruída vem, geralmente, acompanhada de alguma distorção. Assim, a eficiência de um algoritmo de compactação é medida não só por sua habilidade na compactação de dados e por sua complexidade de implementação, mas também pela distorção resultante.

Entendendo-se qualquer cena visual como uma imagem, que pode ser representada por um conjunto finito de números, suponhamos então, que uma imagem  $I$  possa ter valores contínuos. A *quantização* de  $I$  significa, de uma forma não muito rigorosa, dividir o intervalo de valores contínuos que  $I$  pode assumir em regiões discretas, não necessariamente iguais, que serão então representadas por um número bem determinado no computador.

Muitas das técnicas de compactação de dados utilizadas antigamente

se valiam da *quantização escalar*. No entanto, estudos mais recentes passaram a utilizar a *quantização vetorial* e até a *quantização de cena*, por serem mais eficazes [10] [13] [22] .

As aplicações de compactação de imagens consistem basicamente em transmitir e armazenar informação. Para transmissão, as técnicas de compactação são muitas vezes restritas por tempo real e considerações *on line*, que tendem a limitar severamente o tamanho e a complexidade do hardware. Para aplicações de armazenamento, os requerimentos do compressor são menos severos porque muito do pré-processamento pode ser feito *off line*. A decodificação ou descompactação tem que ser rápida e eficiente para minimizar o tempo de resposta. As aplicações de transmissão são por exemplo, em emissora de TV, sensoriamento remoto via satélite, radar, sonar e transmissão de fax, enquanto que o armazenamento de imagens é mais utilizado para tratar de documentos educacionais e de trabalho, imagens médicas em sistemas de monitoração de paciente, entre outros.

## IV.2 Alguns meios para avaliar a fidelidade da imagem descompactada

Conforme o método usado para realizar a compactação/descompactação pode ocorrer uma maior ou menor degradação da imagem resultante em relação à fornecida. Assim, a distorção obtida pode ser avaliada pelo *erro médio quadrático* (MSE) e pela *taxa sinal-e-ruído* (SNR)<sup>1</sup>, entre outras.

A MSE é uma medida comum de distorção [22], que para uma imagem de tamanho N x M, por exemplo, é dada por

$$MSE = \frac{1}{NM} \sum_{k=1}^N \sum_{j=1}^M (I_{k,j} - I'_{k,j})^2,$$

onde  $I_{k,j}$  e  $I'_{k,j}$  representam as imagens original e restaurada, respectivamente. No entanto, valer-se da MSE nem sempre é interessante, especialmente quando se está tratando com imagens que têm que ser avaliadas visualmente. Isso ocorre pois não é muito fácil encontrar um parâmetro de comparação que ajude a estabelecer se o

---

<sup>1</sup>Signal-to-Noise Ratio

MSE obtido está muito ou pouco elevado.

Um critério de avaliação menos significativo é a SNR [13], que está relacionado com o poder de sinal e ruído de uma imagem. Portanto, a SNR é definida por

$$SNR = 10 \log_{10} \frac{(\text{valor pico-a-pico da imagem original})^2}{MSE}$$

Para imagens com escalas em cinza, o valor pico-a-pico é a diferença entre o valor máximo e o valor mínimo da intensidade do pixel encontrado nas imagens. Assim, por exemplo, se forem utilizados os 256 níveis de cinza possíveis, o valor da SNR vai ser dado por  $10 \log_{10}(\frac{256^2}{MSE})$ .

### IV.3 Componentes Principais e Redes Neurais

Consideremos uma amostra de uma imagem  $R$  que tenha sido quantizada e a representemos por uma matriz  $[R]$  de dimensão  $N \times N$ . Pode-se escrever uma *transformação linear* geral sobre uma matriz de imagem  $[R]$  como

$$[\alpha] = [U]^t[R][V], \quad (\text{IV.1})$$

onde  $[\alpha]$  é denominado o domínio da transformada unitária da imagem,  $[U]$  e  $[V]$  são os operadores unitários e  $t$  representa a tranposta da matriz. Como  $[U]$  e  $[V]$  são unitários seguem-se

$$[U][U]^t = [I] \quad e \quad [V][V]^t = [I] \quad (\text{IV.2})$$

onde  $[I]$  é a matriz identidade. Portanto, de (IV.1) e (IV.2), temos que

$$[R] = [U][\alpha][V]^t.$$

Se escrevermos  $[U] = [u_1, \dots, u_N]$  e  $[V] = [v_1, \dots, v_N]$ , onde  $u_i$  e  $v_i$  são vetores dispostos nas colunas de  $[U]$  e  $[V]$ , respectivamente, e ainda, tomarmos  $[\alpha]$  como

$$[\alpha] = \begin{bmatrix} \alpha_{11} & 0 & \dots & 0 \\ 0 & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & \cdot & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & \alpha_{12} & \dots & 0 \\ 0 & 0 & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & \cdot & \dots & 0 \end{bmatrix} + \dots + \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & \cdot & \dots & \alpha_{NN} \end{bmatrix}$$

teremos enfim que

$$[R] = \sum_{i=1}^N \sum_{j=1}^N \alpha_{ij} u_i v_j^t \quad (\text{IV.3})$$

Cada produto  $P_{ij} = u_i v_j^t$ , onde  $1 \leq i, j \leq N$ , pode ser interpretado como uma imagem, de forma que, a soma de todos os  $P_{ij}$  adequadamente reforçados por  $\alpha_{ij}$  fornece a imagem original  $[R]$ .

Em termos de uma análise dos *graus de liberdade*, a equação (IV.3) sugere que uma matriz de imagem  $[R]$  tem um número total máximo de graus de liberdade igual a  $N^2$ , o que corresponde aos  $N \times N$  pixels que formam a imagem. Os graus de liberdade independentes são convertidos, através da equação (IV.3), para  $N^2$  coeficientes  $\alpha_{ij}$ . Os valores de  $u_i$  e  $v_j$  são determinísticos, e não trabalham adicionando ou retirando informação da imagem, mas tão-somente rearrumando-a.

Como uma ilustração gráfica deste processo, temos a figura (IV.1). Pode-se observar que, a imagem  $[R]$  é decomposta numa soma de  $N^2$  matrizes de posto<sup>2</sup> 1, balanceadas pelo coeficiente  $\alpha_{ij}$  adequado, ou seja,

$$[R] = \alpha_{11} u_1 v_1^t + \dots + \alpha_{ij} u_i v_j^t + \dots + \alpha_{NN} u_N v_N^t.$$

Na verdade, estas matrizes de posto 1 representarão *imagens bases* bidimensionais que irão compor a matriz original  $[R]$ .

Este processo [21] apenas revela uma forma bastante simples de representar uma imagem através de seus componentes principais. Pode-se ainda recorrer a Transformada de Fourier, do Cosseno e a várias outras formas de representação.

A Análise de Componentes Principais (PCA) tem sido um método bastante usado na extração da melhor representação dos dados originais quando não há restrições impostas [23]. Desta forma, poderia ser interessante comparar o uso de PCA com o uso de redes neuronais na compactação de imagens.

Em [7], faz-se este estudo. É verificado e analisado o uso de PCA e o de redes neuronais com backpropagation na compactação de um conjunto de

---

<sup>2</sup>Matrizes de posto 1 são aquelas cujo o número de linhas independentes é igual a 1

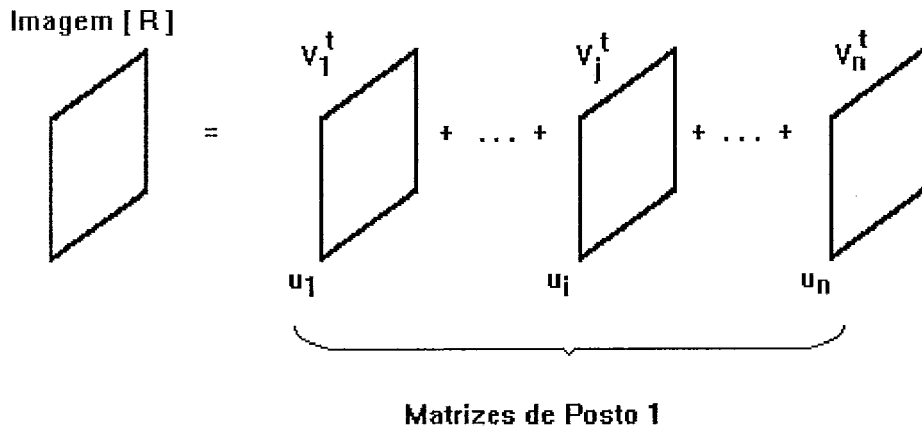


Figura IV.1: Representação da imagem original  $[R]$  através de *imagens bases*

imagens para diagnóstico médico, apesar de existirem razões teóricas para se assumir que PCA pode superar o backpropagation no caso em que a média dos dados a serem compactados é nula, pois desta forma o backpropagation equivale à uma transformação linear [8]. No entanto, se ao serem comparadas com PCA as redes não mostrarem um aumento de erro significativo, elas podem ser consideradas mais adequadas, uma vez que treinar a rede pode ser mais rápido que usar PCA, principalmente se um hardware próprio for utilizado.

Realizaram-se testes com imagens do pulmão. Parte destas imagens caracterizava dois tipos de anormalidades similares, enquanto que uma segunda parte dessas imagens era absolutamente normal. Assim, uma vez mapeados os valores dos pixels no intervalo  $[0, 1]$ , usaram-se PCA e redes neurais para comprimir segmentos de  $8 \times 8$  e de  $32 \times 16$  pixels. No caso de PCA, os primeiros 128 ou os primeiros 16 componentes principais foram usados para uma redução de dimensão de 512 para 128 e de 64 para 16, respectivamente. No caso de redes neurais, utilizou-se uma arquitetura de três camadas:  $64 \times 16 \times 64$ , para os segmentos de  $8 \times 8$  pixels e  $512 \times 128 \times 512$ , para os de  $32 \times 16$  pixels.

Concluída a compactação, os segmentos foram recuperados utilizando-se, no caso do PCA, uma inversa da matriz PCA sobre o conjunto reduzido de dados, e no caso da rede neuronal, utilizando-se o output da camada de saída. Nestes casos,

a fidelidade da imagem final foi avaliada usando-se a *soma total do erro quadrático (TSSE)*, que é uma medida similar ao MSE.

Segundo os testes realizados, pode-se concluir que PCA é similar às redes neuronais na compactação de dados que tenham sido aprendidos, mas é superior na compactação de imagens que não tenham sido previamente aprendidas, sendo essa diferença mais pronunciada em segmentos de 32 x 16 pixels. Os resultados também mostraram que a capacidade de generalização da rede diminui quando se trata de segmentos maiores.

## IV.4 Compactação de imagem utilizando modelos de Redes Neuronais

Em [6] é avaliada a capacidade de compactação de dados de imagem de um modelo de rede neuronal. O sistema utiliza uma arquitetura de três camadas do tipo *feed-forward*. É interessante notar que, ao se adotar um modelo de rede neuronal multicamadas para reduzir a dimensão dos dados de imagem, o número de unidades na camada intermediária deve ser menor que o número de unidades nas camadas de input e output [20].

O sinal de input da rede é obtido da conversão, feita a partir de dados de 8-bits da figura original, em valores contínuos pertencentes ao intervalo  $(0, 1)$ . Assim, utilizando o algoritmo de aprendizado *Backpropagation*, espera-se que a primeira fase da rede aprenda a eliminar redundância de sinal, ao passo que na segunda fase seja possível aprender uma regra de transformação inversa. Além disso, um digitalizador é colocado na saída da camada intermediária, realizando-se então a quantização, cujos símbolos gerados são mapeados em mensagens para transmissão. Embora se espere que grande parte da redução dos dados seja obtida pela quantização e esquemas simbólicos de código, o estudo realizado se deteve na redução da dimensão para avaliar a capacidade de compactação do modelo utilizado.

Assim, realizado o aprendizado, ocorre a etapa da transmissão dos dados, na qual a rede é dividida em duas partes: CODER, em que a imagem com-

pactada é quantizada e mapeada em mensagens para transmissão, e a DECODER, em que a imagem é reconstruída ou descompactada. Enfim, ao final de todo o processo, a qualidade da imagem decodificada ou reconstruída é avaliada utilizando-se a SNR.

A fim de tornar mais eficiente o processo de aprendizado, o modelo de rede utilizado particiona a imagem de entrada em pequenos blocos, sendo cada sub-imagem ou bloco fornecido a uma sub-rede neuronal como dado de entrada. Desta forma, concorrentemente, as sub-redes realizam as computações necessárias ao aprendizado. Por exemplo, uma imagem de 128 x 128 pixels é dividida em 256 blocos, de 8 x 8 p ixéis cada, de forma que 256 subredes, concorrentemente, executam o aprendizado para cada bloco da imagem.

Os testes realizados utilizam imagens de 128 x 128 pixels e permitiram concluir que com um aumento no número de imagens aprendidas, a SNR média diminui lentamente o que ocorre devido a uma interferência entre as imagens que já foram aprendidas e as que ainda serão. No entanto, uma SNR alta para imagens aprendidas pode ser obtida com um aumento no número de unidades da camada intermediária ou com uma elevação do número de iterações de aprendizado, que possibilita conhecer o que é necessário a fim de que a qualidade da imagem seja preservada.

O método proposto em [10] permite o trabalho com uma grande quantidade de dados , compactando-os para armazenamento ou transmissão e restaurando-os. A arquitetura da rede é do tipo *feed-forward* com quatro camadas, e utiliza como algoritmo de aprendizado um *Nested Training Algorithm* (NTA).

O sistema proposto permite compactação simultânea de uma grande quantidade de dados, em oposição as quantizações escalar e vetorial, o que faz com que o requerimento de armazenamento ou da capacidade do link de comunicação sejam bem menores que no uso de quantização escalar ou quantização vetorial.

O sistema consiste de três seções: uma seção de camada de entrada, uma seção de camada intermediária e uma seção de camada de saída. A seção de camada de entrada consiste de um número de sub-redes que servem como camada

de input para a rede. A seção de camada intermediária consiste de três partes: *combiner*, *compressor* e *decombiner*. O *combiner*, cujo número de nós é menor que o número de nós da camada de entrada, atua como um multiplexador para as subredes de input. O *compressor* realiza a compressão da saída do *combiner*, produzindo uma representação interna da imagem, isto é, a imagem compactada, que será transmitida ou armazenada. O *decombiner*, que possui o mesmo número de nós que o *combiner*, atua como um demultiplexador, tendo sua saída dividida em subcampos conforme a ligação com as subredes da camada de saída. Finalmente, cada subcampo é aplicado a uma subrede que produz uma subimagem final de saída.

Pela estrutura descrita, o modelo de rede utilizado é simétrico, o que permite treinar a rede *outside-in*, isto é, treinar primeiro as camadas mais externas, para só então treinar as camadas mais internas. Daí a razão do nome Nested Training Algorithm, que nada mais é do que uma versão estendida do Backpropagation para redes simétricas com múltiplas camadas que possui um tempo de processamento bem reduzido em comparação com o do Backpropagation padrão.

Pelos testes realizados para imagens aprendidas de 512 x 512 pixels, foi possível concluir que com um aumento no número de nós no compressor ocorre uma diminuição da MSE e conseqüentemente, um aumento da SNR, representando um bom nível na qualidade da imagem.



# Capítulo V

## O método de compactação utilizado

O método aqui proposto é bem intuitivo e como constitui um módulo à parte (figura I.1), pode ser processado a qualquer momento que se queira compactar/descompactar uma ou mais imagens previamente ensinadas a uma rede neuronal analógica de Hopfield. É interessante notar que, apesar deste método para compactar imagens estar intimamente ligado à rede neuronal, existem outros métodos [27] que não necessitam da rede para realizar tal tarefa.

Suponhamos uma imagem  $I$  constituída por  $n$  pixels ou neurônios. Ao compactarmos  $I$ , surgem duas questões: o *quanto* de  $I$  e *quais* dos elementos que a constituem devemos armazenar a fim de que ao reconstruí-la haja o mínimo possível de distorção. Estabelecida uma *taxa de armazenamento* de neurônios  $t_a$ , sabemos que dos  $n$  pixels ou neurônios que formam a imagem, apenas  $t_a * n$  serão armazenados. Este método propõe, então, que sejam escolhidos os  $t_a * n$  neurônios que recebem maior carga de informação, ou seja, são recrutadas as unidades mais fortes, aquelas que adquiriram maior conhecimento a respeito de  $I$ .

A fase de reconstituição ou descompactação de  $I$  é simples, pois consiste em fornecer à rede a imagem  $I_c$  compactada, que constitui parte da informação total. Como a propriedade fundamental de uma rede neuronal de Hopfield é a memória de acesso pelo conteúdo (CAM), basta fornecermos à rede (que já possui todo o conhecimento necessário armazenado) parte da imagem, isto é, a imagem

compactada  $I_c$ , para que ela evolua até atingir um padrão estável, ou seja, uma imagem  $I_f$  descompactada.

Cabe agora compararmos a imagem final  $I_f$  com a imagem  $I_o$  previamente concebida no início de todo o processo. Tal comparação é feita analisando-se unidade a unidade, isto é, o pixel ou neurônio  $N_1^o$  de  $I_o$  e o pixel ou neurônio  $N_1^f$  de  $I_f$ , e assim sucessivamente. Quando as unidades  $N_i^o$  e  $N_i^f$ , para  $1 \leq i \leq n$ , são distintas, computamos o valor do *número de neurônios errados* (NNE), que embora seja uma medida simples de avaliação, é uma forma adequada e prática se considerarmos o aspecto visual da imagem.

## V.1 O algoritmo

```

para  $i \leftarrow 1$  ate  $n$ -neuronios faça
     $peso_i \leftarrow 0$ 
    para  $j \leftarrow 1$  ate  $n$ -neuronios faça
         $peso_i \leftarrow peso_i + |w_{ij}|$ 
    fim-do-para
     $indice_i \leftarrow i$ 
fim-do-para
ordenar  $indice_i$  decrescentemente de acordo com  $peso_i$ 
para  $p \leftarrow 1$  ate  $n$ -padroes faça
    para  $i \leftarrow 1$  ate  $t_a * n$ -neuronios faça
         $vcomp_{pi} \leftarrow v_{pindice_i}$ 
    fim-do-para
fim-do-para
para  $p \leftarrow 1$  ate  $n$ -padroes faça
    para  $i \leftarrow 1$  ate  $n$ -neuronios faça
         $v_{pi} \leftarrow 0.5$ 
    fim-do-para
    para  $i \leftarrow 1$  ate  $t_a * n$ -neuronios faça
         $v_{pindice_i} \leftarrow vcomp_{pi}$ 

```

fim-do-para

*simular a rede ate que a diferenca entre duas energias consecutivas esteja bem proxima de zero ou que o tempo de simulacao atinja um maximo satisfatorio*

*comparar a imagem descompactada com a imagem original*

*obtendo o numero de neurônios errados*

fim-do-para

## V.2 Implementação

O método de compactação proposto foi implementado através da linguagem de programação C. Foram também utilizadas as estações de trabalho SUN pelos motivos já alegados na seção III.5.

## V.3 Os testes de compactação

As imagens originais  $I_o$  foram criadas com 400 pixels nas cores preto, cinza e branco. Na verdade, cada uma dessas unidades ou pixels constitui um neurônio  $N_i$  com um estado de ativação  $a_i(t)$ , pertencente ao intervalo real  $(0, 1)$ . Dessa forma, foi preciso relacionar cada uma das cores utilizadas na representação das imagens com um valor neste intervalo. O primeiro passo então, foi dividi-lo em três subintervalos, onde cada cor era atribuída a um valor bem preciso, ou seja, o meio do subintervalo.

No entanto, na representação das imagens aprendida e descompactada, o processo foi inverso, uma vez que os estados de ativação deviam ser mapeados em uma das cores usadas. Neste caso então, a cor foi fornecida conforme o subintervalo do intervalo  $(0,1)$  em que se encontrava os estados de ativação, uma vez que cada subintervalo podia ser relacionado com uma das cores usadas.

A compactação foi realizada para grupos de até nove imagens previamente aprendidas pela rede neuronal, utilizando-se para as taxas de compactação valores que variaram de 0.9 a 0.1, conforme a necessidade em armazenar, respecti-

vamente, um número menor ou maior de neurônios.

Podemos observar que, à medida que aumentamos o número de imagens a serem compactadas, a taxa média de compactação em que o número de neurônios (pixels) errados atinge zero, diminui progressivamente (figura V.1). Isso ocorre porque, durante o processo de aprendizado a rede neuronal obteve conhecimento acerca de todas as imagens simultaneamente, não guardando necessariamente, as características específicas de cada uma delas. Além disso, ao se dar o processo de compactação são guardados os neurônios mais fortes que compõem cada imagem, podendo em muitos casos, os estados de ativação relativos aos neurônios mais marcantes de uma imagem estarem bem próximos das ativações dos neurônios mais significativos de outras imagens. Portanto, quanto maior o número de imagens no processo de compactação, maior a quantidade necessária de pixels a serem armazenados, a fim de haja o mínimo possível de "superposição de imagens" e consequentemente, um bom nível na qualidade das imagens restauradas.

Considerando a relação entre a taxa de compactação e a média do número de neurônios errados, dado um número  $k$  de imagens que foram simultaneamente aprendidas, onde  $2 \leq k \leq 9$ , temos pelos gráficos que se seguem que, à medida que diminuimos a taxa de compactação, a *média do número de neurônios* ou pixels *errados* (MNNE) também diminui, até atingir zero. No caso de  $k = 1$ , a MNNE foi nula logo ao considerarmos 90% de compactação. Ao se aumentar o número  $k$  de imagens a serem compactadas e computando a média do número de neurônios errados, temos que esta média tende a baixar mais lentamente à medida que incrementamos o número de imagens. Portanto, são necessárias taxas cada vez menores, conforme se queira utilizar um número maior de imagens, e ainda assim obter uma MNNE bem baixa.

