

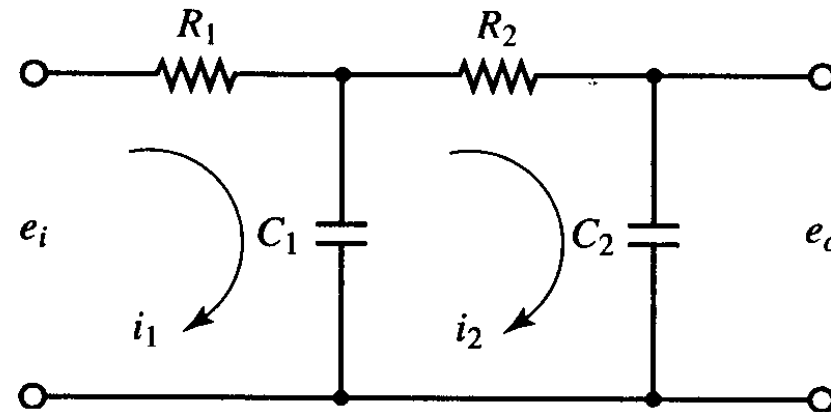
Redes Neurais Recorrentes (Parte 1)

Índice Geral

1	Princípios de Computação Analógica	2
2	O Poder da Simulação Computacional	8
3	Neurocomputador como um <i>hardware</i> ‘flexível’	11
4	As Quatro Gerações de Redes Neurais Artificiais.....	16
5	Princípios Básicos de Sistemas Dinâmicos Não-Lineares	19
5.1	Introdução	19
5.2	Sistemas Não-Lineares Multidimensionais	21
5.3	Análise de Sistemas Não-Lineares.....	25
5.4	Exemplos de Comportamentos Dinâmicos Não-Lineares.....	25
5.5	Estado Estacionário em Sistemas Não-Lineares Autônomos	28
5.5.1	Definição de Trajetória	29
5.5.2	Pontos Fixos (ou Pontos de Equilíbrio).....	32
5.5.3	Soluções Periódicas.....	34
5.5.4	Soluções Quase-Periódicas	36
5.5.5	Caos.....	37
6	Exemplos de Comportamentos Dinâmicos: Retrato de Fase	43
7	Modelagem de Sistemas Dinâmicos Lineares	53
8	Modelagem de Sistemas Dinâmicos Não-Lineares.....	54
9	Excursão por Outras Arquiteturas de Redes Neurais Recorrentes	55
10	O bloco LSTM (<i>Long Short-Term Memory</i>)	60
11	O Efeito da Recorrência no Processo de Adaptação.....	64
12	Treinamento Supervisionado para Redes Recorrentes	67
13	<i>Backpropagation Through Time</i> (BPTT)	69
14	O problema dos gradientes explosivos e que se anulam.....	72
15	Predição de Séries Temporais: Um Passo ou Múltiplos Passos.....	74
16	Referências Bibliográficas	77
17	Referências Adicionais e links relevantes.....	82

1 Princípios de Computação Analógica

- Considere o seguinte circuito elétrico:



- As equações que regem seu comportamento ao longo do tempo são dadas na forma:

$$\frac{1}{C_1} \int (i_1 - i_2) dt + R_1 i_1 = e_i$$

$$\frac{1}{C_2} \int i_2 dt = e_o$$

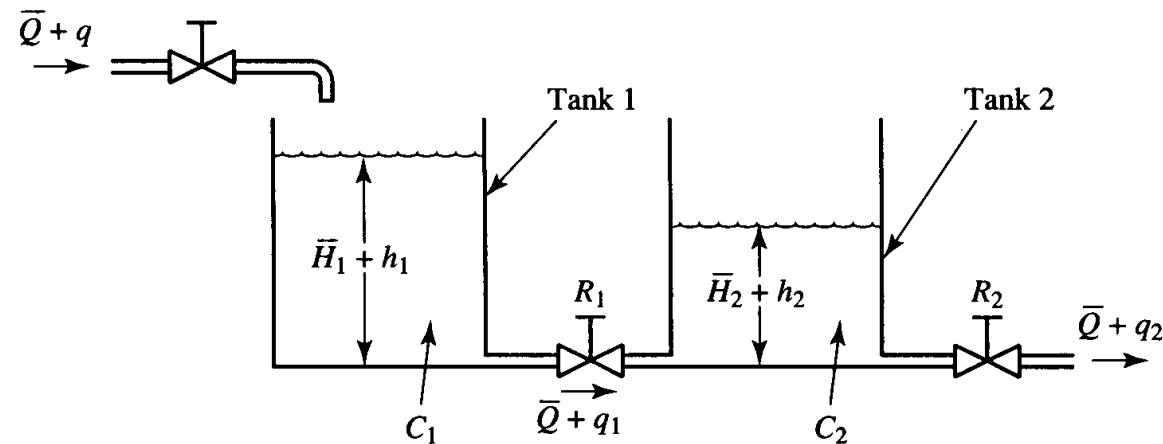
$$\frac{1}{C_1} \int (i_2 - i_1) dt + R_2 i_2 + \frac{1}{C_2} \int i_2 dt = 0$$

- Utilizando a transformada de Laplace, obtém-se:

$$E_0(s) = G(s)E_i(s)$$

com função de transferência $G(s) = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_1 C_2)s + 1}$.

- Considere agora o seguinte sistema hidráulico, composto por tanques comunicantes:



onde \bar{Q} e \bar{H}_i ($i=1,2$) são fluxo e altura em estado estacionário.

- O propósito é mostrar que existem funções de transferência análogas entre este sistema hidráulico e o circuito elétrico apresentado anteriormente, mesmo tratando-se de sistemas dinâmicos de natureza distinta.
- Definição 1: A resistência R em uma válvula é dada pela razão entre a altura h e o fluxo q de vazão, produzindo:

$$R \left[\frac{\text{m}}{\text{m}^3/\text{s}} \right] = \frac{h}{q}.$$

- Definição 2: A capacitância de um tanque é dada pela mudança em quantidade de líquido armazenado necessária para ocasionar uma mudança unitária na altura, produzindo:

$$C \left[\frac{\text{m}^3}{\text{m}} \right] = \frac{\text{mudança na quantidade de líquido armazenado}}{\text{mudança na altura}}.$$

- Repare que capacitância (medida em m^2) não é capacidade (medida em m^3).

- Se o diâmetro do tanque for constante, então a capacitância (que, por definição, é a área de uma seção horizontal no tanque) será constante.
- Das duas definições acima, resultam as duas equações a serem empregadas na modelagem do sistema de tanques comunicantes:

$$\frac{h}{R} = q \qquad C \frac{dh}{dt} = (q_i - q_o)$$

- Sendo assim, as equações que regem o comportamento ao longo do tempo do sistema de tanques comunicantes são dadas na forma:

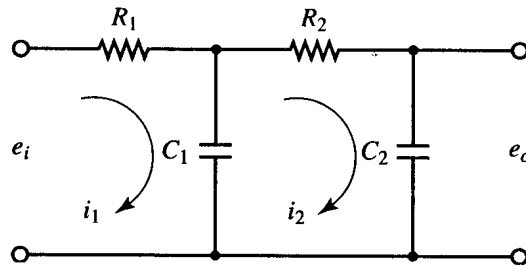
$$\begin{aligned} \frac{h_1 - h_2}{R_1} &= q_1 & C_1 \frac{dh_1}{dt} &= q - q_1 \\ \frac{h_2}{R_2} &= q_2 & C_2 \frac{dh_2}{dt} &= q_1 - q_2 \end{aligned}$$

- Utilizando a transformada de Laplace, obtém-se:

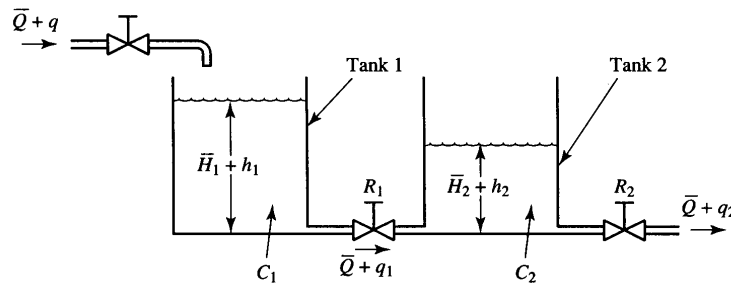
$$Q_2(s) = G(s)Q(s)$$

com função de transferência $G(s) = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_2 C_1)s + 1}$.

- Comparando-se as duas funções de transferência, constata-se que elas não são idênticas (diferem apenas nos termos que indicam a interação de tanques e a interação de capacitores).



$$G(s) = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_1 C_2)s + 1}$$



$$G(s) = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_2 C_1)s + 1}$$

- No entanto, dado um dos sistemas dinâmicos apresentados, **existem valores que podem ser adequadamente definidos para os parâmetros do outro sistema de modo que, para uma mesma condição inicial, o comportamento ao longo do tempo seja idêntico para ambos os sistemas.**
- Sendo assim, tomando-se aquele sistema mais simples de ser projetado, que ocupa menos espaço, que apresenta menor custo, cujos parâmetros podem ser ajustados de forma simples e confiável e cujas variáveis de estado podem ser facilmente medidas, é possível estudar o comportamento dinâmico do outro sistema, por analogia.
- Este é o princípio da computação analógica.
- Quando as redes neurais artificiais apresentam funções de ativação contínuas, elas implementam neurocomputadores que representam computadores analógicos (SIEGELMANN, 1999).

2 O Poder da Simulação Computacional

- Uma das mais importantes realizações científicas do último século é, sem dúvida, a descoberta do poder da simulação, ou seja, a capacidade de extrair informações sobre um determinado dispositivo mesmo antes deste existir fisicamente, se é que ele possa vir a existir algum dia.
- Em 1953, Enrico Fermi e dois de seus colegas do Laboratório Científico Los Alamos (onde foi descoberta a bomba atômica), John Pasta e Stanislaw Ulam, inventaram o conceito de um ‘experimento de computador’.
- A partir de então, o computador se transformou em uma nova forma de exercer ciência, por realizar uma função semelhante àquela do microscópio e outros equipamentos que ampliam a capacidade sensorial humana, ao revelar peculiaridades de fenômenos que jamais seriam perceptíveis por meio de procedimentos tradicionais de experimentação em laboratório.

