SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 19 . 02 . 2003

Assinatura : Ara. Paulo L. M. Silva

Reconhecimento de Padrões Multi-valorados por Redes Neurais Caóticas

Juan Carlos Gutiérrez Cáceres

Orientação: Professor Dr. Zhao Liang

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências, na área de Ciências de Computação e Matemática Computacional.

USP - São Carlos

Fevereiro-2003

A Comissão Julgadora:

Prof. Dr. Zhao Liang

Handwritten signature of Zhao Liang in black ink, written over a horizontal line.

Profa. Dra. Roseli Aparecida Francelin Romero

Handwritten signature of Roseli Aparecida Francelin Romero in black ink, written over a horizontal line.

Prof. Dr. Elbert Einstein Nehrer Macau

Handwritten signature of Elbert Einstein Nehrer Macau in black ink, written over a horizontal line.

A minha mãe,
Eleana

Agradecimentos

A **Deus** e a **família**: minha mãe Eleana, meu pai Juan, minhas irmãs Glenda, Candy e Ivonne, e minha avó Manuela, pelo apoio e carinho que sempre recebi.

Ao meu orientador, Prof. Dr. Zhao Liang, por acreditar em mim, e pela amizade que me brindou.

Aos compatriotas que ajudaram-me a sentir como em casa, em especial ao Ernesto, pelo apoio e pelos muitos conselhos concedidos.

Ao pessoal administrativo do ICMC, em especial à Beth, Laura, Ana Paula e Marília, pela atenção que sempre demonstraram.

Aos colegas e novos amigos que fiz em São Carlos, pela amizade e apoio que me deram. Obrigado de verdade!

Ao CNPq, pelo apoio financeiro.

Juan Carlos

Resumo

Nas últimas décadas, estudos em neurobiologia evidenciam a existência de comportamentos caóticos no cérebro humano e animal, tanto a nível microscópico (neurônio) quanto a nível macroscópico (atividade global de cérebro). Essas evidências motivam a exploração de sistemas caóticos em redes neurais artificiais. Diante deste contexto, o presente trabalho tem como objetivo estudar as redes neurais caóticas existentes e desenvolver novas redes neurais caóticas para reconhecimento de padrões multi-valorados. Dois novos modelos são propostos, cujos funcionamentos são divididos em duas fases: fase de armazenamento e fase de reconhecimento. Na primeira fase, um conjunto de padrões é armazenado em pontos fixos pelo algoritmo de aprendizado de matriz pseudo-inversa. Na segunda fase, a dinâmica periódica e caótica que existem em mapas caóticos são utilizadas. Em ambos modelos propostos, a órbita periódica representa um padrão recuperado, enquanto a órbita caótica oferece um mecanismo de busca eficiente. Uma das vantagens dos modelos propostos em relação às redes neurais caóticas existentes é que os primeiros não só podem reconhecer padrões binários, mas também podem reconhecer padrões multi-valorados, o que é uma característica importante em aplicações práticas.

Abstract

In the last decades, neurobiological researches yield evidences of chaotic behavior in animal and human brains, both in microscopic (neuron) and macroscopic (global brain activity) levels. Such evidences motivate the exploration of chaotic systems in artificial neural networks. In this context, the objectives of the present work are to study the existing chaotic neural networks and to develop new chaotic neural networks for multi-value pattern recognition. Two new models are proposed in this work. The working mechanism of both of them is divided in two phases: storing and recognition. In the former phase, a set of patterns are stored in fixed points by the pseudo-inverse matrix learning algorithm. In the latter one, the periodic and chaotic dynamic existing in chaotic maps are employed with the periodic orbit representing a retrieved pattern and the chaotic orbit providing an efficient searching mechanism. One advantage of the proposed models over the existing chaotic neural networks lies in that the new models recognize not only binary, but also multi-value patterns, which is an important feature in practical applications.

Índice

1	Introdução	1
1.1	Objetivos	3
1.2	Estrutura da Monografia	4
2	Sistemas Dinâmicos	5
2.1	Considerações Iniciais	5
2.2	Sistema Dinâmico	5
2.3	Ponto Fixo	7
2.3.1	Busca de Pontos Fixos	8
2.3.2	Estabilidade de Ponto Fixo	9
2.4	Ciclo Limite	12
2.4.1	Estabilidade dos Ciclos Limites	14
2.5	Quase-Periodicidade	15
2.6	Caos	16
2.7	Os expoentes característicos de Lyapunov	19
2.8	Crise	25
2.9	Considerações Finais	27
3	Redes Neurais Modeladas por Sistemas Dinâmicos	29
3.1	Considerações Iniciais	29
3.2	Redes Neurais Artificiais (RNA)	29
3.2.1	Estrutura do Neurônio Artificial	30
3.2.2	Arquitetura da Rede	31
3.2.3	Algoritmos de Aprendizado de uma RNA	33
3.3	O Modelo de Hopfield	34
3.3.1	Modelo de Hopfield Discreto	35
3.3.2	Modelo de Hopfield Contínuo	39
3.3.3	Uma Aplicação de Reconhecimento de Faces Usando o Modelo de Hopfield	41
3.3.4	Limitações do Modelo Hopfield	45
3.4	Redes Neurais Celulares	46
3.4.1	Topologia de uma Rede Neural Celular	46
3.4.2	Rede Neural Celular de Tempo Contínuo	47

3.4.3	Rede Neural Celular em Tempo Discreto	49
3.4.4	Aplicações das Redes Neurais Celulares	50
3.5	Considerações Finais	59
4	Memórias Associativas Caóticas	61
4.1	Considerações Iniciais	61
4.2	Rede Neural Caótica de Adachi e Aihara	61
4.2.1	O Neurônio Caótico	62
4.2.2	A Rede Caótica	64
4.2.3	Experimentos por Computador	65
4.2.4	Observações	71
4.3	Cluster Dinâmica Para Reconhecimento de Padrões	71
4.3.1	Mapas Globalmente Acoplados	71
4.3.2	Aplicações para Reconhecimento de Padrões	76
4.3.3	Observações	79
4.4	Rede de Mapa Senoidal Parametricamente Acoplada	79
4.4.1	O Mapa Senoidal	79
4.4.2	Modelo de Memória Associativa Utilizando Mapa Senoidal	80
4.4.3	Observações	81
4.5	Considerações Finais	83
5	Reconhecimento de Padrão Multi-valorados	85
5.1	Considerações Iniciais	85
5.2	Descrição do Modelo I	86
5.2.1	A Etapa de Aprendizado	86
5.2.2	A Etapa de Reconhecimento	86
5.2.3	Simulações por Computador	89
5.3	Descrição do Modelo II	91
5.3.1	A Etapa de Aprendizado	91
5.3.2	A Etapa de Reconhecimento	92
5.3.3	Simulações por Computador	94
5.4	Características dos Modelos	101
5.5	Considerações Finais	102
6	Conclusões e Trabalhos Futuros	103
6.1	Considerações iniciais	103
6.2	Conclusões	103
6.3	Trabalhos futuros	104
	Referências Bibliográficas	105

Lista de Figuras

2.1	Parábola da função do mapa logístico com $A = 2.5$	7
2.2	Exemplo de ponto fixo para o mapa logístico. (a) Diagrama <i>Cobweb</i> do ponto fixo, para $A = 2.5$; (b) Série temporal para $A = 2.5$	8
2.3	(a) Ponto fixo estável; (b) ponto fixo instável o repulsor; (c) Ponto de Sela. . . .	11
2.4	(a) Espiral estável; (b) Espiral instável; (c) Centro.	12
2.5	Exemplo de solução periódica para quando $A = 3.4$. (a) Diagrama <i>Cobweb</i> ; (b) Série temporal.	13
2.6	Exemplo de solução periódica com $A = 3.5$ apresentando 4 períodos. (a) Diagrama <i>cobweb</i> ; (b) Série temporal.	14
2.7	Exemplo de solução quase-periódica para $A = 3.82$. (a) Diagrama <i>cobweb</i> ; (b) Série temporal.	16
2.8	Dependência sensível às condições iniciais.	17
2.9	Exemplo de comportamento caótico para $A = 3.9$. (a) Diagrama <i>cobweb</i> ; (b) Série temporal.	18
2.10	(a) Diagrama de Bifurcação do mapa logístico; (b) O expoente de Liapunov do mapa logístico.	19
2.11	Evolução de um elemento de volume esférico de raio $\varepsilon_0(x_0)$ em torno de um ponto inicial x_0 . Depois de um tempo t , a esfera torna-se uma elipse com eixos principais $\varepsilon_1(t)$ e $\varepsilon_2(t)$ para o caso bidimensional.	20
2.12	Sinais dos expoentes de Lyapunov para seus atratores em $\mathbb{R}^3$. (a) Ponto fixo; (b) Ciclo limite; (c) Toro; (d) Atrator caótico.	22
2.13	Crise exterior, para o mapa quadrático quando $A = 2$	25
2.14	Crise interior pode-se ver que o atrator caótico incrementa de tamanho.	26
2.15	(a) Atrator caótico, quando $\eta = 0.97$ ou seja antes do ponto de crise; (b) Atrator caótico quando $\eta = 1.2$ ou seja depois do ponto de crise, acontecendo a expansão do atrator.	26
3.1	Modelo de um Neurônio Artificial	30
3.2	Funções de ativação. (a) Função degrau; (b) Função tangente hiperbólica	31
3.3	Modelo de uma Rede Neural Artificial	32
3.4	Redes <i>FeedForward</i>	32
3.5	Redes Recorrentes	32
3.6	Idéia básica do modelos de Hopfield	34

3.7	Topologia da Rede Hopfield	36
3.8	Padrões utilizados para o treinamento da rede Hopfield.	38
3.9	A imagem correspondente às saídas da rede em cada iteração até reconhecer um padrão.	38
3.10	Modelo Contínuo do Neurônio	39
3.11	Função de ativação dos neurônios para a rede de Hopfield-caso contínuo, com diferentes valores de λ	40
3.12	Processo de redução da dimensionalidade da imagem de uma face; (a) Imagem original de 256×256 ; (b) Primeiro nível de redução; (c) Teceiro nível de redução.	41
3.13	(a) Imagem sem gestos; (b) Imagem depois de aplicar <i>wavelets</i> na imagem anterior; (c) Imagem com sorriso; (d) Imagem depois de aplicar <i>wavelets</i> na imagem anterior.	42
3.14	Processo de binarização aplicando um <i>threshold</i> $\theta = 70$ para uma imagem de uma face.	42
3.15	Resultado do Processo de normalização das transformações geométricas de duas imagens com tamanho, posição e rotações diferentes.	44
3.16	Resultados obtidos no processo de reconhecimento de faces	45
3.17	Exemplos de malhas de uma rede neural celular.	47
3.18	Exemplos vizinhança para $r = 1, 2, 3$, em uma matriz de 9×9	47
3.19	Função de saída da rede neural celular de tempo contínuo.	48
3.20	Operação de convolução.	49
3.21	Função de saída da rede neural celular em tempo discreto.	50
3.22	Imagens para a remoção do ruído.	51
3.23	Resultado das variáveis x e y depois da primeira iteração.	51
3.24	Resultado das variáveis x e y para a iteração 2 e 3.	52
3.25	Comportamento do estado dos neurônios depois de 10 iterações.	52
3.26	Remoção de ruídos, (a) Imagem com ruído, tamanho de 235×235 ; (b) Imagem depois de 2 iterações; (c) Imagem depois de 4 iterações; (d) Imagem depois de 8 iterações; (e) Imagem depois de 20 iterações.	53
3.27	Imagem usada para a extração do contorno.	54
3.28	Valor das variáveis x e y depois da primeira iteração.	54
3.29	Valor das variáveis x depois da 2 e 3 iteração.	54
3.30	Comportamento do estado dos neurônios depois de 10 iterações.	55
3.31	Extrair o contorno da imagem, (a) Imagem depois de aplicar o operador Roberts; (b) Imagem depois de aplicar o operador Sobel; (c) Imagem depois de aplicar o operador Prewitt; (d) Imagem depois de aplicar o operador Laplaciano.	56
3.32	(a) Imagem original; (b) Filtro passa baixa; (c) Filtro passa alta.	57
3.33	Padrões armazenados.	59
3.34	Processo de reconhecimento da rede neural celular.	59
4.1	Amostras de soluções da Eqn. 2, com valores de $k = 0.50$, $\epsilon = 0.015$, $a = 0.50$ e $\alpha = 1.00$	63
4.2	Bifurcação da Eqn. 2 variando o parâmetro a , com $k = 0.51$, $\epsilon = 0.015$ e $\alpha = 1.00$	63
4.3	Padrões utilizados para a simulação do modelo apresentado.	65
4.4	Seqüência do espaço temporal, quando $\alpha = k_r = k_f = a = 0.00$	66

4.5	Uma amostra da sequência do espaço temporal, com $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$	66
4.6	Uma amostra da sequência espaço-temporal, com $k_f = 0.20$, $\alpha = 10.00$ e $k_r = 0.90$ e $a = 2.00$	66
4.7	Série temporal de alguns elementos quando $k_f = k_r = \alpha = a = 0.00$	67
4.8	Distância de Hamming quando $k_f = k_r = \alpha = a = 0.00$	68
4.9	Quase energia quando $k_f = k_r = \alpha = a = 0.00$	68
4.10	Série temporal de alguns elementos quando $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$	69
4.11	Distância de Hamming quando $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$	69
4.12	Quase energia quando $k_f = 0.15$, $\alpha = 8.00$, $k_r = 0.50$ e $a = 1.75$	70
4.13	Série temporal de alguns elementos quando $k_f = 0.20$, $\alpha = 10.00$ e $k_r = 0.90$ e $a = 2.00$	70
4.14	Distância de Hamming quando $k_f = 0.20$, $\alpha = 10.00$, $k_r = 0.90$ e $a = 2.00$	71
4.15	Quase energia quando $k_f = 0.20$, $\alpha = 10.00$, $k_r = 0.90$ e $a = 2.00$	72
4.16	(a) Mapa logístico com $\alpha = 2.00$ com só um ponto extremo; (b) Mapa cúbico com $\alpha = 4.00$, quando $\alpha > 2.0$ o mapa apresenta dois extremos.	73
4.17	(a) Bifurcação do mapa quadrático com $1.00 < \alpha < 2.00$; (b) Bifurcação do Mapa cúbico com $2.50 < \alpha < 4.00$	73
4.18	Diagrama de fase de mapa globalmente acoplado com variações dos parâmetros α de 3.00 a 4.00 e ϵ que varia de 0.00 a 0.40 (S. Ishii, 1996).	74
4.19	Exemplo da fase coerente com $\alpha = 2.60$ e $\epsilon = 0.60$	74
4.20	Exemplo da formação de dois clusters quando a rede se encontra na fase ordenada (2), com $\alpha = 3.20$ e $\epsilon = 0.20$	75
4.21	Exemplo com quatro clusters quando $\alpha = 3.20$, $\epsilon = 0.02$ e $N = 100$ e qual corresponde ao número de elementos utilizados.	75
4.22	Exemplo com valores pares de tempo $t = 2, 4, 6, \dots$ e $\alpha = 3.40$, $\epsilon = 0.10$ e $N = 100$	76
4.23	Exemplo da fase de turbulência do rede quando $\alpha = 4.00$ e $\epsilon = 0.01$	76
4.24	Exemplo de simulação controlando o parâmetro α	78
4.25	Valores da energia para cada instante de tempo $t = \{1, 2, \dots\}$ do aplicação de reconhecimento de padrão	78
4.26	Valores do parâmetro α da aplicação de reconhecimento de padrão para cada instante de tempo $t = \{1, 2, \dots\}$	79
4.27	Dois diagramas de bifurcação correspondentes a diferentes estados iniciais (a) Quando o estado inicial é um valor positivo $x(0) = 0.1$; (b) Quando o valor do estado inicial é negativo $x(0) = -0.1$	80
4.28	Exemplo de reconhecimento de padrão quando o valor de $\eta = 0.97$ $\alpha = 1.00$ e $\epsilon = 0.14$	81
4.29	Valores da energia obtido pelo reconhecimento do padrão anterior, para cada instante de tempo $t = \{1, 2, \dots\}$	82
4.30	Valores do parâmetro η obtido para cada instante de tempo $t = \{1, 2, \dots\}$ no reconhecimento do padrão anterior.	82
5.1	(a) Comportamento da função Gaussiana variando valor de α , $\theta = 1$; (b) Comportamento da função Gaussiana variando valor de θ , $\alpha = 0.5$	87

5.2	Escolha de n_{min} determinando os pontos pp_1 , pp_2 , pp_3 e pp_4 do mapa senoidal, a serem armazenados, assim como a determinação do ponto n_c , que corresponde a uma órbita caótica depois do ponto de crise.	88
5.3	Padrões armazenados no modelo, cada um com quatro níveis de cinza.	89
5.4	Processo de reconhecimento de um padrão corrompido.	89
5.5	Evolução do parâmetro n_i	90
5.6	Evolução do estado de ativação.	90
5.7	Distância de Hamming entre o estado da rede e cada padrão armazenado. . . .	91
5.8	Diagrama de bifurcação e escolha dos pontos pp_1 , pp_2 , pp_3 e pp_4 do mapa logístico.	93
5.9	Diagrama <i>cobweb</i> e série temporal do mapa logístico, quando o valor de $A = 4$. . .	93
5.10	Padrões binários armazenados na rede	94
5.11	Processo de reconhecimento de um padrão.	95
5.12	Evolução de u_i para $i = 1, 2, \dots, N$	95
5.13	Evolução de u_i para $i = 1, 2, \dots, N$	96
5.14	Evolução de u_i para $i = 1, 2, \dots, N$	96
5.15	Processo de reconhecimento da borboleta.	97
5.16	Evolução de u_i para $i = 1, 2, \dots, N$	97
5.17	Série temporal dos neurônios do sistema.	98
5.18	Distância de Hamming entre o estado da rede e cada padrão armazenado. . . .	98
5.19	Seqüências de DNA utilizadas na etapa de armazenamento.	99
5.20	Processo de reconhecimento de uma seqüência de DNA.	99
5.21	Evolução de u_i para $i = 1, 2, \dots, N$	100
5.22	Série temporal dos neurônios do sistema.	100
5.23	Distância de Hamming entre o estado da rede e cada padrão armazenado. . . .	101

Introdução

No início da década de 1980, aumentou o interesse na arquitetura de computação e processamento de informação tipo não Von Neumann. Este fato foi motivado pelo processamento paralelo e a flexibilidade, tipicamente observada em sistemas biológicos, como por exemplo o cérebro humano. O cérebro é um sistema de processamento de informação altamente complexo, não-linear e paralelo, que possui capacidade de organização dos neurônios para realizar algumas computações, tais como, percepção e reconhecimento de padrão, muitas vezes mais rápido do que o computador mais poderoso existente na atualidade. A pesquisa para desenvolver arquiteturas e modelos de computação inspirados no cérebro é chamada Redes Neurais Artificiais (RNA) (Haykin, 1994).

Nos últimos 20 anos, estudos em neurobiologia evidenciam a existência de comportamentos caóticos tanto em nível microscópico (neurônio) como em nível macroscópico (atividade global de cérebro). Em nível microscópico, oscilações periódicas, quase-periódicas e caóticas nas atividades do potencial de membrana do axônio da lula gigante foram identificadas pelo grupo de pesquisa de Aihara (K. Aihara, 1986). Em nível macroscópico, Babloyantz et al. registraram o eletroencefalograma (EEG) do cérebro humano durante o sono (A. Babloyantz, 1995). Análises sobre esses registros mostram a presença de um atrator caótico durante o sono profundo de uma pessoa perturbado por baixo ruído. O grupo de pesquisa de Freeman descobriu o comportamento caótico em registros de EEG de bulbos olfativos dos coelhos (W. Freeman, 1987). Os padrões de atividades espaciais indicados pelas potências EEG diferem para diferen-

tes odores. Isso significa que o padrão parcial de atividade do receptor pode causar um padrão especial de atividades dos neurônios no bulbo, e conseqüentemente, transmitir as informações de odor-específico para o córtex olfativo. Atividades de comportamento periódico (ciclo limite) ocorrem para um odor específico percebido. Atividades caóticas ocorrem quando o odor é desconhecido (novo odor).

Por outro lado, as redes neurais artificiais recorrentes usualmente são sistemas dinâmicos não-lineares de alta dimensão. Portanto, é possível obter-se vários comportamentos ajustando-se seus parâmetros: convergência na direção de pontos de equilíbrio, de soluções periódicas e de atratores caóticos (M. Adachi, 1997) (P. Das, 1995) (H. Sompolsky, 1988) (M. Thomas, 1996).

Essas evidências motivam muitos pesquisadores a explorar sistemas dinâmicos caóticos em redes neurais artificiais. A pesquisa para descobrir regras de funcionamento e o papel do caos em redes neurais é chamada Redes Neurais Caóticas (RNC). Os modelos de redes neurais caóticas são importantes não apenas como sistemas não-lineares com vários graus de liberdade, mas também do ponto de vista de processamento de informação. Atualmente, as RNCs desenvolvidas são usadas principalmente como memória associativa para reconhecimento de padrões.

Do ponto de vista de aplicações reais, uma memória associativa deve possuir pelo menos as seguintes características (P. Thiran, 1994): 1) Capacidade razoável de armazenamento: onde um número de padrões pode ser armazenado no sistema. 2) Evitar estados espúrios: os parâmetros do sistema podem ser ajustados de forma que só os estados correspondentes aos padrões armazenados sejam os únicos estados estáveis. 3) Recuperação do padrão: ante a apresentação de algum dos padrões armazenados a saída do sistema deverá sempre convergir para ele mesmo. 4) Capacidade de correção de erro: ante a apresentação de um padrão parecido com algum dos padrões armazenados (uma versão incompleta ou com ruído), o sistema sempre dará como saída o padrão mais parecido possível.