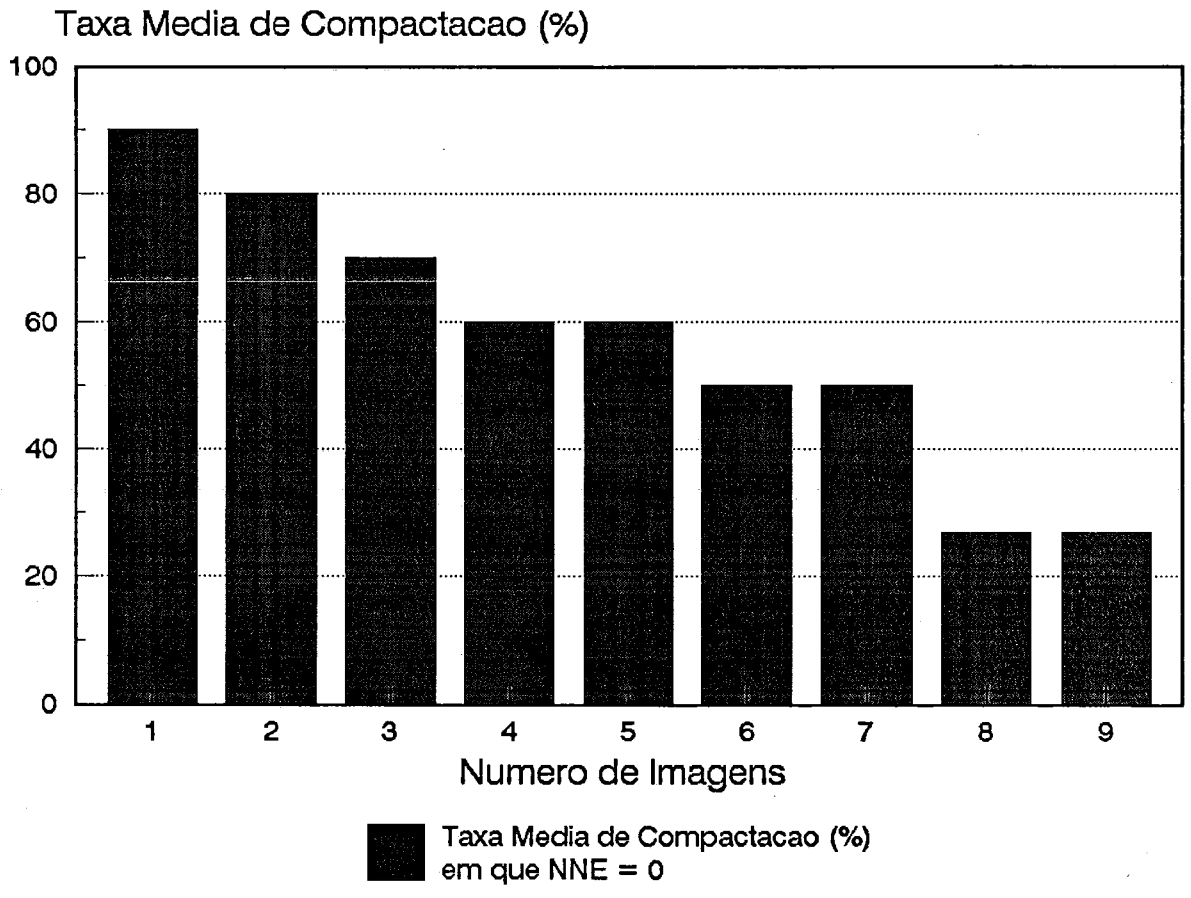


Figura V.1: Gráfico 1

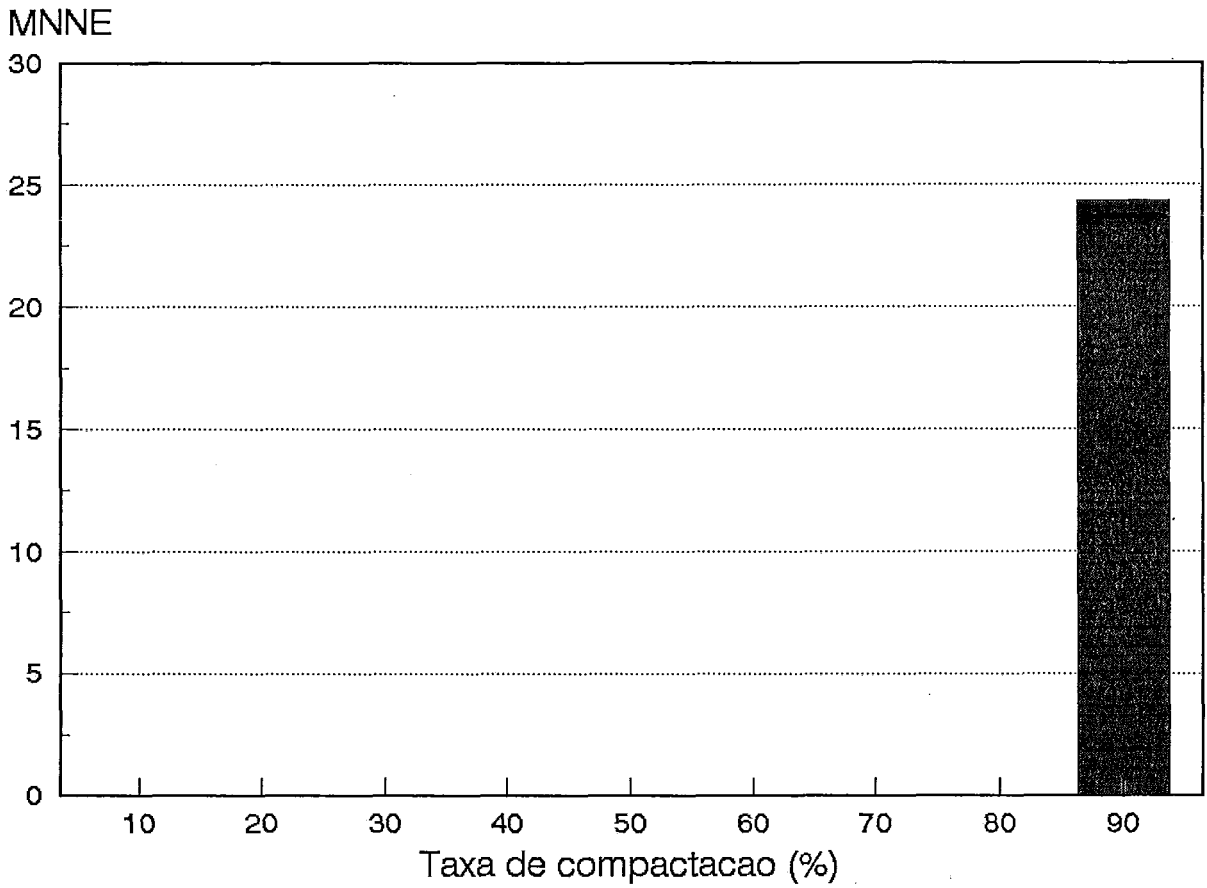


Figura V.2: Gráfico 2 - número de imagens = 2

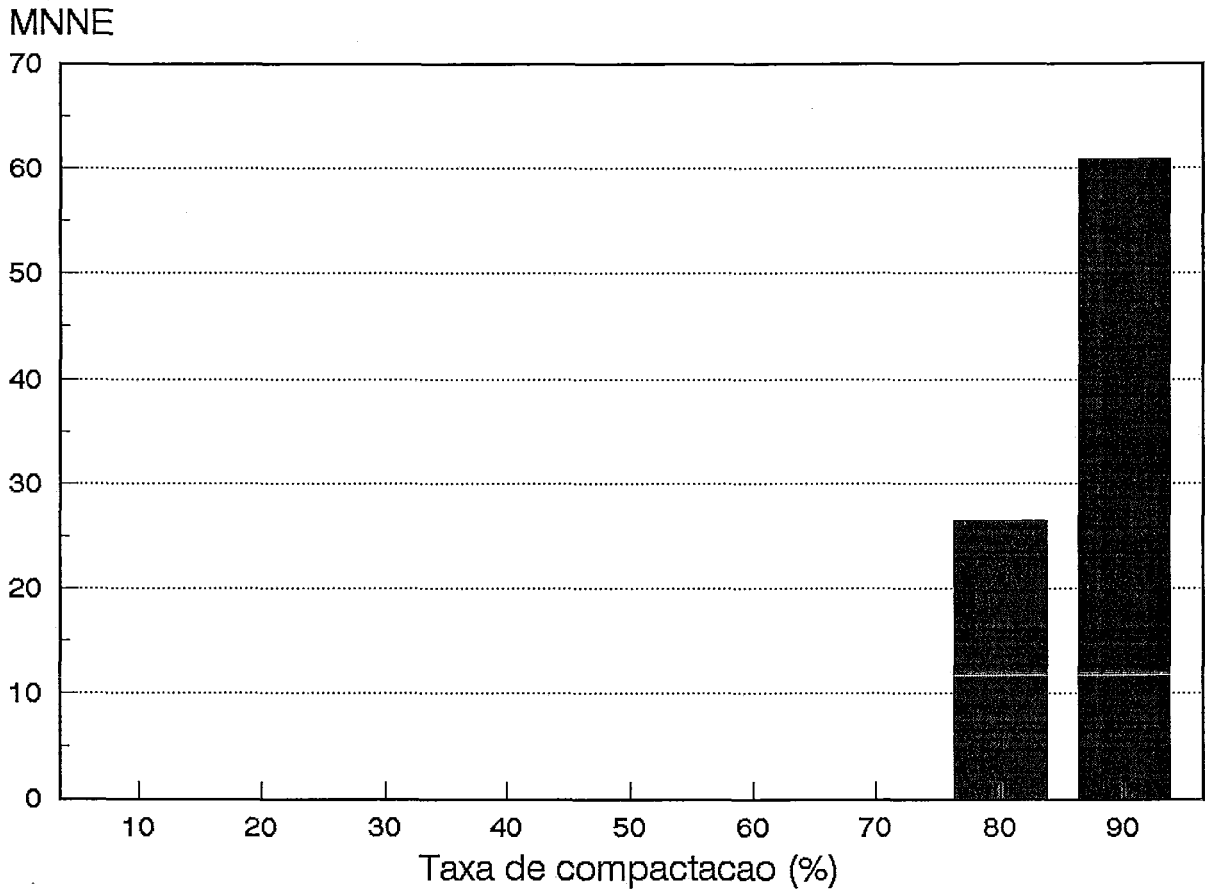


Figura V.3: Gráfico 3 - número de imagens = 3

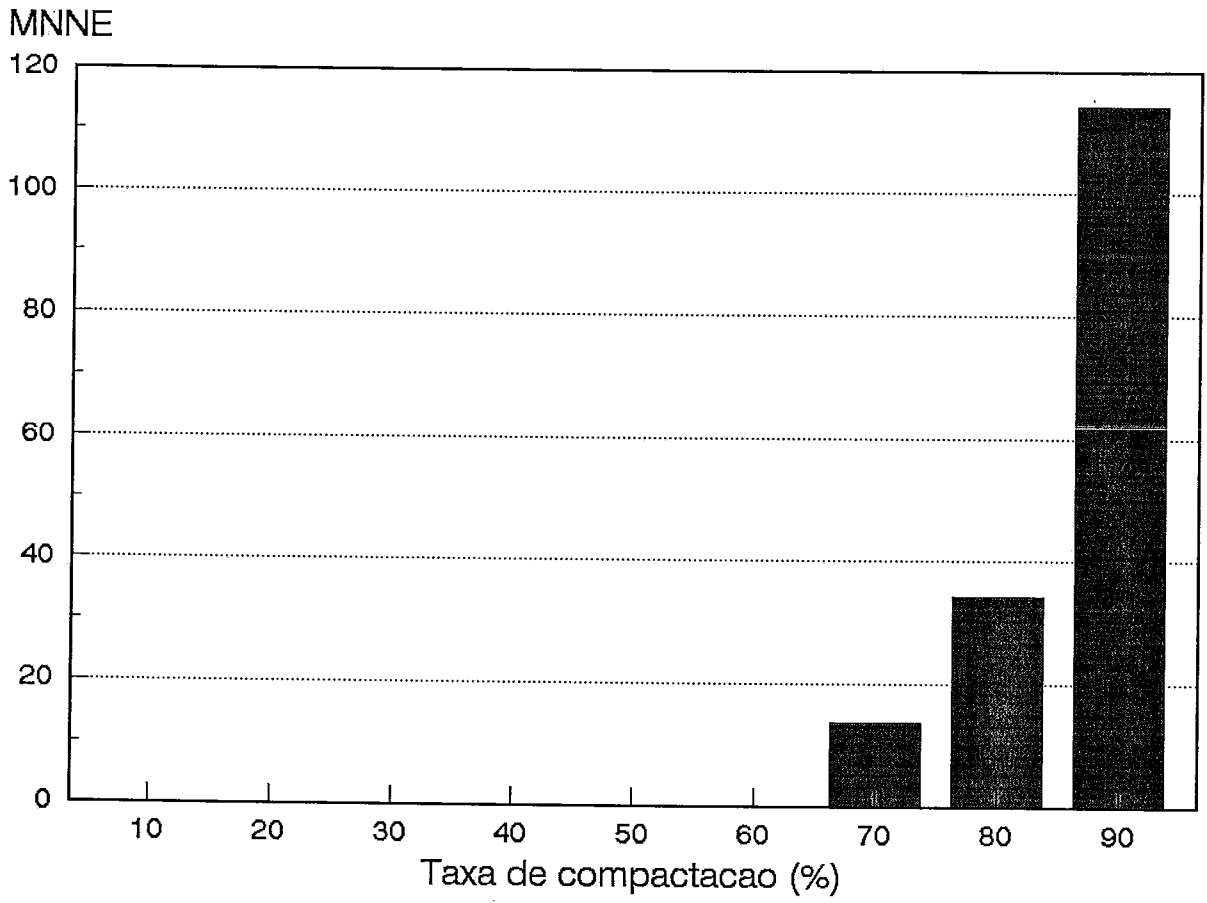


Figura V.4: Gráfico 4 - número de imagens = 4



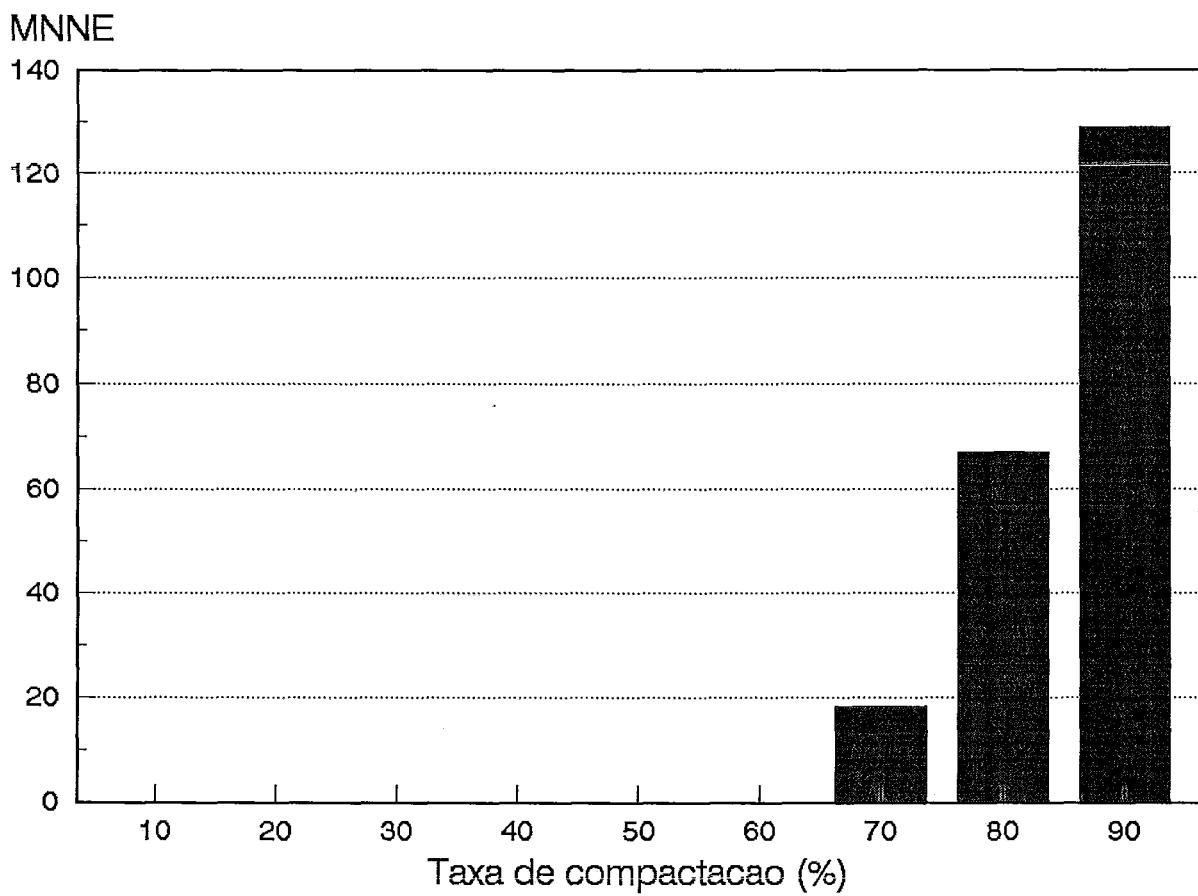


Figura V.5: Gráfico 5 - número de imagens = 5

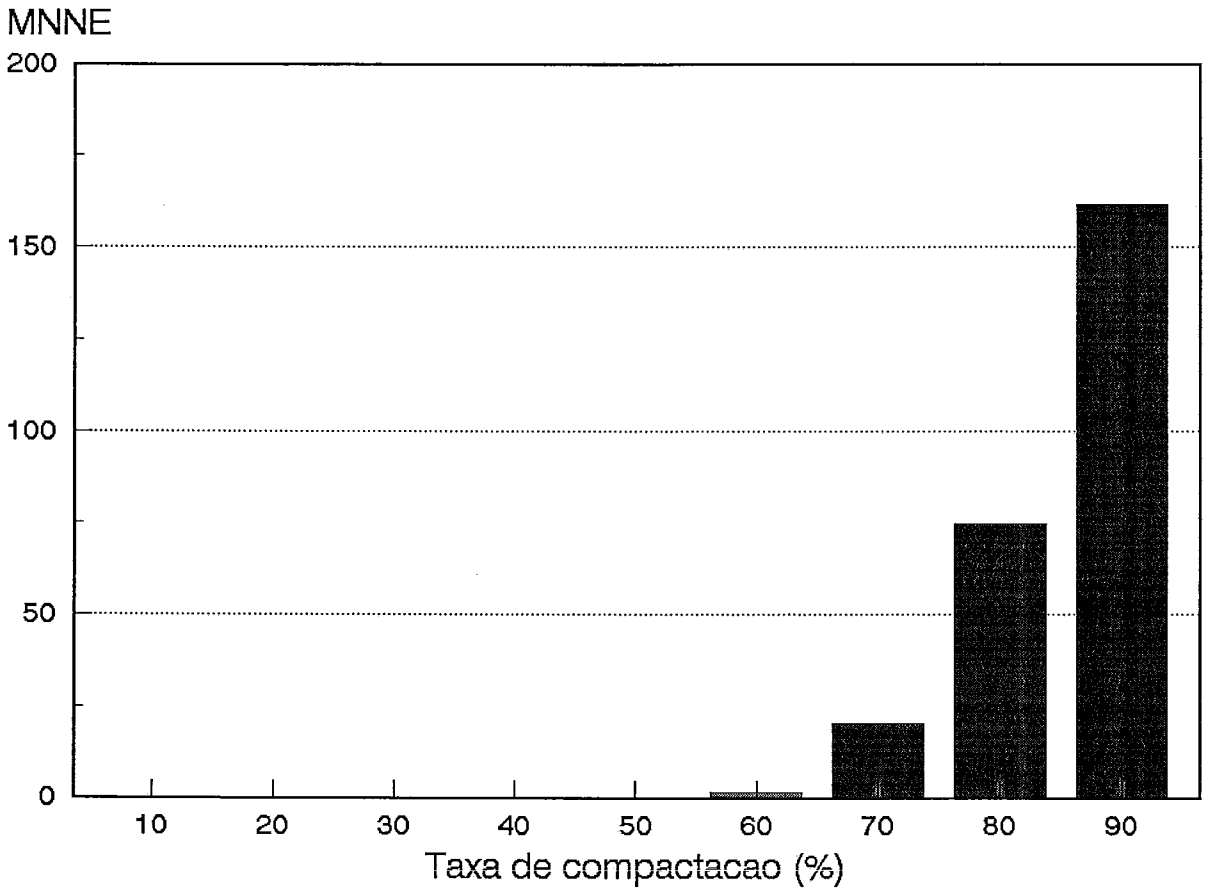


Figura V.6: Gráfico 6 - número de imagens = 6

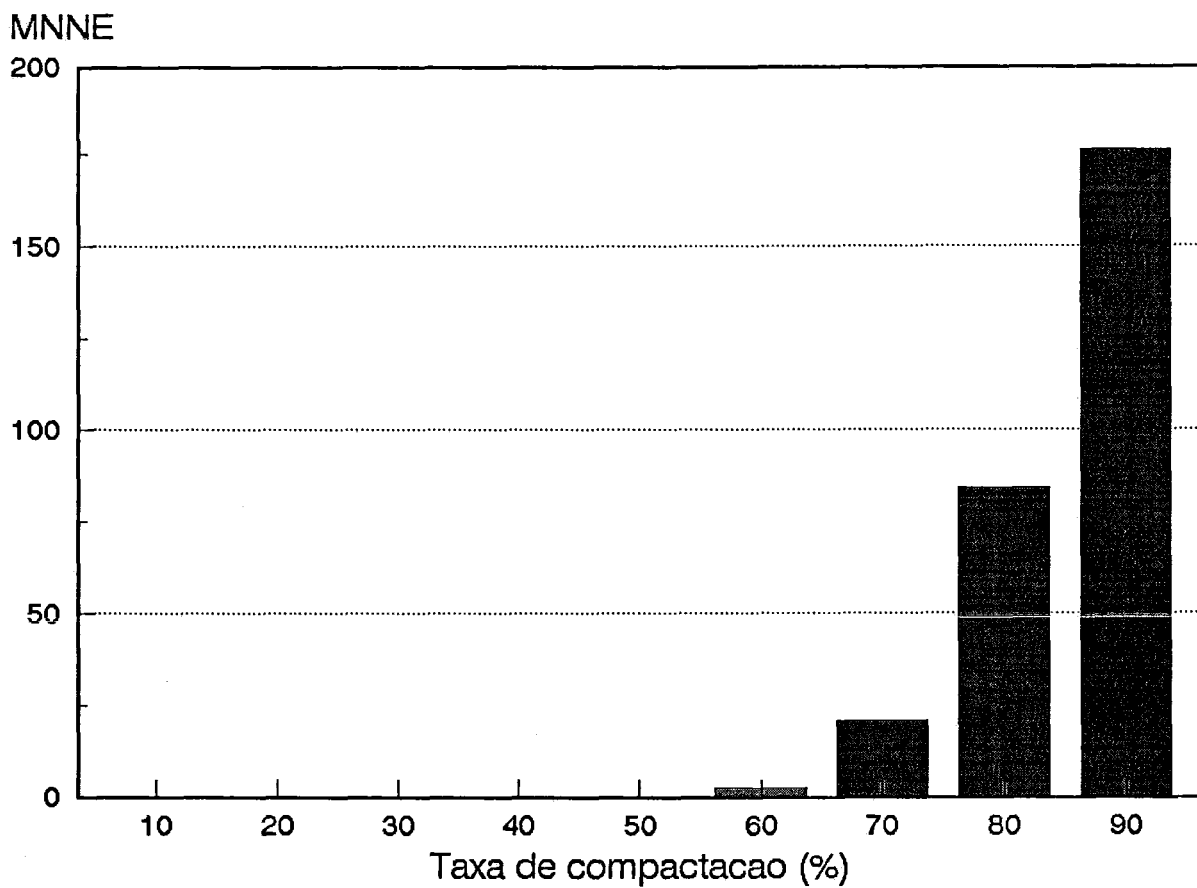


Figura V.7: Gráfico 7 - número de imagens = 7

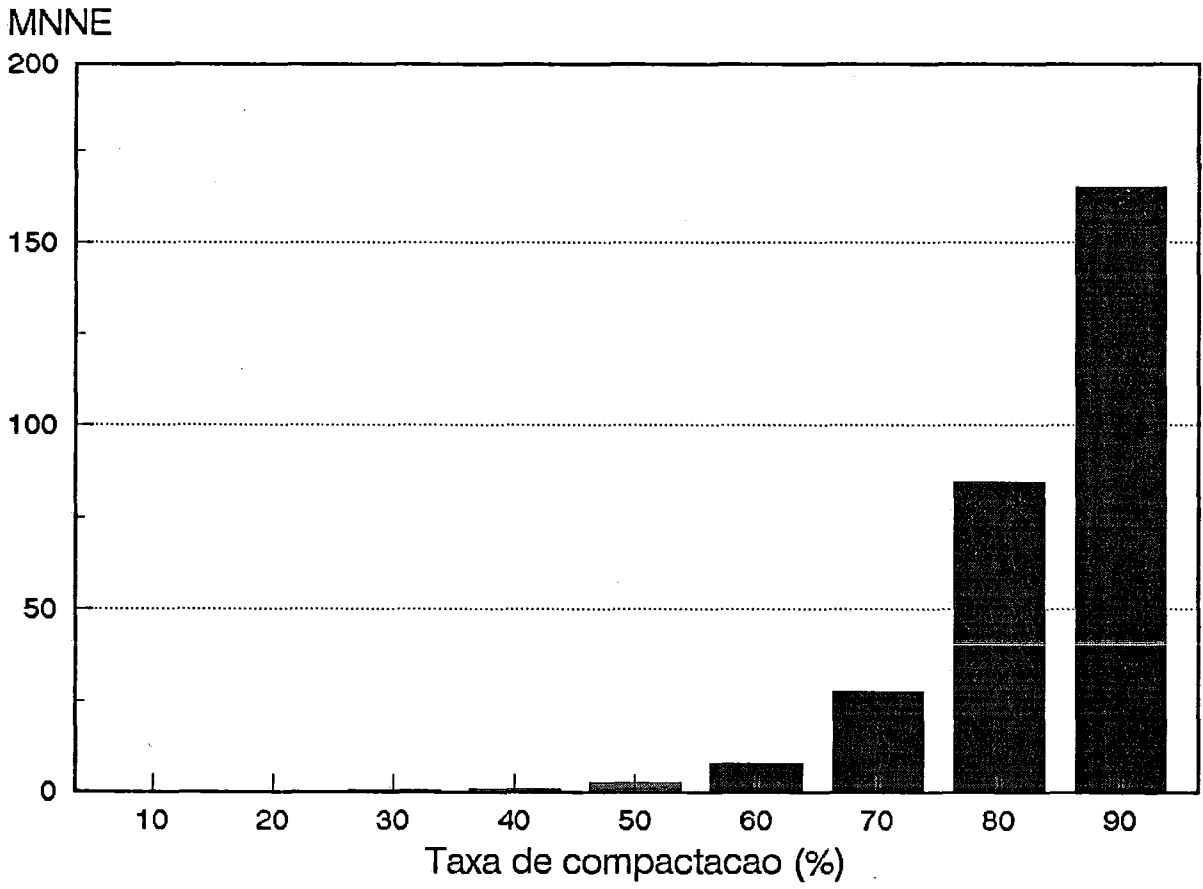


Figura V.8: Gráfico 8 - número de imagens = 8

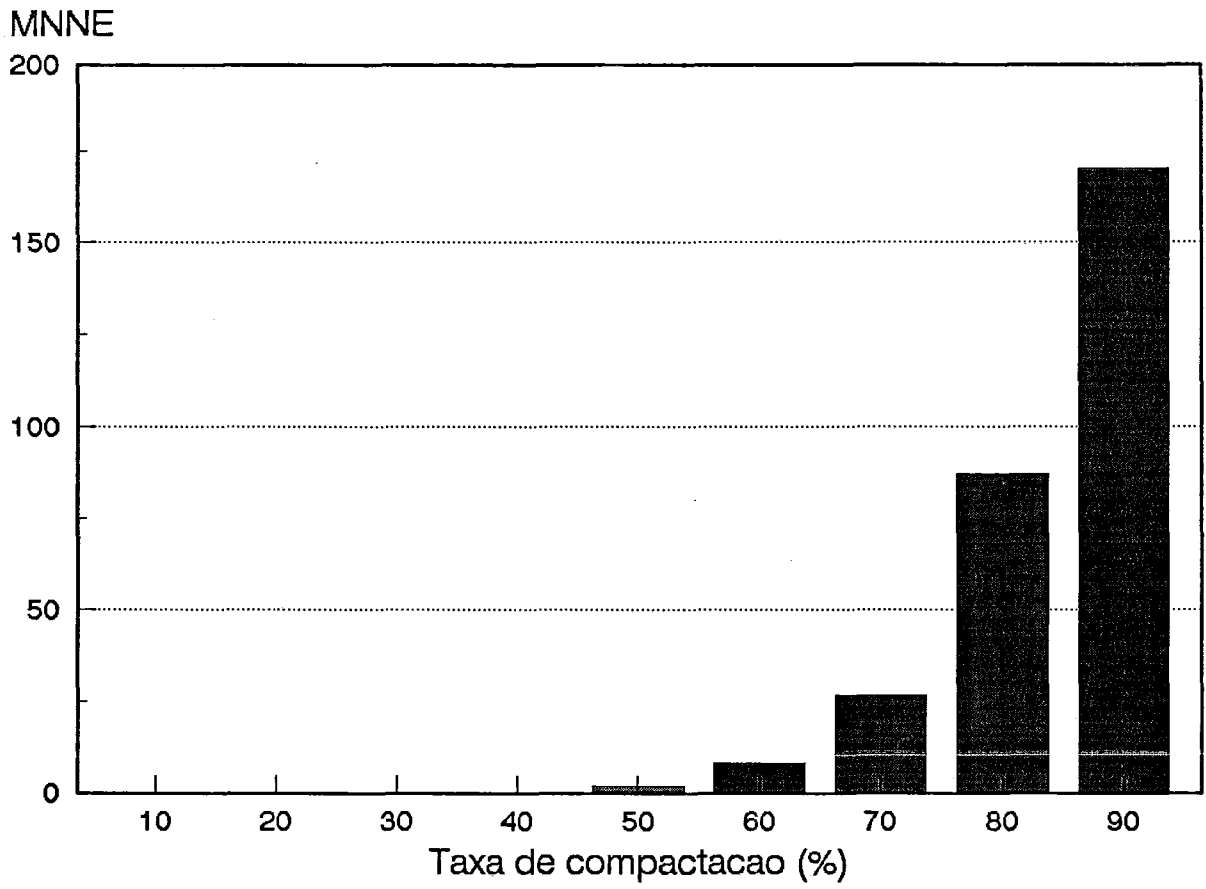


Figura V.9: Gráfico 9 - número de imagens = 9

# Capítulo VI

## Conclusão

Até o momento não era possível adotar um bom algoritmo de aprendizado para redes neuronais analógicas de Hopfield, pois o que normalmente acontecia era recorrer a regra de Hebb, e assim obter um número máximo de padrões estáveis em torno de 15% do número de neurônios da rede.

Os testes realizados com os três algoritmos de aprendizado propostos para redes neuronais analógicas de Hopfield, mostraram de forma empírica, que mesmo o mais simples desses algoritmos pode superar os resultados existentes até então. Dessa forma, o algoritmo 3, que obteve melhor desempenho entre os algoritmos propostos pôde, dados uma rede com 50 neurônios e 15 padrões de atividade gerados aleatoriamente, ensiná-los todos em 50% dos casos<sup>1</sup>, e assim superar ainda mais os resultados já obtidos.

A etapa de aplicação da tese, que consistiu na compactação de imagens (padrões) de 400 pixels (neurônios) nas cores preto, cinza e branco, previamente aprendidas<sup>2</sup> por uma rede neuronal analógica de Hopfield, também produziu bons resultados. Utilizando um algoritmo de compactação bem simples e intuitivo, que se detém, principalmente, em armazenar os pixels mais "fortes", foi preciso para grupos com nove imagens de 400 pixels cada, por exemplo, armazenar em média 75% dos pixels de cada imagem, a fim de que a imagem pós-descompactação fosse exatamente igual a imagem fornecida no início da aplicação. No entanto, obter uma

---

<sup>1</sup>No pior caso foi possível ensinar 11 padrões dos 15 fornecidos

<sup>2</sup>É interessante lembrar que todas as imagens fornecidas nesta etapa foram 100% aprendidas