- Fermi e seus colegas introduziram a simulação computacional como uma forma eficaz de se buscar uma maior compreensão do conceito de entropia e, desde então, inúmeros experimentos científicos e tecnológicos têm sido realizados usando simulação computacional.
- Devido ao seu poder de processamento, os computadores digitais são hoje definidos como **ferramentas de simulação de propósito geral**, já que a partir do mesmo *hardware* dotado de poder computacional suficiente, é possível desenvolver vários produtos de *software* que conduzam o computador a desempenhar tarefas específicas de complexidade variada.
- Portanto, o programa de computador (*software*) é o responsável pela flexibilidade associada ao papel do computador como uma ferramenta de simulação de propósito geral.

- Um exemplo recente deste poder da simulação computacional está vinculado a testes de confiabilidade de neurocomputadores obtidos a partir de *deep learning* (KATZ et al., 2017).
- KATZ et al. (2017) propõem uma ferramenta computacionalmente eficiente para superar um obstáculo conhecido: obtenção de garantias formais de comportamento / desempenho quando operando em sistemas que envolvem níveis críticos de segurança, como sistemas de controle de tráfego aéreo autônomos de próxima geração.
- Com isso, um computador digital pode simular um neurocomputador, por si só, e também pode, através de mecanismos de otimização e de simulação de todas as possibilidades operacionais factíveis para o neurocomputador, fornecer garantias formais de confiabilidade na solução de problemas.

3 Neurocomputador como um *hardware* ‘flexível’

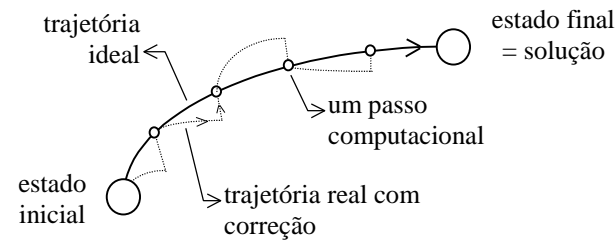
- A notável evolução de processos de síntese de mapeamentos estáticos e dinâmicos utilizando redes neurais artificiais recorrentes e não-recorrentes representa uma nova perspectiva na geração de ferramentas de simulação de propósito geral.
- Esta nova concepção de dispositivos computacionais, denominada neurocomputação, busca dividir a responsabilidade pela flexibilidade do dispositivo computacional entre o *software* e o *hardware*.
- Com isso, ao abrir mão de um *hardware* de propósito geral, pode-se **projetar dispositivos de *hardware* dedicados**, específicos para cada aplicação.
- Como visto na seção anterior, computadores digitais podem ser prontamente utilizados no processo de implementação de neurocomputadores. Com isso, faz-se necessário apresentar razões pelas quais se mantém o interesse em projetar e construir arquiteturas dedicadas que resultem em máquinas de neurocomputação:

- Devido à sua arquitetura inerentemente paralela, o tempo de processamento ganho na utilização de neurocomputadores, em lugar de computadores digitais, torna possível a solução de problemas até então intratáveis pelos mais poderosos computadores digitais atualmente disponíveis;
- Máquinas dedicadas induzem o pesquisador a adotar outras estratégias de abordagem para um determinado problema a ser resolvido com base em ferramentas computacionais;
- Já existem tecnologias e iniciativas bem-sucedidas para implementar neurocomputadores em *hardware* (MISRA & SAHA, 2010), incluindo *deep learning* (<https://www.nvidia.com/en-us/data-center/dgx-systems/>).
- Como será visto ainda neste tópico do curso, a programação de um neurocomputador na forma de uma rede neural de Hopfield não envolve a elaboração de um programa caracterizado por uma sequência de instruções.

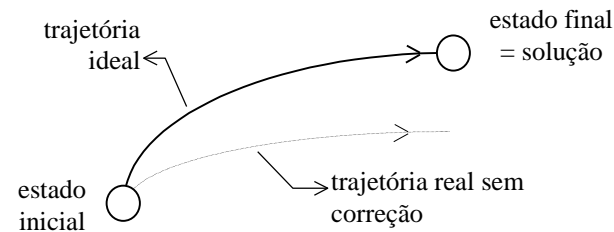
- O que se busca, na verdade, é delinear o problema no espaço de estados do neurocomputador, conduzindo a uma superfície de energia a ser minimizada com base na dinâmica de relaxação produzida pelo acoplamento entre neurônios.
- Fica evidenciada aqui a existência de três tipos básicos de variáveis associadas à dinâmica de um neurocomputador: (1) os estados dos neurônios; (2) seu padrão de conectividade; e (3) a intensidade das conexões sinápticas da rede neural.
- O ajuste dinâmico da intensidade das conexões sinápticas corresponde à fase de aprendizado ou incorporação de conhecimento, enquanto que a dinâmica dos estados, ou dinâmica de relaxação, corresponde ao processo de minimização da superfície de energia, que conduz à solução do problema.
- PERETTO (1992) apresentou uma comparação pertinente a esta discussão, envolvendo neurocomputadores, computadores analógicos e computadores digitais, conforme é apresentado na Figura 1:

- Computadores digitais: Executam automaticamente, a cada passo do processo de computação, correções da trajetória de seu estado físico por meio da representação binária (explora redundância) do estado interno de seus componentes;
- Computadores analógicos: Não são capazes de realizar correção de trajetória, sendo que a correção / confiabilidade dos resultados produzidos depende da acurácia dos componentes (quanto maior a acurácia, maior a confiabilidade dos resultados) e da extensão dos cálculos (quanto maior a extensão dos cálculos, menor a confiabilidade dos resultados);
- Neurocomputadores: Embora executem cálculos analógicos, estando portanto sujeitos às mesmas limitações de confiabilidade dos computadores analógicos, a trajetória do estado interno dos neurocomputadores é apreendida em uma base local e atraída pela solução do problema (associada àquela base).

Computadores digitais



Computadores analógicos



Neurocomputadores

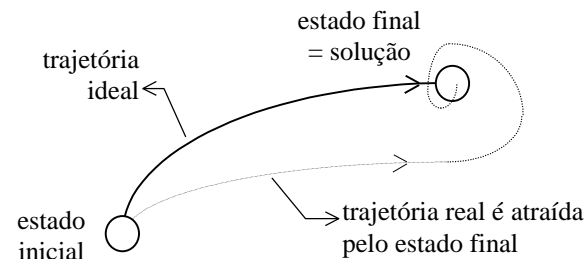


Figura 1 – Comparação entre computadores digitais, computadores analógicos e neurocomputadores (adaptado de PERETTO, 1992)

4 As Quatro Gerações de Redes Neurais Artificiais

- Na literatura, são identificadas quatro gerações de redes neurais artificiais:
 1. **Primeira Geração**: Fundamentada nos neurônios padrão MCCULLOCH & PITTS (1943), com ativação binária. Incluem redes MLP e de Hopfield, quando estas consideram tais ativações para seus neurônios. Apresentam capacidade de aproximação universal de funções booleanas.
 2. **Segunda Geração**: As redes desta geração empregam funções de ativação que assumem valores em intervalos contínuos. Incluem redes MLP, RBF e de Hopfield. Apresentam capacidade de aproximação universal de funções booleanas, funções contínuas de argumento compacto e dinâmicas lineares e não-lineares matematicamente representáveis. Admitem algoritmos de ajuste de conexões sinápticas via retropropagação. A interpretação biológica do grau de ativação está geralmente vinculada à frequência de pulsação dos neurônios biológicos.

3. **Terceira geração**: As redes neurais aqui empregam neurônios pulsantes, também conhecidos como neurônios de integração-disparo. Nas redes pulsantes, é possível codificar informação no intervalo de tempo entre pulsos e é válido considerar sincronismo de pulsos e defasagem entre trens de pulso (MAASS, 1997).
4. **Quarta geração**: Inclui redes neurais com organizações hierárquicas e dinâmicas oscilatórias que sintetizam os conjuntos-K de Freeman (FREEMAN, 1975; KOZMA, 2008). Em neurociência, a ideia de modulação de um sinal sensorial em diferentes frequências no sistema nervoso é bem conhecida e já foi comprovada em vários contextos (KELSO, 1995).
- A plausibilidade biológica tende a aumentar ao longo das gerações. Há correntes em neurofisiologia que defendem que o sistema nervoso é constituído de unidades osciladoras, e que também dentro da célula vários mecanismos diferentes, envolvendo genes ou não, podem atuar como osciladores (GOLDBETER, 1996).

- A hipótese sugerida é que, dentro da célula, assim como no sistema nervoso, a coordenação é regida por um vasto número de osciladores. Por exemplo, em FRANÇOIS & HAKIM (2004), mais de 10 possíveis estruturas ainda inéditas de osciladores genéticos e proteicos são propostas. Essas estruturas foram evoluídas artificialmente em computador.