Considerando uma RNC como memória associativa, as informações podem ser representadas por um estado estável da rede. Pesquisas desenvolvidas focalizam as relações existentes entre a estrutura do atrator caótico e a transição caótica da rede com o armazenamento de informação e o processo de aprendizado. É fácil perceber que as RNCs são extensões normais dos modelos neurodinâmicos tradicionais, tal como modelo de Hopfield. Como sabemos, o processo de associação (armazenamento e retorno de padrões) do modelo de Hopfield corresponde à minimização de função de Lyapunov da rede. Nesse sentido, o modelo emprega dinâmica de equilíbrio. Por outro lado, em RNCs, a dinâmica caótica leva o sistema a estados não-equilibrados. Assim, o espaço de fase das RNCs pode geralmente ser dividido em duas partes: fase caótica e fase não caótica. Na fase não caótica, a rede comporta-se como uma memória associativa convencional. Mas em fase caótica, aparecem órbitas itinerárias, chamada itinerária

caótica, que visitam quase todos padrões armazenados (M. Adachi, 1997) (Hayashi, 1997). Isto é caracterizado por uma transição itinerária do estado dinâmico de neurônios entre atratores de relativamente baixa dimensão e *manifolds* instáveis de baixa dimensionalidade, distribuídos no espaço de fase de alta dimensão. Itinerária caótica é um mecanismo com alta habilidade de busca de informações. Naturalmente, as RNCs podem ser obtidas estendendo-se o modelo de Hopfield por um dos seguintes métodos ou uma combinação entre eles: 1) utilizando neurônios complexos (M. Adachi, 1997) (Hayashi, 1997); 2) variando topologia de conexões sinápticas dinamicamente (S. Nara, 1997); 3) combinando duas ou mais redes (Tsuda, 1994); 4) conexões assimétricas (Parisi, 1997); 5) combinando fases de aprendizado e de revogação (Tsuda, 1994).

Ao invés de se obter RNCs estendendo-se o modelo de Hopfield, recentemente vem-se incrementando o interesse em desenvolver novas redes neurais caóticas, utilizando diretamente as propriedades dinâmicas do caos. Ishii et. al. (S. Ishii, 1993) (S. Ishii, 1996) desenvolveram um modelo de reconhecimento de padrões utilizando a dinâmica de cluster caótica gerada por sistemas de mapas globalmente acoplados (Kaneko, 1990). O modelo é dividido em dois estados: estado de cluster e estado de turbulência. O estado de cluster representa um padrão memorizado, enquanto o estado de turbulência permite que os neurônios da rede possam mudar seus estados para minimizar funções parciais da energia. Lee e Farhat (G. Lee, 2001) desenvolveram uma outra rede neural caótica usando mapas senoidal caóticos e fenômeno de crise. O modelo é dividido em estado de pré-crise e pós-crise. O estado de pré-crise representa um padrão recuperado, enquanto o estado de pós-crise permitirá aos elementos mudar seu sinal (positivo ou negativo) e assim minimizar a função parcial de energia.

Mas, no modelo de Hopfield e nas RNCs desenvolvidas, somente padrões binários podem ser processados. Isso limita muito os modelos em aplicações reais. Por exemplo, uma sequência de DNA é naturalmente representada por um padrão de 4 valores diferentes; uma imagem de níveis de cinza é geralmente representada por um padrão numérico de 256 valores diferentes. Neste trabalho, desenvolvemos redes neurais para reconhecimento de padrões multi-valorados, utilizando dinâmica periódica e caótica em um sistema de mapas acoplados.

1.1 Objetivos

O objetivo principal deste trabalho é desenvolver redes neurais caóticas para reconhecimento de padrões multi-valorados. Nestes modelos, podem ser distinguidas duas etapas: Na primeira etapa, os padrões são armazenados em pontos fixos pelo algoritmo de aprendizado da matriz pseudo-inversa. Na segunda etapa, é feito o reconhecimento, onde a dinâmica periódica e caótica que existem em mapas caóticos são usadas. A órbita periódica representa o padrão recupe-

rado, enquanto a órbita caótica representa um estado de busca. O caos oferece um mecanismo eficiente e robusto para buscar estados desejáveis, coberto pelo próprio atrator caótico.

O objetivo principal será atingido, conforme se tinham alcançado objetivos específicos, os quais são:

- Estudo de conceitos básicos de sistemas dinâmicos não-lineares, assim como de redes neurais artificiais;
- Estudo e implementação do modelo de Hopfield e a Rede Neural Celular. Esses dois tipos de redes neurais também são sistemas dinâmicos não-lineares;
- Estudo e implementação das redes neurais caóticas existentes, analisando os distintos comportamentos dinâmicos;
- Desenvolvimento de redes neurais caóticas para reconhecimento de padrões multi-valorados.

1.2 Estrutura da Monografia

Este documento está organizado em 6 capítulos:

Capítulo 2: apresenta-se os conceitos básicos de sistemas dinâmicos. Quatro tipos de comportamentos dinâmicos, tais como: ponto fixo, oscilação periódica, quase-periódica e caos, são revisados.

Capítulo 3: apresentam-se conceitos fundamentais de redes neurais artificiais, resumindo as características principais, as diferentes arquiteturas e os algoritmos e paradigmas de aprendizado. Ainda neste capítulo é revisado, em certa nível de detalhes, duas redes neurais artificiais: o modelo de Hopfield e Rede Neural Celular.

Capítulo 4: apresenta-se uma revisão dos modelos de memórias associativas caóticas representativas, mostrando se as diferentes regras de funcionamento do caos e analisando-se a dinâmica das redes.

Capítulo 5: são propostos dois modelos de redes neurais caóticas para reconhecimento de padrões multi-valorados. Também são apresentados os resultados de simulações por computador.

Finalmente, o Capítulo 6: são apresentadas as conclusões e futuros trabalhos.

Sistemas Dinâmicos

2.1 Considerações Iniciais

Neste capítulo, será apresentada uma revisão de alguns conceitos básicos de sistemas dinâmicos. Mais especificamente, serão descritos as dinâmicas de ponto fixo, oscilação periódica, quase periódica e caos. Também será apresentado o expoente de Lyapunov, que é uma ferramenta importante para quantificar o caos e o conceito de crise.

2.2 Sistema Dinâmico

Um sistema dinâmico pode ser definido como uma fórmula matemática que descreve a evolução do estado de um sistema no decorrer do tempo. O tempo pode ser uma variável contínua ou discreta.

Um sistema dinâmico de tempo contínuo autônomo de primeira ordem é dado por:

$$\begin{aligned}\frac{dx_1}{dt} &= F_1(x_1, x_2, \dots, x_N) \\ \frac{dx_2}{dt} &= F_2(x_1, x_2, \dots, x_N)\end{aligned}$$

$$\frac{dx_N}{dt} = F_N(x_1, x_2, \dots, x_N) \quad (1)$$

t denota o tempo, N é a dimensão do sistema. Esta equação pode ser reescrita em forma vetorial:

$$\begin{aligned} \dot{\mathbf{x}} &= F(\mathbf{x}, \mu) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned} \quad (2)$$

com $\mathbf{x} \in U \subset \mathbb{R}^N$ é o vetor de estados, $\mu \in V \subset \mathbb{R}^P$ é o vetor de parâmetros e U e V são conjuntos abertos em $\mathbb{R}^N$ e $\mathbb{R}^P$, respectivamente. $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$. O ponto sobre $\mathbf{x}$ na Eqn. 2 significa d/dt , $\mathbf{x}_0$ é o estado inicial e a função F descreve como o sistema evolui no tempo.

Para qualquer estado inicial de $\mathbf{x}_0$ no tempo t_0 , o sistema alcançará um futuro estado $\mathbf{x}(t, t_0, \mathbf{x}_0)$ para $t > 0$. O caminho gerado pela mudança contínua de estado no espaço de fase é referenciado como uma *órbita* ou *trajetória*. Se F é uma função contínua diferenciável, existe uma única solução para a Eqn. 2 dada uma condição inicial $\mathbf{x}_0$ (Ott, 1993).

No caso do tempo ser discreto, um sistema dinâmico pode ser especificado pela seguinte equação:

$$\begin{aligned} \mathbf{x}(t+1) &= F(\mathbf{x}(t), \mu) \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \quad (3)$$

com $\mathbf{x} \in \mathbb{R}^N$, $\mu \in \mathbb{R}^P$, $t \in \mathbb{Z}^+$ e a função $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$. Onde $\mathbf{x}$ representa o vetor de estados, μ representa o vetor de parâmetros, t denota o tempo que corresponde a um número inteiro, $\mathbf{x}_0$ é o estado inicial e a função F descreve como o sistema evolui no tempo. Esta equação corresponde a um sistema dinâmico de tempo discreto autônomo, homogêneo.

Os sistemas dinâmicos tanto contínuo quanto discreto podem ser lineares ou não-lineares dependendo da natureza da função F , i.e., se F é uma função linear, temos um sistema dinâmico linear; caso contrário, temos um sistema dinâmico não-linear.

Sistemas dinâmicos também podem ser classificados em autônomos (tempo invariável) ou não-autônomos (tempo variável). Um sistema dinâmico discreto não autônomo pode ser representado pela seguinte equação:

$$\mathbf{x}(t+1) = f(t, \mathbf{x}, \mu) \quad (4)$$

com $f : \mathbb{Z}^+ \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. Enquanto que para o caso autônomo ele é dado por:

$$\mathbf{x}(t+1) = f(\mathbf{x}, \mu) \quad (5)$$

com $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$

Neste trabalho, focalizamos em sistemas dinâmicos discretos, autônomos e de primeiro ordem. Em grande maioria dos casos, trabalhamos com mapas unidimensionais.

Um dos interesses principais no estudo de sistemas dinâmicos é analisar o comportamento assintótico, ou seja, o comportamento de um sistema dinâmico quando $t \rightarrow \infty$. Neste contexto, sistemas dinâmicos podem apresentar vários tipos de *atratores*. Qualitativamente, um *atrator* é uma região ou um ponto no espaço de estados do sistema para onde as trajetórias convergem a partir de um conjunto de condições iniciais. Os atratores mais comuns são: ponto fixo, ciclo limite, atrator quase-periódico e do caos.

A seguir, ilustramos cada tipo desses atratores usando mapa logístico como exemplo. Mapa logístico é um sistema dinâmico discreto, não-linear, unidimensional e autônomo, que é representado pela seguinte equação:

$$x(t+1) = Ax(t)(1 - x(t)) \quad (6)$$

onde t representa tempo discreto, x é a variável de estado e A é o parâmetro de bifurcação. A Fig. 2.1 mostra a função $f(x) = Ax(1 - x)$ que corresponde a uma parábola quando $A = 2.5$.

Figura 2.1: Parábola da função do mapa logístico com $A = 2.5$.

2.3 Ponto Fixo

Um *ponto fixo* também é chamado um *ponto de equilíbrio*. Um ponto fixo é um estado invariante sob a dinâmica do sistema no qual o sistema não perturbado permanece indefinidamente neste estado uma vez que o tenha atingido.

Para um sistema dinâmico contínuo, os pontos fixos são aqueles pontos $\mathbf{x}$ quando:

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t)) = 0 \quad (7)$$

Para um sistema dinâmico discreto, os pontos fixos $\mathbf{x}$ são aqueles que:

$$\mathbf{x}(t+1) = \mathbf{x}(t) \quad (8)$$

A Fig. 2.2 mostra um exemplo em que o mapa logístico converge para um ponto fixo quando $A = 2.5$. Aqui, o ponto fixo corresponde à intersecção da parábola do mapa logístico com a linha diagonal. Neste exemplo, a iteração do sistema é a partir da condição inicial $x(0)$. A Fig. 2.2 (a) é chamada de diagrama *cobweb*, que é um método gráfico para a representação das iterações de um sistema discreto. Neste diagrama a parábola corresponde à função quadrática e à linha diagonal corresponde a $x(t+1) = x(t)$. A Fig. 2.2(b) é uma série temporal. Em ambas as figuras, podemos ver que o mapa logístico converge para um ponto fixo.

Figura 2.2: Exemplo de ponto fixo para o mapa logístico. (a) Diagrama *Cobweb* do ponto fixo, para $A = 2.5$; (b) Série temporal para $A = 2.5$.

2.3.1 Busca de Pontos Fixos

Do gráfico do sistema dinâmico discreto, é fácil encontrar pontos fixos, como se pode apreciar na Fig. 2.2 (a), ponto fixo corresponde à intersecção da parábola com a linha diagonal.

Matematicamente, pontos fixos são encontrados pela resolução da Eqn. 7 ou da Eqn. 8.

Para um sistema linear discreto, por exemplo, $x(t+1) = Ax(t)$, x^* é um ponto fixo se e somente se $x^* = Ax^*$. Uma solução para esta equação é quando $x^* = 0$. Isto significa que a

origem é um ponto fixo para este sistema linear. Quando $A = 1$, podemos perceber que todos os pontos (x) são pontos fixos.

Para um sistema não-linear, encontraremos pontos fixos da mesma forma. Por exemplo, para o mapa logístico é preciso encontrar as raízes da equação quadrática $x^2 + x(1 - A)/A = 0$, obtendo como resultado $x^* = 0$ e $x^* = (A - 1)/A$.

2.3.2 Estabilidade de Ponto Fixo

Pontos fixos podem ser estáveis ou instáveis. A estabilidade de um ponto fixo é definida a seguir.

Definição 2.3.1 Um ponto fixo x^* é dito estável se dado $\epsilon > 0$, existe um $\delta > 0$, tal que $|x - x^*| < \delta$ implica $|f^t(x) - x^*| < \epsilon$ para todos $t > 0$. Se x^* não é estável, então é dito instável.

Definição 2.3.2 O ponto fixo x^* é dito de atração se existe um $\eta > 0$, tal que $|x - x^*| < \eta$ implica $\lim_{t \rightarrow \infty} x(t) = x^*$.

Definição 2.3.3 O ponto fixo x^* é dito assintoticamente estável se ele é estável e de atração.

Qualitativamente, um ponto fixo é estável se a proximidade entre ele e seus vizinhos são mantidas ao longo do tempo; um ponto fixo é assintoticamente estável se uma vizinhança dos pontos converge a ele.

Uma técnica para determinar a estabilidade de um sistema dinâmico não-linear é a linearização. O propósito da linearização é fornecer um método para determinar se o ponto fixo é estável, tanto para o sistema discreto quanto contínuo. Se a função F é linear, i.e., da forma $F(\mathbf{x}) = \mathbf{A}\mathbf{x}$, fica mais fácil determinar a estabilidade do sistema, analisando os autovalores da matriz Jacobiana. A idéia é aproximar F perto do ponto fixo $\mathbf{x}^*$ por uma função linearizada.

No primeiro caso, consideramos um sistema dinâmico contínuo:

$$\dot{\mathbf{x}} = F(\mathbf{x}) \quad (9)$$

com $\mathbf{x} \in \mathbb{R}^N$, temos que:

$$\mathbf{y} = \mathbf{x} - \mathbf{x}^*(t) \quad (10)$$

onde $\mathbf{x}^*$ é o ponto fixo, então:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}^*(t) - \dot{\mathbf{y}} \quad (11)$$

usando a expansão do Taylor em $\mathbf{x}^*$ temos:

$$\dot{\mathbf{x}} = F(\mathbf{x}^*(t)) + DF(\mathbf{x}^*(t))\mathbf{y} + O(|\mathbf{y}^2|) \quad (12)$$

como $\dot{\mathbf{x}}^*(t) = F(\mathbf{x}^*(t))$ temos que:

$$\dot{\mathbf{y}} = DF(\mathbf{x}^*(t))\mathbf{y} + O(|\mathbf{y}^2|) \quad (13)$$

como a diferença $\mathbf{y}$ é pequena, então o estudo da estabilidade do sistema original poderia ser respondida estudando o sistema linear associado:

$$\dot{\mathbf{y}} = DF(\mathbf{x}^*(t))\mathbf{y} \quad (14)$$

então:

$$\dot{\mathbf{y}} = e^{DF(\mathbf{x}^*)t}\mathbf{y}(0) \quad (15)$$

desde que $DF(\mathbf{x}^*(t)) = DF(\mathbf{x}^*)$ então:

$$\dot{\mathbf{y}} = A\mathbf{y}(0) \quad (16)$$

onde A corresponde à matriz Jacobiana dada por:

$$A = DF(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1}(x) & \dots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix}. \quad (17)$$

sendo os autovalores da equação:

$$Ae = \lambda e. \quad (18)$$

$$y(t) = \sum_{i=1}^N A_i e_i \exp(\lambda_i t) \quad (19)$$

onde A_i são determinados da condição inicial $\mathbf{y}(0) = \sum_{i=1}^N A_i e_i$ e $\lambda_i (i = 1, 2, \dots, N)$ são os autovalores de A .

Pode-se observar que:

- Se algum dos valores da parte real dos autovalores é $\lambda_i > 0$ é dito de ser um ponto fixo instável;
- Se todos os valores da parte real dos autovalores são $\lambda_i < 0$, o ponto fixo é estável;

- Se a parte real dos autovalores são $\lambda_i = 0$ significa que as soluções de $\mathbf{x}$ para $t \rightarrow \infty$ não se afastam nem se aproximam do ponto fixo, permanecendo em sua vizinhança.

Uma classificação de pontos fixos em sistemas contínuos bidimensionais pode ser apresentada pela seguinte forma.

- Se os autovalores são números reais negativos é dito ser um ponto estável. Esse tipo de estabilidade pode ser visto na Fig. 2.3 (a);
- Se os autovalores são números reais positivos é dito ser um ponto instável ou repulsor. A Fig. 2.3 (b) mostra este tipo de estabilidade;
- Se os autovalores são reais e têm sinais diferentes, temos que existe um estável *manifold* e outro instável *manifold* e é chamado de ponto fixo instável o ponto sela. O espaço de estados do ponto **sela** exibe propriedades de atração e repulsão, dependendo da direção de aproximação dos estados vizinhos. O ponto **sela** só pode ocorrer em sistemas dinâmicos cujo espaço de estados apresenta mais de uma dimensão (Kolen, 1994). Na Fig. 2.3 (c) pode ser visto uma representação da estabilidade do ponto sela;

Figura 2.3: (a) Ponto fixo estável; (b) ponto fixo instável o repulsor; (c) Ponto de Sela.

- Se os autovalores são conjugados complexos e a parte real é negativa então trata-se de um espiral estável como pode ser visto na Fig. 2.4 (a);
- Se os autovalores são conjugados complexos e a parte real é positiva, é um espiral instável como pode ser visto na Fig. 2.4 (b);
- Se os autovalores são conjugados complexos e a parte real é 0 então é um centro. A Fig. 2.4 (c) mostra esse tipo de estabilidade.

Para o sistema dinâmico de tempo discreto o procedimento é parecido, onde temos que:

$$\mathbf{y}(t) = \mathbf{x}(t) - \mathbf{x}^*, \quad (20)$$

Figura 2.4: (a) Espiral estável; (b) Espiral instável; (c) Centro.

então

$$\mathbf{x}(t) = \mathbf{x}^* + \mathbf{y}(t), \quad (21)$$

utilizando a expansão de Taylor temos:

$$\mathbf{y}(t+1) = DF(\mathbf{x}^*)\mathbf{y}(t) + O(y(t)^2), \quad (22)$$

como a diferença $\mathbf{y}$ é pequena, e $DF(\mathbf{x}^*) = \mathbf{A}$ então a estabilidade do sistema pode ser dado pelo sistema linear:

$$\mathbf{y}(t+1) = \mathbf{A}\mathbf{y}(t) \quad (23)$$

sendo que:

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e} \quad (24)$$

finalmente temos:

$$y(t+1) = \sum_{k=1}^t A_k e_k \lambda_k^N \quad (25)$$

Pode-se observar que:

- Se algum dos $|\lambda_i| > 1$, o ponto fixo é instável, onde os pontos saem da vizinhança do ponto fixo, e se distanciam do ponto fixo.
- Se todos os $|\lambda_i| < 1$, o ponto fixo é estável, de modo que os pontos próximos se aproximam ao ponto fixo depois de cada iteração.

2.4 Ciclo Limite

Um ciclo limite é um tipo de atrator, no qual o comportamento do sistema se repete ao longo do tempo. O número de iterações entre repetições é chamado o período do ciclo. Os pontos de um ciclo limite, são chamados de pontos periódicos. Um ciclo de período n ocorre se a seguinte condição é satisfeita:

$$x(t+n) = x(t) \text{ mas } x(t+j) \neq x(t) \text{ para } j = 1, 2, \dots, n-1 \quad (26)$$

Por exemplo, a condição do ciclo com período 2 é:

$$x(t+2) = x(t) \text{ e } x(t+1) \neq x(t) \quad (27)$$

substituindo em $x(t+1) = f(x(t))$, pode-se escrever o valor de $x(t+2)$ como:

$$x(t+2) = f(x(t+1)) = f(f(x(t))) \quad (28)$$

como é um ciclo de período 2, então $x(t) = f(f(x(t)))$. Assim da Eqn. 28, pode-se dizer que existe uma analogia entre a busca de pontos fixos e os ciclos. Se o sistema $x(t+1) = f(x(t))$ tem um ciclo de período 2, então a função $f(f(x(t)))$ tem pelo menos 2 pontos fixos. Assim podemos procurar ciclos de período 2, resolvendo a equação $x(t) = f(f(x(t)))$.

Ciclos mais longos podem ser encontrados da mesma forma. Um ciclo de período n é encontrado resolvendo a seguinte equação:

$$x(t) = \underbrace{f(f(\dots f(x(t))))}_{n \text{ tempos}} \quad (29)$$

Considerando a equação do mapa logístico para $A = 3.4$, a equação fica:

$$x(t+1) = f(x(t)) = 3.4(1-x(t))x(t) \quad (30)$$

onde se apresenta um ciclo com período 2. A Fig. 2.5 corresponde a um ciclo periódico do mapa

Figura 2.5: Exemplo de solução periódica para quando $A = 3.4$. (a) Diagrama *Cobweb*; (b) Série temporal.

logístico da Eqn 6. A Fig. 2.5 (a) apresenta o diagrama *cobweb* e a Fig. 2.5(b) corresponde à série temporal. As imagens correspondem a uma solução com período 2 quando $A = 3.4$. A Fig. 2.6 também representa uma solução periódica mas para $A = 3.5$, onde podem ser vistos um ciclo de período 4.

Figura 2.6: Exemplo de solução periódica com $A = 3.5$ apresentando 4 períodos. (a) Diagrama *cobweb*; (b) Série temporal.

2.4.1 Estabilidade dos Ciclos Limites

Assim como um ponto fixo pode ser estável ou instável, um ciclo pode também ser estável ou instável. Um ciclo estável atrai soluções próximas, mas um ciclo instável rejeita soluções próximas.

Os passos mencionados anteriormente para determinar a estabilidade de um ponto fixo podem ser usados também para os ciclos. Podemos dizer que x^* denota uma solução para a equação $x(t) = f(f(x(t)))$. Seguindo os passos da seção anterior, a estabilidade para o ciclo limite de $x(t+2) = f(f(x(t)))$ depende de:

$$\left. \frac{df(f(x(t)))}{dx(t)} \right|_{x^*} \quad (31)$$

usando a regra da cadeia temos que:

$$\left. \frac{df(f(x(t)))}{dx(t)} \right|_{x^*} = \left. \frac{df}{dx(t)} \right|_{f(x^*)} \left. \frac{df}{dx(t)} \right|_{x^*} \quad (32)$$

Assim a estabilidade de um ciclo limite de período 2 depende da tangente da função $f(x(t))$, tanto para o ponto x^* como para $f(x^*)$. O mesmo procedimento pode ser aplicado para determinar a estabilidade de ciclos de períodos maiores.