imagem final descompactada igual a imagem original inicialmente fornecida, é em geral, desnecessário, pois nem sempre imagens com distorções mínimas se tornam inviáveis. Assim, é possível armazenarmos uma quantidade menor de neurônios e ainda obter uma imagem, pelo menos em termos visuais, com um nível de distorção aceitável.

Embora o objetivo inicial da tese fosse também tratar com cinco cores (preto, branco e três tons degradê de cinza), a inviabilidade de tempo nos levou a utilizar apenas as cores preto, cinza e branco. Isso no entanto, produziu um afrouxamento, principalmente no processo de descompactação, pois neste caso, a probabilidade de um pixel mudar de cor é menor que no caso de serem utilizadas cinco ou mais cores.

É interessante notar que, apesar de termos tratado com imagens bem pequenas de apenas 400 pixels, por limitações de tempo e de máquina, podemos encarar estas imagens, em termos de trabalhos futuros, como subimagens formando imagens muito maiores que as utilizadas neste trabalho.

# Apêndice

As imagens estão distribuídas em nove grupos conforme o número de imagens que tenham sido simultaneamente aprendidas. Cada grupo é formado por três arquivos de dados, para os quais os valores das taxas de compactação chegaram a variar de 0.9 a 0.1, conforme a necessidade em se obter uma imagem final descompactada com o mínimo de degradação possível.

No entanto, apresentaremos um arquivo por grupo, além de apenas considerarmos os grupos formados por três até nove imagens. É interessante notar que, cada imagem é flagrada em três momentos básicos: no início de todo o processo (imagem original), após o aprendizado (imagem aprendida) e ao final da descompactação (imagem descompactada).

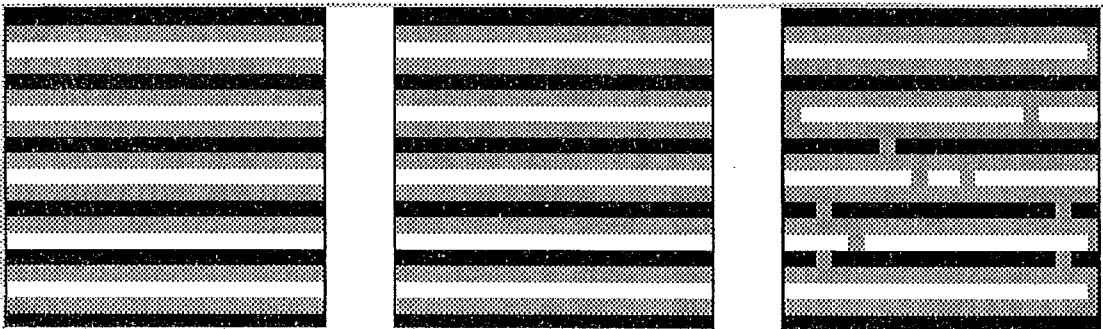
## IMAGENS

### GRUPO 1

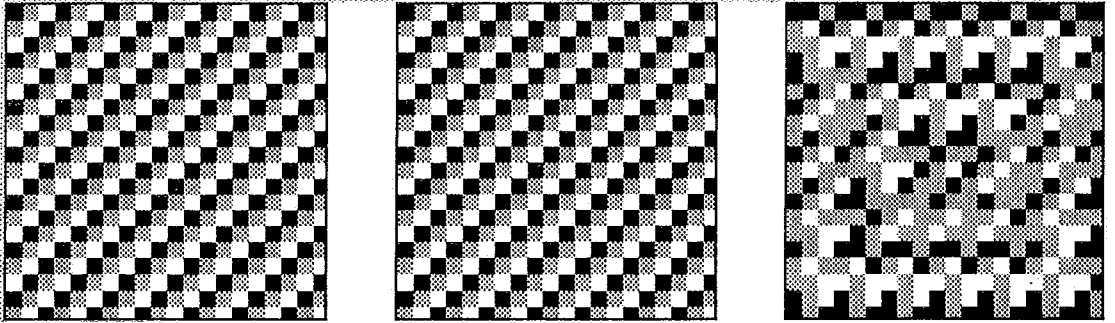
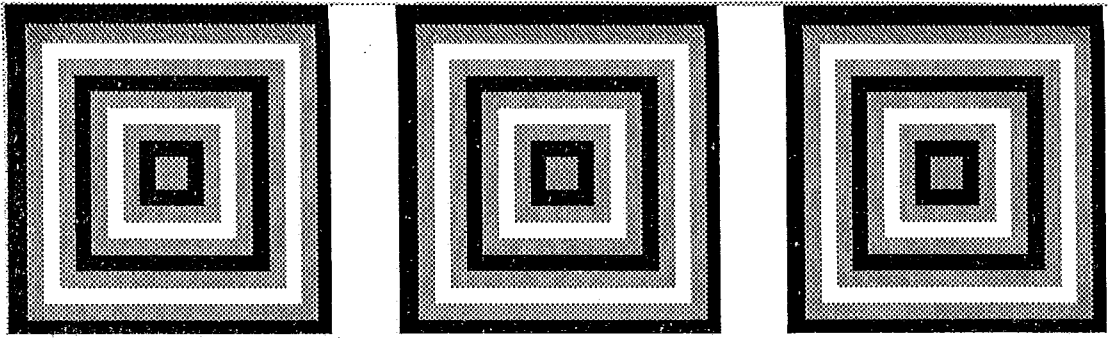
ORIGINAL

APRENDIDA

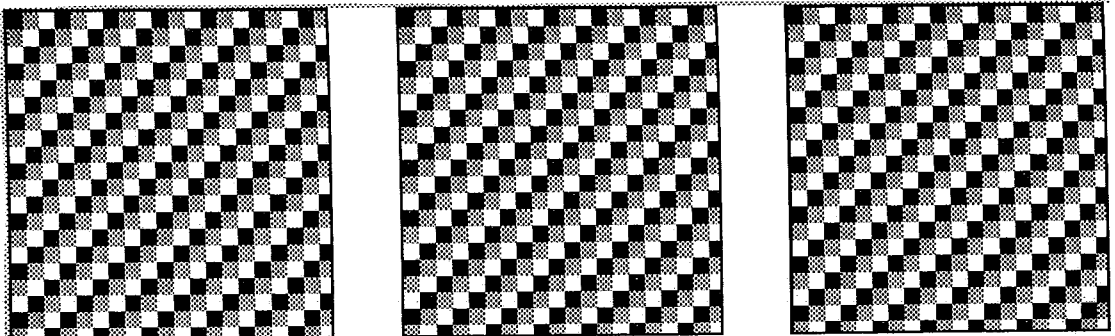
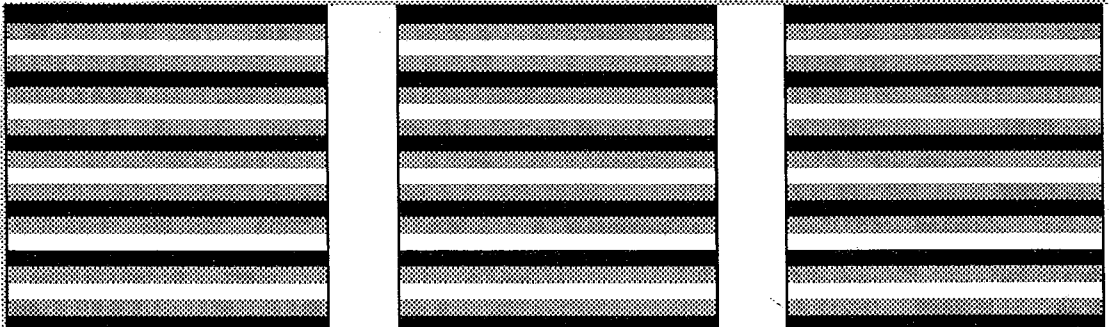
DESCOMPACTADA

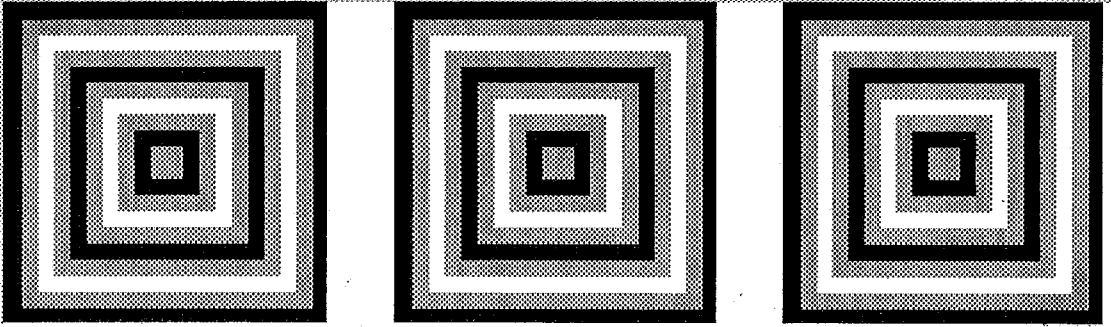






compactação = 0.90





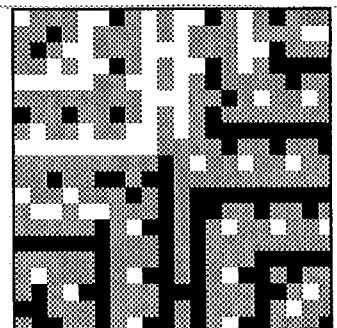
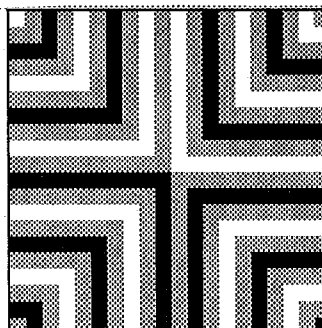
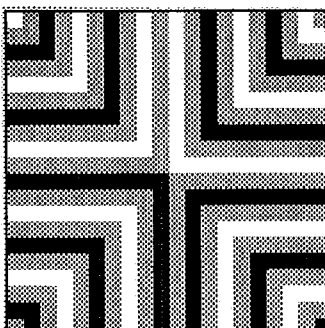
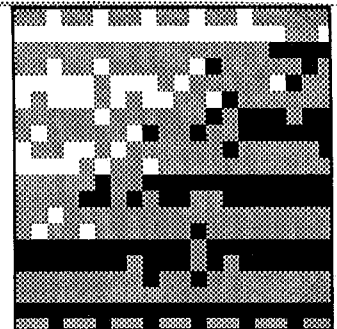
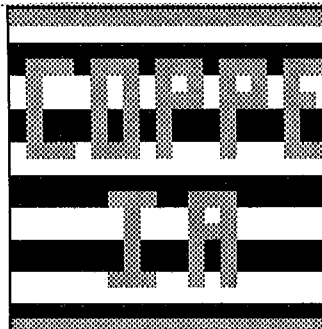
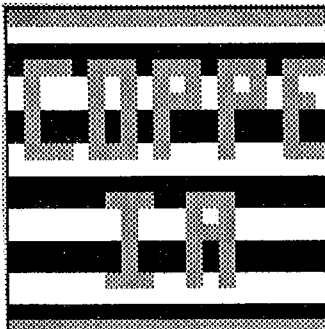
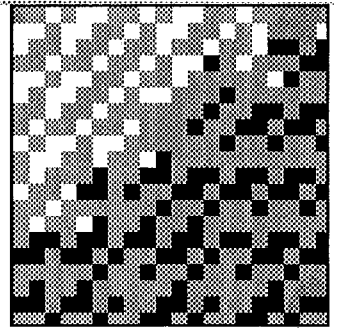
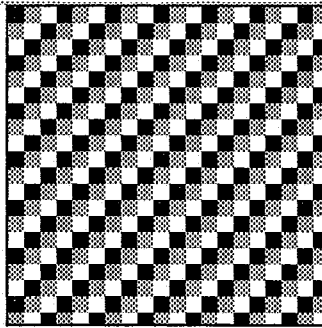
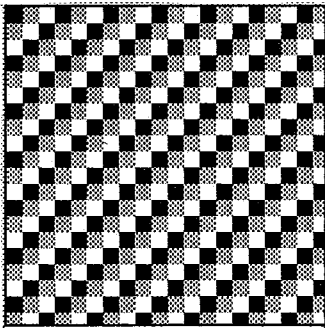
Taxa de compactação = 0.70