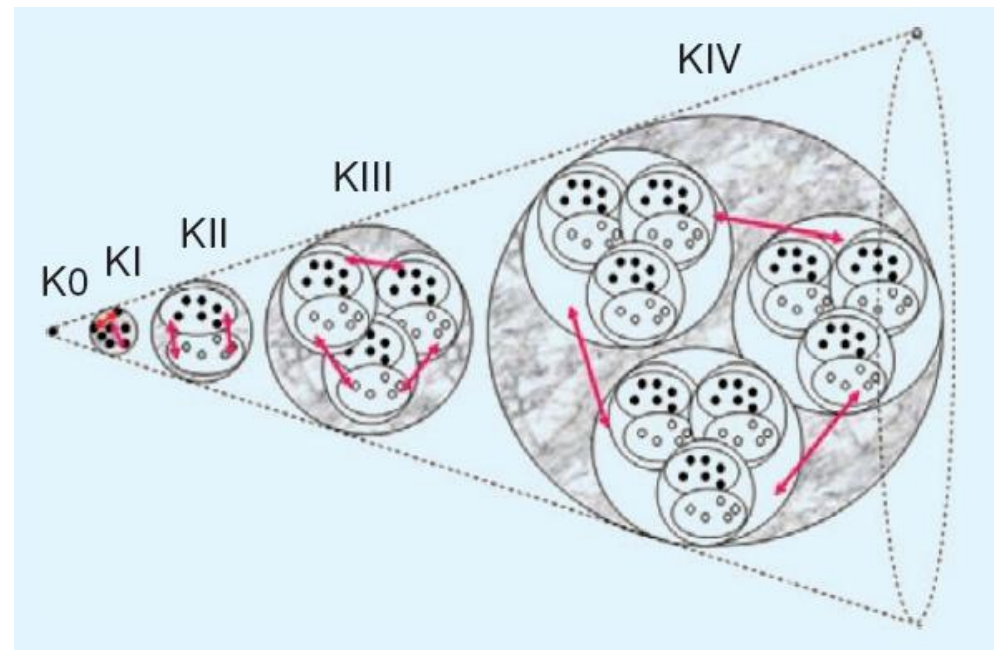


Figura 2 – Progressão de modelos de complexidade crescente, seguindo os níveis organizacionais do cérebro (figura extraída de KOZMA *et al.*, 2007).

5 Princípios Básicos de Sistemas Dinâmicos Não-Lineares

5.1 Introdução

- A teoria de sistemas dinâmicos se ocupa em descrever matematicamente sistemas que dependem do tempo, permitindo classificar e prever seu comportamento.
- O comportamento temporal de sistemas dinâmicos pode depender tanto de variáveis observáveis como de variáveis não-observáveis.
- Um sistema dinâmico consiste de duas partes: um estado e uma dinâmica.
- O estado descreve a condição atual do sistema, na forma de um vetor de variáveis parametrizadas em relação ao tempo, sendo que o conjunto de estados possíveis é denominado **espaço de estados** do sistema.
- A dinâmica descreve como o estado do sistema evolui no tempo, sendo que a sequência de estados exibida por um sistema dinâmico durante sua evolução no tempo é denominada **trajetória no espaço de estados**.

- Hipótese: a dinâmica é determinística (em oposição à estocástica), ou seja, para cada estado do sistema, a dinâmica especifica unicamente o próximo estado (dinâmica discreta) ou então a direção de variação do estado (dinâmica contínua).
- Neste caso, **um sistema dinâmico é uma prescrição matemática determinística para a evolução de um estado no tempo.**
- Entradas externas podem influir na determinação do próximo estado.

Tabela 1 – Taxonomia dos sistemas dinâmicos (KOLEN, 1994)

		ESPAÇO DE ESTADOS	
		contínuo	discreto
DINÂMICA	contínua	sistema de equações diferenciais	vidros de spin
	discreta	sistema de equações a diferenças	autômato

- Quando um sistema dinâmico não apresenta a propriedade de linearidade (princípio da superposição de efeitos) ele é denominado sistema dinâmico não-linear. Os sistemas físicos são inerentemente não-lineares.
- No entanto, quando a faixa de operação do sistema é pequena e as não-linearidades são suaves, um sistema dinâmico não-linear pode ser representado aproximadamente por seu correspondente sistema linearizado, cuja dinâmica é descrita por um conjunto de equações diferenciais ou a diferenças lineares.

5.2 Sistemas Não-Lineares Multidimensionais

- f_i, g_i ($i=1, \dots, n$) são funções não-lineares e n é finito (uma série de restrições devem ser impostas a estas funções, a fim de permitir o tratamento dos sistemas resultantes com base no conhecimento e nas ferramentas disponíveis)
- x_1, \dots, x_n : variáveis de estado (memória que o sistema tem do seu passado)
- u_1, \dots, u_p : entradas do sistema (externas e/ou por realimentação)

- Número finito de equações dinâmicas acopladas (tempo contínuo e discreto)

$$\bullet \quad \begin{cases} \dot{x}_1 = f_1(t, x_1, \dots, x_n, u_1, \dots, u_p) \\ \vdots \\ \dot{x}_n = f_n(t, x_1, \dots, x_n, u_1, \dots, u_p) \end{cases} \quad \begin{cases} x_1(k+1) = g_1(k, x_1(k), \dots, x_n(k), u_1(k), \dots, u_p(k)) \\ \vdots \\ x_n(k+1) = g_n(k, x_1(k), \dots, x_n(k), u_1(k), \dots, u_p(k)) \end{cases}$$

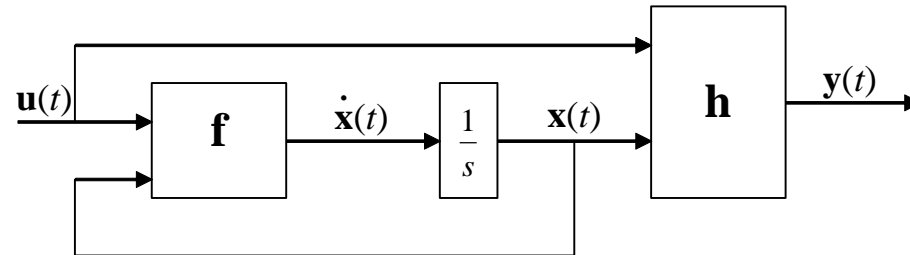
- Representar estas equações em forma mais compacta: notação vetorial

$$\bullet \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \quad \mathbf{f}(t, \mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_1(t, \mathbf{x}, \mathbf{u}) \\ \vdots \\ f_n(t, \mathbf{x}, \mathbf{u}) \end{bmatrix} \quad \mathbf{g}(k, \mathbf{x}, \mathbf{u}) = \begin{bmatrix} g_1(k, \mathbf{x}, \mathbf{u}) \\ \vdots \\ g_n(k, \mathbf{x}, \mathbf{u}) \end{bmatrix}$$

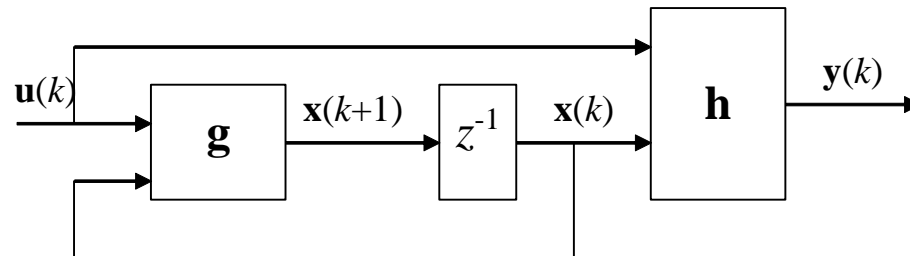
- Equações de estado: $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \quad \mathbf{x}(k+1) = \mathbf{g}(k, \mathbf{x}, \mathbf{u})$
- Equação de saída: $\mathbf{y} = \mathbf{h}(t, \mathbf{x}, \mathbf{u}) \quad \mathbf{y} = \mathbf{h}(k, \mathbf{x}, \mathbf{u})$
- $\mathbf{y} \in \mathbb{R}^p$ é denominado vetor de saída, e geralmente contém variáveis de interesse particular na análise de um sistema dinâmico, como variáveis que podem ser fisicamente medidas ou variáveis cujo comportamento deve ser monitorado ou controlado.

- Modelo de espaço de estados (ou modelo de estados) de um sistema não-linear

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$



$$\begin{cases} \mathbf{x}(k+1) = \mathbf{g}(k, \mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \mathbf{u}(k)) \end{cases}$$



- Quando é necessária a equação de saída? → Representação interna; variáveis de estado × variáveis de saída
- O modelo de espaço de estados é a forma mais geral que um sistema não-linear pode assumir em uma representação matemática. Existem infinitas formas de se representar o vetor de estados de um sistema dinâmico, sendo que algumas delas são chamadas de formas canônicas.

- Muito da eficiência da teoria de controle moderno (casos linear e não-linear) pode ser atribuída à generalidade e versatilidade dos modelos de espaço de estados. Mas nem todos os sistemas físicos de dimensão finita podem ser representados nesta forma.
- Efeito do uso de um modelo de estado para um sistema não-linear: força a análise do sistema sob o ponto de vista da interação entrada/estado e estado/saída (a entrada geralmente não vai atuar *diretamente* sobre a saída).
- \mathbf{f} e \mathbf{g} são também denominadas campos vetoriais e geralmente são consideradas funções suaves em seus argumentos, com derivadas parciais, de qualquer ordem, contínuas (Objetivo: garantir a existência e unicidade de solução). Uma restrição mais forte seria supor que as funções são analíticas em seu domínio de definição.
- Uma função é analítica quando ela é diferenciável em um ponto \mathbf{z}_0 de seu domínio e em uma vizinhança de \mathbf{z}_0 , de modo que ela pode ser representada por uma série de potência baseada em potências de $(\mathbf{z} - \mathbf{z}_0)$.

5.3 Análise de Sistemas Não-Lineares

- Envolve conceitos matemáticos mais avançados (análise funcional, geometria diferencial) que o caso linear (álgebra matricial), e o número de abordagens é muito maior que no caso linear, já que as ferramentas são predominantemente específicas ou não-universais.
- Quando formos estudar redes neurais de Hopfield, nos restringiremos a equações de estado não-forçadas e invariantes no tempo (sistema autônomo):

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \qquad \mathbf{x}(k+1) = \mathbf{g}(\mathbf{x}(k))$$

5.4 Exemplos de Comportamentos Dinâmicos Não-Lineares

- A dinâmica de um sistema não-linear é muito mais rica que a de um sistema linear, ou seja, há fenômenos dinâmicos significativos que só ocorrem na presença de não-linearidades, não podendo assim serem descritos ou preditos por modelos lineares.