Se existe um ciclo estável de período n ele deveria ter pelo menos n pontos periódicos associados ao ciclo. Se $\{p_1, p_2, \dots, p_n\}$ denota um ciclo de período n , então pela regra da cadeia temos que:

$$\begin{aligned} (f^n)'(p_n) &= (f(f^{n-1}))'(p_1) \\ &= f'(f^{n-1}(p_1))(f^{n-1})'(p_1) \\ &= f'(f^{n-1}(p_1))f'(f^{n-2}(p_1))\dots f'(p_1) \\ &= f'(p_n)f'(p_{n-1})\dots f'(p_1) \end{aligned} \tag{33}$$

então tem-se que:

- Se $|f'(p_n)\dots f'(p_1)| > 1$, o ciclo é instável;
- Se $|f'(p_n)\dots f'(p_1)| < 1$, o ciclo é estável.

2.5 Quase-Periodicidade

Quase-Periodicidade é um tipo de dinâmica que apresenta uma trajetória, na qual nenhum estado é repetido até $t \rightarrow \infty$.

Para ver como este tipo de dinâmica aparece, pode-se considerar a seguinte equação:

$$x(t+1) = f(x(t)) = \text{mod1}(x(t) + b) \tag{34}$$

onde mod1 corresponde ao operador módulo um, que toma a parte fracionária de um número (i.e., $\text{mod1}(3.67) = 0.67$). Para iterar esta equação calculamos $x(t) + b$ e tomamos a parte fracionária. Por exemplo, se $x(t) = 0.9$ e o parâmetro $b = 0.3$, então $x(t) + b = 1.2$ e $\text{mod1}(x(t) + b) = 0.2$. Agora considere a segunda iteração a qual pode ser:

$$\begin{aligned} x(t+2) &= \text{mod1}(x(t+1) + b) \\ &= \text{mod1}((x(t) + b) + b) \\ &= \text{mod1}(x(t) + 2b) \end{aligned}$$

de forma similar pode-se mostrar que:

$$x(t+n) = f^n(x(t)) = \text{mod}1(x(t) + nb) \quad (35)$$

consequentemente. Se $\text{mod}1(nb) = 0$ então todos os valores estão sobre o ciclo de período n , senão, nenhum valor estará em nenhum ciclo.

O conceito de número irracional pode nos ajudar a entender melhor a quase-periodicidade. Primeiro um número racional pode ser representado como a proporção entre dois números inteiros (p/q). Mas um número irracional não pode ser expresso da mesma forma. Se π é um número irracional, $1/\pi$ também será um número irracional. Pode-se observar então que $\text{mod}1(n/\pi)$ nunca será 0 para qualquer valor de n . Portanto, nunca apresentará um ciclo periódico.

A Fig. 2.7 mostra um comportamento de quase-periodicidade do mapa logístico da Eqn 6. A Fig. 2.7(a) é o diagrama *cobweb* e a Fig. 2.7(b) a série temporal. As figuras correspondem a uma solução quase-periódica quando $A = 3.82$.

Figura 2.7: Exemplo de solução quase-periódica para $A = 3.82$. (a) Diagrama *cobweb*; (b) Série temporal.

2.6 Caos

O caos é um fenômeno comum em muitos sistemas dinâmicos. Um aspecto importante foi o fato de que Henri Poincaré (Kolen, 1994) analisou o comportamento de um conjunto de órbitas originadas a partir de condições iniciais distintas, ao invés de tratar órbitas individuais. Com isso, ele foi capaz de mostrar que órbitas com propriedades dinâmicas desconhecidas (hoje denominadas órbitas caóticas) podiam ser geradas.

Qualitativamente, O caos é definido por ser não periódico, dinamicamente limitado em um sistema determinista, com dependência sensível sobre as condições iniciais (Ott, 1993). Cada termo tem um significado específico, os quais são descritos a seguir:

- **Sensibilidade dependente das condições iniciais**, significa que dois pontos que são bem próximos terão trajetórias diferentes com o decorrer do tempo (divergem). Isto é um aspecto essencial do caos. Isso significa que se pode prever o que acontecerá em curto tempo mas para longos períodos de tempo a predição é impossível desde que num sistema real não se tem certeza do valor exato da condição inicial $x(0)$. A Figura 2.8 mostra um exemplo de tal comportamento. Nela, pode-se ver duas trajetórias caóticas que possuem condições iniciais bem próximas, mas que divergem rapidamente uma da outra no decorrer do tempo. Este exemplo foi gerado usando o mapa logístico no qual quando o valor do parâmetro $A = 4$, a trajetória representada pela linha contínua possui condição inicial $x(0) = 0.1$. A trajetória representada pela linha pontilhada possui condição inicial $x(0) = 0.10001$. As duas trajetórias se separam após 15 iterações;

Figura 2.8: Dependência sensível às condições iniciais.

- **Aperiódico**, significa que o estado atual não é repetido mais de uma vez. A Fig. 2.9 mostra este tipo de comportamento, mas na prática se utilizarmos um computador de precisão finita, eventualmente o comportamento pode ser repetido. Ainda que o sistema possa apresentar comportamento periódico por ciclos extensos ou comportamento não periódico, este é só uma evidência parcial da existência do caos;
- **Limitado**, significa que em sucessivas iterações o estado fica numa faixa finita (i.e., ele não tende ao infinito). Por exemplo, se utilizarmos o mapa logístico com valores iniciais entre $0 < x(0) < 1$, subseqüentes iterações também estarão entre 0 e 1;

- **Determinístico**, significa que existe uma regra definida, não aleatória, que governa a dinâmica do sistema. Para um sistema unidimensional, determinístico significa que para cada valor $x(t)$ existe só um valor $x(t + 1)$.

Um exemplo do comportamento caótico do mapa logístico é apresentado na Fig. 2.9 quando $A = 3.9$.

Figura 2.9: Exemplo de comportamento caótico para $A = 3.9$. (a) Diagrama *cobweb*; (b) Série temporal.

Dinâmicas lineares podem apenas expandir, comprimir ou rotacionar o espaço de estados, de forma que apenas pontos fixos e ciclos periódicos são possíveis. Pontos fixos estáveis resultam da aplicação repetida de operadores de compressão no limite do tempo. Estas operações reduzem subconjuntos do espaço de estados a um único ponto. Ciclos limites surgem a partir da rotação do espaço de estados. Por outro lado, quando a dinâmica não-linear repetidamente expande, dobra e (possivelmente) comprime o espaço de estados, emerge o comportamento caótico.

O objeto geométrico no espaço de estados para o qual uma trajetória caótica é atraída é denominado atrator estranho sempre que sua dimensão for fracionária (geometria fractal). Da mesma forma que atratores periódicos podem ser diferenciados por seus períodos, atratores estranhos podem ser diferenciados por suas dimensões. Pontos fixos de tempo discreto são atratores de dimensão zero, pois são constituídos apenas por um número finito de pontos. Ciclos limites de tempo contínuo são atratores de dimensão inteira. Por exemplo, quando o ciclo limite é uma curva que liga suas extremidades, sua dimensão é um. Já atratores caóticos podem apresentar dimensão fracionária.

Uma forma de representar de forma gráfica as bifurcações de uma equação não-linear é plotar os valores assintóticos da variável, como uma função de um parâmetro variável que é chamado de diagrama de bifurcação, como pode ser visto na Fig. 2.10 (a), que corresponde ao

diagrama de bifurcação do mapa logístico, onde diferentes tipos de comportamentos são apresentados. Na Fig. 2.10(a), pode-se observar o comportamento do sistema $x(t)$ para variações do parâmetro A . No diagrama podem ser vistos os intervalos para os diferentes tipos de atratores tanto para pontos fixos quanto ciclos periódicos, quase-periódicos e caos. Quando $0 < A < 3$ o sistema converge para um ponto fixo. Quando $A > 3$ começa a dinâmica periódica e o período do ciclo aumenta 2, 4, 8..., até chegar a um estado quase-periódico e caótico.

O expoente de Lyapunov mede a média de taxa de convergência entre trajetórias inicialmente próximas. O valor positivo do expoente de Lyapunov indica uma dinâmica caótica; o valor negativo indica uma dinâmica não caótica. Na Fig. 2.10(b), pode-se ver de forma gráfica que o expoente de Lyapunov é negativo quando $A < 3.7$; quando $A > 3.7$ aproximadamente, o sistema entra num estado caótico exceto para algumas janelas periódicas correspondendo ao expoente de Lyapunov negativo.

Figura 2.10: (a) Diagrama de Bifurcação do mapa logístico; (b) O expoente de Liapunov do mapa logístico.

2.7 Os expoentes característicos de Lyapunov

Como se sabe, um atrator caótico pode ser visto como resultado da combinação de dobras com um número infinito de expansões em pelo menos uma direção e contrações em outras direções. Como consequência, é extremamente difícil, senão quase impossível na prática, seguir a evolução de um fluxo caótico quando a divergência das trajetórias sobre atrator torna-se rápida.

Uma ferramenta importante para a análise de sistemas dinâmicos é o expoente de Lyapunov, que mede a média da taxa de divergências entre trajetórias inicialmente próximas. Há tantos expoentes de Lyapunov quanto dimensões no espaço de estados, sendo que o maior expoente é geralmente o mais importante.

Figura 2.11: Evolução de um elemento de volume esférico de raio $\varepsilon_0(x_0)$ em torno de um ponto inicial x_0 . Depois de um tempo t , a esfera torna-se uma elipse com eixos principais $\varepsilon_1(t)$ e $\varepsilon_2(t)$ para o caso bidimensional.

Considera inicialmente sistemas contínuos (m equações diferenciais ordinárias). Imagine-mos um pequeno hiper-esférico contendo pontos y_0 (raio $\varepsilon_0(x_0)$) em uma vizinhança do ponto inicial x_0 em uma linha de fluxo (como pode ser visto na Fig. 2.11), i.e.,

$$|y_0 - x_0| \leq \varepsilon_0(x_0). \quad (36)$$

Com o passar do tempo, o fluxo se deforma da hiper-esfera para um hiper-elipsoidal com eixos principais $\varepsilon_k(t)$, $k = 1, \dots, m$. Os expoentes de Lyapunov que medem o crescimento exponencial dos eixos principais $\varepsilon_k(t)$ são definidos por:

$$\lambda_i = \lim_{t \rightarrow \infty} \lim_{\varepsilon_0(x_0) \rightarrow 0} \frac{1}{t} \ln \frac{\varepsilon_i(t)}{\varepsilon_0(x_0)}, \quad i = 1, \dots, m. \quad (37)$$

Da Eqn. 37 é imediato que:

$$\varepsilon_i(t) \sim \varepsilon_0(x_0) e^{\lambda_i t}. \quad (38)$$

Pode-se concluir então que:

- A existência de um ou mais expoentes de Lyapunov positivos define uma instabilidade orbital nas direções associadas;

- Uma solução caótica implica na existência de pelo menos um expoente de Lyapunov positivo;
- Para uma solução periódica ou quase periódica pode-se esperar que os deslocamentos na direção perpendicular ao movimento diminua com o tempo, enquanto que ao longo da trajetória eles não se devem alterar, correspondendo a um simples deslocamento do ponto inicial. Segue portanto da Eqn. 38, que no caso de uma solução periódica e uma solução quase-periódica, $\lambda_i < 0$ nas direções perpendiculares ao movimento e $\lambda_i = 0$ ao longo da trajetória.

Num instante t , o elemento de hiper-volume no espaço de fases é escrito

$$\delta V(t) = \prod_{i=1}^m \varepsilon_i(t) \quad (39)$$

Substituindo Eqn. 38 por Eqn. 39 tem-se

$$\delta V(t) = \delta V(0) \exp \left(\sum_{i=1}^m \lambda_i t \right). \quad (40)$$

Existem, portanto, duas situações para as quais o hiper-volume no espaço de fases não diverge:

- Quando $\sum_{i=1}^m \lambda_i = 0$, situação em que $\delta V(t) = \delta V(0)$ e o sistema é conservativo.
- Quando $\sum_{i=1}^m \lambda_i < 0$, que resulta em $\delta V(t) < \delta V(0)$ (diminuição do hiper-volume no espaço de fases) e o sistema é dissipativo.

É possível identificar um atrator pelo sinal dos expoentes de Lyapunov. É uma boa ilustração de tal fato considerar-se os sinais dos expoentes associados com os possíveis atratores em um espaço de fases tridimensional ($m=3$). Tem-se:

- Ponto fixo: neste caso os sinais dos expoentes são $(-, -, -)$. Com efeito, as trajetórias convergem para um único ponto, cessando qualquer deslocamento;
- Ciclo limite: tem-se $(0, -, -)$ correspondendo o expoente nulo à direção ao longo do tempo da trajetória;
- Toro T^2 : existem agora duas direções ao longo das quais processam-se deslocamentos. Os sinais são $(0, 0, -)$;

- Atrator caótico: um dos expoentes deve ser positivo, uma vez que existe dependência das condições iniciais. Ao longo da trajetória associa-se um expoente nulo. Sendo o sistema dissipativo, a desigualdade $\sum_{i=1}^m \lambda_i < 0$ é válida e o terceiro expoente de Lyapunov deve ser necessariamente negativo. Resulta, portanto, $(+, 0, -)$.

Na Fig. 2.12 estão representados os atratores e os respectivos sinais. Observe-se que para os atratores de ponto fixo, ciclo limite e quase-periódico, a dimensão é dada pelo número de expoentes de Lyapunov nulos.

Figura 2.12: Sinais dos expoentes de Lyapunov para seus atratores em $\mathbb{R}^3$. (a) Ponto fixo; (b) Ciclo limite; (c) Toro; (d) Atrator caótico.

Para a ocorrência de atratores caóticos em sistemas contínuos, três condições precisam ser satisfeitas:

- Existência de pelo menos um expoente de Lyapunov $\lambda_i > 0$;
- $\sum_{i=1}^m \lambda_i < 0$, o que garante a contração do volume no espaço de fases (sistema dissipativo);
- Dimensão do espaço de fases $m \geq 3$.

A última condição pode ser mostrada pelo absurdo. Admitamos que seja possível obter um atrator estranho em duas dimensões ($m = 2$). Nesse caso, um dos expoentes de Lyapunov é necessariamente positivo. Ao mesmo tempo, ao longo da direção paralela ao fluxo, o expoente

associado é nulo. Então teria-se $(0, +)$ como sinais para os expoentes e resultaria $\sum_{i=1}^2 \lambda_i > 0$ e o elemento do volume no espaço de fases divergiria, o que não é possível. Deve-se portanto ter um espaço de fases pelo menos tridimensional.

A definição dos expoentes de Lyapunov para sistemas dinâmicos discretos (mapas) é bastante similar. Seja o mapa unidimensional:

$$x_{n+1} = F(x_n). \quad (41)$$

Sejam dois pontos iniciais x_0 e y_0 e a distância inicial entre eles:

$$\delta = y_0 - x_0. \quad (42)$$

Admitamos que depois de uma iteração a nova distância seja

$$\delta' = y_1 - x_1. \quad (43)$$

tal que

$$\delta' = e^L \delta. \quad (44)$$

Então L mede a taxa exponencial de expansão da distância δ até a distância δ' como resultado de uma única iteração. Mas a Eqn. 43 pode ser escrita como:

$$\begin{aligned} \delta' &= F(y_0) - F(x_0) \\ &= F(x_0 + \delta) - F(x_0) \end{aligned} \quad (45)$$

onde na última passagem utilizou-se a Eqn. 42. Usando as Eqn. 45 e 44 resulta $|F(x_0 + \delta) - F(x_0)| = |\delta|e^L$. Iterando-se o mapa N vezes tem-se $|F^N(x_0 + \delta) - F^N(x_0)| = |\delta|e^{NL}$ (com $F^N(x) = F(F \dots F(x) \dots)$, N vezes) que pode ser escrita como

$$L = \frac{1}{N} \ln \left| \frac{F^N(x_0 + \delta) - F^N(x_0)}{\delta} \right|. \quad (46)$$

mas L depende de N e δ . Consideramos então uma distância inicial infinitesimal ($\delta \rightarrow 0$) depois de um número infinito de iterações ($N \rightarrow \infty$). Isso leva a

$$\lambda(x_0) \equiv L(x_0)$$

$$\begin{aligned}
&= \lim_{N \rightarrow \infty} \lim_{\delta \rightarrow 0} \frac{1}{N} \ln \left| \frac{F^N(x_0 + \delta) - F^N(x_0)}{\delta} \right| \\
&= \lim_{N \rightarrow \infty} \frac{1}{N} \ln \left| \frac{dF^N(x_0)}{dx_0} \right|
\end{aligned} \tag{47}$$

que é por definição o expoente de Lyapunov do mapa e constitui uma medida da divergência exponencial ($\lambda > 0$) ou da contração ($\lambda < 0$). A Eqn. 47 pode ser comparada com a Eqn. 38 (caso contínuo) desde que se faça $t \leftrightarrow N$ e $\varepsilon_0(x_0) \leftrightarrow \delta$.

Reescrevendo a definição fornecida pela Eqn. 47 pela regra da cadeia:

$$\frac{d}{dx_0} F^N(x_0) = \frac{d}{dx_0} F(x_{N-1}) \frac{d}{dx_0} F(x_{N-2}) \dots \frac{d}{dx_0} F(x_0), \tag{48}$$

onde $x_i(x_0) = F^i(x_0)$. Substituindo Eqn. 48 em Eqn. 47 obtém-se

$$\begin{aligned}
\lambda(x_0) &= \lim_{N \rightarrow \infty} \frac{1}{N} \ln \left| \prod_{i=0}^{N-1} \frac{d}{dx_0} F(x_i) \right| \\
&= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln |F'(x_i)|
\end{aligned} \tag{49}$$

A generalização para m dimensões é imediata. Seja o mapa $\mathbf{x}_{n+1} = F(\mathbf{x}_n)$, com $\mathbf{x} \in \mathbb{R}^m$ e, $F: \mathbb{R}^m \rightarrow \mathbb{R}^m$. Define-se o espectro de expoentes característicos de Lyapunov como:

$$(e^{\lambda_1}, e^{\lambda_2}, \dots, e^{\lambda_m}) = \lim_{N \rightarrow \infty} \left\| \prod_{i=0}^{N-1} J(\mathbf{x}_i) \right\|^{1/N} \tag{50}$$

onde $J(\mathbf{x}_i)$ é a matriz Jacobiana do mapa calculada em $\mathbf{x}_i = F^i(\mathbf{x}_0)$, isto é:

$$J(\mathbf{x}_i) = \frac{\partial F}{\partial \mathbf{x}} \bigg|_{\mathbf{x}_i} = \frac{\partial (F_1, F_2, \dots)}{\partial (x_1, x_2, \dots)} \bigg|_{\mathbf{x}_i}, \tag{51}$$

e

$$\prod_{i=0}^{N-1} J(\mathbf{x}_i) = J(\mathbf{x}_{N-1}, \mathbf{x}_{N-2}, \dots, \mathbf{x}_0). \tag{52}$$

Como exemplo, temos o expoente de Lyapunov para o mapa logístico, que é dada pela Eqn. 53. Quando o expoente de Liapunov é $\lambda > 0$ indica que a dinâmica é caótica. Quando ele é $\lambda = 0$ a dinâmica é quase-periódica e quando é $\lambda < 0$ é periódica. Na Fig. 2.10(b), pode-se ver de forma gráfica que o expoente de Lyapunov é negativo quando $A < 3.7$; quando $A > 3.7$

aproximadamente, o sistema entra num estado caótico exceto para algumas janelas periódicas correspondendo o expoente de Lyapunov negativo.

$$\lambda_x = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=t_0}^k \log |A(2x(t) - 1)| \quad (53)$$

2.8 Crise

Um dos problemas em teoria do caos é decifrar a sequência de bifurcações através do qual um sistema em particular entra em comportamento caótico quando a variável paramétrica é modificada. Podendo-se observar que o atrator caótico sofre de estados críticos, os quais são chamados apropriadamente de **crises**. A **crise** pode ser definida como a colisão entre um atrator caótico e um ponto fixo instável coexistente ou uma órbita periódica. Existem diferentes tipos de crise: crise interior, crise exterior e crise de união de atratores (*Attractor-merging crisis*)

- Crise exterior: Acontece quando o atrator caótico colide com o ponto fixo instável, desaparecendo o atrator caótico. A Fig. 2.13 mostra a crise exterior para o mapa quadrático $x(t+1) = A - x(t)^2$, podendo-se observar a colisão do atrator com o ponto fixo instável quando $A = 2$.

Figura 2.13: Crise exterior, para o mapa quadrático quando $A = 2$.

- Crise interior: Acontece quando o atrator caótico colide com o ponto fixo instável dentro da bacia de atração do atrator caótico. O atrator incrementa de tamanho abruptamente, como pode ser visto na Fig. 2.14, que mostra a crise interior para o mapa quadrático.

Figura 2.14: Crise interior pode-se ver que o atrator caótico incrementa de tamanho.

- *Attractor-merging crisis*: É um tipo de bifurcação que não é observado no mapa logístico (C. Grebogi, 1987), e acontece quando dois atratores caóticos são simétricos com respeito ao ponto repulsor, expandido-se até se chocarem simultaneamente. Este fenômeno pode ser visto no mapa senoidal. Quando o valor do parâmetro é menor que o ponto de crise, a órbita caótica fica de um só lado, como pode ser visto na Fig. 2.15 (a), e quando η excede o valor do ponto crítico $\eta_c \equiv 1$ o atrator caótico sofre uma expansão, como pode ser visto na Fig. 2.15 (b).

Figura 2.15: (a) Atrator caótico, quando $\eta = 0.97$ ou seja antes do ponto de crise; (b) Atrator caótico quando $\eta = 1.2$ ou seja depois do ponto de crise, acontecendo a expansão do atrator.

2.9 Considerações Finais

Neste capítulo uma revisão dos principais conceitos sobre sistemas dinâmicos foi feita. Definindo o sistema dinâmico tanto para tempo contínuo como discreto, também foi descrito os diferentes tipos de comportamentos em sistemas dinâmicos, tais como: ponto fixo, ciclo limite, oscilação quase-periódica e caos.

Sendo o ponto fixo um comportamento importante, foi feita uma descrição detalhada do processo de busca de pontos fixos e da determinação da estabilidade de um ponto fixo,

Finalmente foi apresentado uma caracterização do comportamento caótico assim como a determinação dos expoentes de Lyapunov e o conceito de crise.