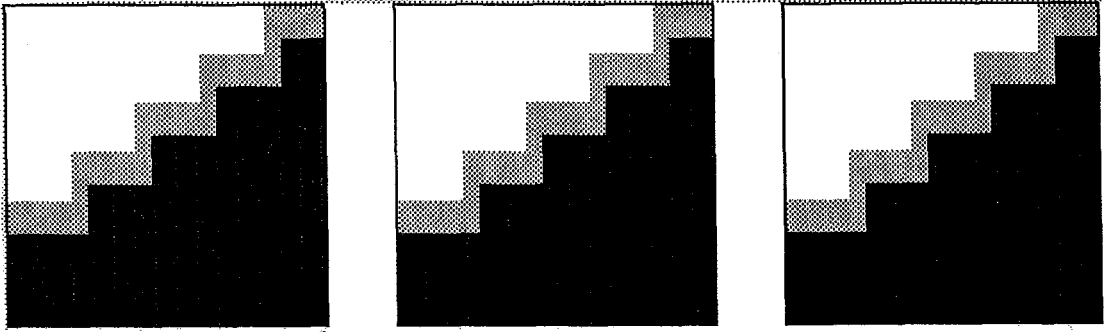
GRUPO 2

ORIGINAL

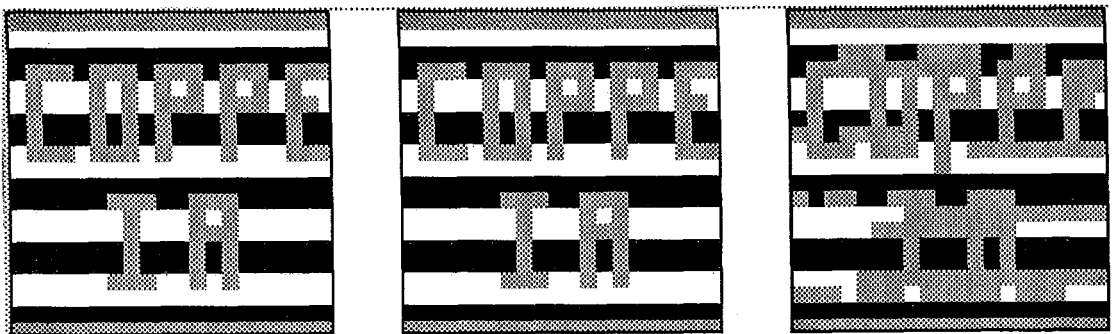
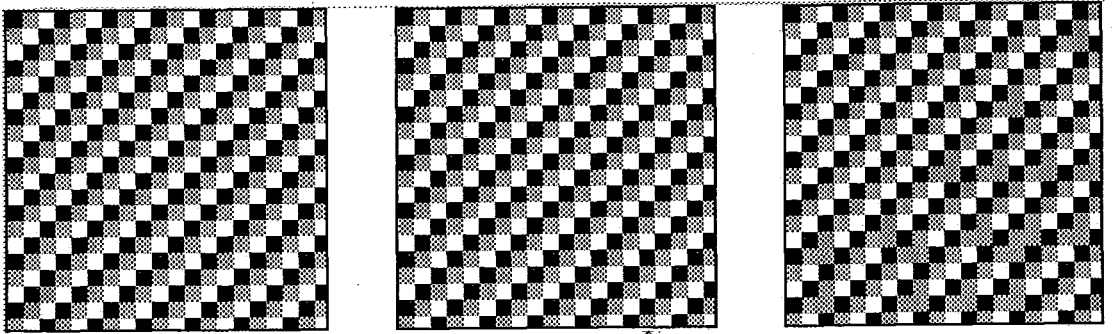
APRENDIDA

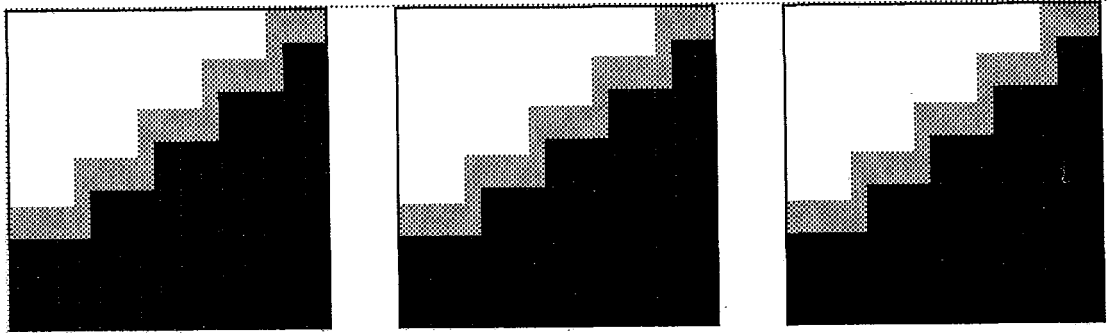
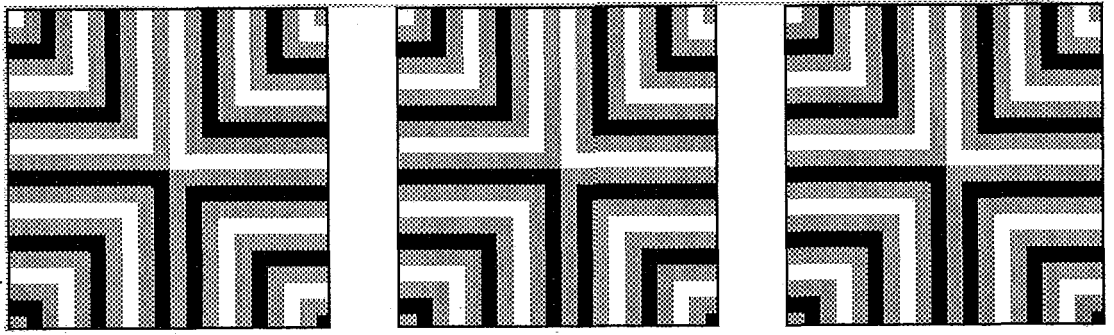
DESCOMPACTADA





Taxa de compactação = 0.90





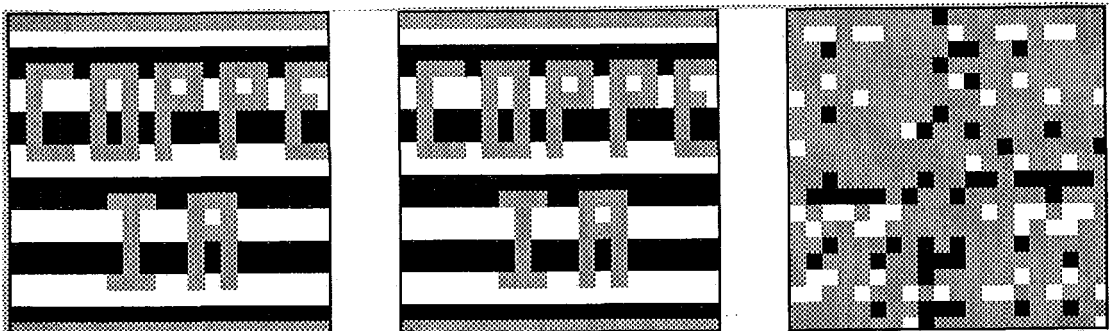
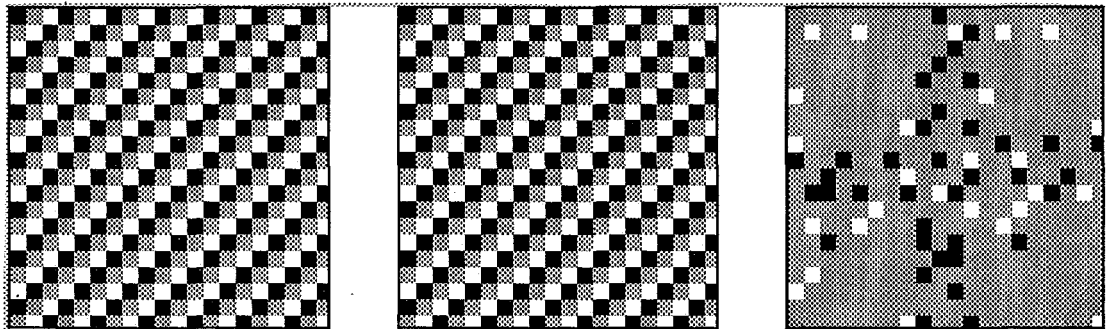
Taxa de compactação = 0.70

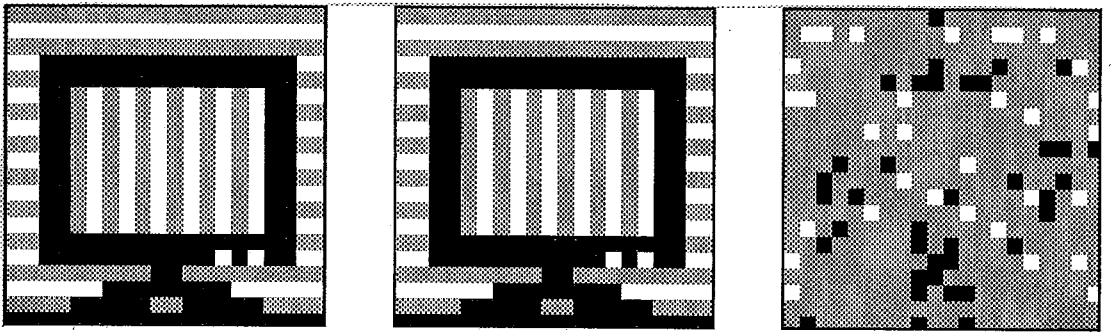
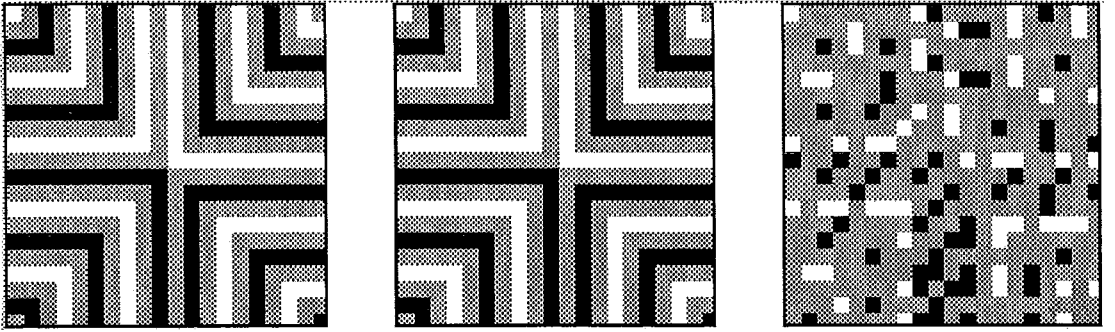
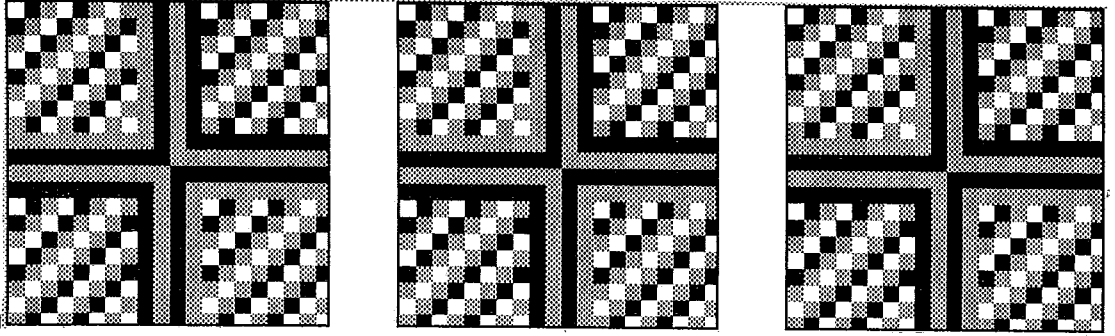
GRUPO 3

ORIGINAL

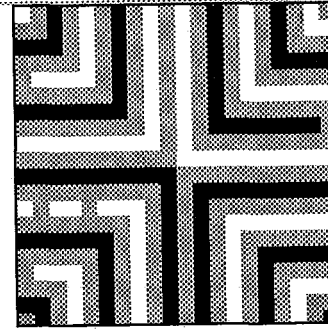
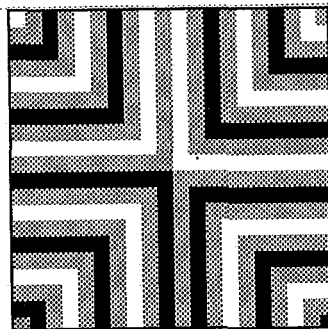
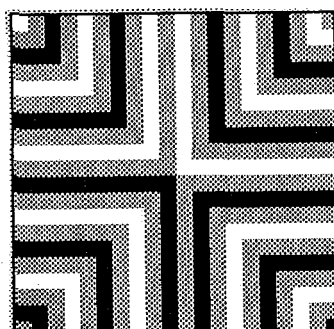
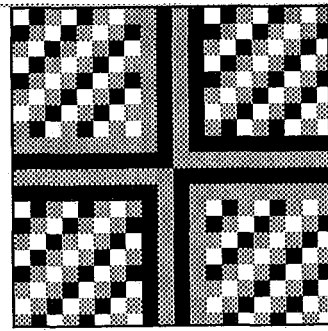
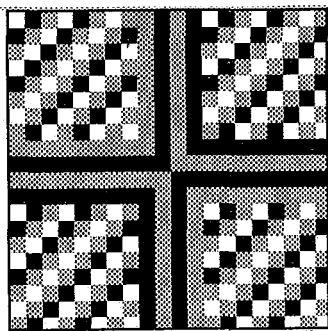
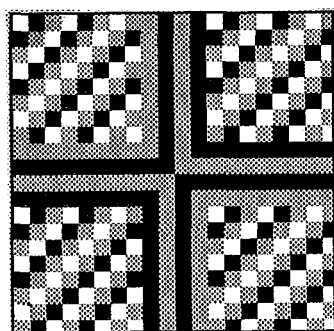
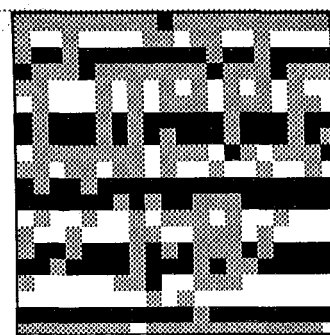
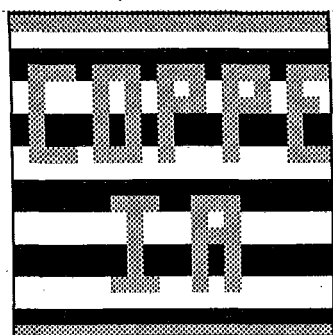
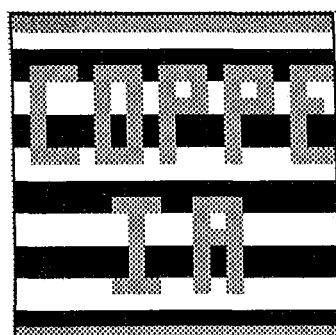
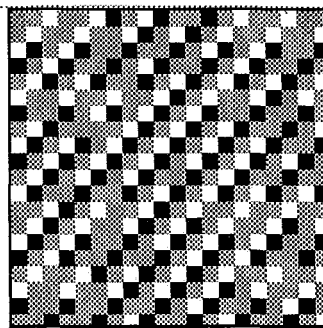
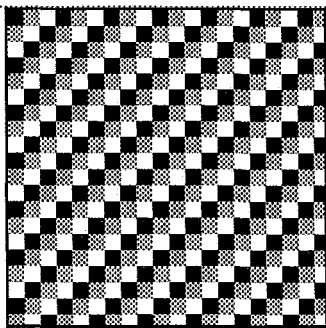
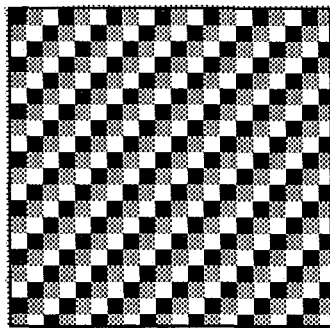
APRENDIDA

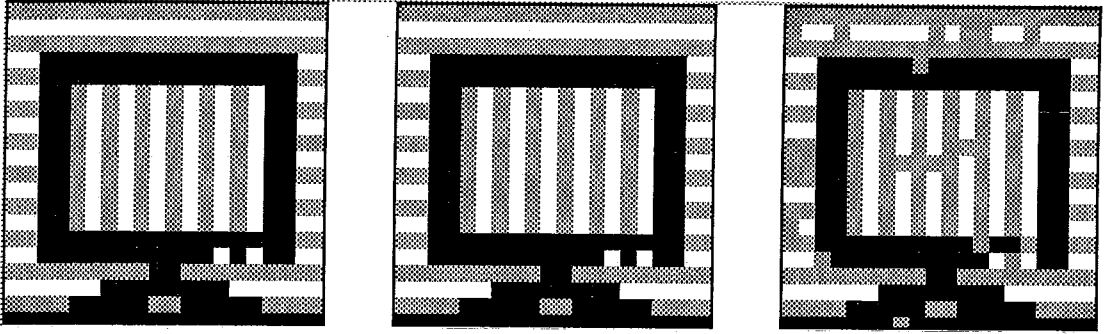
DESCOMPACTADA





Taxa de compactação = 0.90





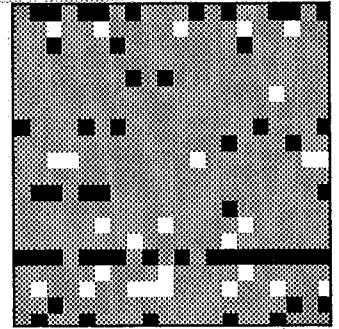
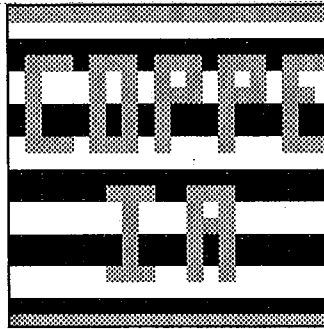
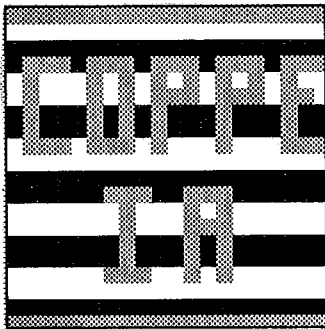
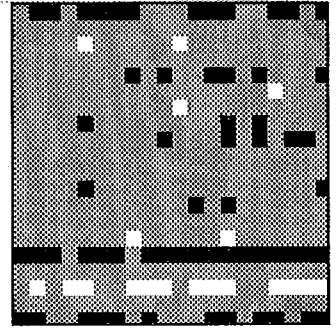
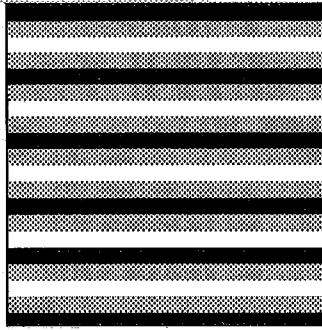
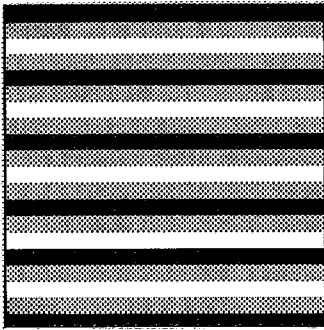
Taxa de compactação = 0.70

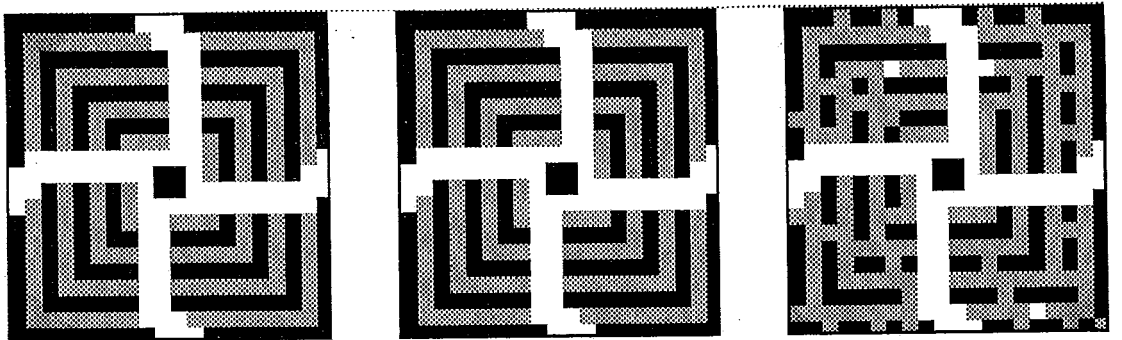
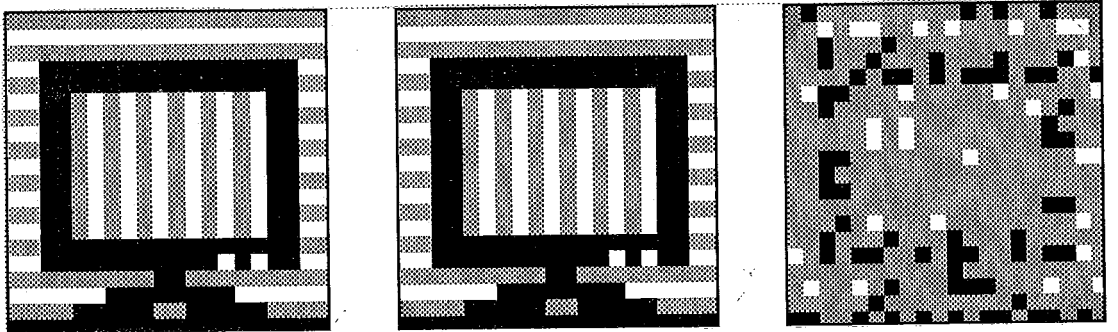
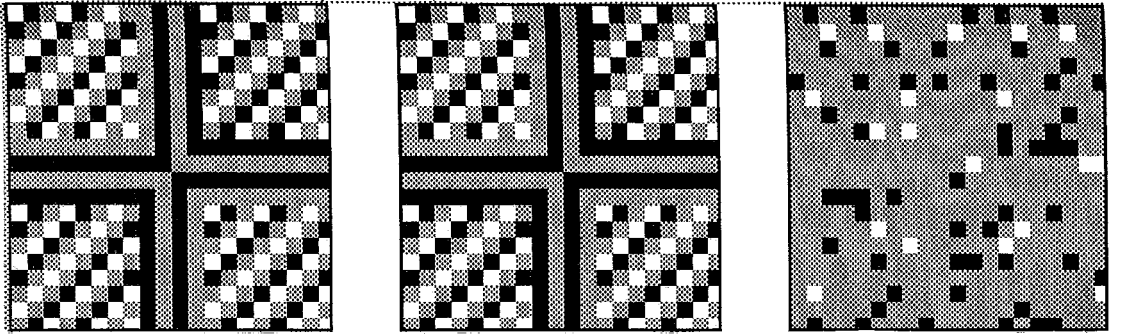
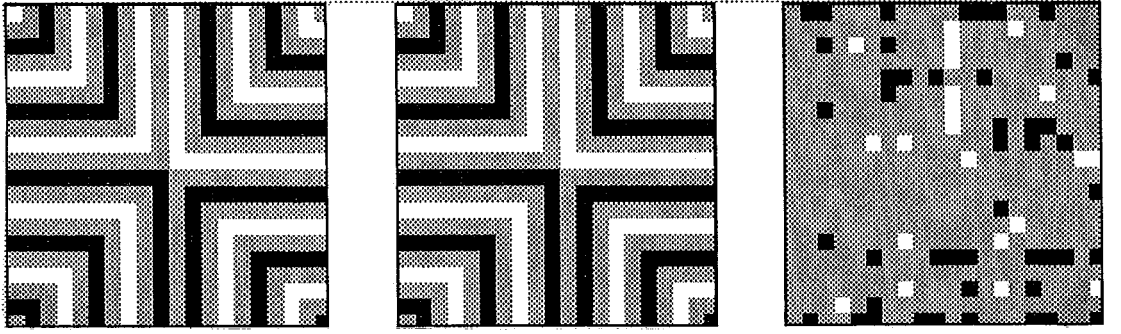
GRUPO 4