Exemplos de fenômenos essencialmente não-lineares:

- **Tempo de escape finito**: O estado de um sistema linear instável vai para infinito quando o tempo tende a infinito. No entanto, existem sistemas não-lineares instáveis cujo estado pode ir para infinito em tempo finito.
- **Múltiplos pontos de equilíbrio isolados**: Um sistema autônomo linear vai apresentar apenas um ponto de equilíbrio isolado, indicando a existência de apenas um ponto de operação em estado estacionário, o qual atrai o estado do sistema, independente da condição inicial. Já no caso não-linear, podem existir múltiplos pontos de equilíbrio isolados. Com isso, *o ponto de operação em estado estacionário vai depender da condição inicial*.
- **Ciclos limites**: Para um sistema linear invariante no tempo apresentar oscilação permanente, ele deve apresentar um par de autovalores no eixo imaginário. Esta condição é uma impossibilidade prática na presença de perturbações. Mesmo

considerando apenas a possibilidade teórica, a *amplitude da oscilação vai depender da condição inicial*. **Na prática, oscilações estáveis só podem ser produzidas por sistemas não-lineares.** Há sistemas não-lineares que atingem oscilações de amplitude e frequência fixas, independente da condição inicial. Este tipo de oscilação é denominada ciclo limite.

- **Oscilações sub-harmônicas, harmônicas e quase-periódicas**: Um sistema linear estável, sujeito a uma entrada periódica, produz uma saída de mesma frequência. Um sistema não-linear, sujeito a uma excitação periódica pode oscilar com frequências que são sub-múltiplos ou múltiplos da frequência de excitação. Pode ser gerada inclusive uma quase-oscilação, formada pela soma de oscilações periódicas cujas frequências não são múltiplos entre si.
- **Caos**: Um sistema não-linear pode apresentar um comportamento de estado estacionário que não é equilíbrio, nem oscilação periódica, nem oscilação quase-periódica, sendo denominado caos (OTT, 1993).

- **Múltiplos modos de comportamento:** É comum que múltiplos modos de comportamento dinâmico, dentre os descritos acima, possam ser exibidos por um mesmo sistema dinâmico não-linear, mesmo sem a presença de excitação. Com excitação, as mudanças de modo de comportamento podem ser descontínuas em relação a mudanças suaves na amplitude e frequência da excitação.

5.5 Estado Estacionário em Sistemas Não-Lineares Autônomos

- Nesta seção, os **sistemas dinâmicos autônomos** são classificados em termos de seu comportamento de estado estacionário (ou de regime), um conceito definido para $t \rightarrow \infty$.
- Embora haja uma diferença qualitativa marcante entre sistemas dinâmicos autônomos de tempo discreto e contínuo, *ambos exibem os mesmos tipos de comportamento dinâmico em estado estacionário*. Com isso, todos os resultados desta seção são **válidos tanto para tempo contínuo como para tempo discreto**.

5.5.1 Definição de Trajetória

Definição 1: A sequência de estados exibida por um sistema dinâmico durante sua evolução no tempo é denominada trajetória.

- Embora um mesmo sistema dinâmico possa experimentar trajetórias iniciais completamente diferentes, estas trajetórias geralmente convergem para um comportamento característico quando $t \rightarrow \infty$, conhecido como comportamento de estado estacionário (ou de regime).
- O estado estacionário corresponde ao comportamento assintótico de um sistema quando $t \rightarrow \infty$, tendo sentido estudar apenas estados estacionários finitos.

Definição 2: Para um dado sistema dinâmico, a diferença entre sua trajetória de estado e seu comportamento de estado estacionário é denominada transiente ou transitório.

- Como os resultados a seguir são válidos tanto para dinâmica discreta como para dinâmica contínua, vamos definir aqui uma notação de trajetória que seja comum para os dois casos (uma formalidade necessária para os estudos que serão considerados na sequência).

Definição 3: Seja para o sistema de tempo contínuo $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t))$, $\mathbf{x}(t_0) = \mathbf{x}_0$, ou então para o sistema de tempo discreto $\mathbf{x}(k+1) = \mathbf{g}(\mathbf{x}(k))$, $\mathbf{x}(0) = \mathbf{x}_0$, a solução vai depender explicitamente da condição inicial, sendo denominada fluxo e denotada por $\phi_t(\mathbf{x}_0)$. Embora, para o caso discreto, a notação mais adequada fosse $\phi_k(\mathbf{x}_0)$, vamos utilizar t em lugar de k , deste ponto em diante, seja para tempo contínuo ou discreto.

Definição 4: O conjunto de pontos $\{\phi_t(\mathbf{x}_0) : -\infty < t < +\infty\}$ é denominado trajetória através de x_0 .

Definição 5: Um atrator é uma região do espaço de estados para onde as trajetórias convergem a partir de uma região maior do espaço de estados. A região do espaço de estados a partir da qual o sistema evolui para o atrator é denominada base de atração.

- Estudo de conjuntos limites atratores é de interesse especial já que conjuntos limites não-atratores (repulsores) não podem ser observados em sistemas físicos ou mesmo em simulações.
- No caso de sistemas lineares assintoticamente estáveis, *existe um único conjunto limite e a base de atração do conjunto limite é todo o espaço de estados*, ou seja, o estado estacionário é independente da condição inicial do sistema. Por outro lado, sistemas não-lineares típicos podem apresentar vários conjuntos limites, com bases de atração distintas. Neste caso, a condição inicial vai determinar qual dos conjuntos limites será alcançado pelo estado estacionário.

- A seguir, são apresentados quatro tipos distintos de atratores, em ordem crescente de complexidade: ponto fixo, soluções periódicas, soluções quase-periódicas e caos.

5.5.2 Pontos Fixos (ou Pontos de Equilíbrio)

Definição 6: Um ponto fixo (ou ponto de equilíbrio) \bar{x} de um sistema autônomo, dado pela equação $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t))$, $\mathbf{x}(t_0) = \mathbf{x}_0$ ou $\mathbf{x}(k+1) = \mathbf{g}(\mathbf{x}(k))$, $\mathbf{x}(0) = \mathbf{x}_0$ é uma solução constante $\bar{\mathbf{x}} = \phi_t(\bar{\mathbf{x}})$, para todo t .

- Assim, diz-se que $\bar{\mathbf{x}}$ é um ponto de equilíbrio de um sistema dinâmico se, uma vez que o estado do sistema atinge $\bar{\mathbf{x}}$, o estado permanece em $\bar{\mathbf{x}}$ indefinidamente.
- Para sistemas invariantes no tempo, pontos de equilíbrio são caracterizados pela solução de equações algébricas:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) \leftrightarrow 0 = \mathbf{f}(\bar{\mathbf{x}})$$

$$\mathbf{x}(k+1) = \mathbf{g}(\mathbf{x}(k)) \leftrightarrow \bar{\mathbf{x}} = \mathbf{g}(\bar{\mathbf{x}})$$

- Um ponto de equilíbrio pode então ser visto como um ponto no espaço de estados que é invariante sob a dinâmica associada ao sistema.
- Existem três variedades de pontos fixos, **pontos estáveis**, **pontos instáveis** e **pontos de sela**, definidas como segue:

Ponto fixo estável ou atrator: Dado um ponto fixo representando o estado atual de um sistema dinâmico, então pequenas perturbações impostas ao estado do sistema resultam em uma trajetória que retorna ao ponto fixo após um transiente inicial. Ou seja, a distância entre o estado do sistema (em uma vizinhança do ponto fixo) no instante t e o ponto fixo estável vai se tornando arbitrariamente pequena para t suficientemente elevado.

Ponto fixo instável ou repulsor: Nem todos os pontos fixos atraem seus estados vizinhos. Os pontos fixos instáveis repelem os estados vizinhos. Se o estado de um sistema no instante t_1 coincide com um ponto fixo instável, o sistema vai permanecer neste estado para todo $t > t_1$. No entanto, qualquer perturbação junto a este estado de

equilíbrio instável faz com que a dinâmica associada ao sistema mova seu estado no sentido de afastá-lo do ponto fixo instável.

Ponto de sela: Entre os pontos fixos estáveis e instáveis existem os pontos de sela. Estes pontos do espaço de estados exibem propriedades de atração e repulsão, dependendo da direção de aproximação dos estados vizinhos. Portanto, pontos de sela só podem ocorrer em sistemas dinâmicos cujo espaço de estados apresenta mais de uma dimensão.

5.5.3 Soluções Periódicas

Definição 7: $\phi_t(\mathbf{x}^*)$ é uma solução periódica de um sistema autônomo se, para todo t ,

$$\phi_t(\mathbf{x}^*) = \phi_{t+T}(\mathbf{x}^*)$$

para algum período mínimo $T > 0$. Este resultado é válido para qualquer estado \mathbf{x}^* pertencente à solução periódica, sendo que a escolha de um estado \mathbf{x}^* específico implica apenas na fixação da origem do tempo.

- A restrição $T > 0$ é necessária para prevenir a classificação de um ponto fixo como solução periódica. Enquanto os pontos fixos estão associados à ausência de movimento, as soluções periódicas correspondem a um sistema em movimento constante e uniforme, que visita periodicamente um conjunto de estados.

Definição 8: Uma solução periódica é dita ser isolada se ela possui uma vizinhança que não contém nenhuma outra solução periódica. Uma solução periódica isolada é comumente denominada ciclo limite.

- Oscilações sustentadas também podem ocorrer em sistemas lineares, mas estas não correspondem a ciclos limites pois as oscilações em sistemas lineares dependem da condição inicial, não são isoladas e são muito sensíveis a qualquer tipo de mudança nos parâmetros do sistema correspondente.
- Os ciclos limites, da mesma forma que os pontos fixos, podem atrair estados vizinhos, repeli-los, ou apresentar uma combinação destas duas condições. Ciclos

limites atratores produzem oscilações que são estáveis em relação a pequenas perturbações. Perturbações junto a ciclos limites repulsores acabam movendo o sistema em direção a outros atratores.

5.5.4 Soluções Quase-Periódicas

Definição 9: Uma função quase-periódica é tal que pode ser expressa por uma soma de p funções periódicas na forma:

$$x(t) = \sum_{i=1}^p h_i(t),$$

onde h_i ($i=1,...,p$) tem período mínimo T_i e frequência $f_i = 1/T_i$ e a razão entre quaisquer pares de períodos leva a um número irracional.