Redes Neurais Modeladas por Sistemas Dinâmicos

3.1 Considerações Iniciais

Na primeira parte deste capítulo será apresentada uma *descrição das características gerais* de redes neurais, assim como dos diferentes tipos de aprendizagem e arquiteturas que as redes podem possuir.

Como foi descrito no capítulo anterior, o caos é um fenômeno comum em sistemas dinâmicos não-lineares. Portanto, na segunda parte deste capítulo, serão apresentadas, em certo nível de detalhes, duas redes neurais modeladas não só como modelos de processamento de informação, mas também como sistemas dinâmicos não-lineares e de alta dimensão. As quais são Modelo de Hopfield e Rede Neural Celular. O primeiro possui uma arquitetura com interação global entre os neurônios e na segunda, cada neurônio interage com seus vizinhos.

3.2 Redes Neurais Artificiais (RNA)

Uma rede neural artificial (RNA) é um modelo computacional inspirado nas redes neurais biológicas e pode ser considerada como um sistema de processamento de informação com ca-

racterísticas salientes, tais como aprendizagem através de exemplos, adaptabilidade, robustez, capacidade de generalização e tolerância a falhas (D. Hush, 1993).

A RNA pode ser definida como uma estrutura distribuída, de processamento paralelo, formada de neurônios artificiais (ou chamados elementos de processamento), interligados por um grande número de conexões (sinapses), os quais são utilizados para armazenar conhecimento que está disponível para ser utilizado (Haykin, 1994).

Uma rede neural é caracterizada pelas propriedades de seus neurônios, pela arquitetura da rede (a topologia que a rede pode possuir) e pelos algoritmos de aprendizado, os quais são descritos a seguir.

3.2.1 Estrutura do Neurônio Artificial

Um neurônio artificial é uma unidade de processamento de informação de redes neurais. O modelo de neurônio mais conhecido é chamado neurônio de McCulloch-Pitts (A. Braga, 1998) e pode ser visto na Fig. 3.1. Nessa figura podemos ver N sinais de entradas representadas pelas variáveis $x_1, x_2, x_3 \dots x_N$. A ligação de um neurônio com índice i ao neurônio j é representada por um peso w_{ij} , que determina o nível de influência do neurônio j para o neurônio i . Se o valor de w_{ij} é positivo é dito que a sinapse é excitatória e é inibitória se for negativo.

Figura 3.1: Modelo de um Neurônio Artificial

Existem duas etapas de processamento para cada neurônio: soma e ativação.

Na primeira etapa, os sinais de entrada x_j e os pesos w_{ij} são combinadas pelo somatório:

$$y_i = \sum_{j=1}^N w_{ij} x_j \quad (1)$$

onde y_i é chamado *estado interno* do neurônio i .

Na segunda etapa, a saída do neurônio é gerada através de aplicação de uma função de ativação:

$$x_i = f(y_i) \quad (2)$$

onde a saída do neurônio é representada por x_i e f é a função de ativação aplicada ao estado interno do neurônio, que tem como objetivo limitar o nível de ativação do neurônio. Geralmente, $x_i \in [-1, 1]$ ou $x_i \in [0, 1]$, no caso de x_i ser um valor contínuo e $x_i \in \{-1, 1\}$ ou $x_i \in \{0, 1\}$, no caso discreto.

Figura 3.2: Funções de ativação. (a) Função degrau; (b) Função tangente hiperbólica

Existem vários tipos de funções de ativação. A Fig. 3.2 mostra duas funções de ativação mais usadas: a função degrau e a tangente hiperbólica.

Como pode ser visto na Fig. 3.1, a saída de um neurônio pode ser uma entrada de outro. Geralmente, uma rede neural é formada por muitos neurônios de alguma forma acoplados.

3.2.2 Arquitetura da Rede

A definição da arquitetura é um ponto importante na modelagem de rede neural, porque ela restringe o tipo de problema que pode ser tratado. Por exemplo as redes de uma camada como por exemplo o *Perceptron* só podem resolver problemas linearmente separáveis (R. Aggarwal, 1997). Uma rede também pode estar formada por múltiplas camadas, as quais podem ser classificadas em três grupos: camada de entrada, camadas intermediárias ou ocultas e camada de saída. Como se pode ver na Fig. 3.3.

Baseado em fluxo dos sinais, redes neurais também podem ser classificadas em dois tipos: *FeedForward* e redes *Recorrentes*.

Figura 3.3: Modelo de uma Rede Neural Artificial

- Redes *FeedForward*

Como pode ser visto na Fig. 3.4 a estrutura de uma rede *FeedForward* consiste em camadas de neurônios na qual a saída de um neurônio de uma camada, alimenta todos os neurônios da camada seguinte. O aspecto fundamental desta estrutura é que não existem laços de realimentação. A rede *MultiLayer Perceptron* (MLP) é um tipo de rede *feedforward* (D. Rumelhart, 1986).

Figura 3.4: Redes *FeedForward*

- Redes *Recorrentes*

Redes recorrentes são aquelas que possuem conexões de realimentação, como pode ser visto na Fig. 3.5, as quais proporcionam comportamento dinâmico. O modelo de Hopfield é um exemplo de rede neural recorrente e será apresentado mais para frente.

Figura 3.5: Redes Recorrentes

Em geral, os seguintes parâmetros são importantes para definir a arquitetura de uma rede neural: número de camadas, número de neurônios em cada camada e tipo de conexão entre os neurônios, que define a rede se é *feedforward* ou *Recorrentes*.

3.2.3 Algoritmos de Aprendizado de uma RNA

Uma propriedade importante das redes neurais é a habilidade de aprender a partir de seu ambiente. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos de conexões entre os neurônios, denominado *treinamento*. Existem muitos algoritmos de aprendizado. Cada um serve para determinadas redes neurais. Entre os principais tem-se:

- **Aprendizado por Correção de Erro:** algoritmo muito conhecido baseado na regra Delta, que busca minimizar a função de erro usando o gradiente descendente. Este é o princípio usado no algoritmo *BackPropagation*, muito utilizado para o treinamento de redes de múltiplas camadas como a *Multilayer-Perceptron* (MLP) (James A. Freeman, 1991);
- **Aprendizado Competitivo:** no qual os neurônios de uma camada competem entre si pelo privilégio de permanecerem ativos, tal que o neurônio com maior atividade seja o único a participar do processo de aprendizado. É usado em Mapas de Kohonen (Kohonen, 1988) e redes ART (Gail A. Carpenter, 1992);
- **Aprendizado Hebbiano:** se dois neurônios estão simultaneamente ativos a conexão entre eles deve ser fortalecida caso contrário será enfraquecida (Hebb, 1949). Utilizada no Modelo de rede Hopfield (Hopfield, 1982);
- **Aprendizado de Boltzmann:** é uma regra de aprendizagem estocástica obtida a partir de princípios da teoria da informação e da termodinâmica. O objetivo da aprendizagem de Boltzmann é ajustar os pesos das conexões de tal forma que o estado das unidades visíveis satisfaça uma distribuição de probabilidades desejada em particular (D. Ackley, 1985);

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente. Nesse contexto existem os seguintes paradigmas de aprendizado:

- **Aprendizado Supervisionado:** é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Reforço:** é uma variante da aprendizagem supervisionada na qual se fornece à rede somente uma crítica da correção da saída da rede e não a resposta correta em si;

- **Aprendizado Não Supervisionado (auto-organização):** não existe um agente externo indicando a resposta desejada para os padrões de entrada. Este tipo de aprendizado é utilizado nos modelos de Mapas de Kohonen (Kohonen, 1988), redes ART1, ART2 (Gail A. Carpenter, 1992) (G. Carpenter, 1987).

3.3 O Modelo de Hopfield

Um dos principais responsáveis pelo desenvolvimento das redes neurais foi J. Hopfield, que desenvolveu um modelo de rede recorrente inspirado em conceitos de dinâmica não-linear e física estatística (Hopfield, 1982). O modelo pode ser usado como *memória associativa* ou *memória endereçável por conteúdo*. A idéia essencial do modelo de Hopfield é mapear uma memória fundamental em um ponto fixo estável de um sistema dinâmico representado em forma de uma rede recorrente.

Figura 3.6: Idéia básica do modelos de Hopfield

A idéia básica do modelo de Hopfield para armazenamento e reconhecimento de padrões é a seguinte: qualquer sistema físico cuja dinâmica em um espaço de fases é dominado por um número substancial de estados localmente estáveis aos quais são atraídos, podendo ser considerado como uma memória geral endereçada por conteúdo. O modelo de Hopfield pode ser considerado, como um sistema físico (sistema dinâmico não-linear) por várias coordenadas $X_1, X_2, \dots, X_n$, que são os componentes do vetor de estados $\mathbf{X}$. Supondo que o sistema possui pontos fixos localmente estáveis $\mathbf{X}_a, \mathbf{X}_b, \dots$, então se o sistema é iniciado perto o suficiente de qualquer ponto fixo $\mathbf{X}_a$, como em $\mathbf{X} = \mathbf{X}_a + \delta$, ele irá proceder no tempo até que $\mathbf{X} = \mathbf{X}_a$. A informação é armazenada no sistema como os vetores $\mathbf{X}_a, \mathbf{X}_b, \dots$. O ponto inicial $\mathbf{X}$ representa um conhecimento parcial do item $\mathbf{X}_a$, e a partir disto o sistema então gera a informação total de $\mathbf{X}_a$. A interesse central desta idéia é a construção de uma rede neural estável. O interesse

central desta idéia é a construção de uma rede neural estável. A Fig. 3.6 mostra a idéia deste modelo.

No modelo de Hopfield todos os neurônios são interligados entre si e as ligações são simétricas, i.e., o peso da conexão do neurônio i para j é igual ao peso da conexão de j para i . Hopfield demonstrou que se os pesos das ligações ou sinapses entre os neurônios são simétricos com diagonal zero, é possível garantir que a rede converge para um ponto fixo.

Na próxima seção serão apresentado o mecanismo de funcionamento de modelo de Hopfield discreto e contínuo.

3.3.1 Modelo de Hopfield Discreto

Esse modelo foi desenvolvido por Hopfield em 1982 (Hopfield, 1982), utilizando o neurônio de McCulloch-Pitts assumendo valores binários para saídas dos neurônios da rede.

3.3.1.1 Estrutura do Neurônio e da Rede

O modelo de Hopfield discreto consiste em uma rede de uma única camada formada por unidades de processamento não-lineares (neurônios), cujos estados de ativação podem ser valores binários ou bipolares: $\{0, 1\}$ e $\{-1, 1\}$ respectivamente. A função de ativação do neurônio utilizada é a *degrau*. As seguintes equações definem o processamento de soma e ativação de cada neurônio, respectivamente:

$$y_i(t+1) = \sum_{j=1}^N w_{ij}x_j(t) + \theta_i \quad (3)$$

$$x_i(t+1) = \begin{cases} 1 & y_i(t+1) > 0 \\ -1 & y_i(t+1) < 0 \\ x_i(t) & y_i(t+1) = 0 \end{cases} \quad (4)$$

na qual $x_i(t)$ e $y_i(t)$ representam a saída e o estado interno do neurônio i na iteração t , para $i = 1, 2, \dots, N$. w_{ij} é o peso de conexão de neurônio i ao neurônio j .

A Fig. 3.7 mostra a topologia de uma rede Hopfield, no qual os neurônios estão totalmente interconectados sendo que um elemento qualquer não possui conexões com ele mesmo (não tem conexões auto-concorrentes). Neste caso, o conjunto de pesos podem ser representados por uma matriz W com as seguintes propriedades: uma matriz quadrada, simétrica ($w_{ij} = w_{ji}$) e os elementos de seu diagonal nulos ($w_{ii} = 0$).

Figura 3.7: Topologia da Rede Hopfield

O Hopfield define a energia da rede como segue

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} x_i x_j + \sum_{i=1}^N x_i \theta_i \quad (5)$$

Por simplicidade assume-se que todos os valores limiares θ são zeros, a Eqn. 5 da energia é dada por:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} x_i x_j \quad (6)$$

Quando este modelo é usado como memória associativa, ele apresenta duas fases de processamento: a fase de armazenamento e a fase de reconhecimento. Ambas são realizadas por minimização dessa função de energia, dirigindo o sistema para um estado estável.

3.3.1.2 A Fase de Aprendizado

A fase de aprendizado serve para armazenar os padrões nos estados estáveis da rede, através de ajuste dos pesos de conexões entre os neurônios. Este aprendizado é realizado baseado na regra de Hebb (Hebb, 1949), que define os pesos de sinapses com a seguinte propriedade: "Quando o axônio do neurônio i está perto do neurônio j e de forma persistente toma parte na excitação do neurônio j , certo processo de crescimento ou alteração metabólica acontece em um ou ambos neurônios, de forma que a conexão ou peso da ligação entre i e j é incrementada". A regra de aprendizado do modelo de Hopfield é dada pela seguinte equação:

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^M \xi_{\mu,i} \xi_{\mu,j} & \text{se } i \neq j \\ 0 & \text{se } i = j \end{cases} \quad (7)$$

onde w_{ij} corresponde ao peso entre os neurônios i e j , N é o número de elementos de cada padrão, M é o número total de padrões a ser armazenados e $\xi_{\mu,i}$ representa o i -ésimo elemento do μ -ésimo padrão.

O processo de armazenamento corresponde a minimização da função de energia Eqn. 6 e conseqüentemente coloca os padrões a ser armazenados nos estados estáveis do sistema.

Suponhamos que queremos armazenar os padrões $\xi^s = \xi_1^s, \xi_2^s, \dots, \xi_N^s$, onde $\xi_i^s = \pm 1$. Precisamos minimizar a função de energia da rede da equação. A seguinte equação mostra que a energia é decrescente (o somatório é negativo) no processo de aprendizagem.

$$\begin{aligned}
 E &= -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} x_i x_j \\
 &= -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} \xi_{\mu,i} \xi_{\mu,j} \\
 &= -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N (\xi_{\mu,i})^2 (\xi_{\mu,j})^2
 \end{aligned} \tag{8}$$

3.3.1.3 Fase de Recuperação dos Padrões

A recuperação dos padrões armazenados é feita por simplesmente iterar o sistema definido pelos Eqn. 3 e Eqn. 4. Um vetor N -dimensional $x_i(0)$, que representa uma versão corrupta de um padrão armazenado, é tomado como estado inicial da rede. A seguinte equação mostra que sucessivas iterações leva a energia decrescer.

$$\begin{aligned}
 \Delta_k E &= E(t+1) - E(t) \\
 &= -\frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} x_i(t+1) x_j + \frac{1}{2} \sum_{i=1}^N \sum_{(j=1)(j \neq i)}^N w_{ij} x_i(t) x_j \\
 &= -\Delta x_i(t) \sum_{(j=1)(j \neq i)}^N w_{ij} x_j
 \end{aligned} \tag{9}$$

onde três coisas podem acontecer em $\Delta x_i(t)$: primeiro que estado de x alterne de +1 para -1, então $\Delta x_i(t) < 0$ e toma o valor de -1, então a soma é negativa, o que implica que a energia diminui. Segundo se a energia varia de -1 a +1, então $\Delta x_i(t) > 0$ e toma o valor de 1, então somatório é maior que 1, o que implica que a energia também diminui. Terceiro se $\Delta x_i(k) = 0$,

a energia não muda. Isso garante que o sistema sempre vai cair para um mínimo local. Como o padrão de entrada e seu correspondente padrão armazenado são similares, eles provavelmente estão em uma mesma *bacia de atração* (um conjunto de condições iniciais que levam o sistema para um mesmo atrator). Isso significa que o sistema geralmente não cairá para um mínimo local qualquer, mas cairá o mínimo local que corresponde o padrão a ser reconhecido.

Um exemplo de armazenamento e reconhecimento de padrões utilizando o modelo de Hopfield é apresentado a seguir. A Fig. 3.8 mostra os padrões utilizados no treinamento da rede. A Fig. 3.9 apresenta um processo de reconhecimento iterando o modelo de Hopfield a partir de uma condição inicial mostrado na primeira imagem da sequência. Esta condição inicial é uma versão corrupta do padrão armazenado "USP". Portanto, podemos ver que a iteração do sistema converge a um estado estável que corresponde o padrão "USP".

Figura 3.8: Padrões utilizados para o treinamento da rede Hopfield.

Figura 3.9: A imagem correspondente às saídas da rede em cada iteração até reconhecer um padrão.

Existem dois tipos de atualização dos neurônios que produzem ligeiras diferenças no comportamento da rede. A primeira forma de atualização pode ser feita para todos os neurônios simultaneamente, i.e., cada neurônio calcula seu estado da seguinte iteração usando os valores dos outros neurônios da iteração atual. Essa forma de iteração é conhecida como *atualização síncrona*. A segunda forma de atualização é a assíncrona. Neste caso, cada vez um neurônio é aleatoriamente selecionado e seu estado da próxima iteração é calculado. A atualização para

próximos neurônios selecionados usará os valores dos estados da próxima iteração dos neurônios anteriormente selecionados. Esse processo é repetido para todos os neurônios da rede. Na forma de atualização assíncrona, o cálculo da saída de um neurônio pode afetar o cálculo das saídas dos seguintes neurônios. Isso significa que a ordem de atualização dos neurônios afeta o comportamento da rede. Isso não acontece na forma de atualização síncrona. Mas tanto as atualizações síncronas quanto as assíncronas compartilham as características gerais do modelo de Hopfield.

3.3.2 Modelo de Hopfield Contínuo

O modelo contínuo de Hopfield compartilha as principais propriedades com o modelo discreto (Hopfield, 1994). A estrutura do neurônio para o modelo contínuo pode ser visto na Fig. 3.10.

Figura 3.10: Modelo Contínuo do Neurônio

Pode se ver que o modelo é apresentado como circuito elétrico, onde:

- Os pesos $w_{j1}, w_{j2}, \dots, w_{jN}$ representam a condutores;
- as entradas $x_1(t), x_2(t), \dots, x_N(t)$ representam os potenciais de energia;
- N corresponde ao número de entradas;
- v_j corresponde ao potencial desenvolvido pelo neurônio;
- φ é uma função continua não-linear;
- R_j é a resistência do neurônio j ;
- e C_j é a capacitance do neurônio j .

Na Fig. 3.10 pode ser verificado que o fluxo de corrente total esta dada por:

$$C_j \frac{v_j(t)}{R_j} + \frac{dv_j(t)}{dt} \quad (10)$$

onde o primeiro elemento corresponde à resistência para uma entrada e o segundo ao potencial que pode ser livrado pelo condensador. Das leis Kirchoff sabemos que, o fluxo de corrente total a qualquer nó de um circuito elétrico é zero. Conseqüentemente, aplicando as leis de Kirchoff ao neurônio da Fig. 3.10, temos que:

$$C_j \frac{dv_j(t)}{dt} + \frac{v_j(t)}{R_j} = \sum_{i=1}^N w_{ji} x_i(t) + I_j \quad (11)$$

esta equação corresponde ao estado de ativação $v_j(t)$ do neurônio j , e através deste pode ser determinado a saída do neurônio aplicando uma função de ativação φ . φ deve ser uma função não-linear e contínua, portanto diferenciável no tempo t . Uma função comum para representar φ é a tangente hiperbólica dada por:

$$x_j = \varphi(v_j) = \tanh(\lambda v_j) \quad (12)$$

onde λ é chamado de parâmetro de ganância. A Fig. 3.11 apresenta o comportamento da função tangente hiperbólica, para diferentes valores de λ , pode-se observar que é essa função é não-linear, monotonicamente crescente e continua.

Figura 3.11: Função de ativação dos neurônios para a rede de Hopfield-caso contínuo, com diferentes valores de λ .

Como sabemos que $x_i(t) = \varphi_i(v_i(t))$ a Eqn. 11, pode ser reescrita como:

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^N w_{ji} \varphi_i(v_i(t)) + I_j \quad (13)$$

Nesta equação se deve tomar em consideração as seguintes condições:

- A matriz de pesos deve ser simétrica, i.e., $w_{ij} = w_{ji}$, para todo i e j ;
- Cada neurônio tem uma função de ativação não-linear φ_i ;
- O inverso da função de ativação existe, e pode ser escrita como $v = \varphi_i^{-1}(x)$.

A função de energia para o modelo contínuo de Hopfield (Hopfield, 1994) está dado por:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j + \sum_{j=1}^N \frac{1}{R_j} \int_0^{x_j} \varphi(x) dx - \sum_{j=1}^N I_j x_j \quad (14)$$

Hopfield demonstrou que a função de energia é uma função monótona decrescente, limitada e estável em Lyapunov. i.e., que a rede consegue estabilizar num mínimo local. Portanto, o modelo contínuo também pode ser usado como memória associativa.

3.3.3 Uma Aplicação de Reconhecimento de Faces Usando o Modelo de Hopfield

Nesta sessão se apresenta uma aplicação em reconhecimento de faces utilizando o modelo de Hopfield. Esta aplicação forma parte do estudo e desenvolvimento desta pesquisa.

Para o reconhecimento de faces duas etapas são propostas (Cáceres, 2002): extração de características e reconhecimento. A primeira é realizada mediante a utilização da transformada de *wavelets* (Daubechies, 1992) (Gomez et al., 1997) para a redução da dimensão da imagem, assim o processo computacional é menos pesado, depois a imagem resultante é binarizada e é aplicado um processo de normalização das transformações geométricas (translação, rotação e escala). Na segunda etapa o modelo de Hopfield é usado para o reconhecimento da face.

Figura 3.12: Processo de redução da dimensionalidade da imagem de uma face; (a) Imagem original de 256×256 ; (b) Primeiro nível de redução; (c) Teceiro nível de redução.

A Fig. 3.12 mostra o processo de redução da dimensão de uma imagem, usando a transformada Daubechies *wavelet* (Daubechies, 1992). A Fig. 3.12 (a) mostra a imagem original; a Fig. 3.12 (b) mostra a primeira decomposição da imagem original em quatro sub-bandas onde: a primeira decomposição (imagem acima a direita da Fig. 3.12 (b)) corresponde à imagem de dimensão reduzida e corresponde à convolução da imagem original com uma função de escala das *wavelet* (componentes de baixa frequência), as outras 3 imagens corresponde as os detalhes

da imagem (componentes de alta frequência). Pode ser percebido que a primeira imagem é mais similar à imagem original e é utilizada em decomposições seguintes, como pode ser visto na Fig. 3.12 (c).

A aplicação da *wavelet* ajuda a atenuar as diferenças de gestos que uma pessoa pode ter. A Fig. 3.13 mostra o resultado de aplicar *wavelets* para duas imagens de uma mesma face, mas para gestos diferentes. Pode se ver que os gestos das duas imagens são atenuados.