ORIGINAL

APRENDIDA

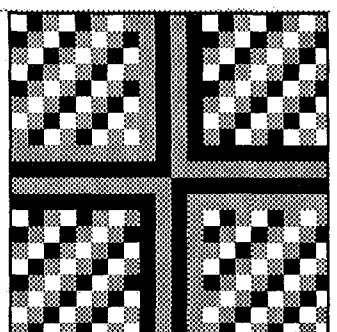
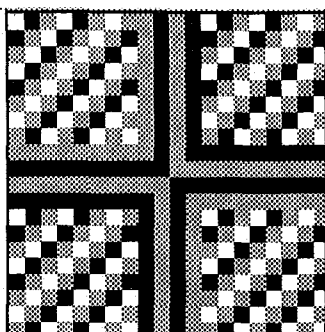
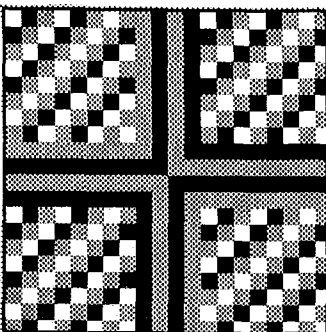
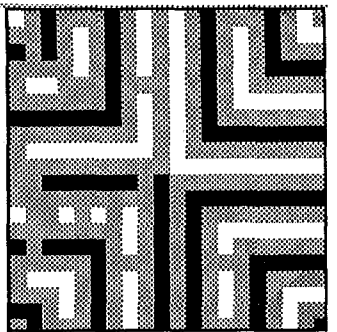
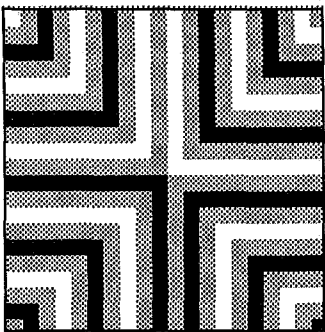
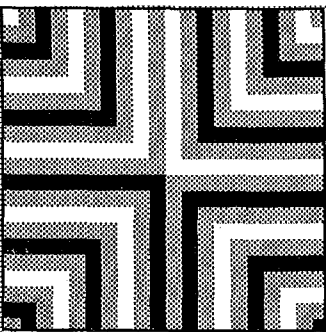
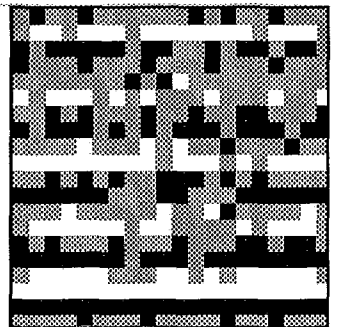
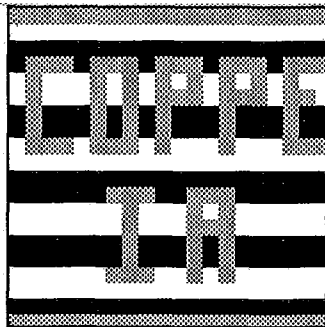
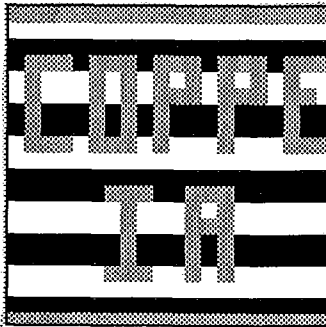
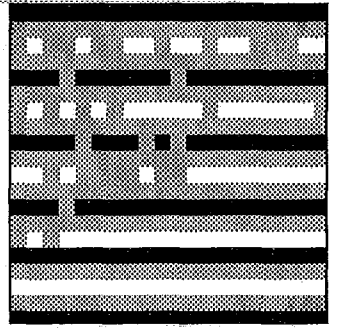
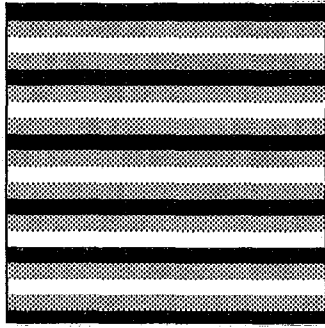
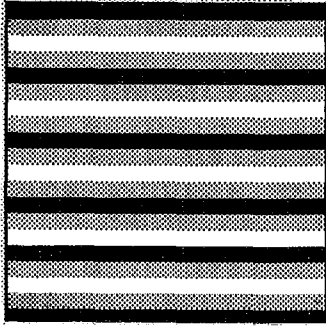
DESCOMPACTADA

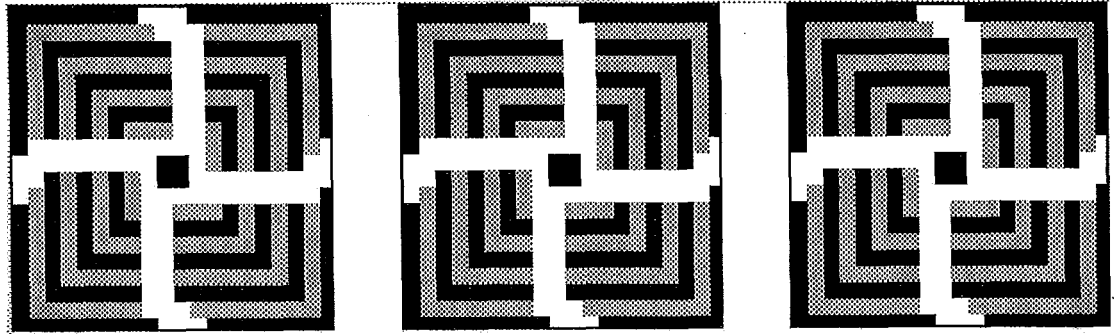
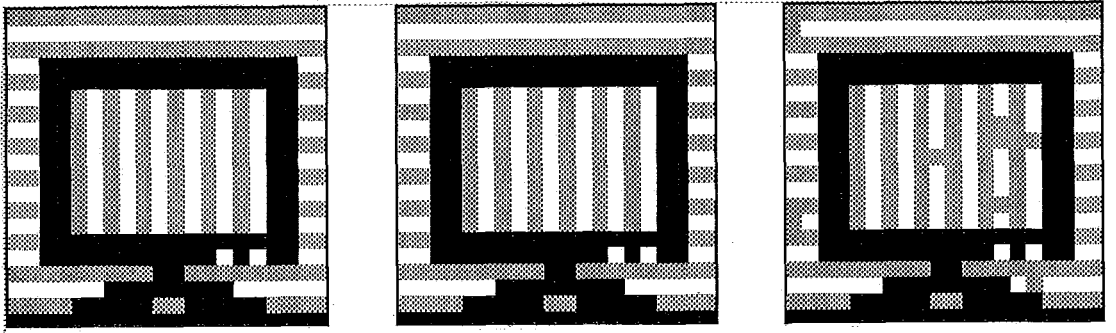




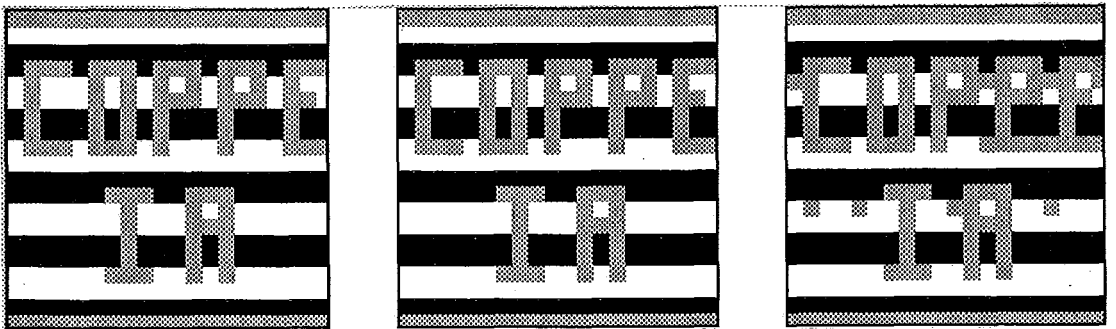
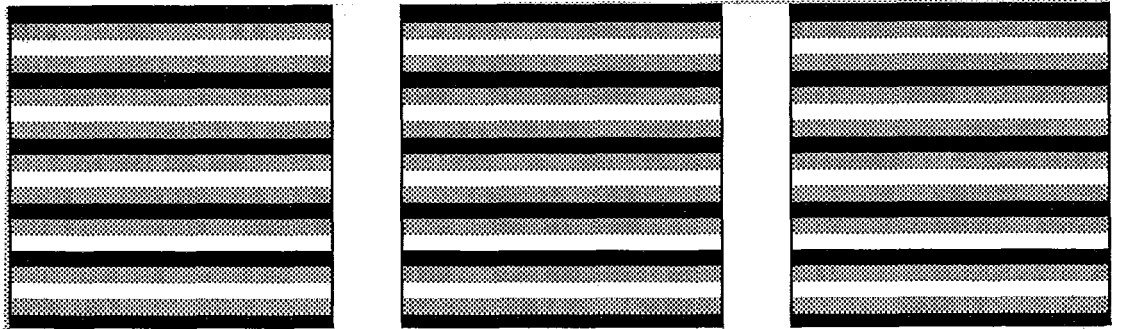
Taxa de compactação = 0.90

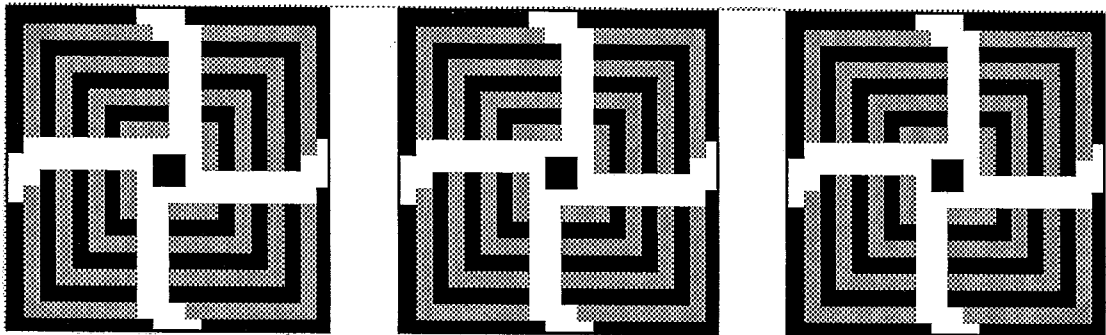
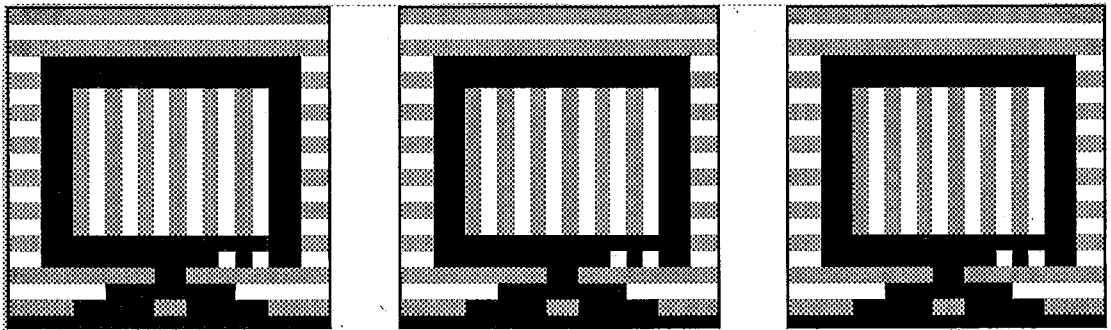
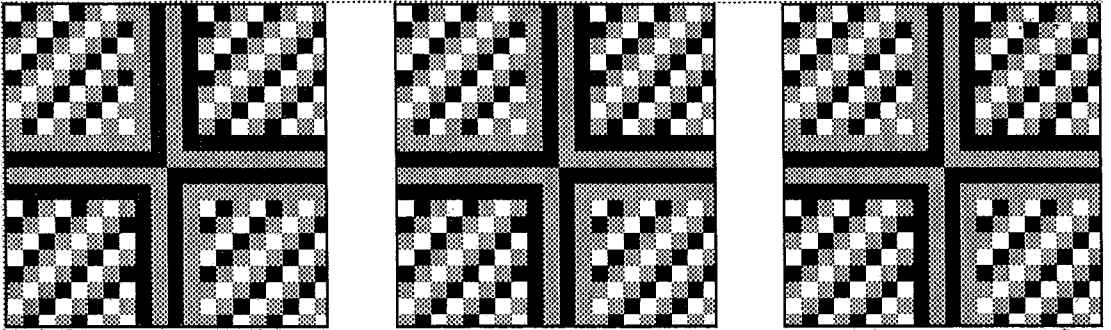
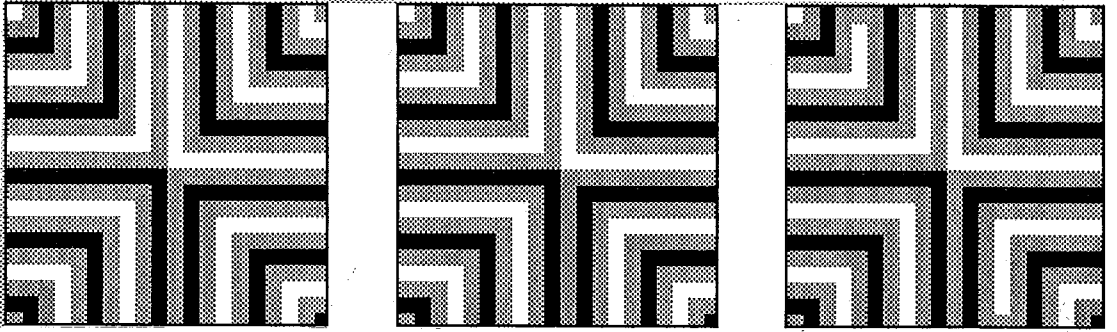






Taxa de compactação = 0.70





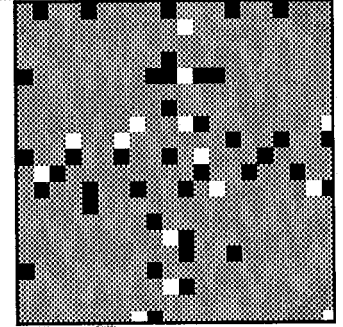
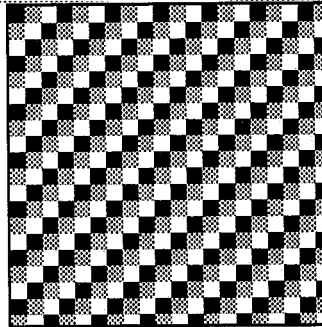
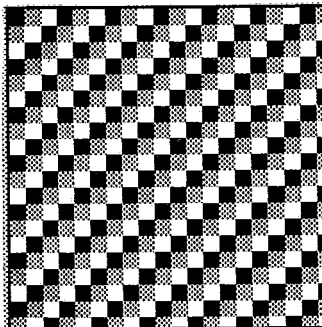
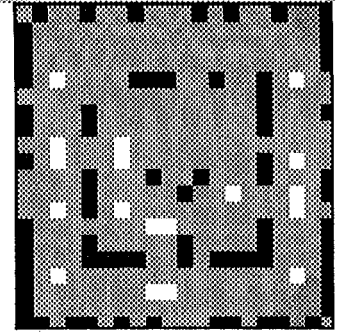
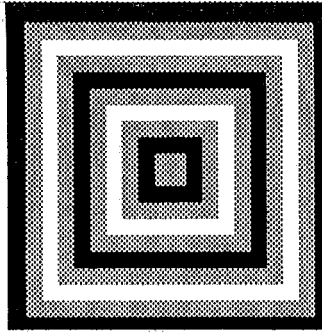
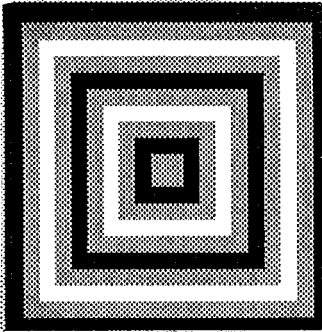
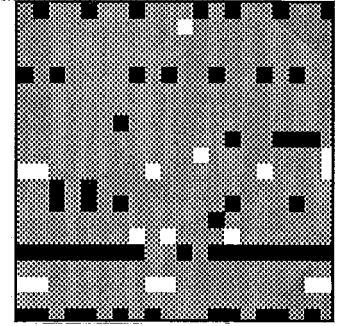
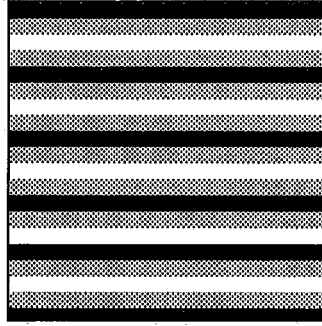
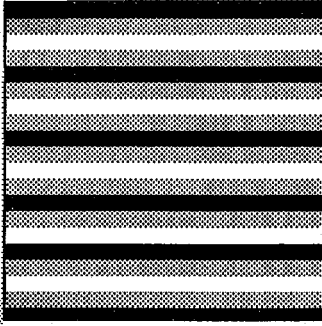
Taxa de compactação = 0.60