- Para $p = 2$, considere a seguinte função no tempo:

$$x(t) = h_1(t) + h_2(t),$$

onde h_1 e h_2 são funções periódicas arbitrárias com períodos T_1 e T_2 , respectivamente.

- A função $x(t)$ é quase-periódica se T_1 e T_2 forem tais que T_1/T_2 é um número irracional. Caso contrário, existiriam dois inteiros q e r tais que $qT_1 = rT_2$, fazendo com que $x(t)$ fosse periódica, com período $T = qT_1$.
- Soluções quase-periódicas também podem representar regimes atratores, repulsores, ou uma combinação destes.

5.5.5 Caos

- Não há uma definição geral para caos que seja aceita em toda sua extensão nas mais diversas áreas envolvidas no estudo deste fenômeno não-linear. Em termos práticos, como a trajetória de um sistema dinâmico caótico é aperiódica, pode-se atribuir comportamento caótico a todo comportamento estacionário que não possa ser inserido em nenhum dos três casos anteriores: pontos fixos, soluções periódicas ou soluções quase-periódicas.

- É certo que sistemas dinâmicos lineares não podem exibir comportamento caótico, ou seja, atratores ou repulsores caóticos só ocorrem no caso de sistemas dinâmicos não-lineares.
- Também é sabido que caos não ocorre caso a ordem do sistema dinâmico seja inferior a 3, no caso de dinâmica contínua, ou 2, no caso de dinâmica discreta (se o mapeamento for não-inversível, então ordem 1 basta).
- 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, 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.

- Pode-se dizer que a dinâmica caótica começou a ser estudada a partir dos trabalhos do matemático francês Henri Poincaré, por volta da virada do século passado. Um dos problemas analisado por Poincaré correspondia à descrição das órbitas de três corpos celestiais (uma estrela e dois planetas) experimentando atração gravitacional mútua. Um aspecto importante foi o fato de Poincaré analisar 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.
- Um sistema dinâmico caótico converge para um comportamento de regime, mas que não gera uma trajetória periódica nem quase-periódica. À primeira vista, a dinâmica caótica se assemelha a muitos processos aleatórios, não seguindo nenhum tipo de padrão.

- No entanto, sistemas caóticos são descritos por equações determinísticas, enquanto que um processo inerentemente aleatório pode ser caracterizado apenas em termos de propriedades estatísticas.
- A questão é que não é possível realizar previsões de longo prazo para uma trajetória caótica, ou seja, o horizonte de previsibilidade cresce apenas com o logaritmo da precisão de medida realizada junto à condição inicial: sensibilidade à condição inicial (ou a uma pequena perturbação que ocorra a qualquer instante de tempo), dada por um valor positivo para o expoente de Lyapunov (WOLF *et al.*, 1985; ECKMANN *et al.*, 1986; MCCAFFREY *et al.*, 1992).
- O expoente de Lyapunov mede a taxa com que trajetórias vizinhas no espaço de estados convergem ou divergem. Há tantos expoentes de Lyapunov quanto dimensões, sendo que o maior expoente é geralmente o mais importante.

- Pelo exposto acima, fica evidente que caos não tem a ver com desordem (ao menos em dinâmica não-linear). Trata-se de um **comportamento dinâmico não-linear, determinístico, mas imprevisível a longo prazo.**
- Estes aspectos gerais associados ao comportamento caótico conduzem aos três requisitos matemáticos utilizados para demonstrar que um sistema dinâmico apresenta comportamento caótico:
 - sensibilidade à condição inicial;
 - ergodicidade (dado um período prolongado de tempo, o tempo de permanência da trajetória em uma dada região do espaço de estados é proporcional ao volume da região, ou seja todos os estados acessíveis pela trajetória são igualmente prováveis ao longo de um período de tempo prolongado);
 - presença de um conjunto denso de pontos pertencentes a soluções periódicas instáveis.
- 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). É possível obter atratores caóticos que não são estranhos.

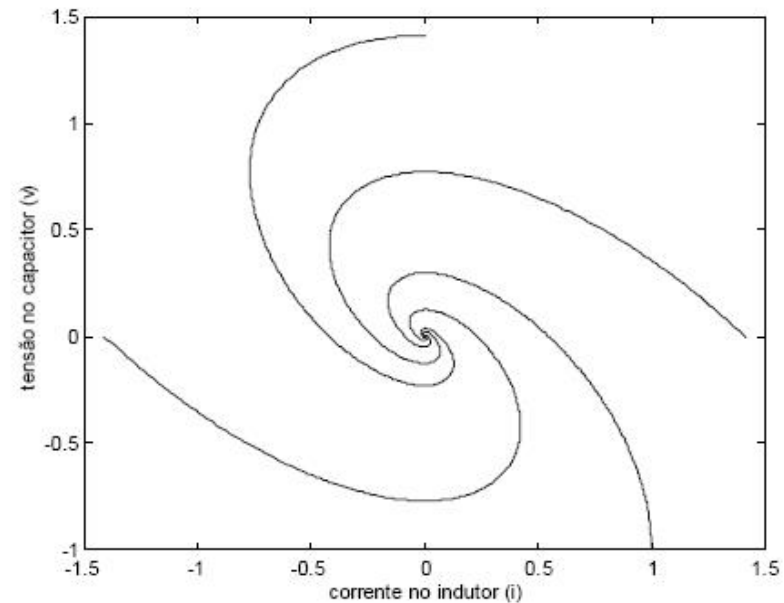
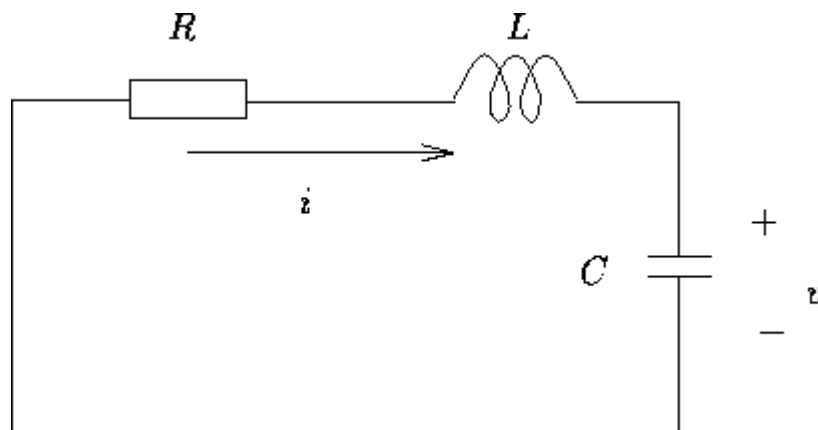
- Da mesma forma que atratores periódicos podem ser diferenciados por seus períodos, atratores estranhos podem ser diferenciados por sua dimensão. Pontos fixos e ciclos limites de tempo discreto são atratores de dimensão zero, pois são constituídos apenas por um número finito de pontos. Ciclos limites de tempo contínuo são atratores de dimensão inteira. Por exemplo, quando o ciclo limite é uma curva que se liga em suas extremidades, sua dimensão é um. Já atratores caóticos podem apresentar dimensão fracionária.
- Observação: Existem métodos para o cálculo da dimensão de um atrator estranho e também há atratores estranhos que não são produzidos por dinâmicas caóticas.
- É possível que computadores simulem comportamentos caóticos? (BINDER & JENSEN, 1986)
- Resposta simplificada: quantitativamente não, pois a representação numérica aproximada da máquina cria uma impossibilidade teórica (e, portanto, prática), mas qualitativamente sim (o atrator tem as mesmas propriedades básicas).

6 Exemplos de Comportamentos Dinâmicos: Retrato de Fase

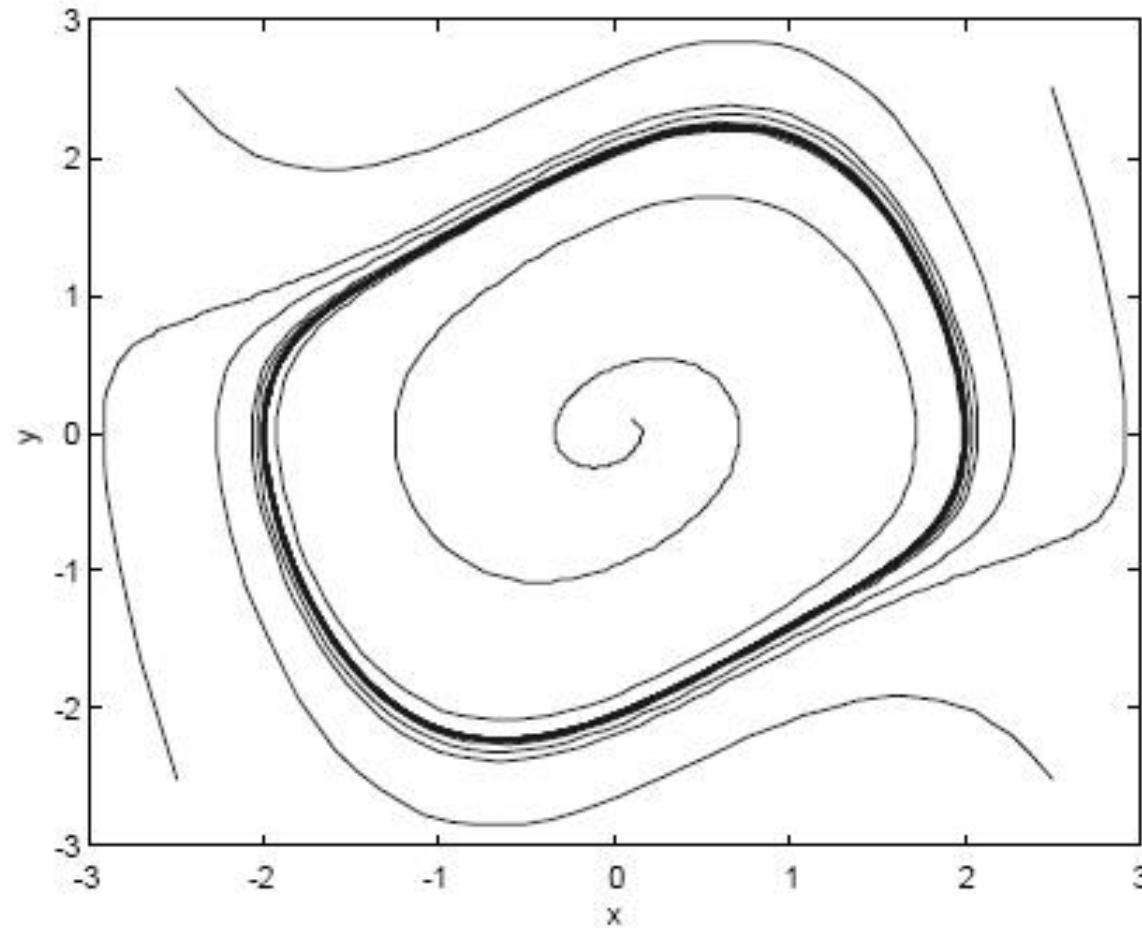
- Nota: Boa parte das figuras desta seção foram extraídas de ATTUX (2001).