Figura 3.13: (a) Imagem sem gestos; (b) Imagem depois de aplicar *wavelets* na imagem anterior; (c) Imagem com sorriso; (d) Imagem depois de aplicar *wavelets* na imagem anterior.

Uma vez obtida a imagem reduzida, é binarizada aplicando um *threshold* θ à imagem. A Fig. 3.14, mostra o resultado de aplicar *thresholding* à imagem de uma face.

Figura 3.14: Processo de binarização aplicando um *threshold* $\theta = 70$ para uma imagem de uma face.

Um problema nas imagens das faces é que podem sofrer transformações geométricas lineares, ou seja, duas imagens apesar de corresponder à mesma pessoa pode ser diferentes em tamanho, ou sofrer deslocamentos e inclusive rotações, isso dificulta o processo de reconhecimento. Para resolver esse problema é realizado a normalização das transformações geométricas tirando

aquelas diferenças nas imagens de uma mesma face. Para este processo de normalização foram usados os momentos, da forma proposta por Ravichandran e Yegnanarayana (Ravichandran and Yegnanarayana, 1991):

1. Calcular o centro de massa por:

$$x_c = \sum \frac{x_i}{N} \quad y_c = \sum \frac{y_i}{N} \quad (15)$$

os pontos x_i e y_i correspondem as coordenadas dos pixels. x_c e y_c é o centro de massa obtido.

2. Calcular os pesos w_i dados por:

$$w_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (16)$$

3. Calcular os momentos centrais por:

$$\begin{aligned} X_{w1} &= \frac{\sum (w_i x_i)}{\sum w_i} \\ Y_{w1} &= \frac{\sum (w_i y_i)}{\sum w_i} \\ X_{w2} &= \frac{\sum \left(\frac{x_i^2}{w_i} \right)}{\sum \frac{1}{w_i}} \\ Y_{w2} &= \frac{\sum \left(\frac{y_i^2}{w_i} \right)}{\sum \frac{1}{w_i}} \end{aligned} \quad (17)$$

4. Calcular a distância d dada por:

$$d = \sqrt{(X_{w1} - X_{w2})^2 + (Y_{w1} - Y_{w2})^2} \quad (18)$$

5. Os pontos $P_i (i = 1, 2, \dots, N)$ são redefinidos por:

$$\begin{aligned} X_i &= \frac{(X_i - X_{w1})(X_{w1} - X_{w2})}{d} + \\ &\quad \frac{(Y_i - Y_{w1})(Y_{w1} - Y_{w2})}{d} \\ Y_i &= \frac{(X_{w1} - X_i)(Y_{w1} - Y_{w2})}{d} + \\ &\quad \frac{(Y_i - Y_{w1})(X_{w1} - X_{w2})}{d} \end{aligned} \quad (19)$$

P_i corresponde ao valor dos pixels na posição x_i e y_i .

6. Finalmente, cada novo ponto deve ser dividido pela distância d

A Fig. 3.15 mostra os resultados obtidos do processo de normalização de duas imagens com diferentes transformações geométricas.

Figura 3.15: Resultado do Processo de normalização das transformações geométricas de duas imagens com tamanho, posição e rotações diferentes.

Uma vez que as imagens das faces foram normalizadas elas podem ser armazenadas e reconhecidas seguindo os seguintes passos:

- Devido à pouca diferença entre as imagens de faces. O modelo de Hopfield não foi usado diretamente para o armazenamento das faces. Mas para superar essa desvantagem, usamos uma base de dados chamada protótipos, onde são armazenadas as imagens de cada face que se deseja reconhecer. Como é sabido a imagem de uma pessoa pode ter vários gestos, os quais podem ser: alegre, sério, normal, triste, olhos fechados, assustado e de um olho fechado. Então é necessário eleger um daqueles gestos para que represente a face de uma pessoa e poder ser armazenada na base de dados protótipos.
- No processo de reconhecimento, uma face é apresentada ao sistema e é calculada a distância Euclidiana entre a imagem e os protótipos armazenados na base de dados. Escolhem-se os protótipos com menor distância euclidiana para ser armazenados na rede de Hopfield. Finalmente a imagem a ser reconhecida é apresentada ao modelo de Hopfield, sendo a rede a que decide qual dos padrões armazenados é recuperado.

Este processo foi utilizado para o reconhecimento de faces de 15 pessoas da base de dados da universidade Yale que contém imagens de faces com diferentes expressões faciais obtendo-se os seguintes resultados (Cáceres, 2002):

Nro.	Pessoa	Nro de gestos	% Classificação			Protótipo
			correta	errada	%	
1		7	6	1	85
2		8	6	2	75
3		7	6	1	85
4		7	6	1	85
5		7	7	0	100
6		7	7	0	100
7		7	7	0	100
8		6	4	2	66
9		7	7	0	100
10		7	7	0	100
11		7	7	0	100
12		7	7	0	100
13		7	7	0	100
14		7	6	1	85
15		7	4	3	57

Figura 3.16: Resultados obtidos no processo de reconhecimento de faces

3.3.4 Limitações do Modelo Hopfield

Modelo de Hopfield é uma das principais contribuições da área de redes neurais. O Hopfield não só simplesmente sugere um novo modelo de rede neural para reconhecimento de padrão e otimização, mas também desenvolveu uma função de Lyapunov e relaciona redes neurais com outros sistemas dinâmicos. No entanto, o modelo de Hopfield apresenta algumas limitações em aplicações de engenharia:

- Capacidade de armazenamento é altamente limitada. No modelo de Hopfield, quando armazena muitos padrões, pode causar que o sistema não converge para o estado estável desejado, mas converge para um outro estado estável na fase de reconhecimento. Este fenômeno é conhecida como *crosstalk*. Foi mostrado que o número máximo de padrões aleatórios que uma rede de Hopfield pode armazenar é $P_{máx} \equiv 0.15N$ (J. Hertz, 1991). Quando o número de padrões armazenados $p < 0.15N$, uma recuperação quase perfeita pode ser conseguida. Qualitativamente, a capacidade de memória do modelo de Hopfield é limitada porque uma rede com N unidades binárias tem no máximo 2^N estados distintos, e somente poucos desses estados são estáveis. Mais ainda, nem todos os estados estáveis podem ser projetados para armazenar padrões.
- Existência de memórias espúrias. Os padrões armazenados na rede de Hopfield podem ser chamados de *memórias fundamentais*, que correspondem aos estados estáveis da função

de energia. mas, em geral, também existem outros estados estáveis que não correspondem nenhuma memória fundamental. Esses estados são denominados como *estados espúrios* ou *memórias espúrias*. Devida a existência de memórias espúrias, na fase de reconhecimento, o sistema pode não convergir para o estado desejado, mas converge para algum estado espúrio.

3.4 Redes Neurais Celulares

Uma rede neural celular é um conjunto de células localmente acopladas, onde cada célula é um sistema dinâmico com uma entrada, uma saída, um estado e evoluindo de acordo com algumas regras dinâmicas predefinidas (L. Chua, 1988b) (L. Chua, 1988a).

A idéia de uma rede neural celular surgiu a partir de características de redes neurais e de autômatos celulares. Como as redes neurais, uma rede neural celular, é um sistema não-linear que processa informações de forma paralela e distribuída. Como os autômatos celulares, uma rede neural celular, é formada por um conglomerado de elementos regularmente espaçados, chamadas células, as quais se comunicam entre elas diretamente com os vizinhos mais próximos.

As redes neurais celulares compartilham as características das redes neurais artificiais e autômatos celulares. Das RNAs tomaram características como:

- Processamento assíncrono em paralelo;
- Dinâmica em tempo contínuo ou discreto;
- Interação global entre os elementos da rede (em grande maioria dos casos).

e dos autômatos celulares tomaram as características de:

- Distribuição de elementos em malhas regulares;
- Comunicação entre células a nível local;
- Geração de padrão global via interação local das células.

3.4.1 Topologia de uma Rede Neural Celular

A topologia de uma rede neural celular é similar aos autômatos celulares. A Fig. 3.17 mostra algumas formas de conexões entre as células. Em cada um desses casos, cada célula é conectada

somente com alguns vizinhos mais próximos. Então, cada elemento propaga o sinal do seu estado para todas outras células da rede via seus vizinhos conectados.

Figura 3.17: Exemplos de malhas de uma rede neural celular.

Em uma rede neural celular, os vizinhos de uma célula $C(i, j)$ são definidos pela seguinte expressão:

$$N_r(i, j) = \left\{ \begin{array}{l} C(k, l) | \max\{|k - i|, |l - j|\} \leq r, \\ 1 \leq k \leq M; 1 \leq l \leq N, r \in \mathbb{Z}, r > 0; \end{array} \right. \quad (20)$$

Onde N e M são altura e largura da rede, respectivamente. r é um número inteiro positivo que representa o raio da vizinhança como pode ser visto na Fig. 3.18. $r = 0$ significa que cada célula não é conectada com nenhuma outra célula; $r = 1$ significa que cada célula é somente conectada com seus vizinhos mais próximos e $r > 1$ significa que cada célula é conectada com vizinhos r próximos.

Figura 3.18: Exemplos vizinhança para $r = 1, 2, 3$, em uma matriz de 9×9 .

As redes neurais celulares podem ser de tempo contínuo e discreto.

3.4.2 Rede Neural Celular de Tempo Contínuo

A dinâmica de uma rede neural celular é definida pelas seguintes equações:

$$\frac{dx_{ij}(t)}{dt} = -\alpha x_{ij} + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l) * y_{kl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) * u_{kl} + I \quad (21)$$

$$y_{ij} = \frac{1}{2}(|x_{ij}(t) + 1| - |x_{ij}(t) - 1|) \quad (22)$$

onde $x_{ij}(t)$, $y_{ij}(t)$ representam o estado interno e a saída da célula ij na instante t , respectivamente. u_{ij} é a entrada externa da célula ij . $1 \leq i \leq M$ e $1 \leq j \leq N$ e N e M são dimensões horizontal e vertical, respectivamente. A e B são duas matrizes do tamanho da vizinhança, que representam as interações de saídas e entradas externas entre célula ij e seus vizinhos, respectivamente. $*$ representa operação de convolução.

A Fig. 3.19 mostra a função de saída da rede neural celular.

Figura 3.19: Função de saída da rede neural celular de tempo contínuo.

A Fig. 3.20 mostra um exemplo de convolução. G é uma matriz e W chamada *template*. A convolução $W * G(2,2)$ é o produto interno entre a matriz W e uma parte da matriz G com tamanho da W e centralizada na célula $(2,2)$.

As seguintes condições são assumidas:

$$|x_{ij}(0)| \leq 1, \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (23)$$

$$|u_{ij}| \leq 1, \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (24)$$

$$A(i,j;k,l) = A(k,l;i,j), \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (25)$$

$$\alpha > 0 \quad (26)$$

Figura 3.20: Operação de convolução.

Chua e Yang (L. Chua, 1988b) define uma função de Lyapunov para rede neural celular:

$$\begin{aligned}
 E(t) = & \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} A(i,j;k,l) y_{ij}(t) y_{kl}(t) + \frac{1}{2} \alpha \sum_{(i,j)} y_{ij}(t)^2 \\
 & - \sum_{(i,j)} \sum_{(k,l)} B(i,j;k,l) y_{ij}(t) u_{kl} - \sum_{(i,j)} I y_{ij}(t)
 \end{aligned} \quad (27)$$

Eles provaram os valores de $x_{ij}(t)$ e $y_{ij}(t)$ são limitados e a função de Lyapunov é monotonicamente decrescente no tempo t . Portanto, a evolução de uma rede neural celular converge para algum estado estável que corresponde um mínimo local da função de Lyapunov.

3.4.3 Rede Neural Celular em Tempo Discreto

O modelo discreto da rede neural celular é uma simplificação do modelo contínuo. Desta forma, os estados, saídas e entradas externas são representados pelos seguintes mapas:

$$x_{ij}(t) = \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l) y_{kl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) u_{kl} + I \quad (28)$$

e a equação de saída é:

$$y_{ij}(t) = \begin{cases} 1 & \text{se } x_{ij}(t) > 0, \\ -1 & \text{se } x_{ij}(t) < 0; \end{cases} \quad (29)$$

com $1 \leq i \leq M$ e $1 \leq j \leq N$.

A Fig. 3.21 mostra a função de saída.

Figura 3.21: Função de saída da rede neural celular em tempo discreto.

O modelo discreto compartilha as propriedades de estabilidade do modelo contínuo, obtendo as seguintes vantagens: A implementação numérica é mais simples; a convergência se consegue com menos iterações; A matriz de entrada deve conter valores bipolares, ou seja -1 ou 1; os valores da matriz de saída também são bipolares.

3.4.4 Aplicações das Redes Neurais Celulares

Nesta seção serão apresentadas as seguintes aplicações de redes neurais celulares: memória associativa e processamento de imagens, tais como binarização de imagens, remoção de ruído e extração de contornos, dilatação e erosão.

Não é difícil perceber que as aplicações de redes neurais celulares podem ser realizadas através de definição de *templates A* e *B* apropriadas. Serão ilustrados alguns *templates* para diferentes aplicações nas seguintes subseções.

3.4.4.1 Remoção de ruído

A presença de ruído é comum em imagens. Portanto, técnicas de remoção de ruído são importantes em processamento de imagens.

Ruídos geralmente são pixels isolados modificando os valores dos pixels originais. Isso significa que maioria dos vizinhos de um pixel de ruído são pixels normais (não ruído). Portanto,

os ruídos de imagens podem ser removidos através da mudança do estado da célula que corresponde cada pixel de ruído para o estado dos seus maioria dos vizinhos. Baseada essa idéia, os seguintes *templates* são definidos para remoção de ruídos:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad I = 0$$

(30)

A seguir apresentamos um exemplo do processo de remoção de ruído. Na Fig. 3.22 pode ser visto uma imagem original de 5×5 , na qual foi inserido um pouco de ruído, assim como os respectivos valores de u , $x(0)$ e $y(0)$.

Figura 3.22: Imagens para a remoção do ruído.

Como pode ser observado apenas duas células são as que apresentam ruído, mas após iterar o sistema, alguns valores das variáveis x e y mudam, o resultado da primeira iteração pode ser visto na Fig. 3.23. Pode-se observar também que os valores dos neurônios $\{2, 2\}$ (centro) e a $\{0, 4\}$ (abaixo esquerda) mudaram de valores.

Figura 3.23: Resultado das variáveis x e y depois da primeira iteração.

Depois de varias iterações pode ver que os valores das variáveis x e y não mudam mais (Fig. 3.24). Isso significa que os neurônios errados se estabilizaram com os valores correspondentes à média de seus vizinhos, recuperando a imagem original.

$x(2)$	$y(2)$	$x(3)$	$y(3)$
-6,0 -4,0 -4,0 -4,0 -6,0	-1,0 -1,0 -1,0 -1,0 -1,0	-6,0 -4,0 -4,0 -4,0 -6,0	-1,0 -1,0 -1,0 -1,0 -1,0
-4,0 2,0 4,0 2,0 -4,0	-1,0 1,0 1,0 1,0 -1,0	-4,0 2,0 4,0 2,0 -4,0	-1,0 1,0 1,0 1,0 -1,0
-4,0 4,0 6,0 4,0 -4,0	-1,0 1,0 1,0 1,0 -1,0	-4,0 4,0 6,0 4,0 -4,0	-1,0 1,0 1,0 1,0 -1,0
-4,0 2,0 4,0 2,0 -4,0	-1,0 1,0 1,0 1,0 -1,0	-4,0 2,0 4,0 2,0 -4,0	-1,0 1,0 1,0 1,0 -1,0
-6,0 -4,0 -4,0 -4,0 -6,0	-1,0 -1,0 -1,0 -1,0 -1,0	-6,0 -4,0 -4,0 -4,0 -6,0	-1,0 -1,0 -1,0 -1,0 -1,0

Figura 3.24: Resultado das variáveis x e y para a iteração 2 e 3.

O comportamento do estado de todos os neurônios depois de varias iterações pode ser visto na Fig. 3.25.

Figura 3.25: Comportamento do estado dos neurônios depois de 10 iterações.

Várias características podem ser observadas:

- Os estados de todas as células chegam a convergir;
- Existem neurônios que tinham valores positivos, ou seja neurônios que eram pretos, convergiram do lado positivo;
- Existem neurônios que tinham valores negativos, ou seja neurônios que eram brancos, convergiram do lado negativo;

- Os dois neurônios dos quais tinham ruído inserido, se comportam diferente: O neurônio $\{2, 2\}$, começo com valor -1 (branco), passo para o lado positivo e convergiu do lado positivo, ou seja muda de cor. O neurônio $\{0, 4\}$, começo com um valor 1 (preto), passo para o lado negativo e convergiu do lado negativo, ou seja, mudo de cor também.

A Fig. 3.26 mostra resultados de remoção de ruído para uma imagem real. Podemos ver após algumas iterações os valores de pixel não mudam. Isso implica que o sistema está estabilizado. Também podemos ver que a imagem resultante contém ruído significativamente menor do que a imagem original.

Figura 3.26: Remoção de ruídos, (a) Imagem com ruído, tamanho de 235x235; (b) Imagem depois de 2 iterações; (c) Imagem depois de 4 iterações; (d) Imagem depois de 8 iterações; (e) Imagem depois de 20 iterações.

3.4.4.2 Extração de contornos

A detecção de contornos faz parte de segmentação de imagens, na qual obtém-se informação de discontinuidades em uma determinada imagem. Essas discontinuidades geralmente são limites de segmentos, cujas magnitudes de gradiente são valores grandes. Usando os seguintes valores para A , B e I a rede neural celular pode ser usada para extrair contornos.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad I = -1 \quad (31)$$

A seguir, um exemplo de extração do contorno para uma imagem artificial de 5x5 elementos é apresentado na Fig. 3.27.

Pelo fato de ser uma imagem simples, podemos observar que o contorno consiste em todos pixels pretos menos o pixel na posição central $\{2, 2\}$. Através da Fig. 3.28, pode se observar que o valor do neurônio na posição $\{2, 2\}$ muda de preto para branco logo depois da primeira iteração.

Figura 3.27: Imagem usada para a extração do contorno.

x(1)	y(1)
-1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 -1,0 -1,0
-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0
-3,0 1,0 -1,0 1,0 -3,0	-1,0 1,0 -1,0 1,0 -1,0
-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0
-1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 -1,0 -1,0

Figura 3.28: Valor das variáveis x e y depois da primeira iteração.

Pelos valores mostrados na Fig. 3.24, pode ser visto que o contorno desta imagem é corretamente extraído (os elementos do contorno estão com valor de $y = 1$ e os elementos não contorno estão com valor de $y = -1$).

x(2)	y(2)	x(3)	y(3)
-1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 -1,0 -1,0	-1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 -1,0 -1,0
-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0	-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0
-3,0 1,0 -5,0 1,0 -3,0	-1,0 1,0 -1,0 1,0 -1,0	-3,0 1,0 -5,0 1,0 -3,0	-1,0 1,0 -1,0 1,0 -1,0
-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0	-3,0 3,0 1,0 3,0 -3,0	-1,0 1,0 1,0 1,0 -1,0
1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 1,0 -1,0	-1,0 -3,0 -3,0 -3,0 -1,0	-1,0 -1,0 -1,0 -1,0 -1,0

Figura 3.29: Valor das variáveis x depois da 2 e 3 iteração.

O comportamento do estado de todos os neurônios depois de varias iterações pode ser visto na Fig. 3.30. Podemos ver que as variáveis x não mudam mais após algumas iterações. Isso de novo implica que os neurônios estão estabilizados.

Nesta figura, podemos observar os seguintes itens:

- Os estados de todos os neurônios chegam a convergir;

Figura 3.30: Comportamento do estado dos neurônios depois de 10 iterações.

- Existem neurônios, que tinham valores positivos (representa o cor preto), convergem para o lado positivo; também existem neurônios, que tinham valores negativos (representa o cor branco); convergem para o lado negativo;
- O neurônio $\{2, 2\}$ que começa com valor positivo (cor preto), convergem para o lado negativo (representa o cor branco).

Os seguintes *templates* também podem ser utilizados para extrair contornos:

$$B_{Roberts} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{pmatrix} \quad B_{Prewitt} = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad (32)$$

$$B_{Sobel} = \begin{pmatrix} -2 & -2 & 0 \\ -2 & 0 & 2 \\ 0 & 2 & 2 \end{pmatrix} \quad B_{laplaciano} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (33)$$

e para os quatro *templates* B acima, $A = 0$. Ou seja,

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad I = -1 \quad (34)$$

A Fig. 3.31 mostra os resultados de aplicação dos diferentes *templates* citados acima, para extrair contornos.

Figura 3.31: Extrair o contorno da imagem, (a) Imagem depois de aplicar o operador Roberts; (b) Imagem depois de aplicar o operador Sobel; (c) Imagem depois de aplicar o operador Prewitt; (d) Imagem depois de aplicar o operador Laplaciano.

3.4.4.3 Filtros

Filtro é uma técnica importante em processamento de sinais. Qualitativamente, um filtro é projetado para eliminar alguns componentes indesejáveis de um determinado sinal. Frequentemente, filtros são projetados para eliminar componentes em determinada faixa de frequência. Se um filtro elimina alta frequência de um sinal, é chamado um filtro passa baixo; se elimina baixa frequência, é chamado um filtro passa alta; se elimina tanto alta frequência quanto baixa frequência, este filtro é chamado passa banda.

Filtro passa baixo é utilizado para eliminar detalhes de uma imagem. Portanto, pode ser aplicado para remoção de ruídos ou suavização de imagens. Este tipo de filtro pode ser implementado por rede neural celular, configurando os *templates* A , B e I como:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad I = 0 \quad (35)$$

Um filtro passa alta é utilizado para destacar melhor os detalhes de uma imagem. Neste caso, os *templates* podem ser:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \frac{1}{9} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad I = 0 \quad (36)$$

Na Fig. 3.32 poderemos ver o resultado da rede neural celular usada como um filtro passa baixa e passa alta.

Figura 3.32: (a) Imagem original; (b) Filtro passa baixa; (c) Filtro passa alta.

3.4.4.4 Memória Associativa

Se expressamos a rede neural celular de tempo discreto de forma vetorial temos:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{y}(t) + \mathbf{I} \quad (37)$$

$$\mathbf{y}(t) = \mathbf{g}[\mathbf{x}(t)] \quad (38)$$

com $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ sendo o vetor de estados, $\mathbf{y} = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$ é o vetor de saída, $\mathbf{I} = [I_1, I_2, \dots, I_n]^T \in \mathbb{R}^n$ que corresponde a os *bias*, e $\mathbf{g} = [g, g, \dots, g]^T \in \mathbb{R}^n$, onde a função $g : \mathbb{R} \rightarrow \mathbb{R}$ definida por:

$$g(x) = \frac{1}{2} (|2x+1| - |2x-1|) \quad (39)$$

Sendo A a matriz esparsa $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n \times n}$ é a matriz de interconexão, a qual tem as propriedades de conexão local da arquitetura da rede neural celular.