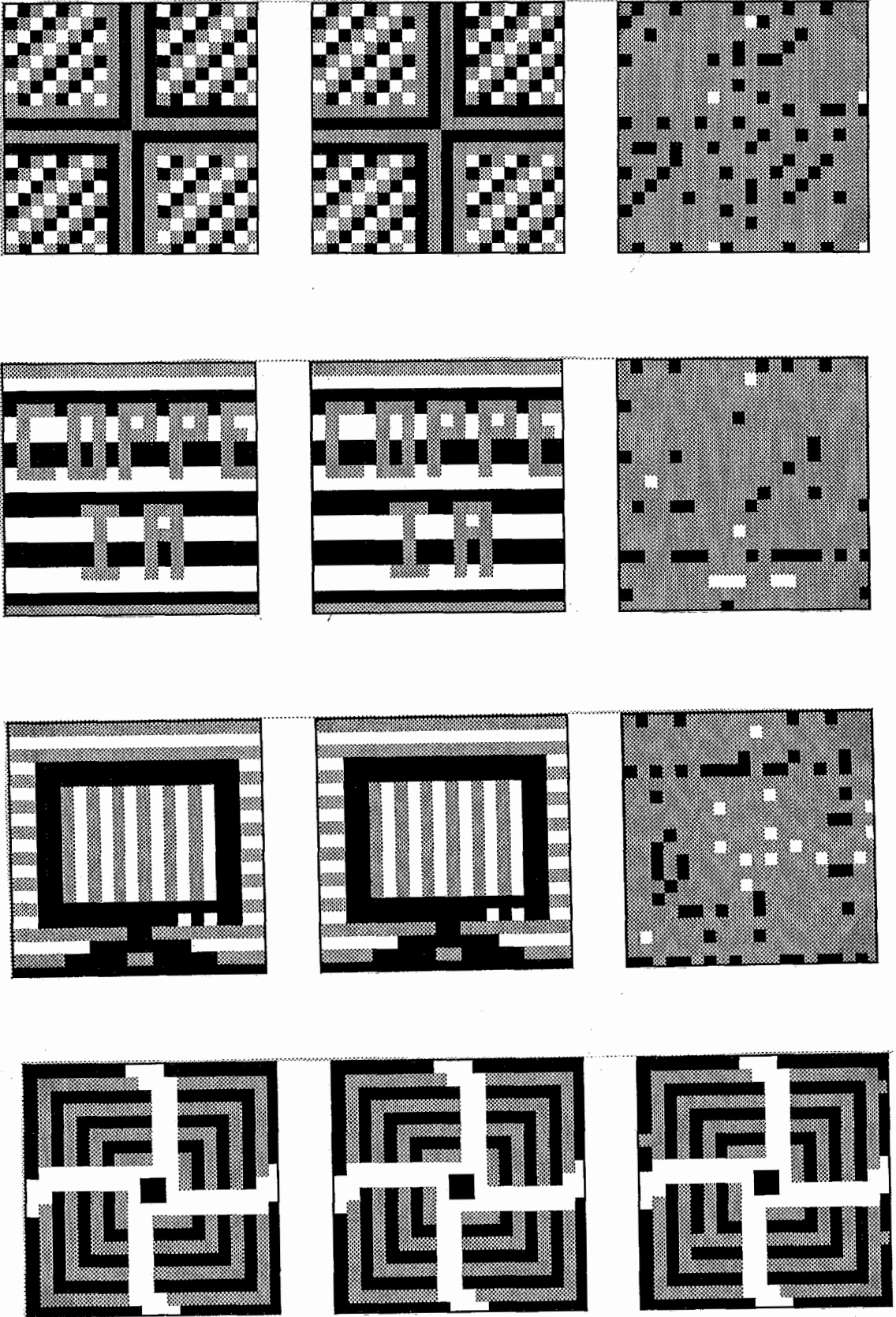
## GRUPO 5

ORIGINAL

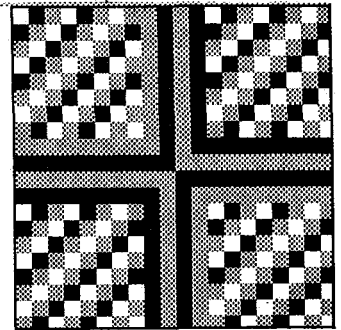
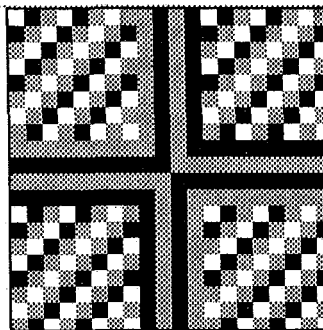
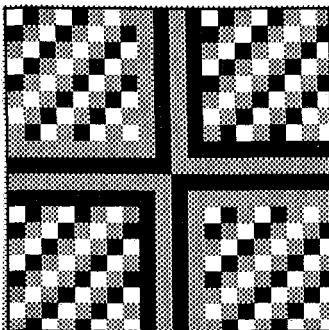
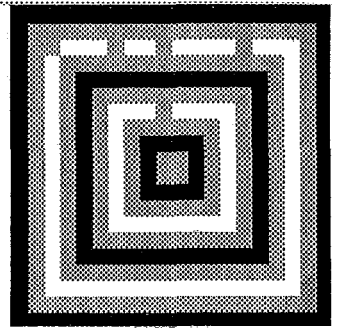
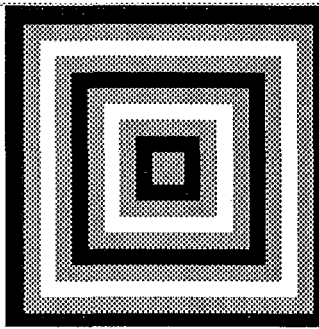
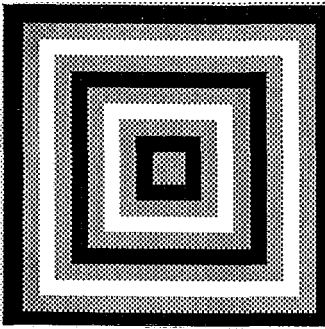
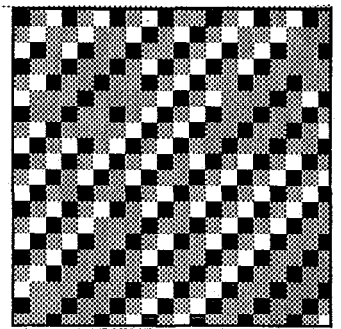
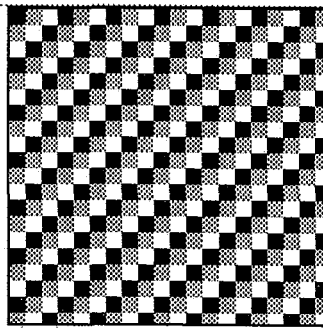
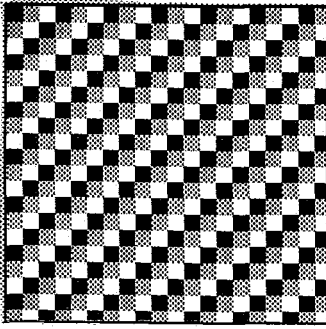
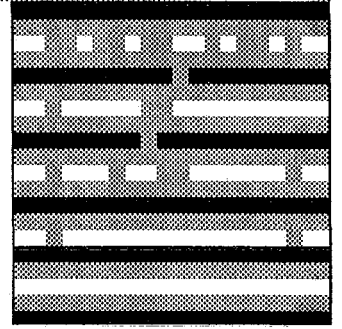
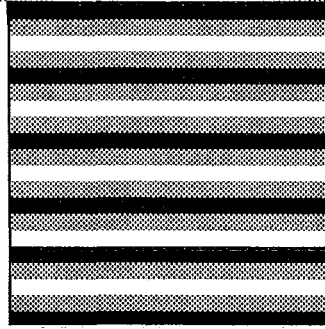
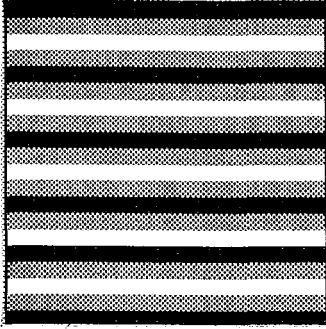
APRENDIDA

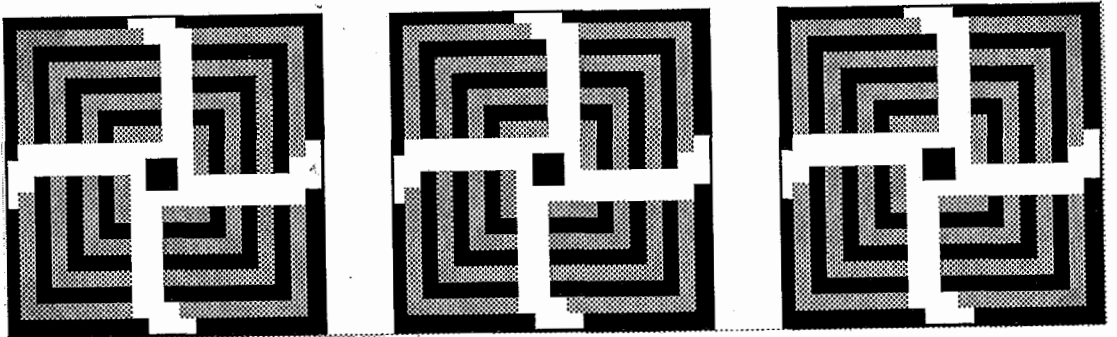
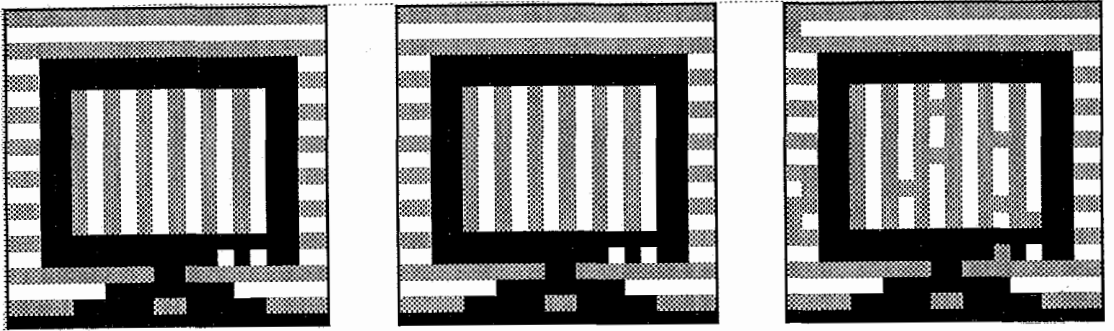
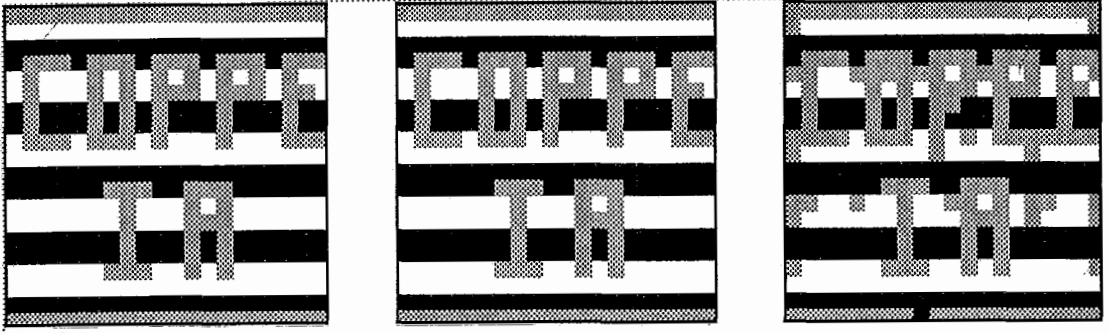
DESCOMPACTADA





Taxa de compactação = 0.90





Taxa de compactação = 0.70

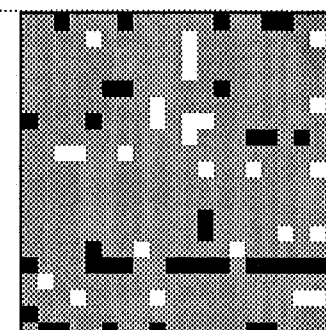
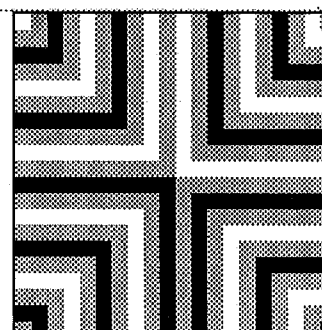
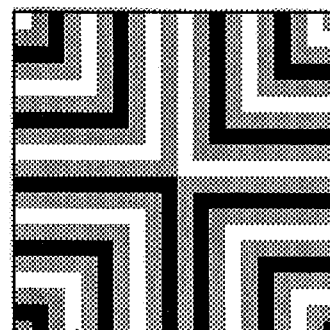
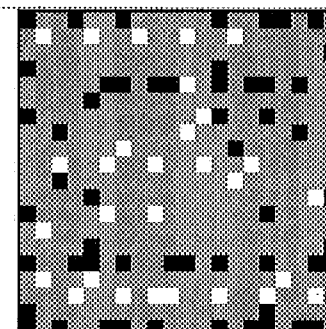
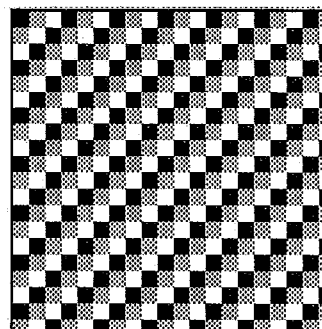
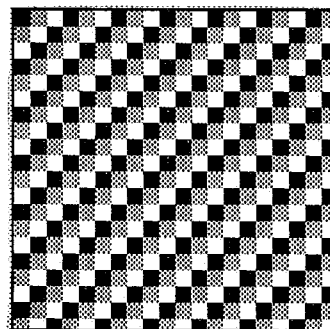
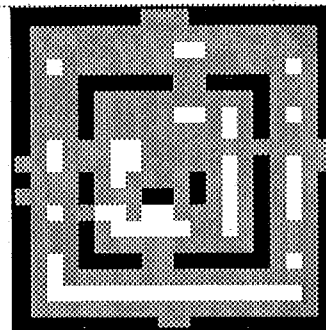
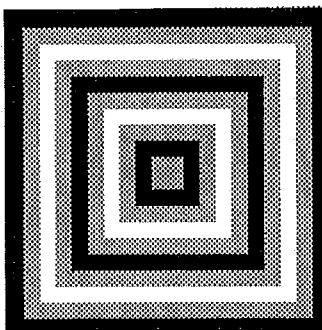
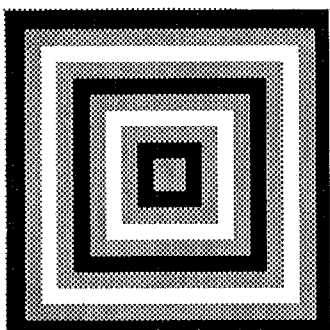
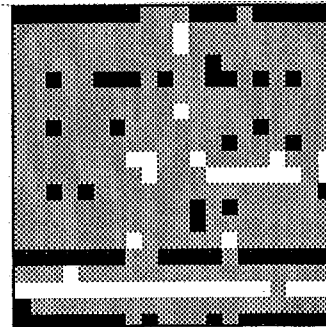
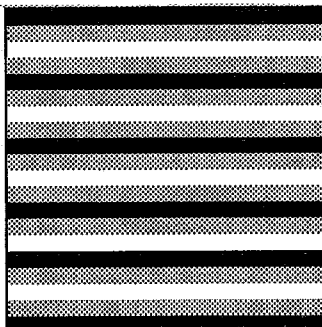
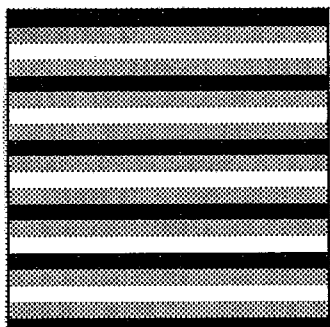


## GRUPO 6

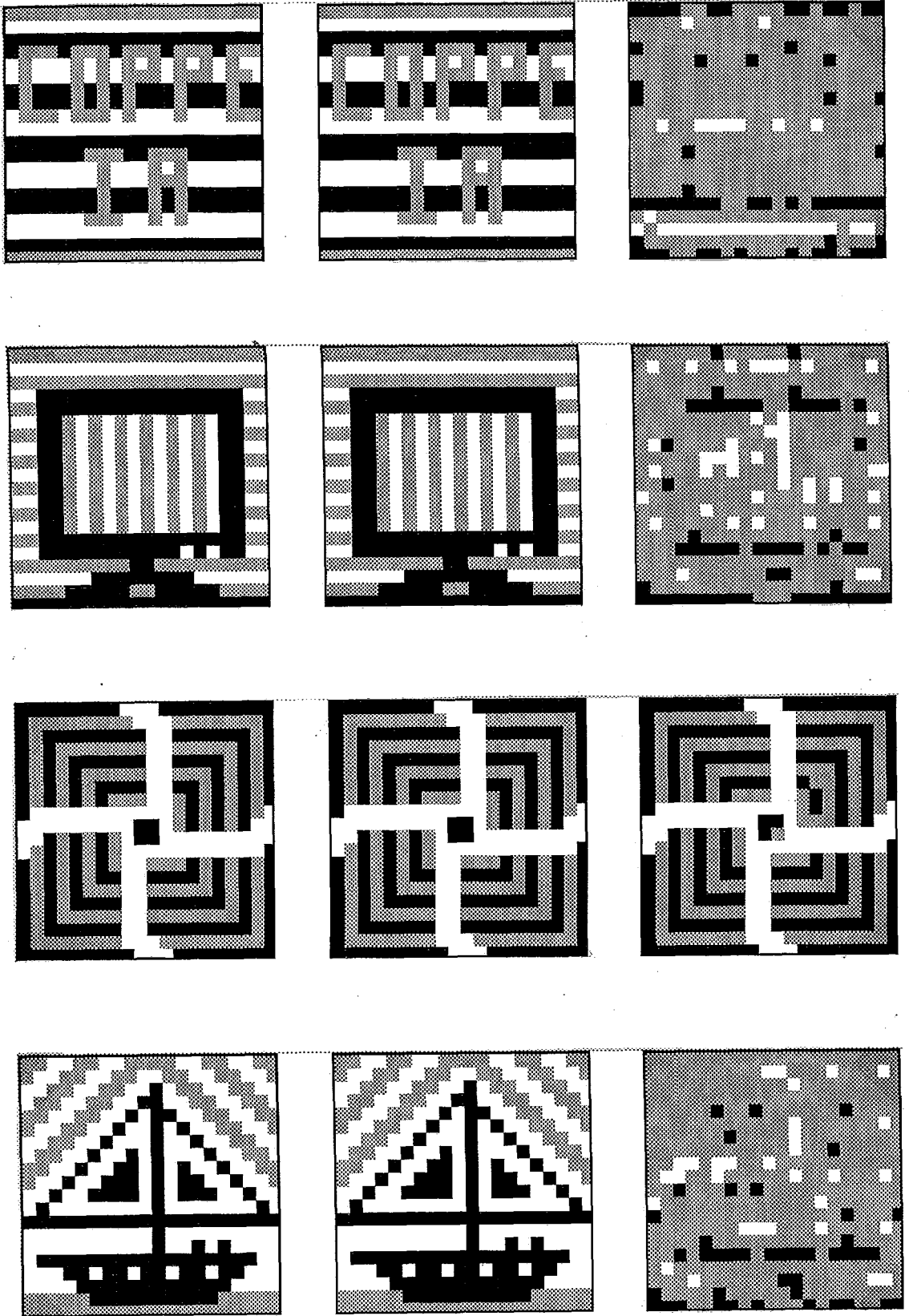
ORIGINAL

APRENDIDA

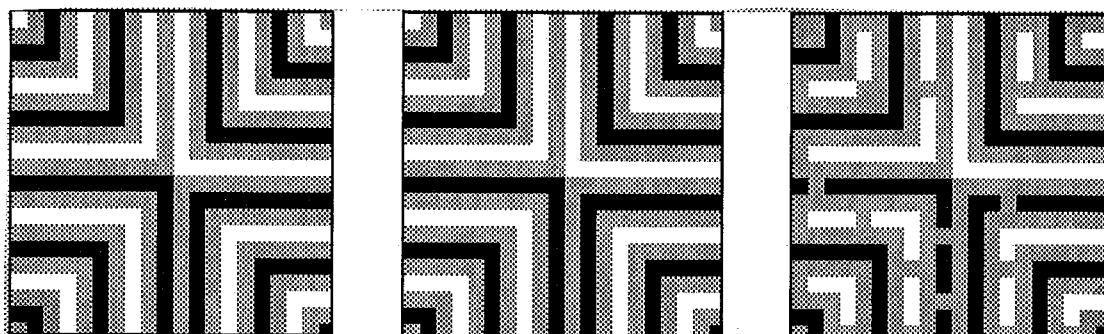
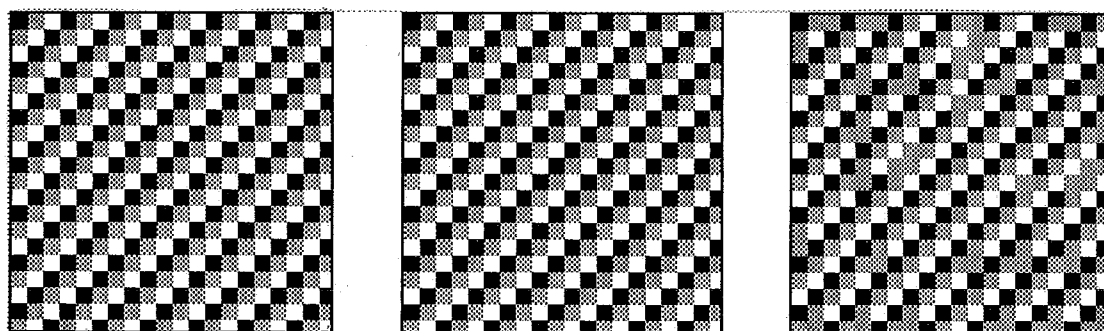
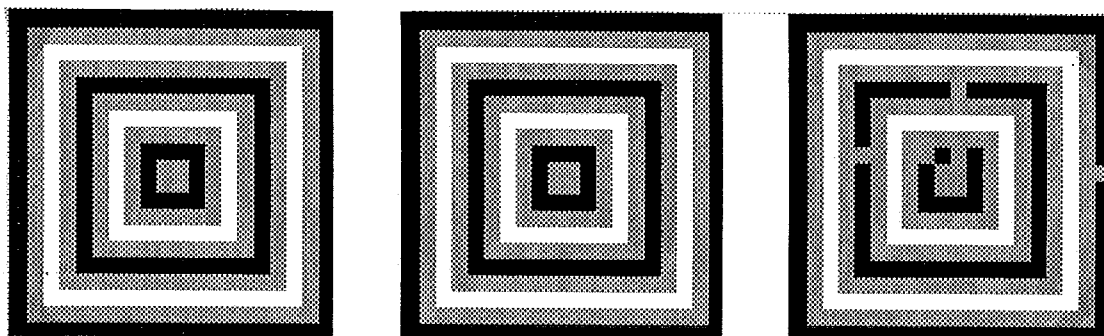
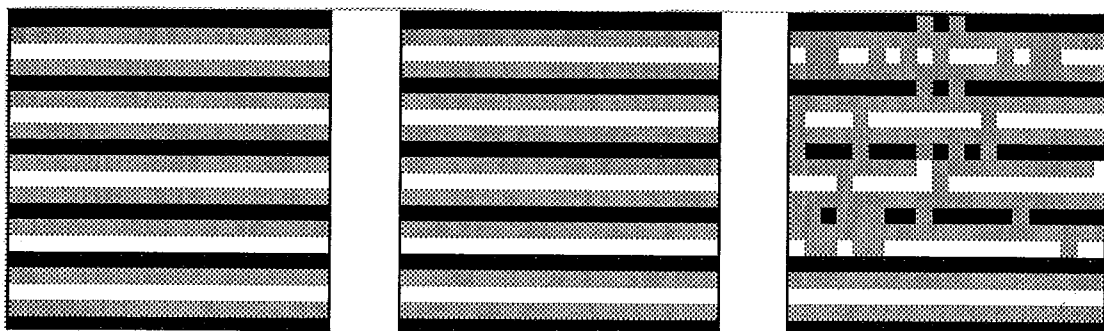
DESCOMPACTADA

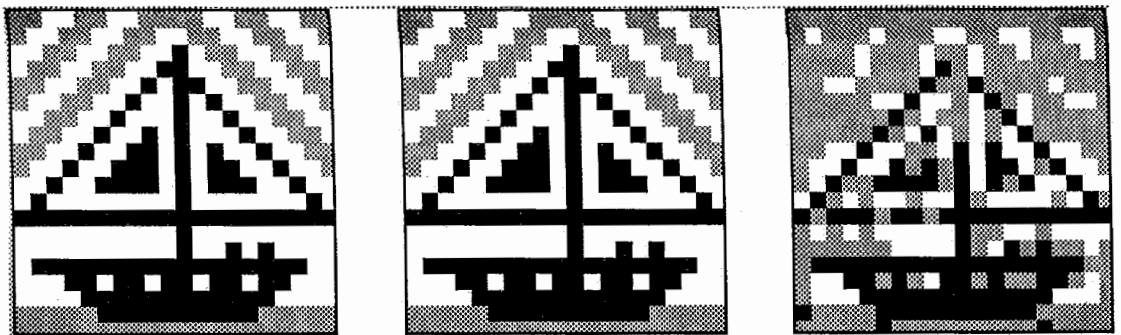
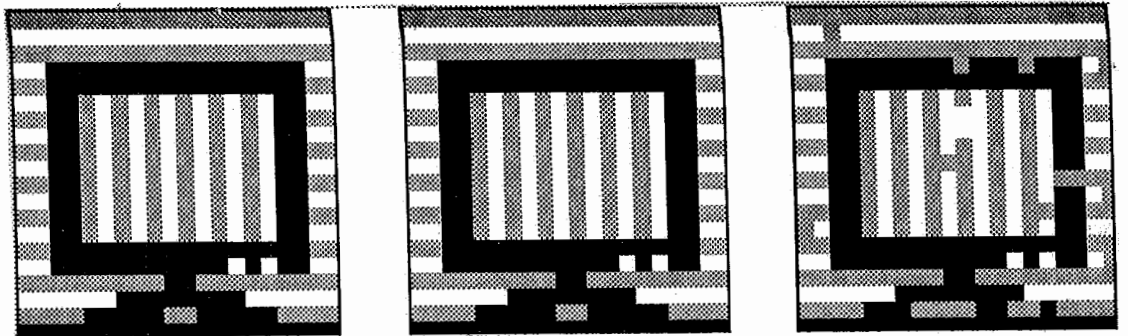
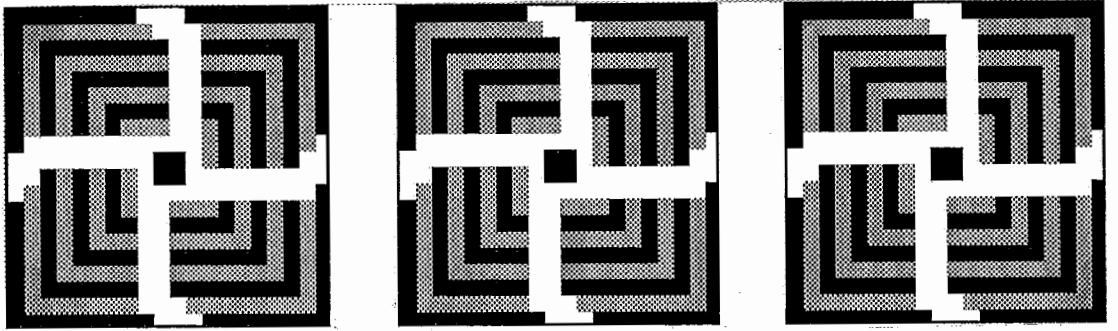
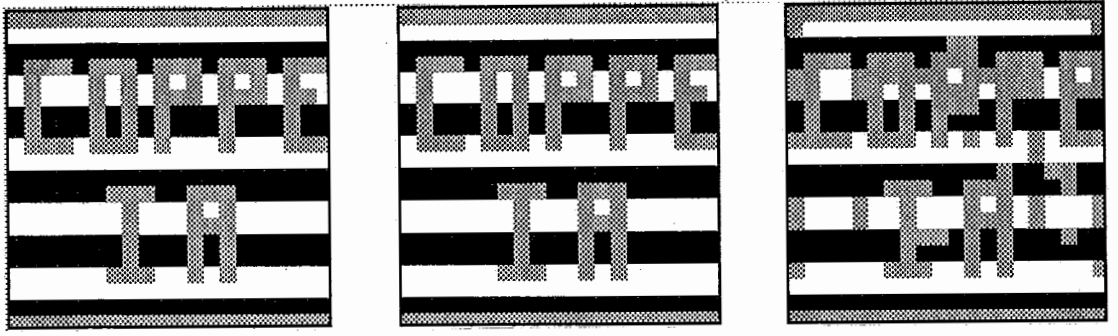