Ponto de equilíbrio estável: Circuito RLC série autônomo

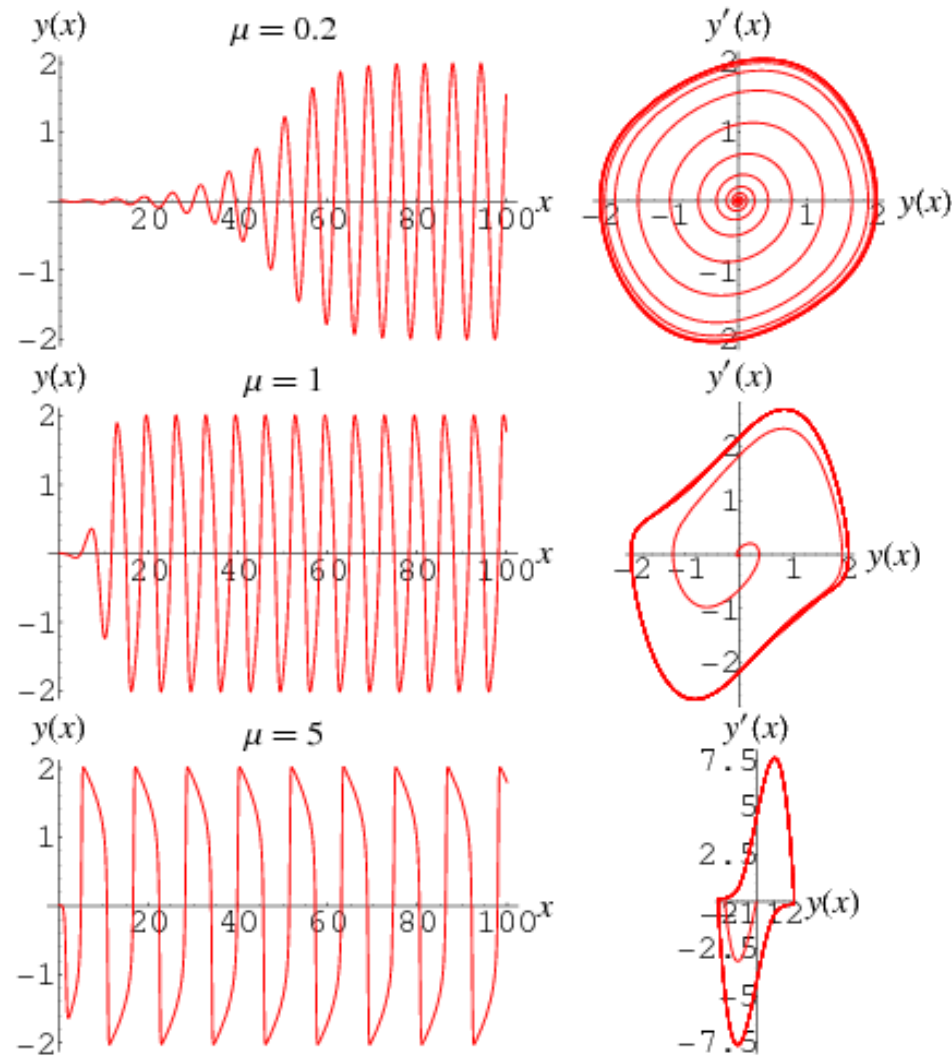
$$\begin{cases} \frac{di}{dt} = -\frac{1}{L}(Ri + v) \\ \frac{dv}{dt} = \frac{i}{C} \end{cases}, \text{ com } L = C = R = 1, \text{ produzindo um foco estável.}$$



Ciclo limite estável: $\begin{cases} \dot{x} = y \\ \dot{y} = -\lambda(1-x^2)y - x \end{cases}$, com $\lambda = 0,5$.

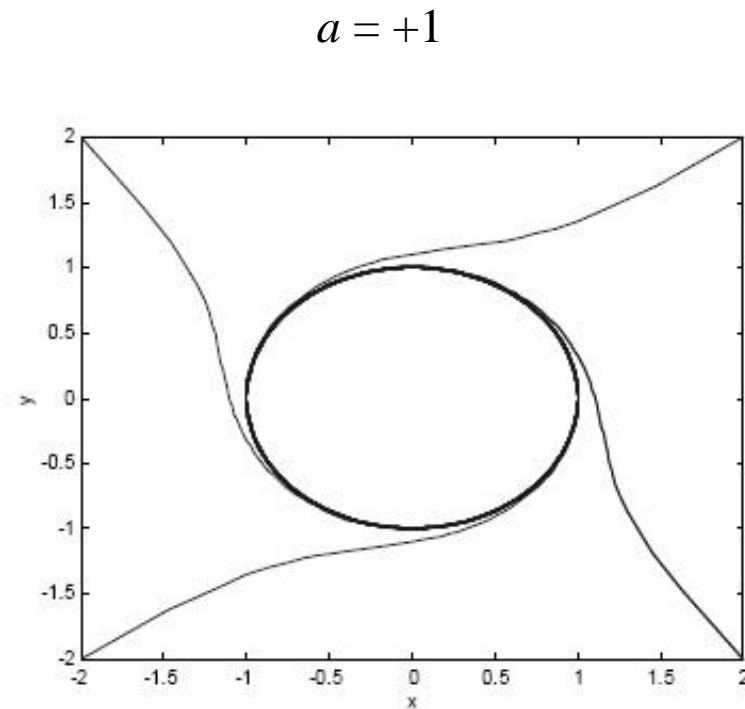
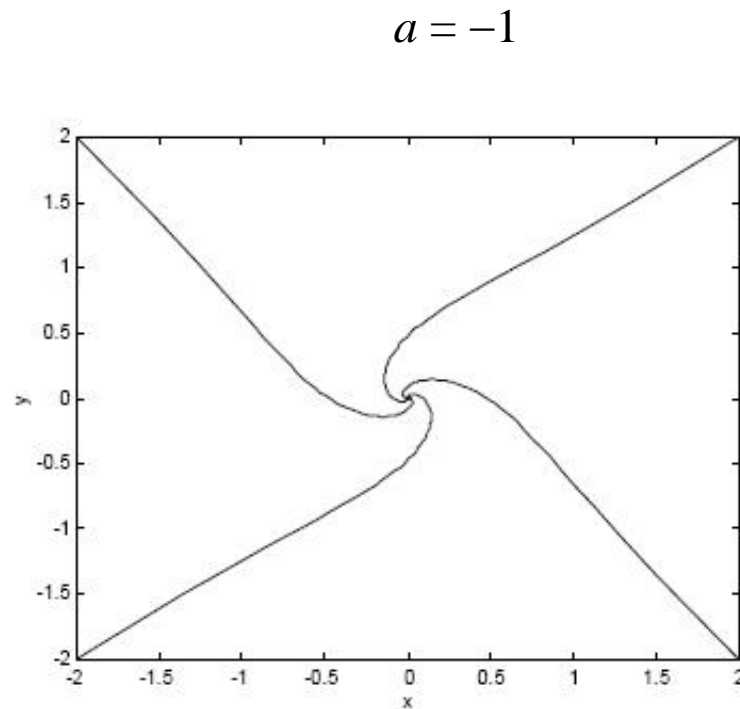


Oscilador de van der Pol: $\ddot{y} - \mu(1 - y^2)\dot{y} + y = 0$



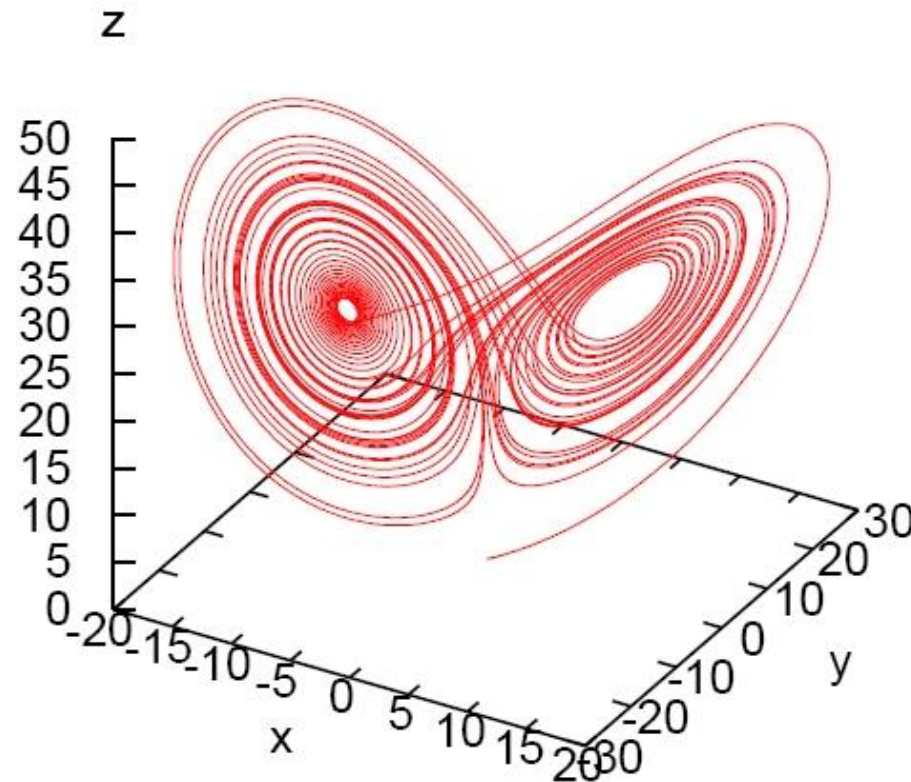
Bifurcação no comportamento do sistema dinâmico quando se varia um parâmetro:

$$\begin{cases} \dot{x} = -y + x(a - x^2 - y^2) \\ \dot{y} = x + y(a - x^2 - y^2) \end{cases}, \text{ passa-se de ponto de equilíbrio a ciclo limite quando } a = 0.$$



Comportamento caótico: Equação de Lorentz,
$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = -xz + rx - y, \text{ com } \sigma = 10, r = 28, \\ \dot{z} = xy - bz \end{cases}$$

 $b = 8/3$ e condição inicial $x(0) = y(0) = z(0) = 0,1$.