Qualquer ponto $\mathbf{x}_0 \in \mathbb{R}^n$ é dito um ponto de equilíbrio da Eqn. 37 se:

$$\mathbf{x}_0 = \mathbf{A}\mathbf{g}(\mathbf{x}_0) + \mathbf{I} \quad (40)$$

Para armazenar padrões em pontos fixos, precisa-se encontrar uma matriz $\mathbf{A}$ adequada.

Se queremos armazenar m padrões na rede, $\mathbf{y}^i, i = 1, 2, \dots, m$, então, a Eqn. 37 deve ser satisfeita:

$$\mathbf{x}^i = \mathbf{A}\mathbf{y}^i + \mathbf{I}, \quad i = 1, 2, \dots, m \quad (41)$$

com $\mathbf{x}^i = [x_1^i, x_2^i, \dots, x_n^i] \in \mathbb{R}$ e $\mathbf{y}^i = [y_1^i, y_2^i, \dots, y_n^i] \in \mathbb{R}$.

Introduz-se as seguintes notações:

$$\begin{aligned}\mathbf{L}_m &= [\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^m] \\ \mathbf{H}_m &= [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m]\end{aligned}\quad (42)$$

A Eqn. 41 pode ser reescrito como:

$$\mathbf{H}_m = \mathbf{A}\mathbf{L}_m + \mathbf{I}' \quad (43)$$

com $\mathbf{I}' = [\mathbf{I}, \mathbf{I}, \dots, \mathbf{I}] \in \mathbb{R}^{n \times m}$.

O objetivo é determinar os valores de $\mathbf{A}$ e $\mathbf{I}$, de forma que a Eqn. 43 seja satisfeita. para atingir esse propósito serão definidas as seguintes matrizes:

$$\begin{aligned}\mathbf{R}_m &= [\mathbf{L}_m^T | \mathbf{J}] \in \mathbb{R}^{m \times (n+1)} \\ \mathbf{W}_{m,j} &= [A_{j1}, A_{j2}, \dots, A_{jn} | I_j] \in \mathbb{R}^{1 \times (n+1)} \\ \mathbf{H}_{m,j} &= [x_j^1, x_j^2, \dots, x_j^m] \in \mathbb{R}^{1 \times m} \quad j = 1, 2, \dots, n \\ \mathbf{S} &= [S_{i,j}] \in \mathbb{R}^{n \times n}\end{aligned}\quad (44)$$

onde $\mathbf{J} = [1, 1, \dots, 1]^T \in \mathbb{R}^{m \times 1}$, $S_{i,j} = 1$ se a j -ésima célula pertence à vizinhança da i -ésima célula e $S_{i,j} = 0$ para qualquer outro caso ($i = 1, 2, \dots, n; j = 1, 2, \dots, n$).

A Eqn. 43 pode ser escrita como:

$$\mathbf{R}_m \mathbf{W}_{m,j}^T = \mathbf{H}_{m,j}^T \quad j = 1, 2, \dots, n \quad (45)$$

Agora, a matriz $\mathbf{R}_{m,j}$, $j = 1, 2, \dots, n$, pode ser obtida da matriz $\mathbf{R}_m$ por eliminar aquelas colunas, cujo índice corresponde aos elementos de valor 0 na j -ésima linha de $\mathbf{S}$. Além disso, um vetor $\tilde{\mathbf{W}}_{m,j}$ pode ser definido como o vetor obtido de $\mathbf{W}_{m,j}$ eliminando seus elementos com valor 0. Assim temos que:

$$\mathbf{R}_{m,j} \tilde{\mathbf{W}}_{m,j}^T = \mathbf{H}_{m,j}^T \quad j = 1, 2, \dots, n \quad (46)$$

$$\tilde{\mathbf{W}}_{m,j}^T = \mathbf{R}_{m,j}^+ \mathbf{H}_{m,j}^T \quad j = 1, 2, \dots, n \quad (47)$$

onde $\mathbf{R}_{m,j}^+$ denota a matriz pseudo-inversa de $\mathbf{R}_{m,j}$. Finalmente, o procedimento é completado expandindo o vetor $\tilde{\mathbf{W}}_{m,j}^T$ com zeros até que o vetor $\mathbf{W}_{m,j}^T$ seja obtido. Assim podem ser armazenados padrões na rede neural celular.

A Fig. 3.33 apresenta 5 letras de tamanho 5×5 armazenadas na rede.

Figura 3.33: Padrões armazenados.

A Fig. 3.34 apresenta o processo de reconhecimento, oferecendo um padrão com ruído como condição inicial à rede (primeira imagem da Fig. 3.34). Assim, o sistema é iterado, até que a rede reconhece o padrão da letra A, como pode ser vista na última imagem da Fig. 3.34, para este exemplo foi usado uma vizinhança de raio 2.

Figura 3.34: Processo de reconhecimento da rede neural celular.

3.5 Considerações Finais

Neste capítulo foram apresentados conceitos básicos das redes neurais como: estrutura do neurônio, arquitetura e alguns algoritmos de aprendizado. Também foram descritos dois modelos de redes neurais, o modelo de Hopfield e rede neural celular.

Para o modelo de Hopfield foi revizado o formalismo matemático e descrição das principais características, assim como uma análise qualitativa do funcionamento e da estabilidade da rede, mostrando as vantagens e limitações quando é usado como memória associativa. Mostrando uma aplicação em reconhecimento de faces.

Em rede neural celular também foi feita uma descrição do funcionamento assim como das principais características. Além disso, algumas aplicações de processamento de imagens tais como: remoção de ruído, extração de contornos, filtros passa baixa e passa alta foram apresen-

tadas. Finalmente foi mostrado como a rede neural celular pode ser utilizada como memória associativa.

Os dois tipos de redes neurais revizadas neste capítulo são também sistemas dinâmicos não-lineares e de alta dimensão. Portanto, o entendimento destas redes neurais são fundamentais para desenvolver redes neurais caóticas, as quais são o principal tópico do próximo capítulo.

Memórias Associativas Caóticas

4.1 Considerações Iniciais

As redes neurais caóticas possuem características e comportamentos diferentes quando comparadas com os modelos tradicionais, como por exemplo o modelo de Hopfield. Sendo o *caos*, uma diferença importante no comportamento de tais redes e a razão pela qual são chamadas de **redes neurais caóticas**.

Neste capítulo são descritas as redes neurais caóticas mais representativas, tais como o modelo Aihara, clusters dinâmicos para o reconhecimento de padrões e a rede de mapa senoidal parametricamente acoplado.

O modelo de Adachi e Aihara (M. Adachi, 1997), pode ser considerada como uma extensão do modelo de Hopfield, utilizando o neurônio mais complexo em lugar do neurônio de McCulloch-Pitts. A dinâmica dos outros dois modelos utilizam diretamente as propriedades e características do caos.

4.2 Rede Neural Caótica de Adachi e Aihara

Esta é uma rede desenvolvida por Adachi e Aihara (M. Adachi, 1997) usada como uma memória associativa com neurônios caóticos interligados. No processo de aprendizagem utilizou uma

matriz auto-associativa de pesos, onde a dinâmica associativa da rede é analisada: no espaço temporal, na função de quase-energia, na distância entre os vetores dos estados internos e na instabilidade orbital.

4.2.1 O Neurônio Caótico

O modelo do neurônio é a modificação e extensão do modelo de tempo discreto proposto por Aihara (Aihara, 1990), sendo que a equação do neurônio caótico é dada por:

$$x(t+1) = f\left(A(t) - \alpha \sum_{d=0}^t k^d g(x(t-d)) - \Theta\right) \quad (1)$$

onde: $\alpha \sum_{d=0}^t k^d g(x(t-d))$ pode ser entendido como o mecanismo de auto-inibição do neurônio. No modelo de McCulloch-Pitts, o próximo estado depende somente do estado atual. Neste modelo de neurônio, o mecanismo de auto-inibição representa influencia não só do estado atual, mas também dos estados anteriores, sendo os estados mais recentes que tem maior influencia. Esta característica de neurônio biológico é representada pelo termo $\alpha \sum_{d=0}^t k^d g(x(t-d))$, sendo $||k|| < 1$ o parâmetro de decaimento e α é o parâmetro de escala para poder ajustar o efeito de auto-inibição. Nesta equação, g é uma função aplicada aos estados anteriores. $x(t)$ é a saída do neurônio com um valor analógico entre 0 e 1. t representa o tempo discreto ($t = 0, 1, 2, \dots$), f é a função de ativação do neurônio, $A(t)$ é a entrada externa na iteração t e Θ é um limiar ou *threshold*.

A Eqn. 1 também pode ser escrita pela seguinte forma:

$$x(t+1) = f(y(t+1)) \quad (2)$$

$$y(t+1) = A(t) - \sum_{d=0}^t k^d g(x(t-d)) - \Theta \quad (3)$$

onde a nova variável $y(t)$ representa o estado interno do neurônio. A Eqn. 3 pode ser definida em função do estado interno y pela seguinte equação (Aihara, 1990):

$$y(t+1) = ky(t) - \alpha g(f(y(t))) + \theta(t) \quad (4)$$

com $\theta \equiv A(t) - kA(t) - \Theta(1-k)$. E a função de saída é definida por:

$$f(y) = \frac{1}{1 + e^{-\left(\frac{y}{\tau}\right)}} \quad (5)$$

Para simplificar a análise do modelo, pode-se assumir que a função $g(x) = x$ e também o parâmetro $\theta(t) = a$ seja um constante.

Este modelo apresenta diferentes comportamentos tais como: ponto fixo, oscilação periódica e comportamento caótico, dependendo dos valores dos parâmetros. A Fig. 4.1 mostra um ploto *cobweb* para solução da Eqn. 2, na qual pode-se ver que as trajetórias não são fechadas, indicando o comportamento caótico.

Figura 4.1: Amostras de soluções da Eqn. 2, com valores de $k = 0.50$, $\epsilon = 0.015$, $a = 0.50$ e $\alpha = 1.00$

A Fig. 4.2 mostra o diagrama de bifurcação da Eqn. 1 no qual pode ser vista as regiões caóticas e as regiões periódicas do neurônio, dependendo dos valores do parâmetro a .

Figura 4.2: Bifurcação da Eqn. 2 variando o parâmetro a , com $k = 0.51$, $\epsilon = 0.015$ e $\alpha = 1.00$

4.2.2 A Rede Caótica

O neurônio caótico pode ser usado como elemento para construir uma rede neural caótica. A rede neural caótica, que consiste de um conjunto de tais neurônios caóticos integrados, pode ser descrita pela seguinte equação (M. Adachi, 1997):

$$x(t+1) = f\left(\sum_{j=1}^M v_{ij} \sum_{d=0}^t k_e^d A_j(t-d) + \sum_{j=1}^N w_{ij} \sum_{d=0}^t k_f^d x_j(t-d) - \alpha \sum_{d=0}^t k_r^d g x_i(t-d) - \Theta\right) \quad (6)$$

- v_{ij} são os pesos entre o neurônio i e a entrada externa j ;
- w_{ij} são os pesos entre o neurônio i e o neurônio j ;
- k_e é o parâmetro de decaimento para as entradas externas,
- k_f é o parâmetro de retroalimentação das entradas,
- e k_r é o parâmetro de auto-inibição.

Da Eqn. 6 pode se dizer que:

O primeiro somatório representa influência das entradas externas a cada neurônio da rede, ou seja, como o mundo externo influencia a rede; o segundo somatório, representa as conexões sinápticas entre os neurônios; o terceiro somatório corresponde ao efeito de auto-inibição, ou seja, como cada neurônio é influenciado por si mesmo. Em resumo, uma rede, que possui somente o primeiro e o segundo somatório, é considerada uma rede neural convencional. A novidade deste modelo é a inclusão do terceiro somatório, o qual possibilita que cada neurônio pode ser caótico.

Considerando os três somatórios como o estado interno do neurônio, a Eqn. 6 pode ser reescrita em seguinte forma padrão de redes neurais:

$$\xi_i(t+1) = \sum_{j=1}^M v_{ij} A_j(t) + k_e \xi_i(t) \quad (7)$$

$$\eta_i(t+1) = \sum_{j=1}^N w_{ij} x_j(t) + k_f \eta_i(t) \quad (8)$$

$$\begin{aligned} \zeta_i(t+1) &= -\alpha g(x(t)) + k_r \zeta_i(t) - \theta_i, \\ \theta_i &\equiv \Theta_i(1 - k_r) \end{aligned} \quad (9)$$

$$x_i(t+1) = f(\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)) \quad (10)$$

Assumindo que o estímulo externo é constante, o termo $\xi_i(t)$ é substituído por uma constante a_i . Assim o somatório $\xi_i(t+1) = \sum_{j=1}^M v_{ij} A_j(t) + k_e \xi_i(t)$ é eliminado e o modelo é reduzido para a seguinte forma:

$$x_i(t+1) = f\left(\eta_i(t+1) + \zeta_i(t+1)\right) \quad (11)$$

$$\eta_i(t+1) = k_f \eta_i(t) + \sum_{j=1}^N w_{ij} x_j t \quad (12)$$

$$\zeta_i(t+1) = k_r \zeta_i(t) - \alpha x_i(t) + a_i \quad (13)$$

onde a_i corresponde tanto à soma do *threshold* quanto as entradas externas para o i -ésimo neurônio.

4.2.3 Experimentos por Computador

Nesta subseção são apresentados resultados de experimentos por computador da rede neural caótica de Adachi e Aihara. Primeiro, apresentamos um exemplo de reconhecimento de padrões, ilustrando o comportamento de convergência pontual, periódico e caótico da rede em diferentes regiões de parâmetros. Depois, analisamos dinâmica tanto em nível da rede quanto em nível de neurônios. A Fig. 4.3 mostra 4 padrões binários armazenados em rede de Adachi e Aihara utilizando a seguinte regra de aprendizado:

$$w_{ij} = \frac{1}{M} \sum_{p=1}^N x_i^p x_j^p \quad (14)$$

onde M corresponde ao número de padrões armazenados e N ao número de neurônios.

Figura 4.3: Padrões utilizados para a simulação do modelo apresentado.

Quando os parâmetros α , k_r , k_f e a tem o valor de zero, o modelo comporta como o modelo de Hopfield. Neste caso, apresentando um padrão, ele converge para um ponto fixo. A Fig.

4.4 mostra o processo de convergência para um ponto fixo, quando um padrão com ruído é apresentado como condição inicial na rede.

Figura 4.4: Sequência do espaço temporal, quando $\alpha = k_r = k_f = a = 0.00$.

Aumentando o valor de k_r , a rede apresenta um curioso comportamento dinâmico. Quando k_r é relativamente pequeno, a rede pode entrar em ciclo limite, oscilando por volta do padrão armazenado, como pode ser apreciado na Fig. 4.5.

Figura 4.5: Uma amostra da sequência do espaço temporal, com $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$.

Quando os parâmetros são modificados de forma apropriada, a rede pode apresentar um comportamento complicado e irregular, como pode ser visto na Fig. 4.6,

Figura 4.6: Uma amostra da sequência espaço-temporal, com $k_f = 0.20$, $\alpha = 10.00$ e $k_r = 0.90$ e $a = 2.00$.

Três tipos de comportamento foram vistos nas Figs. 4.4, 4.5 e 4.6. A primeira trata-se de ponto fixo, onde uma vez que ele é atingido o estado não muda mais. O segundo é chamado de ciclo limite onde a trajetória fica perto da solução. Finalmente no ultimo, a trajetória da rede é diferente para cada iteração, percorrendo todos os estados espaço-temporais. Este fenômeno foi identificado como uma dinâmica caótica (M. Adachi, 1997), portanto, é chamada de *itinerária caótica*.

Para caracterizar as dinâmicas da rede, plotamos três tipos de figuras: séries temporais de alguns neurônios da rede, distância de Hamming entre a saída da rede de cada iteração e cada padrão armazenado e a função chamada *quase energia* para os diferentes comportamentos, ponto fixo, oscilação periódica e itinerária caótica.

Segundo (M. Adachi, 1997), a função de quase energia é definida pela seguinte equação:

$$QE(t) = \frac{1}{2} \sum_{i \neq j} \sum_j w_{ij} x_i(t) x_j(t) - \sum_i a_i x_i(t). \quad (15)$$

Quando a rede converge para um ponto fixo, as séries temporais de alguns elementos da rede podem ser visto na Fig. 4.7. Através desta figura, podemos observar que o estado interno dos elementos estabilizam, alguns estão no lado positivo (interpretado como 1) e outros estão no lado negativo (interpretado como -1).

Figura 4.7: Série temporal de alguns elementos quando $k_f = k_r = \alpha = a = 0.00$

Na Fig. 4.8, pode ser vista a distância de Hamming entre a saída da rede e o padrão 1 cai para zero. A distância de Hamming entre a saída da rede e outros padrões estabilizam em

valores diferente de zero. Isso significa que a rede converge para um ponto fixo e o padrão 1 é reconhecido.

Figura 4.8: Distância de Hamming quando $k_f = k_r = \alpha = a = 0.00$

Na Fig. 4.9, pode ser visto a evolução da quase energia da rede, podendo se observar que ela cai e fica estável quando o padrão é reconhecido.

Figura 4.9: Quase energia quando $k_f = k_r = \alpha = a = 0.00$

Quando o comportamento da rede é periódico, as séries temporais são mostradas na Fig. 4.10. Podemos observar que os estados internos dos elementos não permanecem em um único ponto mas oscilaram entre alguns pontos, correspondendo ao comportamento periódico.

Figura 4.10: Série temporal de alguns elementos quando $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$.

Na Fig. 4.11, pode ser visto que a distância de Hamming também é oscilatória e a distancia entre a saída e o padrão 3 é menor do que outras. Podemos perceber que a solução oscila próximo do padrão armazenado.

Figura 4.11: Distância de Hamming quando $k_f = 0.15$, $k_r = 0.50$, $\alpha = 8.00$ e $a = 1.75$.

Na Fig. 4.12, pode ser visto a evolução da quase energia da rede, podendo observar que também é oscilatória.

Figura 4.12: Quase energia quando $k_f = 0.15$, $\alpha = 8.00$, $k_r = 0.50$ e $a = 1.75$.

Finalmente, quando o comportamento é uma itinerária caótica, as séries temporais dos elementos da rede são altamente irregulares, indicando um comportamento caótico. Isso pode ser visto na Fig. 4.13.

Figura 4.13: Série temporal de alguns elementos quando $k_f = 0.20$, $\alpha = 10.00$ e $k_r = 0.90$ e $a = 2.00$.

Nas Fig. 4.14 e Fig. 4.15, podemos ver que a distância de Hamming e a função de quase energia também são irregulares.

Figura 4.14: Distância de Hamming quando $k_f = 0.20$, $\alpha = 10.00$, $k_r = 0.90$ e $a = 2.00$.

4.2.4 Observações

- O espaço de fase desse tipo de rede pode ser dividido em dois subespaços: caótico e não caótico.
 1. Sub-espaço não caótico representa o armazenamento dos padrões.
 2. Sub-espaço caótico representa um estado de busca. Neste caso, a trajetória caótica visita quase todos os padrões armazenados. Este fenômeno é chamado de itinerária caótica (M. Adachi, 1997).

4.3 Cluster Dinâmica Para Reconhecimento de Padrões

Nesta seção, é apresentada uma memória associativa baseada em cluster dinâmica de um tipo de sistemas dinâmicos espaço-temporal chamado *mapa globalmente acoplado*.

4.3.1 Mapas Globalmente Acoplados

Mapa globalmente acoplado *GCM* é uma rede formada de elementos caóticos. Kaneko é um pesquisador que primeiro estudou esse tipo de sistema e descobriu muitas características interessantes (Kaneko, 1990). Uma dessas características mais salientes é chamado clusterização, que será explicado adiante. Em forma geral, um GCM pode ser definido pelas seguintes equações:

Figura 4.15: Quase energia quando $k_f = 0.20$, $\alpha = 10.00$, $k_r = 0.90$ e $a = 2.00$.

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{N} \sum_{j=1}^N f(x_j(t)) \quad (16)$$

$$f(x) = 1 - \alpha x^2 \quad x \in [-1, 1], \quad (17)$$

onde $x_i(t)$ denota o estado do i -ésimo elemento da rede na iteração t . N é o número de elementos da rede. Este modelo pode ser visto como uma rede de N mapas quadráticos acoplados, sendo α o parâmetro de bifurcação de cada elemento e ϵ o peso de acoplamento.

Ishii propõe um outro GCM, chamado de *mapa simétrico globalmente acoplado (S-GCM)* (S. Ishii, 1996), substituindo o mapa quadrático por um mapa cúbico, que é representado pela seguinte equação:

$$f(x) = \alpha x^3 - \alpha x + x, \quad x \in [-1, 1], \quad (18)$$

A Fig. 4.16 mostra a diferença entre a função quadrática e a função cúbica quando $\alpha < 2$. Podemos ver que só existe um ramo na função quadrática, mas existem dois ramos na função cúbica, os quais são simétricos em relação ao diagonal. Portanto, o mapa cúbico permite órbitas periódicas, nas quais os pontos periódicos são simetricamente distribuídos em relação ao ponto fixo.

A Fig. 4.17 mostra os diagramas de bifurcação para mapa quadrático e cúbico, respectivamente, podendo apreciar a simetria do mapa cúbico. Esta característica de simetria é importante

Figura 4.16: (a) Mapa logístico com $\alpha = 2.00$ com só um ponto extremo; (b) Mapa cúbico com $\alpha = 4.00$, quando $\alpha > 2.0$ o mapa apresenta dois extremos.

para reconhecimento de padrões no sentido se os pontos periódicos de um lado do ponto fixo são utilizados para representar o valor 1, os do outro lado do ponto fixo são naturalmente utilizados para representar o valor 0 (ou -1).

Figura 4.17: (a) Bifurcação do mapa quadrático com $1.00 < \alpha < 2.00$; (b) Bifurcação do Mapa cúbico com $2.50 < \alpha < 4.00$.

A seguir, apresentamos um diagrama de fases do S-GCM, variando o parâmetro α e ϵ , onde α é o parâmetro de bifurcação de cada mapa da rede, que corresponde ao grau de não linearidade do mapa e ϵ é o peso de acoplamento. Na diagrama de fases (Fig. 4.18), pode ser visto vários tipos de dinâmicas: fase coerente, fase ordenada, fase parcialmente ordenada e fase de turbulência.