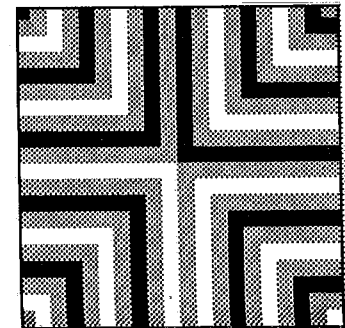
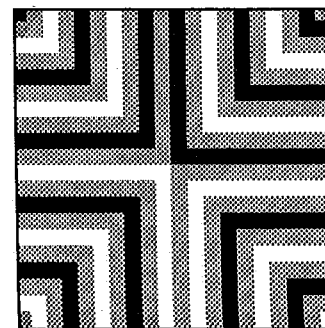
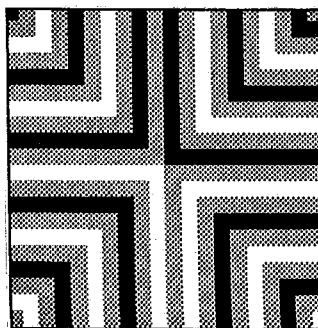
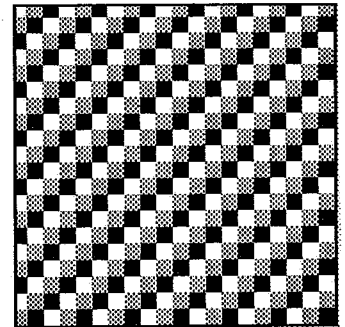
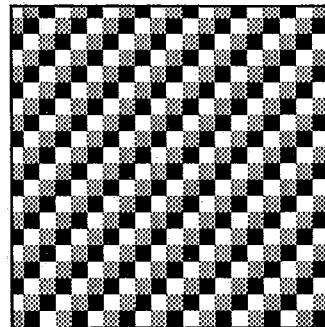
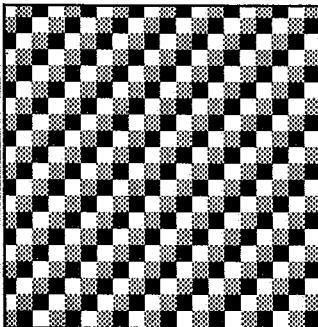
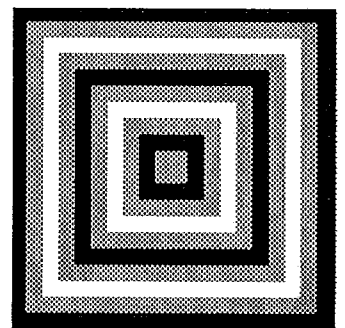
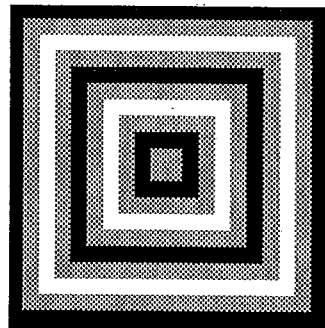
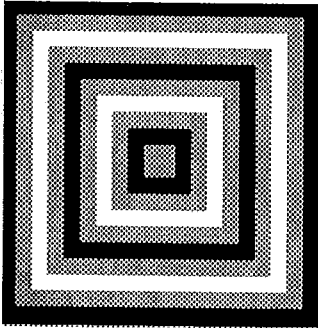
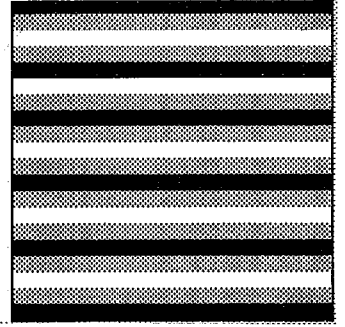
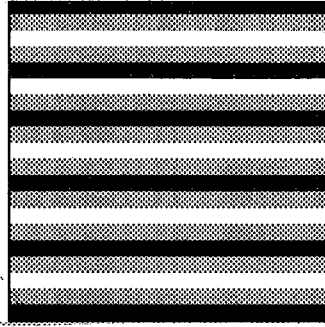
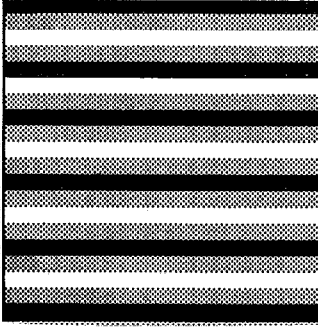


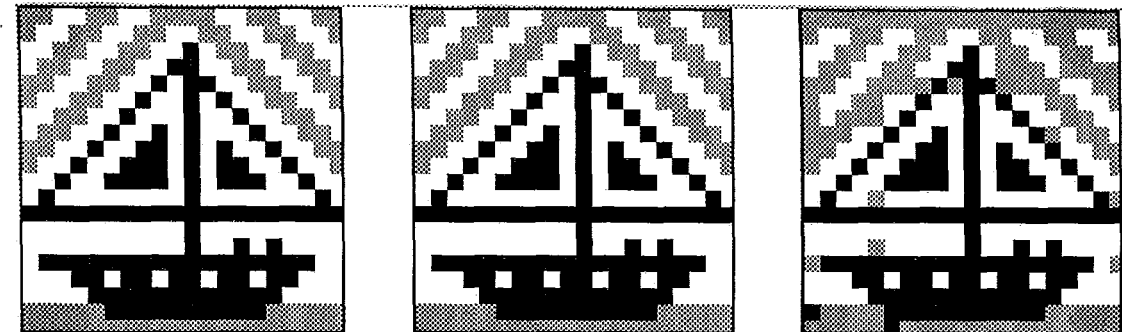
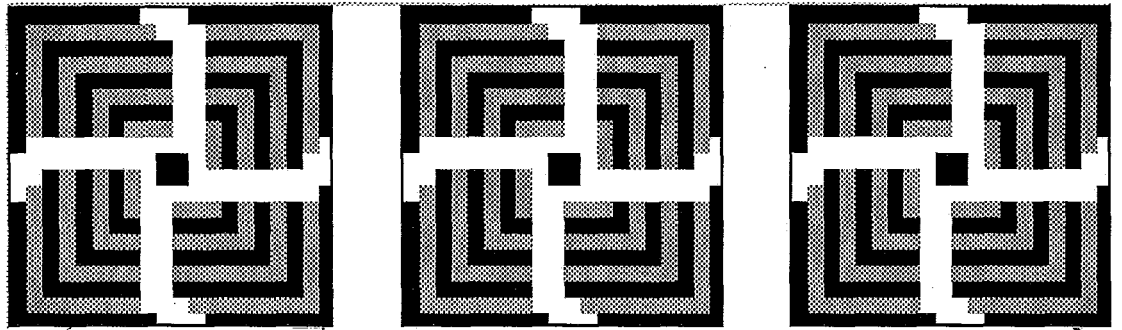
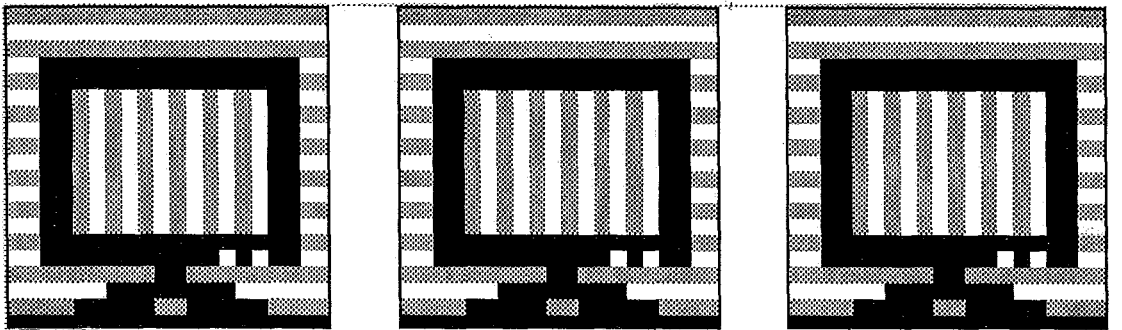
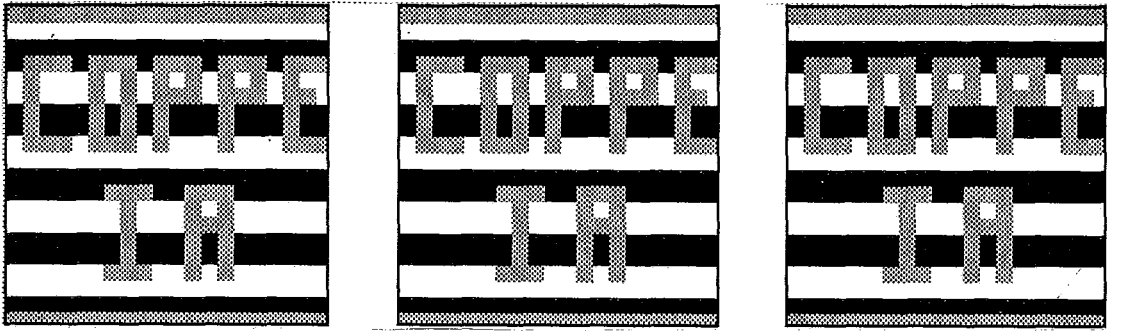
Taxa de compactação = 0.90





Taxa de compactação = 0.70





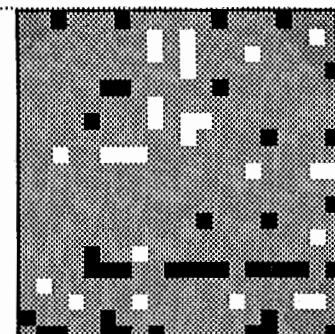
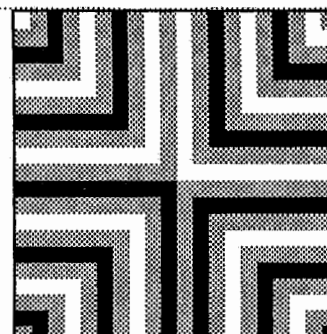
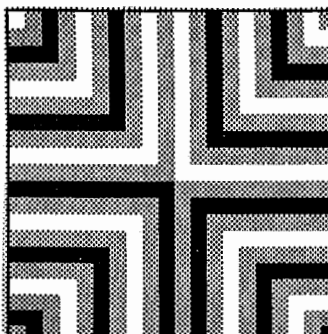
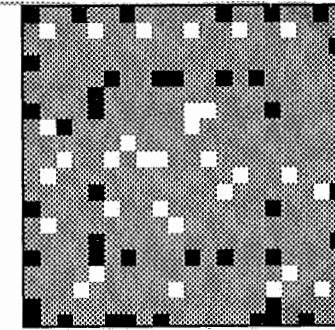
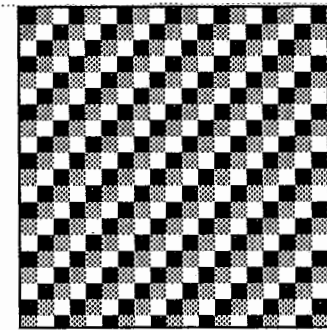
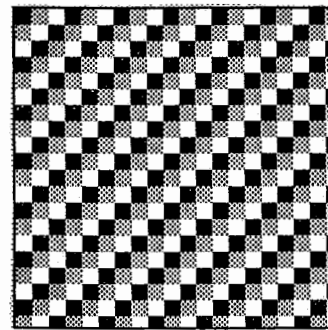
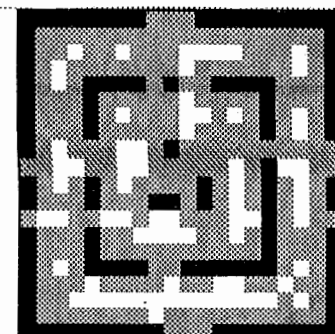
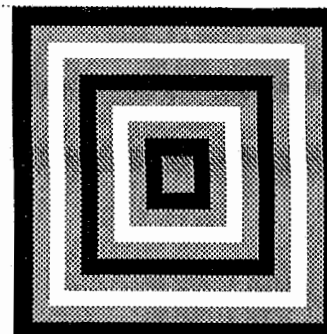
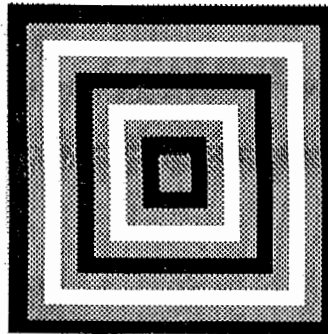
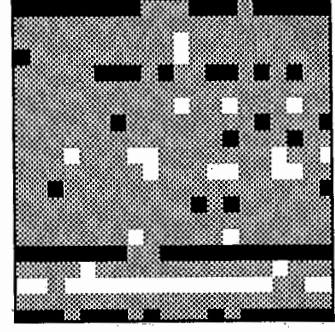
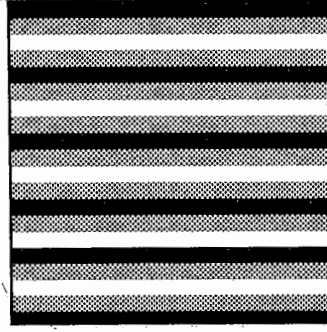
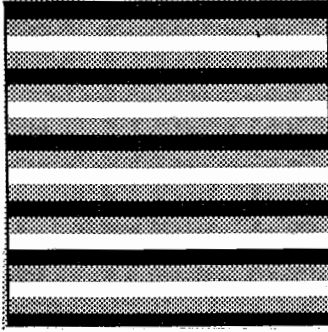
Taxa de compactação = 0.50

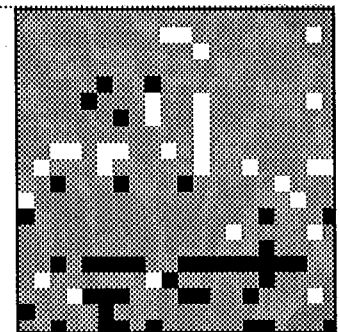
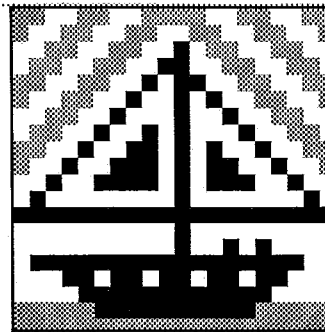
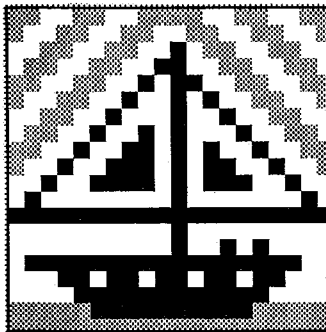
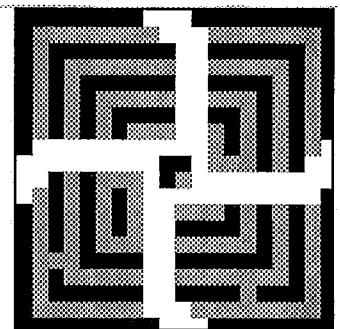
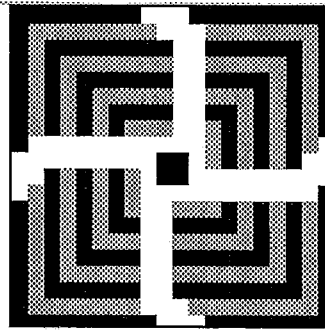
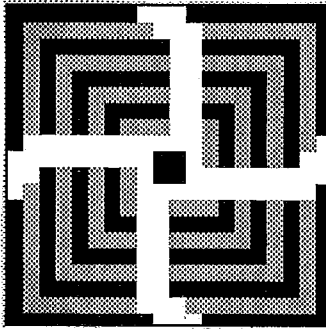
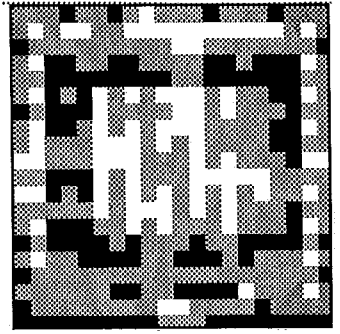
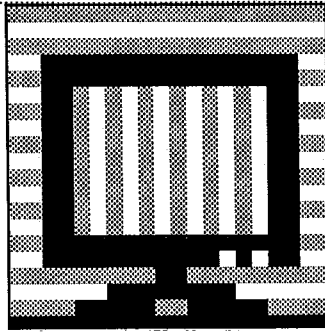
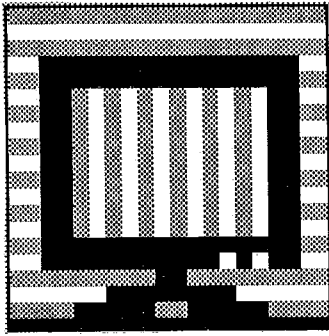
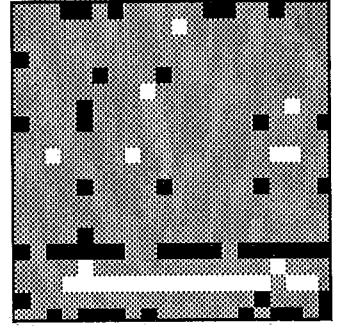
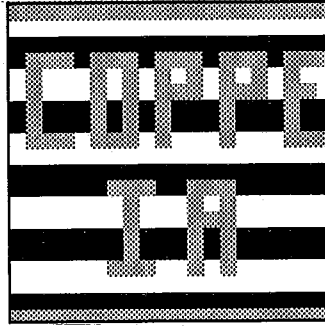
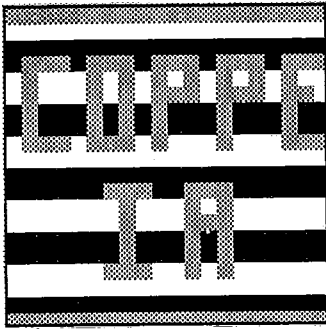
## GRUPO 7

ORIGINAL

APRENDIDA

DESCOMPACTADA

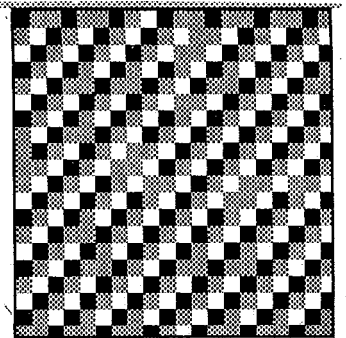
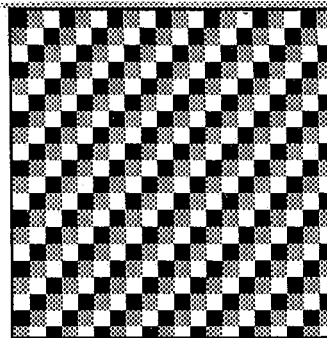
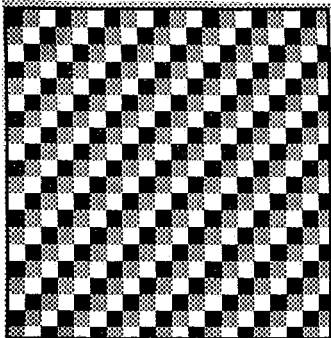
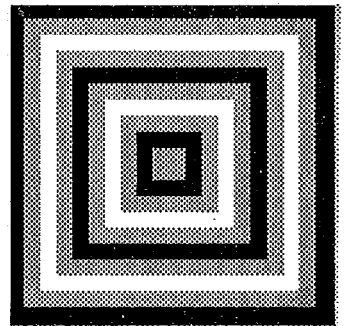
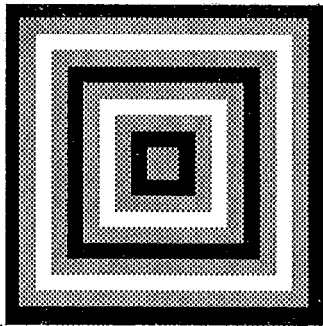
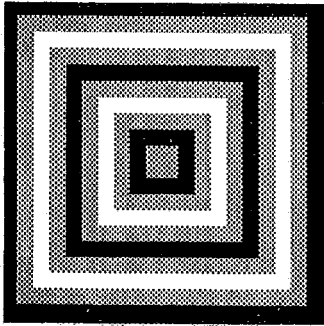
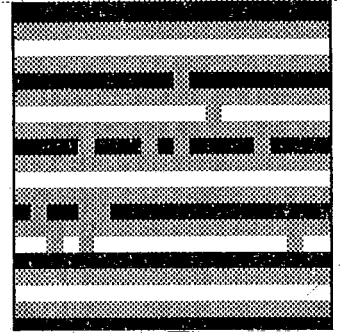
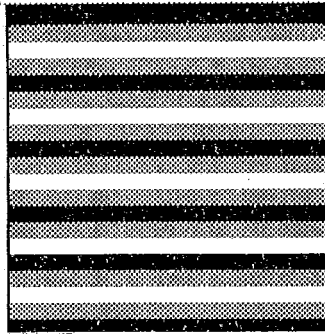