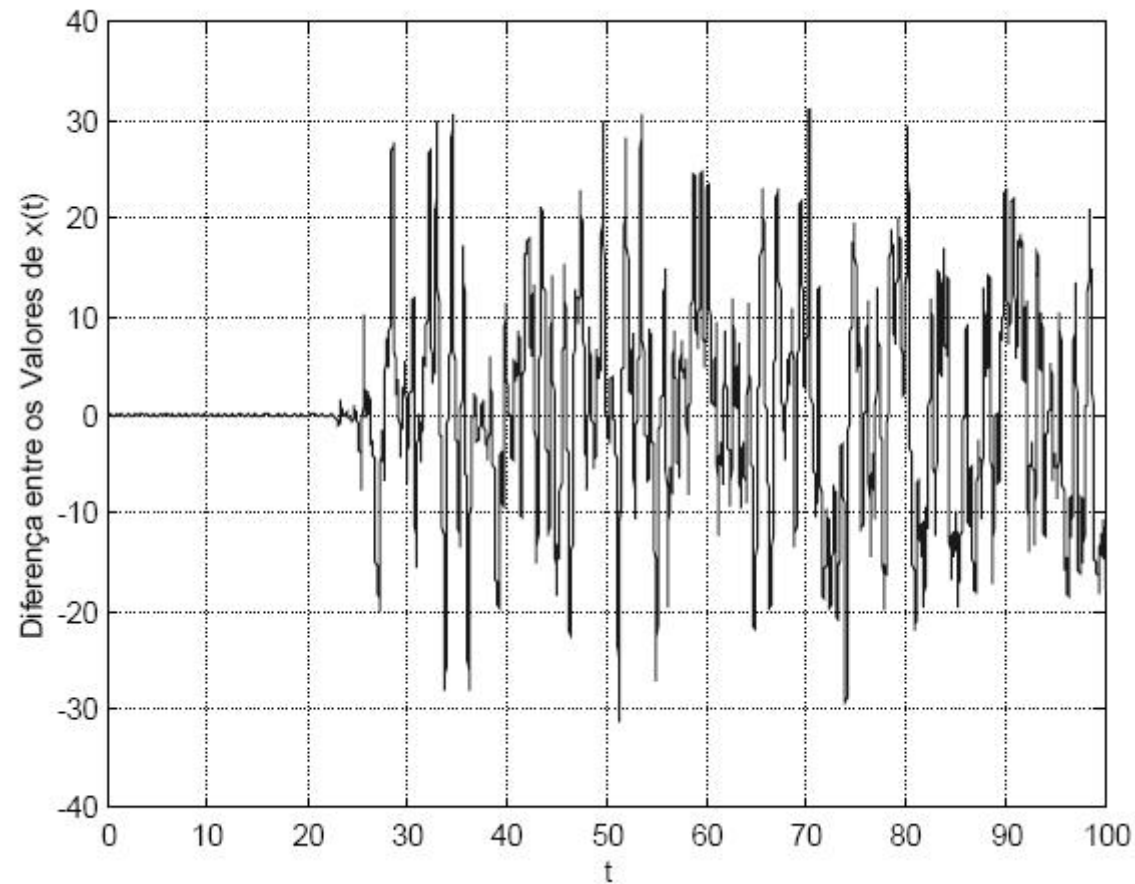


Sensibilidade à condição inicial:

Duas condições iniciais distintas:

$$x(0) = 0,25; y(0) = 2,1; z(0) = 1,1$$

$$x(0) = 0,25; y(0) = 2,1; z(0) = 1,100001$$



Mapa de Hénon: $\begin{cases} x_{k+1} = y_k + 1 - ax_k^2 \\ y_{k+1} = bx_k \end{cases}$, com $a = 1,4$ e $b = 0,3$.

A dimensão do atrator estranho é de aproximadamente 1,25.



Diagrama de bifurcação para o mapeamento logístico, com $x_0 = 0,5$.

$$x_{k+1} = r \cdot x_k \cdot (1 - x_k)$$

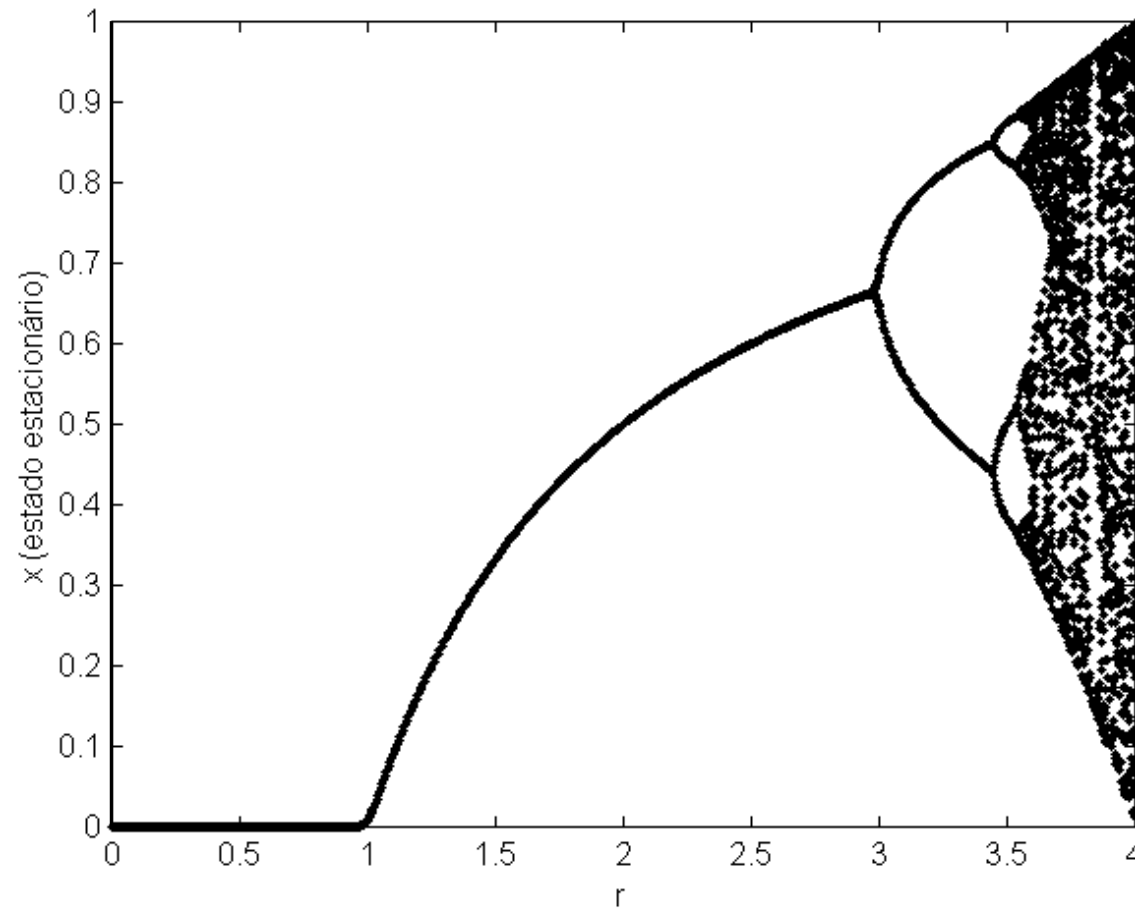


Diagrama de teia (*cobweb*) para $x_{k+1} = r \cdot x_k \cdot (1 - x_k)$, com $r = 3,3$.