Figura 4.18: Diagrama de fase de mapa globalmente acoplado com variações dos parâmetros α de 3.00 a 4.00 e ϵ que varia de 0.00 a 0.40 (S. Ishii, 1996).

1. **Fase coerente**, todos os elementos estão movendo-se para uma mesma trajetória, indicando um comportamento de sincronização completa entre os elementos. Este ocorre quando o peso de acoplamento é forte (ϵ é grande) e a não linearidade do mapa é baixa (α é pequeno) como pode ser visto na Fig. 4.19;

Figura 4.19: Exemplo da fase coerente com $\alpha = 2.60$ e $\epsilon = 0.60$.

2. **Fase ordenada**, mesmo inicializado com valores aleatórios, os elementos da rede são distribuídos em um número pequeno de grupos. Elementos do mesmo grupo têm o mesmo comportamento, ou seja, sincronizados. Tais grupos são chamados clusters. O número de clusters formados nesta fase é independente das condições iniciais e do número de elementos da rede.

A Fig. 4.20 mostra o comportamento do modelo na fase ordenada 2, podendo-se ver que depois de algumas iterações o estado dos neurônios forma duas trajetórias distintas. Cada uma é chamada um *cluster*;

Figura 4.20: Exemplo da formação de dois clusters quando a rede se encontra na fase ordenada (2), com $\alpha = 3.20$ e $\epsilon = 0.20$.

A Fig. 4.21 mostra o comportamento do modelo na fase ordenada 4 plotada em todas as iterações. A Fig. 4.22 mostra o mesma simulação mas plotada cada duas iterações ($t = \{2, 4, 6, \dots\}$), podendo observar-se os 4 clusters formados.

Figura 4.21: Exemplo com quatro clusters quando $\alpha = 3.20$, $\epsilon = 0.02$ e $N = 100$ e qual corresponde ao número de elementos utilizados.

Ainda que os valores dos parâmetros sejam fixados, os números de clusters podem variar, por exemplo, quando $\alpha = 3.96$ e $\epsilon = 0.25$, $x(t)$ assume valores 5,6, etc dependendo dos estados iniciais dos elementos.

3. **Fase parcialmente ordenada (intermitência e *glassy*)**, as trajetórias podem cair em um grande número clusters em alguns casos, ou em um número pequeno de clusters, dependendo das condições iniciais e do número de elementos da rede;
4. **Fase de turbulência**, acontece quando o peso de acoplamento é fraco (ϵ é pequeno) e o parâmetro de bifurcação α é grande. Neste caso, para a grande maioria dos elementos, cada um tem sua própria trajetória e o número de clusters aproxima de N .

Figura 4.22: Exemplo com valores pares de tempo $t = 2, 4, 6, \dots$ e $\alpha = 3.40$, $\epsilon = 0.10$ e $N = 100$.

A Fig. 4.23 mostra o comportamento da rede, quando está em estado de turbulência.

Figura 4.23: Exemplo da fase de turbulência do rede quando $\alpha = 4.00$ e $\epsilon = 0.01$.

4.3.2 Aplicações para Reconhecimento de Padrões

O S-GCM pode ser usado em uma memória associativa, na qual padrões binários podem ser armazenados. Na fase de armazenamento, o algoritmo de aprendizado Hebbiano é utilizado. Na fase de reconhecimento, padrões de entrada (padrões a ser conhecidos) precisam ser adaptados para S-GCM e clusters formados do S-GCM precisam ser interpretados como valor binário. Portanto, um processo completo de fase de reconhecimento é representado pelo seguinte processo:

$$I \rightarrow^V x(0) \rightarrow^{S-GCM} x(T) \rightarrow^C O. \quad (19)$$

Figura 4.26: Valores do parâmetro α da aplicação de reconhecimento de padrão para cada instante de tempo $t = \{1, 2, \dots\}$.

4.3.3 Observações

- O sistema apresenta dois estados: estado de cluster e o estado de turbulência.
 1. Estado de cluster representa os padrões armazenados;
 2. Estado de turbulência que permite aos elementos auto organizarem-se e trocar de grupos para minimizar a energia;
- A capacidade de armazenamento é mais alta que Hopfield (S. Ishii, 1996);
- Este esquema é mais difícil de cair em mínimo local, pois a minimização de energia é feita em cada elemento e não como a média global.

4.4 Rede de Mapa Senoidal Parametricamente Acoplada

Através do modelo apresentado acima, podemos observar que a dinâmica simétrica é desejável para processamento de padrões binários. Devido às propriedades da função trigonométrica, o diagrama de bifurcação do mapa senoidal é simétrica sobre zero para todas as condições iniciais. Portanto, o mapa senoidal é um bom candidato a ser utilizado para reconhecimento de padrões binários.

4.4.1 O Mapa Senoidal

O mapa senoidal é representado pela seguinte equação (Eqn. 26):

$$x(t+1) = \eta \sin(\pi x(t)) \quad (26)$$

onde t é a índice de iterações e η é a amplitude da função senoidal e também o parâmetro de bifurcação do mapa.

O diagrama de bifurcação do mapa senoidal é mostrado na Fig. 4.27, no qual dois diferentes ramos são apresentados os quais a dois diferentes condições iniciais: $x(0) > 0$ e $x(0) < 0$.

Figura 4.27: Dois diagramas de bifurcação correspondentes a diferentes estados iniciais (a) Quando o estado inicial é um valor positivo $x(0) = 0.1$; (b) Quando o valor do estado inicial é negativo $x(0) = -0.1$.

Fazendo um análise ao diagrama de bifurcação, podemos ver que quando $0 \leq \eta < 1/\pi$, o mapa converge no ponto fixo $x = 0$. Para $\eta = 1/\pi$ o mapa senoidal experimenta uma bifurcação decrescente simétrica (Hilborn, 1994), e o valor do ponto fixo para $x = 0$ repentinamente pode chegar a ser instável e dar origem a dois pontos fixos, os quais são simétricos com respeito a sua origem e esses podem dar origem a novas bifurcações e terminar em atratores caóticos (Hilborn, 1994). Quando o valor $\eta = 1$ é chamado de ponto crítico η_c .

4.4.2 Modelo de Memória Associativa Utilizando Mapa Senoidal

Aproveitando a bifurcação de duplicação de período e as propriedades do *attractor merging* crise, Lee and Farhat (G. Lee, 2001) propõem um modelo de memória associativa. Na fase de armazenamento o algoritmo de aprendizado Hebbiano é utilizado novamente. O modelo para reconhecimento de padrões binários é dado pelas seguintes equações:

$$x_i(t+1) = \eta_i(t) \sin(\pi x_i(t)) \quad (27)$$

$$\eta_i(t) = \eta_0 - g \left(\frac{\alpha x_i(t)}{N} \sum_{j=1}^N w_{ij} x_j(t) \right) \quad (28)$$

$$g(x) = \begin{cases} -\epsilon & x < -\epsilon \\ x & -\epsilon \geq x \geq \epsilon \\ \epsilon & x > \epsilon \end{cases} \quad (29)$$

onde $x_i(t)$ é a variável de estado do i -ésimo elemento na rede, para $i = 1, 2, \dots, N$. Pode-se notar que nesta equação que o parâmetro de bifurcação η virou uma variável diferente para cada elemento sendo denotada por $\eta_i(t)$ é governada pela Eqn. 28. η_0 é uma constante que corresponde a um ponto muito próximo e antes do ponto de crise. α é uma constante positiva. w_{ij} é a matriz de pesos obtida na fase de armazenamento. g é uma função que modifica o valor do parâmetro η_i e está dada pela Eqn. 29, onde ϵ é uma constante real positiva de valor pequeno.

O termo $\frac{x_i(t)}{N} \sum_{j=1}^N w_{ij} x_j(t)$ da Eqn. 28 pode ser considerado como a energia parcial e equivale à função da energia de Hopfield.

Esta idéia pode ser utilizada para reconhecimento de padrões como pode ser visto na Fig. 4.28 a qual mostra a reconstrução de um padrão parcial, com valores de $\eta_0 = 0.97$ $\alpha = 1.00$ e $\epsilon = 0.14$. Para o treinamento foram utilizados os mesmos padrões mostrados na Fig. 4.3.

Figura 4.28: Exemplo de reconhecimento de padrão quando o valor de $\eta = 0.97$ $\alpha = 1.00$ e $\epsilon = 0.14$.

A Fig. 4.29 corresponde ao valor de energia de cada neurônio em cada instante de tempo, podemos ver que ela sempre fica oscilando.

A Fig. 4.30 mostra a evolução da variável de bifurcação η em cada instante de tempo t .

4.4.3 Observações

- A idéia para reconhecimento de padrão é similar com o modelo anterior (S-GCM). Do ponto de vista de memória associativa, o sistema é dividido em dois estados: antes da crise e depois da crise. O primeiro representa o estado de armazenamento de padrões. O

Figura 4.29: Valores da energia obtido pelo reconhecimento do padrão anterior, para cada instante de tempo $t = \{1, 2, \dots\}$.

Figura 4.30: Valores do parâmetro η obtido para cada instante de tempo $t = \{1, 2, \dots\}$ no reconhecimento do padrão anterior.

último é um estado de busca, onde os elementos podem trocar grupos para minimizar a energia;

- Este modelo possui a vantagem da dinâmica ser mais local. Isso é verdade porque a bifurcação e crise são dinâmicas de cada elemento da rede. Por outro lado, os clusters no modelo S-GCM são dinâmicas coletivas da rede.

4.5 Considerações Finais

Nesta seção foi realizada uma revisão das características de três redes neurais caóticas representativas, analisando os diferentes comportamentos e mostrando as vantagens e desvantagens de cada modelo.

No modelo de Aihara, uma descrição das características do neurônio caótico e da rede caótica foram feitas, também um análises dos comportamentos de ponto fixo, ciclo limite e itinerária caótica, mostrando as principais diferenças entre tais comportamentos.

No modelo de clusters dinâmicos para o reconhecimento de padrões, uma descrição das características dos mapas globalmente acoplados e a forma como pode ser usado para processamento de informação foi visto.

Finalmente as características de bifurcação de duplicação de período e crise do mapa senooidal parametricamente acoplado foram descritos.

As características e conceitos destas três redes são importantes no desenvolvimento dos modelos propostos no seguinte capítulo.

Reconhecimento de Padrão Multi-valorados

5.1 Considerações Iniciais

No capítulo 3 foram revisados dois tipos de redes neurais: o modelo de Hopfield e Rede Neural Celular. No capítulo 4, foram apresentados vários modelos de redes neurais caóticas. Todos esses modelos de redes neurais têm pelo menos três aspectos em comum: 1) são sistemas dinâmicos; 2) podem ser usados como memórias associativas para reconhecimento de padrões e 3) só armazenam e reconhecem padrões binários.

O último item acima apresenta uma limitação desses modelos em aplicações reais. Neste capítulo, serão propostos dois modelos de redes neurais caóticas para reconhecimento de padrões não só binários, mas também multi-valorados.

O primeiro modelo utiliza a dinâmica do mapa senoidal, a qual é inspirado na Rede de Mapa Senoidal Parametricamente Acoplado revisada no capítulo anterior. O segundo modelo é uma simplificação e melhoramento do primeiro, aproveitando a dinâmica periódica e caótica para reconhecimento de padrões multi-valorados. Uma característica do segundo modelo é que ele não depende de um mapa específico, só do tipo da dinâmica. Portanto, muitos mapas caóticos podem ser usados como elementos da rede. Neste documento, para ilustrar o funcionamento do modelo, utilizaremos o mapa logístico como exemplo.

5.2 Descrição do Modelo I

O funcionamento deste modelo é dividido em duas etapas: etapa de aprendizado e etapa de reconhecimento. Na etapa de aprendizado, o algoritmo de aprendizado por matriz pseudo-inversa é usado para armazenar um conjunto de padrões em pontos fixos. Na etapa de reconhecimento, a dinâmica caótica e periódica do mapa senoidal é utilizada.

5.2.1 A Etapa de Aprendizado

Para qualquer matriz Q de $m \times n$ com $m \geq n$, a matriz pseudo-inversa Q^+ é definida por $Q^+ = (Q^T Q)^{-1} Q^T$. Sendo $Q^T Q$ não singular e $Q^+ Q = I$, onde I é a matriz identidade. Mas, $Q Q^+ \neq I$ não é sempre verdadeira.

Para armazenar os M padrões em pontos fixos, a seguinte condição deve ser satisfeita:

$$WP = P \quad (1)$$

onde W é a matriz de conexão. Então, W pode ser determinada pelo algoritmo de aprendizado por matriz pseudo-inversa (J. Hertz, 1991).

$$W = PP^+ = P(P^T P)^{-1} P^T \quad (2)$$

O algoritmo de aprendizado por matriz pseudo-inversa é uma generalização e melhoramento do algoritmo Hebbiano usado no modelo de Hopfield em dois aspectos: 1) esta regra de aprendizado pode ser usada para armazenar $M < N$ padrões linearmente independentes (N é o número de neurônios da rede), mas o modelo de Hopfield só pode armazenar por volta de $0.14N$ padrões; 2) este algoritmo pode armazenar diretamente padrões V -ários (V são os possíveis valores que pode ter um atributo do padrão), enquanto que o algoritmo de aprendizado Hebbiano somente pode armazenar padrões binários, quando usado no modelo de Hopfield (J. Hertz, 1991).

5.2.2 A Etapa de Reconhecimento

O processo de reconhecimento utiliza a dinâmica do mapa senoidal, aproveitando as características de bifurcação de duplicação de período e crise. Especificamente, este modelo é governado pelas seguinte equações:

$$x_i(t+1) = n_i(\tau) \sin(\pi x_i(n)) \quad (3)$$

$$n_i(\tau) = \begin{cases} n_0 - g\left(\left|x_i(t) - \sum_{j=1}^N w_{ij}x_j(t)\right|\right) & \text{se } S\left(x_i(t), \sum_{j=1}^N w_{ij}x_j(t)\right) = 1 \\ n_c & \text{se } S\left(x_i(t), \sum_{j=1}^N w_{ij}x_j(t)\right) = -1 \end{cases} \quad (4)$$

$$g(x) = \theta * e^{(-x^2/\alpha)} \quad (5)$$

$$S(y, z) = \begin{cases} 1 & \text{se } y \geq z \\ -1 & \text{caso contrário} \end{cases} \quad (6)$$

Onde $x_i(t)$ é a variável de estado do i -ésimo elemento na rede, para $i = 1, 2, \dots, N$. n_0 é um constante que corresponde a um ponto muito próximo e antes do ponto de crise. n_c também é uma constante que corresponde a um ponto após o ponto de crise. $n_i(\tau)$ é o parâmetro de bifurcação do i -ésimo mapa senoidal, que pode tomar um dos seguintes valores: $n_0 - g(|x_i(t) - \sum_{j=1}^N w_{ij}x_j(t)|)$ ou n_c . Se o primeiro caso ocorre, o comportamento do i -ésimo elemento se move para órbita periódica. Se o segundo caso ocorre, o elemento i se mantém em estado caótico, com n_i localizado após o ponto de crise, permitindo ao elemento percorrer o atrator caótico e mudar seu sinal, até que se aproxime do estado correto. Conseqüentemente, a primeira condição da Eqn. 4 será satisfeita, e o elemento será dirigido para região periódica. A variável x é avaliada em cada iteração t , enquanto que a variável n é avaliada a cada $\tau = 2$ iterações. A função Gaussiana na Eqn. 5 determina o passo de avanço do elemento para a órbita periódica. O parâmetro θ representa a amplitude máxima da função Gaussiana, e α determina sua inclinação. A Fig. 5.1, mostra o comportamento da função Gaussiana para diferentes valores de θ e α .

Figura 5.1: (a) Comportamento da função Gaussiana variando valor de α , $\theta = 1$; (b) Comportamento da função Gaussiana variando valor de θ , $\alpha = 0.5$

Deve-se considerar que o valor do parâmetro n_i tem um valor mínimo n_{min} pré-definido, i.e., se $n_i < n_{min}$ então $n_i = n_{min}$, caso contrário seu valor é modificado de acordo as equações acima. Este valor mínimo n_{min} corresponde a uma dinâmica de ciclo de 2 períodos. Considerando o sinal da condição inicial, um mapa senoidal, que está com parâmetro n_{min} , pode cair em um dos dois ciclos de período 2, onde cada ponto periódico é usado para representar um dos quatro possíveis valores dos padrões. Isso significa que um padrão é reconhecido quando todos os mapas estão com parâmetro n_{min} . A Fig. 5.2 corresponde à sobreposição dos dois diagramas de bifurcação do mapa senoidal, para as condições iniciais positiva e negativa. A mesma figura mostra um n_{min} selecionado, onde a dinâmica do mapa pode cair no ciclo pp_1, pp_2 ou no ciclo pp_3, pp_4 .

Figura 5.2: Escolha de n_{min} determinando os pontos pp_1, pp_2, pp_3 e pp_4 do mapa senoidal, a serem armazenados, assim como a determinação do ponto n_c , que corresponde a uma órbita caótica depois do ponto de crise.

O funcionamento do modelo para reconhecimento de padrões é descrito a seguir: primeiramente, um conjunto de padrões são armazenados na memória, usando o algoritmo de aprendizado descrito anteriormente pela Eqn. 2. Na etapa de reconhecimento, selecionamos o valor de n_{min} que corresponde a um ciclo de período 2, e n_c corresponde a uma órbita caótica. Inicialmente, $n_i = n_c$ para todos i , i.e., todos os mapas são caóticos. Quando um padrão a ser reconhecido (denotado por *padrão de entrada*) é apresentado à rede como condição inicial, e devido à propriedade densa do caos, cada elemento pode se aproximar de um estado no qual $x_i(t) - \sum_{j=1}^N w_{ij}x_j(t)$ é pequeno. Isso significa que o componente i do padrão de entrada está aproximando-se do ponto fixo estabelecido na etapa de armazenamento. Neste caso, a primeira condição da Eqn. 4 será satisfeita, então o valor de n_i será diminuído e, conseqüentemente, a

dinâmica do mapa i se aproximará da órbita periódica. Se o elemento i converge para o ponto periódico correto, ele ficará ali. Senão, a segunda condição da Eqn. 4 será satisfeita, dirigindo o mapa de volta para o estado caótico, e também pós-crise. Assim os elementos ficam movendo-se entre os estados periódico e caótico, até encontrarem seus estados corretos, ou seja, a diferença $x_i(t) - \sum_{j=1}^N w_{ij}x_j(t)$ é minimizada. Quando todos os elementos atingem seus pontos periódicos corretos, o padrão armazenado será recuperado.

5.2.3 Simulações por Computador

À seguir, é apresentada simulações do modelo de reconhecimento de padrões quaternários. A Fig. 5.3 apresenta 6 padrões armazenados na memória, mediante o algoritmo de armazenamento em ponto fixo. Nestas simulações, os seguintes valores para os parâmetros são usados: $\alpha = 0.07$, $\theta = 0.2$, $n_0 = 0.97$ e $n_{min} = 0.8$.

Figura 5.3: Padrões armazenados no modelo, cada um com quatro níveis de cinza.

A Fig. 5.4 mostra o processo de reconhecimento. Uma versão corrompida do padrão **cérebro** é mostrada, pela primeira imagem da Fig. 5.4, e é apresentada ao modelo como condição inicial. O sistema é executado, e cada elemento da rede procura um novo estado para sair do estado caótico e permanece no ponto periódico correto, mediante a mudança do valor do parâmetro de controle. Quando todos os elementos ficam no ponto periódico correto, o sistema conserva seu estado, e o padrão armazenado é recuperado, como pode ser visto na últimas imagens da Fig. 5.4.

Figura 5.4: Processo de reconhecimento de um padrão corrompido.

Figura 5.5: Evolução do parâmetro n_i .

A Fig. 5.5 mostra a evolução dos parâmetros n_i para todos os neurônios da rede, podendo se ver que finalmente todos os n_i caem para o valor $n_{min} = 0.8$, o que indica que todos os elementos estão nos pontos periódicos. Conseqüentemente, o sistema inteiro cai para um ponto fixo.

Figura 5.6: Evolução do estado de ativação.

A Fig. 5.6 mostra o estado de ativação de todos os neurônios, onde pode ser visto que os neurônios ficam num dos quatro pontos periódicos depois de algumas iterações. A Fig. 5.7 mostra as distâncias de Hamming entre o estado da rede e cada padrão armazenado, para cada iteração. Podemos ver que uma dessas distâncias cai para o valor zero. Isso indica que a rede convergiu para um ponto fixo, o que corresponde a um dos padrões armazenados.

Figura 5.7: Distância de Hamming entre o estado da rede e cada padrão armazenado.

5.3 Descrição do Modelo II

O funcionamento deste modelo também é dividido em duas etapas: etapa de aprendizado e etapa de reconhecimento. Na etapa de aprendizado, o mesmo algoritmo de matriz pseudo-inversa é usada para armazenar os padrões em pontos fixos. Na etapa de reconhecimento, as dinâmicas caótica e periódica do mapa logístico são utilizadas, sendo que a órbita periódica representa o padrão reconhecido, e a órbita caótica representa um estado de busca. Como foi mencionado, o mapa logístico usado neste modelo pode ser substituído por qualquer mapa que apresente dinâmica caótica e periódica, de período desejado.

5.3.1 A Etapa de Aprendizado

O processo de aprendizado é similar ao apresentado no modelo anterior, utilizando a matriz pseudo-inversa Q^+ .

Para armazenar M padrões, são utilizados os pontos fixos da equação:

$$WP = P \quad (7)$$

onde W é a matriz de conexão, podendo ser determinada pela matriz pseudo-inversa (J. Hertz, 1991).

$$W = PP^+ \quad (8)$$

Neste modelo, antes do processo de armazenamento, os valores originais dos padrões $p_j^{(j)}$ são transformados pela seguinte regra, para poder aproveitar a estrutura simétrica da bifurcação de duplicação de período:

$$Tp_i = 2 \left(i - \frac{V}{2} \right) + 1 \quad (9)$$

onde i é um possível valor dos padrões, $i = 1, 2, \dots, V$. A transformação distribui os valores de forma simétrica sobre zero.