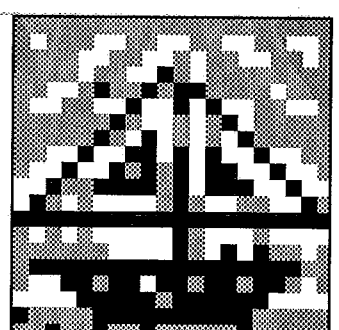
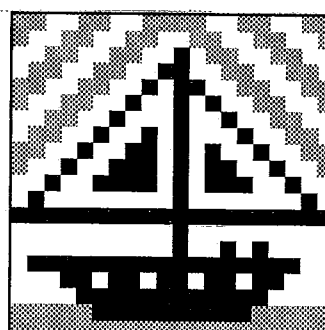
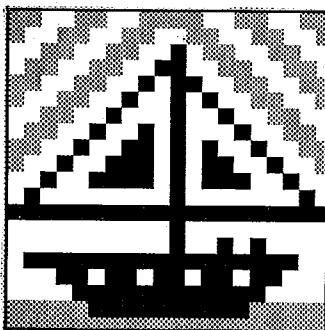
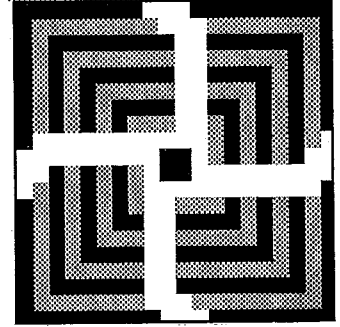
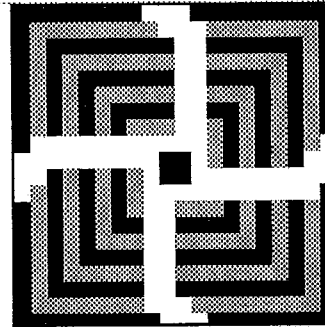
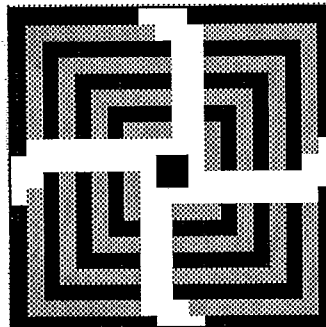
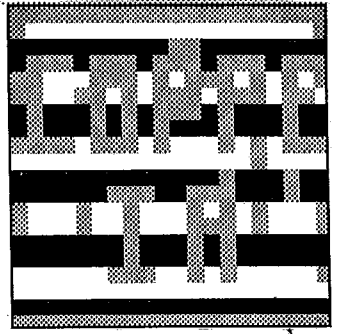
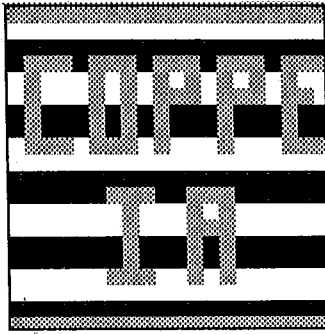
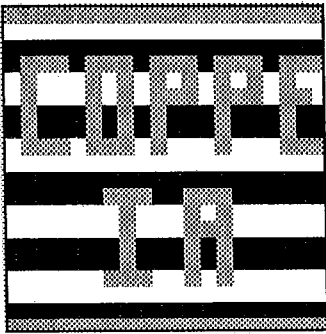
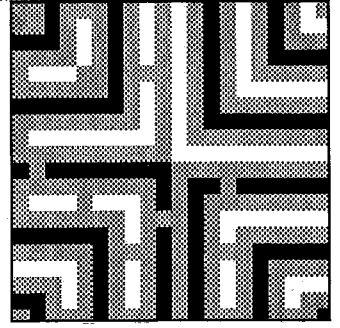
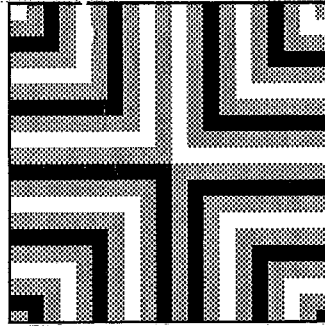
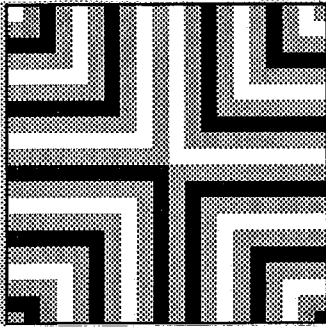


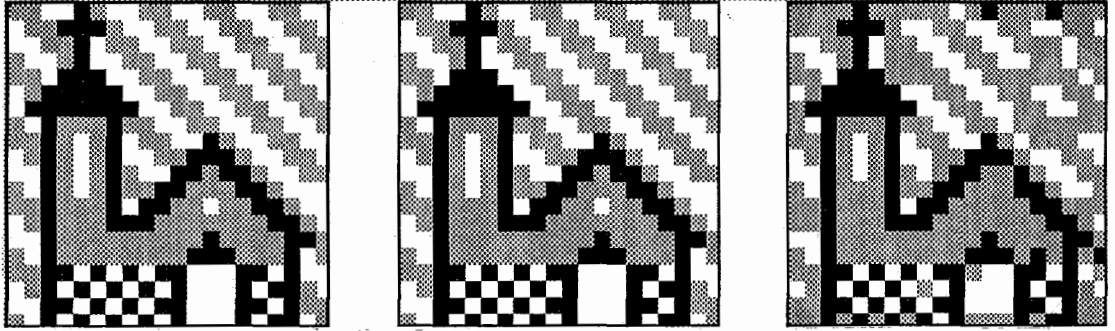
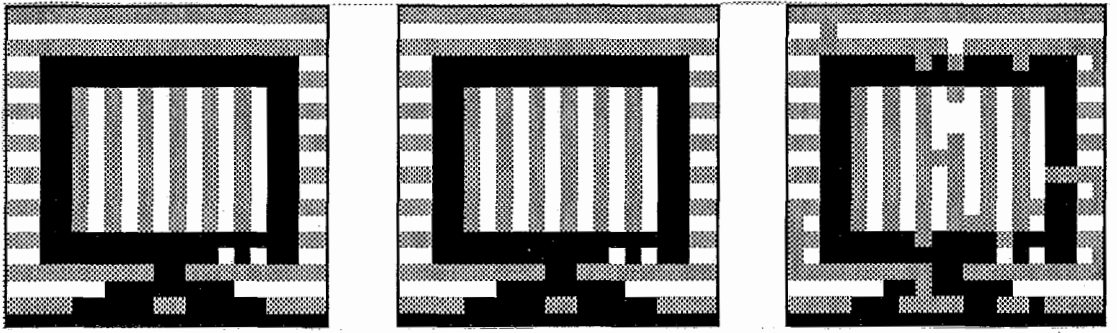


Taxa de compactação = 0.90

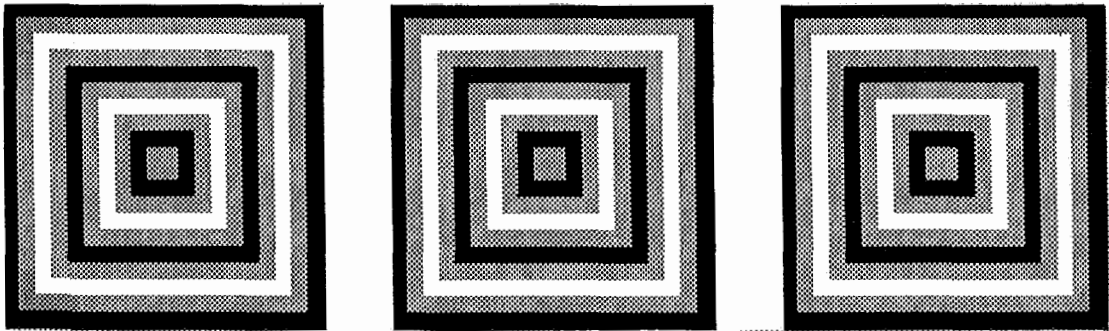
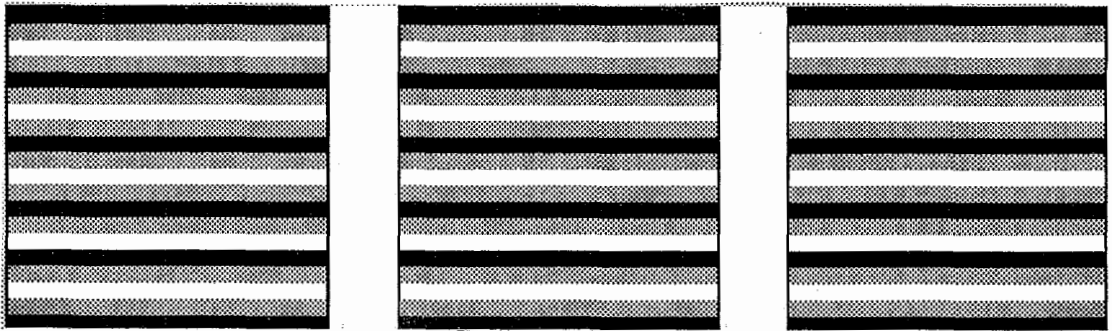


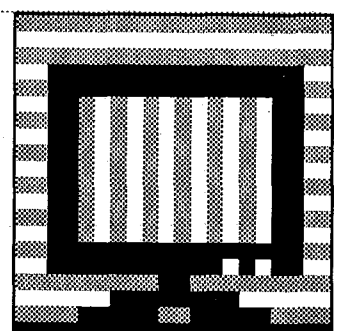
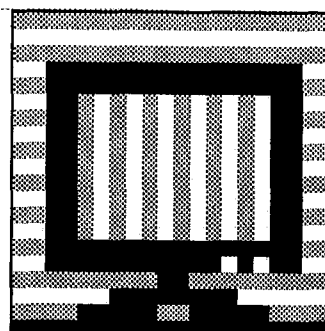
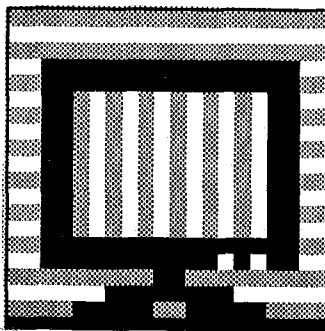
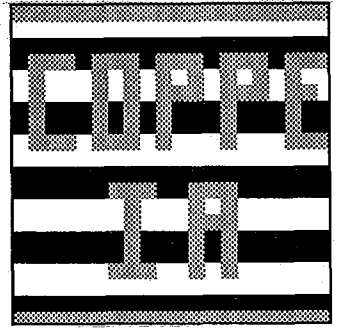
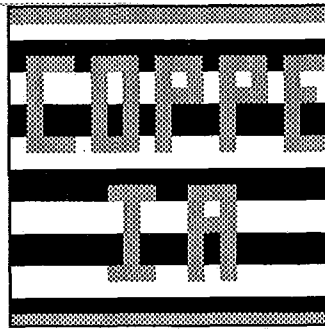
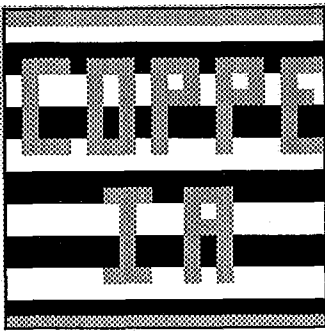
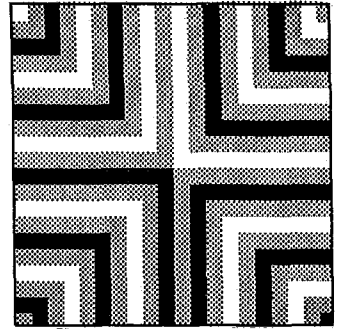
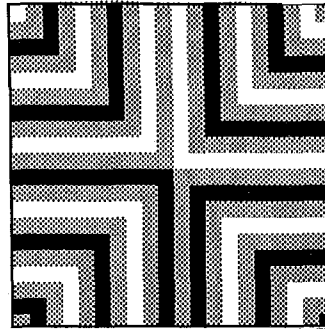
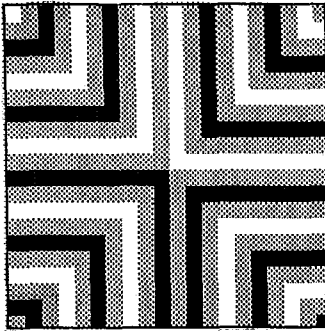
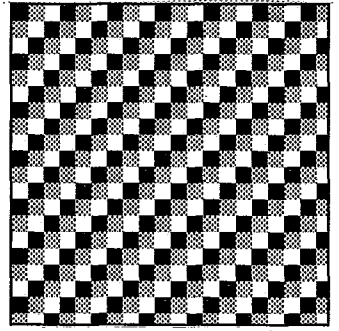
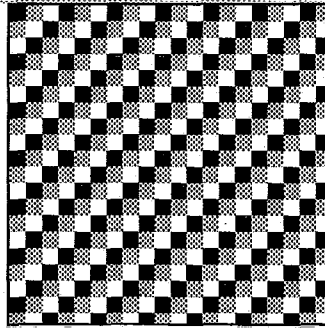
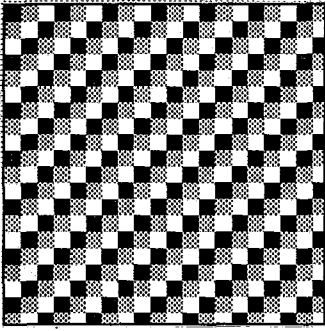


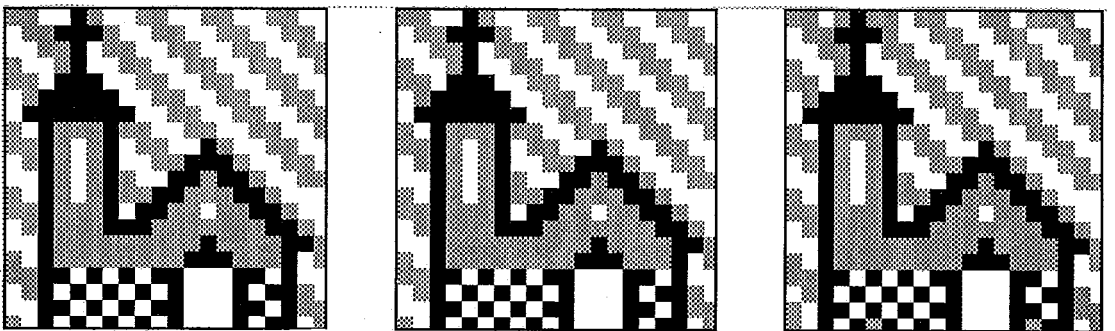
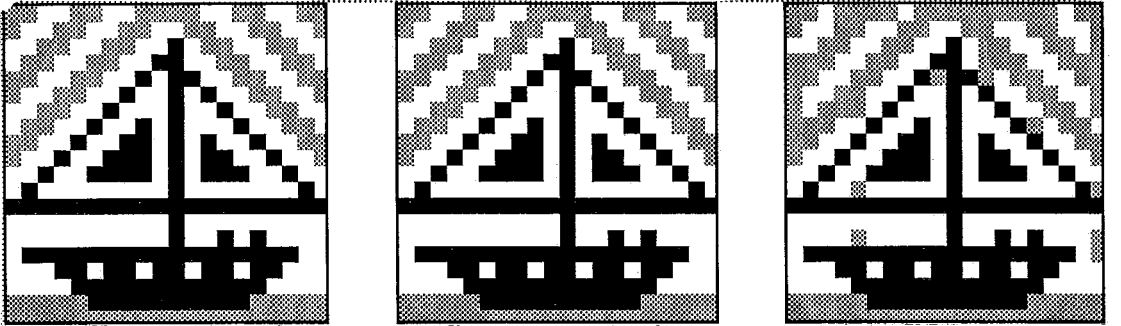
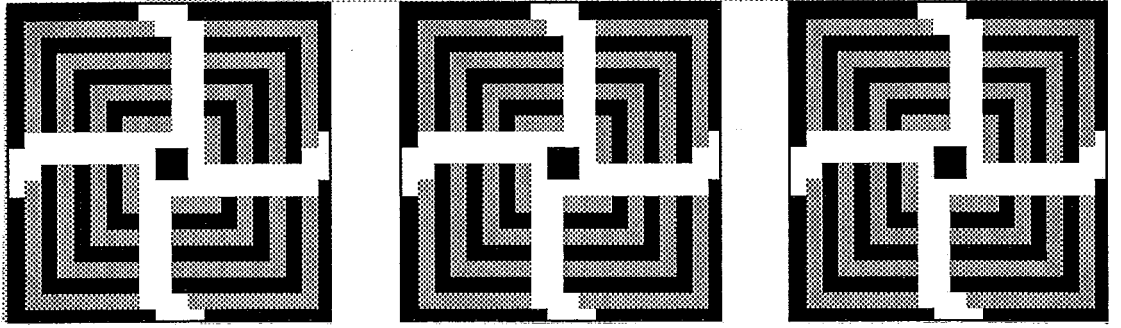




Taxa de compactação = 0.70







Taxa de compactação = 0.5

# Referências Bibliográficas

- [1] Barbosa, V. C., Carvalho, L. A. V., "Learning in Analog Hopfield Networks", *International Joint Conference on Neural Networks - IJCNN* (1991).
- [2] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. National Academy of Sciences USA* 79, 2554-2558 (1982).
- [3] Hopfield, J. J., "Neurons with Graded Response have Collective Computational Properties like those of two-state neurons", *Proc. National Academy of Sciences, USA* , 81, 3088-3092 (1984).
- [4] Hopfield, J. J., Tank, D. W., "Computing with Neural Circuits: A Model", *California Institute of Technology (CA), Molecular Biophysics Research Department , AT & T Bell Laboratories, Murray-Hill, NJ* (1986).
- [5] Carvalho, L. A. V., " Síntese de Redes Neurais com Aplicações à Representação do Conhecimento e à Otimização ", *Tese de Doutorado, COPPE-UFRJ* (1989).
- [6] Sonehara, N., Kawato, M., Myiane, S., Nakane, K., "Image Data Compression using a Neural Network Model", *International Joint Conference on Neural Networks - IJCNN* (1989)
- [7] Anthony, D., Hines, Evor, Barham, J., Taylor, D., "A Comparison of Image Compression by Neural Networks and Principal Components Analysis", *International Joint Conference on Neural Networks - IJCNN* (1989).
- [8] Bourland, H., Kamp, Y., "Auto-association by Multilayer Perceptrons and Singular Value Decomposition", *Biological Cybernetics, v. 59, pp 291-294* (1988).

- [9] Arozullah, M., Namphol, A., "A Data Compression System using Neural Network Based Architecture", *International Joint Conference on Neural Networks - IJCNN* (1990)
- [10] Namphol A., Arozullah, M., Chin, S., "Higher Order Data Compression with Neural Networks", *International Joint Conference on Neural Networks (IJCNN), Seattle, WA* (1991).
- [11] Feldman, J. A., "Connections," *Byte*, pp 277-284, April 1985.
- [12] Simpson, P. K., "Artificial Neural Systems, Foundations, Paradigms, Applications and Implementations", *Neural Networks : Research and application; Pergamon Press, Inc.* (1990).
- [13] Jain, A. K., "Image Data Compression: a review", *Proceedings of the IEEE*, v. 69, no. 3, 349 - 389, March 1981.
- [14] Carvalho, L. A. V., et al., "Redes Neurais Artificiais: A volta do cérebro Eletrônico ?", *Ciência Hoje*, v.12, n.12, pp 12 - 21, Jan./Fev. 1991.
- [15] Carvalho, L. A. V., "Redes Neurais e a Tradição Conexionista da Inteligência Artificial", *Rio de Janeiro, [UFRJ]* (1988).
- [16] McClelland, J. L., Rumelhart, D. E., Hinton, G. E., "The Appeal of Parallel Processing", *Parallel Distributed Processing, Eds. Rumelhart, D.E., McClelland, J.L., The MIT Press, Cambridge, Massachusetts* (1986)
- [17] Rumelhart, D. E., Hinton, G. E., McClelland, J. L., " A General Framework for Parallel Distributed Processing", *Parallel Distributed Processing, Eds. Rumelhart, D.E., McClelland, J.L., The MIT Press, Cambridge, Massachusetts* (1986)
- [18] Rumelhart, D. E., McClelland, J. L., " PDP Models and General Issues in Cognitive Science", *Parallel Distributed Processing, Eds. Rumelhart, D.E., McClelland, J.L., The MIT Press, Cambridge, Massachusetts* (1986)

- [19] Rumelhart, D. E., Zipser, D., "Feature Discovery by competitive Learning", *Parallel Distributed Processing, Eds. Rumelhart, D.E., McClelland, J.L., The MIT Press, Cambridge, Massachusetts* (1986)
- [20] Rumelhart, D. E., Hinton, G. E., Williams, R. J., "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing, Eds. Rumelhart, D.E., McClelland, J.L., The MIT Press, Cambridge, Massachusetts* (1986)
- [21] Andrews, H. C., Hunt, B. R., "Digital Image Restoration", *Prentice-Hall, Inc., Englewood Cliffs, New Jersey* (1977)
- [22] Ahalt, S. C., Prakoon, C., Krishnamurthy, A. K., "Performance analysis of two image vector quantization techniques", *Department of Electrical Engineering, The Ohio State University, International Joint Conference on Neural Networks - IJCNN* (1989)
- [23] Kung, S. Y., "Orthogonal Learning Network for Constrained Principal Component Problem", *Dept. of Electrical Engineering, Princeton University, International Joint Conference on Neural Networks - IJCNN* (1990)
- [24] Lawrence, Jeannette "Jet", "Untangling Neural Nets", *Dr. Dobbs' Journal, April* (1990)
- [25] Abu-Mostafa, Y. S., Jacques, Jeannine-Marie St., "Information capacity of the Hopfield Model," *IEEE*, 1985.
- [26] Pessoa, Luiz Adauto F. C., "Aprendizado não-supervisionado em redes neurais," *Tese de Mestrado, Rio de Janeiro, COPPE/UFRJ* (1990)
- [27] Soares, Luiz Fernando Gomes et al., "Fundamentos de Sistemas Multimídia", *Porto Alegre : Instituto de Informática da UFRGS, 1992, (VIII Escola de Computação)*