5.3.2 A Etapa de Reconhecimento

A dinâmica do modelo é dada pelas seguintes equações:

$$x_i(t+1) = z_i(\tau) x_i(t) (1 - x_i(t)) \quad (10)$$

$$z_i(\tau) = \begin{cases} A_p & \text{se } \left| s(x_i(t)) - \sum_{j=1}^N w_{ij} s(x_j(t)) \right| \\ A_c & \text{caso contrário} \end{cases} \quad (11)$$

$$s(x) = \begin{cases} Tp_1 & \text{se } x > (X_1 + X_2)/2 \\ Tp_2 & \text{se } ((X_1 + X_2)/2 > x) \text{ e } x > ((X_2 + X_3)/2) \\ \dots & \dots \\ Tp_{V-1} & \text{se } ((X_{V-2} + X_{V-1})/2 > x) \text{ e } x > ((X_{V-1} + X_V)/2) \\ Tp_V & \text{se } x < (X_{V-1} + X_V)/2 \end{cases} \quad (12)$$

Onde a Eqn. 10 corresponde ao mapa logístico, sendo $x_i(t)$ a variável de estado do i -ésimo elemento na rede, para $i = 1, 2, \dots, N$. No mapa logístico original, $z_i(\tau)$ corresponde ao parâmetro de bifurcação, mas aqui a variável $z_i(\tau)$ define a dinâmica de cada elemento. $z_i(\tau)$ pode tomar um dos seguintes valores: A_p e A_c . Se $z_i(\tau) = A_p$, o comportamento do i -ésimo elemento corresponde a dinâmica periódica. Se $z_i(\tau) = A_c$, a dinâmica do i -ésimo elemento é caótica. Assim a dinâmica de cada elemento pula entre os estados periódico e caótico, dependendo do valor retornado pela Eqn. 11.

A variável x é avaliada em cada iteração t , enquanto que a variável z é avaliada a cada Vt iterações, onde V corresponde ao número de possíveis valores dos padrões armazenados, e também corresponde à órbita periódica selecionada. Podemos perceber que cada ponto periódico é usado para representar um possível valor do padrão. Por exemplo, se forem armazenados

padrões binários, i.e., $V = 2$, então, A_p será escolhido onde um ciclo de período 2 ocorre. Neste caso, $z_i(\tau)$ é avaliado a cada duas iterações. Um ponto periódico representa o valor 1 e outro representa o valor 0. Finalmente a Eqn. 12 define o valor correspondente entre cada valor transformado Tp_i , para cada variável de estado $x_i(t)$ do mapa logístico.

Figura 5.8: Diagrama de bifurcação e escolha dos pontos pp_1 , pp_2 , pp_3 e pp_4 do mapa logístico.

O funcionamento do modelo, quando $V = 4$, é descrito a seguir. Primeiramente, um conjunto de padrões são armazenados na memória, usando o algoritmo de aprendizado descrito anteriormente pela Eqn. 8. Na etapa de reconhecimento, selecionamos o valor de A_p , que corresponde a uma órbita com período 4, e o valor de A_c , que corresponde a uma órbita periódica, como pode ser visto na Fig. 5.8. Nesta figura, pp_1 , pp_2 , pp_3 , e pp_4 correspondem aos 4 valores da órbita periódica, e representam os 4 valores do padrão Tp_1 , Tp_2 , Tp_3 e Tp_4 , respectivamente. $z_i(0)$ são colocados inicialmente na órbita A_c , então todos os elementos são inicialmente caóticos.

A Fig. 5.9 mostra o diagrama *cobweb* e a série temporal do mapa logístico, quando o valor de $A = 4$. Neste caso, a dinâmica do mapa logístico é caótica.

Figura 5.9: Diagrama *cobweb* e série temporal do mapa logístico, quando o valor de $A = 4$.

Da figura, pode se observar que a trajetória da órbita caótica vaga no atrator e frequentemente se aproxima de todos os pontos da região de atração. Quando um padrão armazenado é apresentado à rede como condição inicial, e devido à propriedade de ergodicidade do caos, cada elemento, em algum instante, pode se aproximar de um estado no qual $|s(x_i(t)) - \sum_{j=1}^N w_{ij}s(x_j(t))|$ é pequeno. Isso significa que o i -ésimo elemento está se aproximando do ponto fixo definido pelo processo de armazenamento. Quando isto acontece, a primeira condição da Eqn. 11 é satisfeita, então o i -ésimo elemento é levado para a órbita periódica. Se esse elemento converge para o ponto periódico correto, ele ficará ali. Senão, a segunda condição da Eqn. 11 será satisfeita, e o mesmo elemento retornará para o estado caótico. Assim, os elementos ficam pulando entre os estados periódico e caótico, até encontrarem seus estados corretos, ou seja, a diferença $|s(x_i(t)) - \sum_{j=1}^N w_{ij}s(x_j(t))|$ é minimizada. Quando todos os elementos atingem seus pontos periódicos corretos, e o padrão armazenado é recuperado.

Em resumo, o modelo apresenta dois estados: o estado periódico, que representa o padrão recuperado, e o estado caótico, que permite ao elemento buscar seu estado correto.

5.3.3 Simulações por Computador

Nesta seção, serão apresentadas várias simulações para reconhecimento de padrões binários e quaternários.

Primeiro, serão apresentados resultados de simulações para reconhecimento de padrões binários, i.e., $V = 2$. Nestas simulações, os parâmetros são fixados em $\delta = 1$, $A_p = 3.4$ e $A_c = 4$.

A Fig. 5.10 mostra 4 padrões armazenados na rede.

Figura 5.10: Padrões binários armazenados na rede .

Inicialmente, uma versão corrompida do padrão X é apresentado ao modelo, como pode ser visto na primeira imagem da Fig. 5.11. Neste caso, a Eqn. 5.11 é avaliada a cada 2 unidades de tempo. Assim, o sistema é iterado, e cada elemento da rede procura seu estado correto, ajustando seu parâmetro de controle. Quando todos os elementos se estabilizam no ponto periódico correto, o sistema conserva seu estado, e o padrão armazenado é recuperado, como pode ser vista nas últimas imagens da Fig. 5.11.

Figura 5.11: Processo de reconhecimento de um padrão.

A Fig. 5.12 mostra a diferença $u_i = |s(x_i(t)) - \sum_{j=1}^N w_{ij}s(x_j(t))|$ de todos os neurônios para cada iteração, podendo-se ver que finalmente todos os u_i caem para zero. Isso indica que todos os elementos estão no ponto periódico correto. Também significa que o sistema caiu para o ponto fixo.

Figura 5.12: Evolução de u_i para $i = 1, 2, \dots, N$.

A Fig. 5.13 mostra o estado de ativação de todos os neurônios, e pode ser visto que os neurônios ficam em um dos 2 pontos periódicos após algumas iterações. Na Fig. 5.14, pode-se ver a distância de Hamming entre o estado da rede e cada padrão armazenado, para cada iteração. Podemos ver que uma dessas distâncias cai para o valor zero, indicando que a rede convergiu para um ponto fixo, que corresponde a um dos padrões armazenados.

No caso de $V = 4$, os parâmetros são fixados em $\delta = 1.2$, $A_p = 3.5$ e $A_c = 4$. Os seis padrões mostrados na Fig. 5.3 são armazenados na rede.

Figura 5.13: Evolução de u_i para $i = 1, 2, \dots, N$. Figura 5.14: Evolução de u_i para $i = 1, 2, \dots, N$.

A Fig. 5.15 mostra o processo de reconhecimento.

Inicialmente, uma versão corrompida do padrão da borboleta é apresentado ao modelo, como pode ser visto na primeira imagem da Fig. 5.15. Neste caso, a Eqn. 11 é avaliada a cada 4 unidades de tempo. Assim, o sistema é executado, e cada elemento da rede procura seu estado correto, modificando o parâmetro de controle. Quando todos os elementos ficam no ponto periódico correto, o sistema conserva seu estado, e o padrão armazenado é recuperado, como pode ser visto nas últimas imagens da Fig. 5.15.

Da Fig. 5.15, pode ser visto que o padrão já aparece em alguns passos intermediários, por exemplo, nas iterações $t = 180$ e $t = 240$. Isso acontece por causa da representação gráfica. Na verdade, nestas iterações, alguns elementos ainda não estão em seus estados corretos, mas

Figura 5.15: Processo de reconhecimento da borboleta.

estão próximos, i.e., o sistema ainda não convergiu para os estados estáveis. Mas, nas últimas iterações, o sistema realmente converge, porque não houve mais mudança.

Figura 5.16: Evolução de u_i para $i = 1, 2, \dots, N$.

A Fig. 5.16 mostra a diferença $u_i = |s(x_i(t)) - \sum_{j=1}^N w_{ij}s(x_j(t))|$, para todos os neurônios em cada iteração, podendo se ver que finalmente todos os u_i caem para zero. Isso indica que todos os elementos estão no ponto periódico correto. Também significa que o sistema caiu para o ponto fixo.

A Fig. 5.17 mostra o estado de ativação de todos os neurônios, onde pode ser visto que os neurônios ficam em um dos quatro pontos periódicos depois de algumas iterações. Na Fig. 5.18, pode-se ver a distância de Hamming entre o estado da rede e cada padrão armazenado,

Figura 5.17: Série temporal dos neurônios do sistema.

para cada iteração. Pode-se ver que uma dessas distâncias cai para o valor zero, indicando que a rede convergiu para um ponto fixo, que corresponde a um dos padrões armazenados.

Figura 5.18: Distância de Hamming entre o estado da rede e cada padrão armazenado.

Naturalmente uma sequência de DNA é representada por um padrão de 4 valores diferentes. Então, quando $V = 4$ o modelo também pode ser usado para armazenar sequências de DNA, sendo cada ponto periódico a representação de um nucleotídeo. Assim uma sequência armazenada pode ser recuperada a partir de um segmento incompleto ou corrompido.

A seguir se mostra um exemplo de como usar nosso modelo para o reconhecimento de sequências de DNA. A Fig. 5.19 mostra 6 possíveis sequências de DNA que são armazenadas na rede. Também pode ser visto que os 4 nucleotídeos são representados por um cor.

Figura 5.19: Sequências de DNA utilizadas na etapa de armazenamento.

A Fig. 5.20 mostra o processo de reconhecimento da sequência 1 do DNA da Fig. 5.19, utilizando os seguintes parâmetros: $\delta = 1.2$, $A_p = 3.5$ e $A_c = 4$.

Figura 5.20: Processo de reconhecimento de uma sequência de DNA.

Inicialmente, uma versão corrompida da sequência 1 é usada como condição inicial e corresponde à primeira imagem da Fig. 5.20. Quando todos os elementos ficam no ponto periódico correto, o sistema fica estável e a sequência armazenada é recuperada, como pode ser visto nas últimas imagens da Fig. 5.20.

A Fig. 5.21 mostra a evolução do parâmetro u , podendo-se ver que finalmente todos os u_i caem para zero. Isso indica que todos os elementos estão no ponto periódico correto.

Figura 5.21: Evolução de u_i para $i = 1, 2, \dots, N$.

Figura 5.22: Série temporal dos neurônios do sistema.

A Fig. 5.22 mostra o estado de ativação de todos os neurônios, onde pode ser visto que os neurônios ficam em um dos 4 pontos periódicos depois de algumas iterações. Na Fig. 5.23, pode-se ver a distância de Hamming entre o estado da rede e cada padrão armazenado, para cada iteração. Pode-se ver que uma dessas distâncias cai para o valor zero, indicando que a rede convergiu para um ponto fixo, que corresponde a um dos padrões armazenados.

Este método pode ser generalizado para outros valores de V como, por exemplo, $V = 8$ ou $V = 16$, precisando-se encontrar os parâmetros adequados para o processo de reconhecimento, principalmente o valor de A_p , que determinará o ciclo periódico e o número de valores que a rede pode reconhecer.

Figura 5.23: Distância de Hamming entre o estado da rede e cada padrão armazenado.

5.4 Características dos Modelos

Nesta tese, foram propostas duas redes neurais caóticas para o reconhecimento de padrões multi-valorados. Em ambos modelos, podem ser distinguidas duas etapas: na primeira, os padrões são armazenados em pontos fixos, pelo algoritmo de aprendizado da matriz pseudo-inversa; na segunda, é feito o reconhecimento, onde as dinâmicas periódica e caótica, presentes no mapa, são usadas. A órbita periódica representa o padrão recuperado, enquanto que a órbita caótica representa um estado de busca. Inicialmente, todos os neurônios são ajustados na região caótica e, devido à propriedade da ergodicidade do caos, a trajetória caótica de cada neurônio se aproximará dos pontos periódicos cobertos pelo atrator caótico, em algum instante. Quando isto ocorre, um mecanismo de controle será ativado, para dirigir a dinâmica de cada neurônio a seu ponto periódico estável correspondente. Neste caso, o caos fornece um mecanismo de busca robusta para cada neurônio, por estabilizá-lo num estado desejável. Consequentemente, não somente os padrões exatamente armazenados podem ser recuperados, mas também versões corrompidas podem ser reconhecidas. O Modelo I utiliza as dinâmicas do mapa senoidal, tais como dinâmica periódica e caótica, crise e dependência às condições iniciais; o Modelo II necessita somente de um estado caótico e um estado periódico de período desejado, sendo, portanto, independente de um mapa específico. Baseado nisso, podemos dizer que o Modelo II é uma simplificação e melhoria do Modelo I. As simulações foram feitas para reconhecimento de padrões binários e quaternários. Mas, através da descrição do mecanismo de funcionamento dos modelos, podemos perceber que ambos modelos podem ser utilizados para reconhecimento de padrões de mais níveis de valores, basta encontrar um ciclo limite de período igual aos níveis de valores de padrões armazenados.

5.5 Considerações Finais

Neste capítulo foram propostos dois modelos de redes neurais caóticas para reconhecimento de padrões multi-valorados.

Pode-se perceber que ambos modelos utilizam a dinâmica periódica e caótica, sendo que a dinâmica periódica representa o padrão recuperado, enquanto que a dinâmica caótica fornece um mecanismo de busca eficiente e robusto.

Conclusões e Trabalhos Futuros

6.1 Considerações iniciais

O estudo e desenvolvimento de redes neurais como sistemas dinâmicos não-lineares é a parte principal deste trabalho. Foram proposto dois novos modelos, os quais aproveitam diretamente as propriedades dos sistemas dinâmicos não-lineares, mais especificamente a dinâmica periódica e caótica.

Neste capítulo são apresentadas as conclusões desta pesquisa e algumas propostas de trabalhos futuros que formam parte de uma continuação desta pesquisa.

6.2 Conclusões

Como resultado do estudo e desenvolvimento das redes neurais caóticas para reconhecimento de padrões, podemos chegar às seguintes conclusões.

- As redes neurais artificiais são originalmente inspiradas pelo funcionamento do cérebro. Nos últimos 20 anos, pesquisas no funcionamento do cérebro conduziram os pesquisadores a concluir que a dinâmica caótica é um elemento essencial no processamento de informação de sistemas neurais biológicos. Portanto, a utilização da dinâmica de caos

em redes neurais apresenta-se como uma área promissora no desenvolvimento de novos modelos de redes neurais.

- Através da revisão e desenvolvimento dos modelos de redes neurais caóticas pode se dizer que a utilização da dinâmica caótica ajuda a superar certas limitações dos modelos tradicionais como por exemplo escapar da memória espúria e aumentar a capacidade de armazenamento quando comparada com o modelo de Hopfield (S. Ishii, 1996).
- Tanto nos modelos de redes caóticas de Ishii (S. Ishii, 1996), Aihara (M. Adachi, 1997) e Lee (G. Lee, 2001) quanto nos modelos propostos, devido a suas propriedades de ergodicidades, o caos fornece aos neurônios um mecanismo de busca eficiente e robusto. Esta característica particular do caos é muito importante e insubstituível por outros tipos de dinâmicas.
- Os modelos apresentados nos Capítulos 3 e 4 são sistemas dinâmicos que podem ser usados como memórias associativas para reconhecimento de padrões. Mas esses modelos (pelo menos nas formas originais) só armazenam e reconhecem padrões binários. Isso limita muito os modelos em aplicações reais. Os modelos propostos neste trabalho são mais robustos nesse sentido porque são capazes de reconhecer não só padrões binários, mas também padrões multi-valorados.

6.3 Trabalhos futuros

Os resultados obtidos no desenvolvimento deste trabalho nos levou a pensar em outras possíveis aplicações, algumas delas mediante a aplicação direta do modelo e outras mediante a modificação do mecanismo de funcionamento onde as propriedades do caos devem ser utilizadas:

- Neste trabalho, foram mostradas simulações de reconhecimento de padrões V -ários com $V = 4$. Em aplicações práticas, valores de níveis de um padrão podem ser grandes. Por exemplo, para uma imagem de nível de cinza, temos $V = 256$ em geral. Neste caso, é necessário modificar os modelos atuais, construindo um mapa que gera um ciclo limite de período 256. Uma vez realizado, vários tipos de aplicações práticas podem ser tratadas pela idéia ilustrada neste trabalho.
- Na atualidade o desenvolvimento de ferramentas é importante em aplicações de biologia, como por exemplo ferramenta para reconhecimento de DNA. Uma sequência de DNA é naturalmente representada por um padrão de 4 valores diferentes. Então, quando a órbita periódica é de período 4, várias sequências de DNA podem ser armazenadas, sendo cada

ponto periódico a representação de um nucleotídeo. Assim uma sequência armazenada pode ser recuperada a partir de um segmento incompleto ou corrupto.

- Um dos problemas das memórias associativas existentes é a capacidade de armazenamento altamente limitada. Como sabemos que o atrator caótico possui infinitas órbitas instáveis as quais poderiam ser aproveitadas para armazenamento de informação, assim grande quantidade de informações poderiam ser processadas sendo uma vantagem considerável em aplicações reais.

Referências Bibliográficas

- A. Babloyantz, J. M. Salazar, C. N. (1995). Evidence of chaotic dynamics of brain activity during the sleep cycle. *Physical Letters*, 111A(3):152–155.
- A. Braga, A. Carvalho, T. L. (1998). *Fundamentos de Redes Neurais*. COPPE/UFRJ for the XI Brazilian Computing School, Rio de Janeiro, Brazil.
- Aihara, K. (1990). Chaotic neural network. *Bifurcation phenomena in nonlinear systems and theory of dynamical systems*, pages 143–161.
- C. Grebogi, E. Ott, J. Y. (1987). Unstable periodic orbits and the dimension of chaotic attractors. *Physics Review*, 36(7):3522–3524.
- Cáceres, G. J., C. C. e. Z. L. (2002). Reconocimiento de rostros utilizando la transformada de wavelets y la red neuronal hopfield. *Congreso Latinoamericano de Sistemas Informática, Telemática, Electrónica e Telecomunicación*, pages CD-ROM.
- D. Ackley, G. Hinton, T. S. (1985). A learning algorithm for boltzman machines. *Cognitive Science*, 9:147–169.
- D. Hush, B. Horne, H. S. (1993). Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 10(1):8–39.
- D. Rumelhart, G. Hinton, R. W. (1986). Learning representations by back-propagating errors. *Nature (London)*, 323:533–536.
- Daubechies, I. (1992). *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics.
- G. Carpenter, S. G. (1987). Self-organization of the stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930.

- G. Lee, N. H. F. (2001). Parametrically coupled sine map networks. *International Journal of Bifurcation and Chaos*, 11(7):1815–1834.
- Gail A. Carpenter, S. G. (1992). *Neural Network for Vision and Image Processing*. Massachusetts Institute of Technology.
- Gomez, J., Velho, L., and Goldenstein, S. (1997). *Wavelets : Teoría, Software e Aplicações*. Instituto de Matemática Pura e Aplicada.
- H. Sompolinsky, A. Crisanti, A. S. (1988). Chaos in random neural networks. *Physical Review Letters*, 61:259–262.
- Hayashi, Y. (1997). Oscillatory neural network and learning of continuously transformed patterns. *Neural Networks*, 7(2):219–231.
- Haykin, S. (1994). *Neural Networks*. Macmillan College Publishing Company.
- Hebb, D. (1949). *The organization of behavior; a neuropsychological theory*. Wiley-Interscience.
- Hilborn, R. (1994). *Chaos in nonlinear dynamics, And introductions for scientists and engineers*. Oxford University Press.
- Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the U.S.A.*, 79:2554–2558.
- Hopfield, J. (1994). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the U.S.A.*, 81:3088–3092.
- J. Hertz, A. Krogh, R. G. P. (1991). *Introduction to the Theory of Neurocomputing*. Addison-Wesley.
- James A. Freeman, D. M. S. (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*. Addison Wexley.
- K. Aihara, G. M. (1986). Chaotic oscillations and bifurcations in squid giant axons. *In Chaos*, 1:257–269.
- Kaneko, K. (1990). Clustering coding switching, hierarchical ordering, and control in a network of chaotic elements. *Physica*, 41:137–172.
- Kohonen, T. (1988). *Self-Organization and Associative Memory*. Springer-Verlag.

- Kolen, J. (1994). *Exploring the computational capabilities of recurrent neural network*. PhD thesis, The Ohio State University.
- L. Chua, L. Y. (1988a). Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, 35(10):1273–1290.
- L. Chua, L. Y. (1988b). Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257–1272.
- M. Adachi, K. A. (1997). Associative dynamics in a chaotic neural network. *Neural Networks*, 10(5):83–98.
- M. Thomas, W. G. Gibson, J. R. (1996). Stability and bifurcations in an associative memory model. *Neural Networks*, 9:53–66.
- Ott, E. (1993). *Chaos in Dynamical System*. Cambridge University Press.
- P. Das, W. S. (1995). A bifurcation analysis of the four-dimensional generalized hopfield neural network. *Physical D*, 88:14–28.
- P. Thiran, M. H. (1994). Information processing using stable and unstable oscillations: a tutorial. *Third IEEE International Workshop on Cellular Neural Networks and their Applications*, pages 127–136.
- Parisi, G. (1997). Asymmetric neural networks and the process of learning. *J. Phys. A: Math. Gen.*, 19:675–680.
- R. Aggarwal, Y. S. (1997). Artificial neural networks in power systems. i. general introduction to neural computing. *Power Engineering Journal*, 11(3):129–134.
- Ravichandran, A. and Yegnanarayana, B. (1991). A two-stage neural network for translation, rotation and size-invariant visual pattern recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:2393–2396.
- S. Ishii, K. Kukumizu, S. W. (1993). A globally coupled map model for information processing. *Proceedings of the International Symposium on Nonlinear theory and its Applications*, pages 1157–1160.
- S. Ishii, K. Kukumizu, S. W. (1996). A network of chaotic elements for information processing. *Neural Networks*, 9(1):25–40.
- S. Nara, P. Davis, H. T. (1997). Memory search using complex dynamics in a recurrent neural network model. *Neural Networks*, 6:963–973.

- Tsuda, I. (1994). Dynamic link of memory-chaotic memory map in nonequilibrium neural networks. *Neural Networks*, 5:313–326.
- W. Freeman, C. S. (1987). Simulation of chaotic eeg patterns with a dynamic model of the olfactory sistem. *Biological Cybernetics*, 56:139–150.