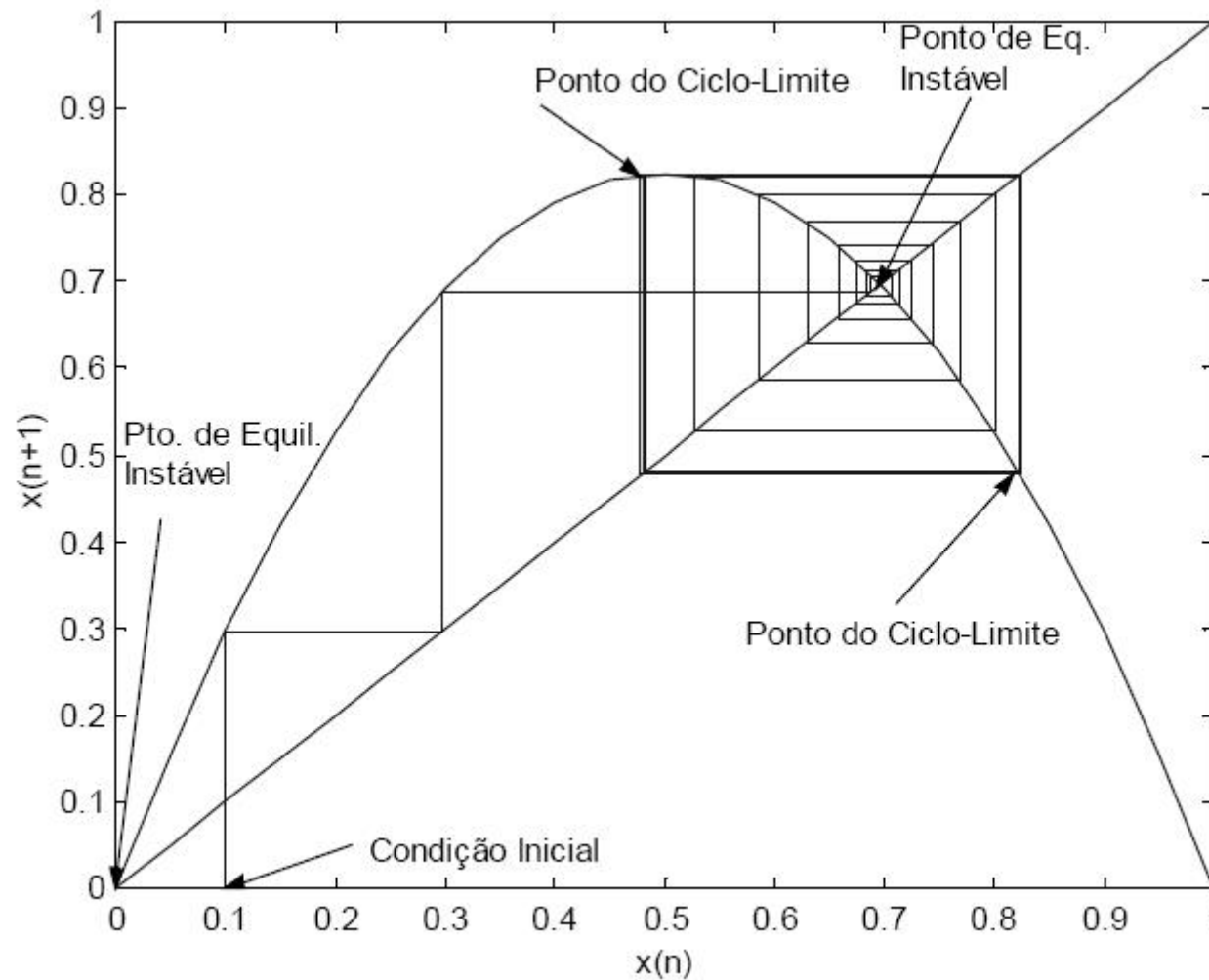
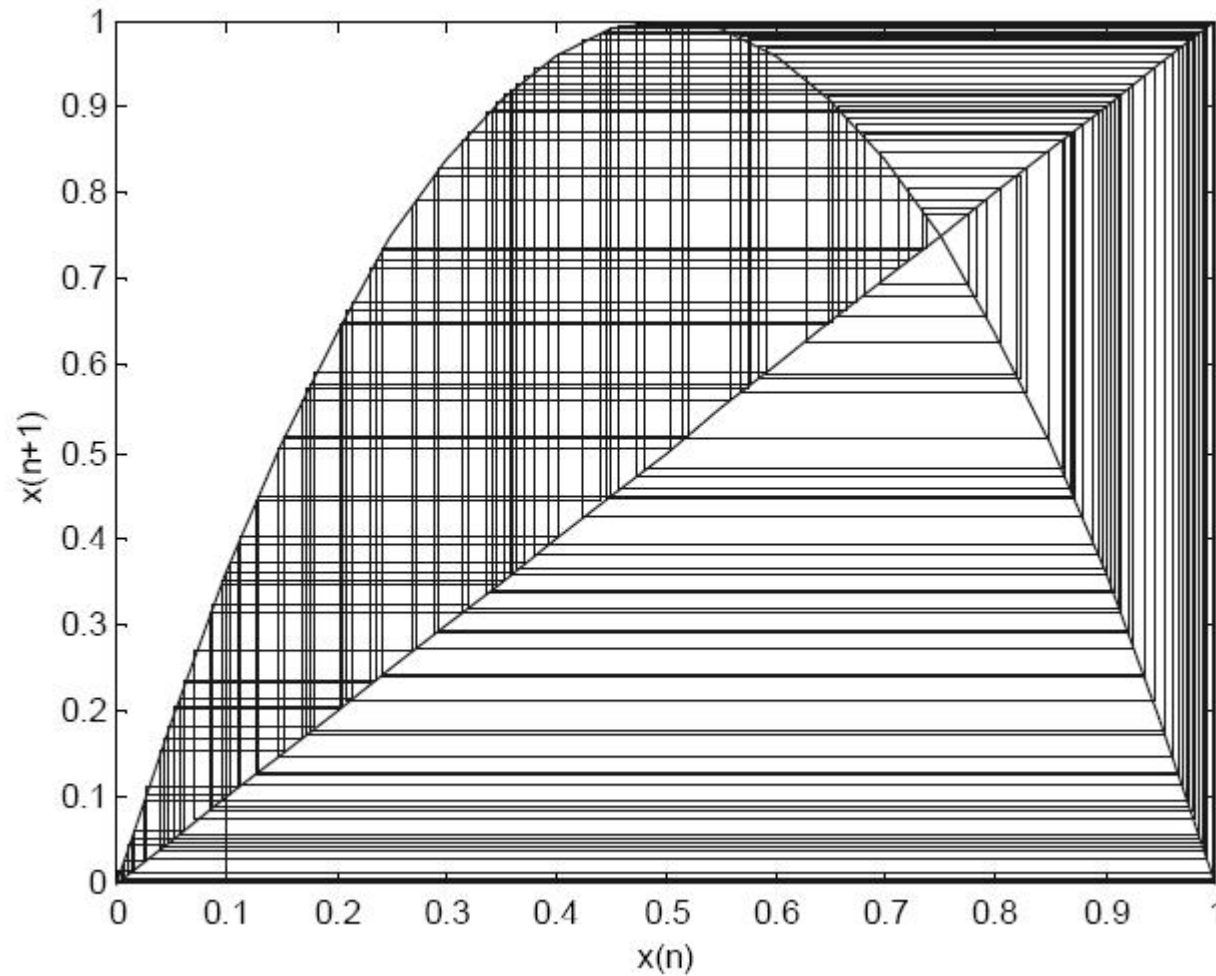


Diagrama de teia (cobweb) para $x_{k+1} = r \cdot x_k \cdot (1 - x_k)$, com $r = 4,4$.



7 Modelagem de Sistemas Dinâmicos Lineares

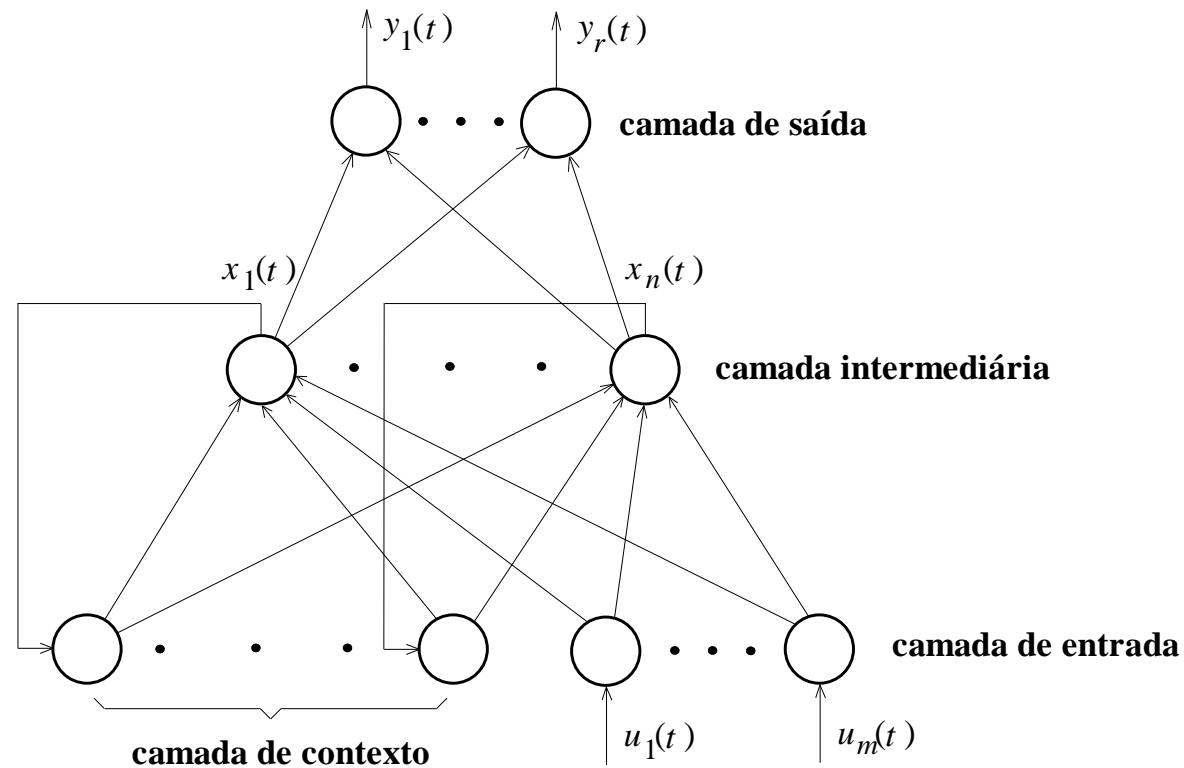


Figura 3 – Estrutura detalhada da Rede de Elman (ELMAN, 1990)

- $$\begin{cases} \mathbf{x}(t) = W_{xx}\mathbf{x}(t-1) + W_{xu}\mathbf{u}(t-1) \\ \mathbf{y}(t) = W_{yx}\mathbf{x}(t) \end{cases} \quad (\text{aproxima qualquer dinâmica linear})$$

8 Modelagem de Sistemas Dinâmicos Não-Lineares

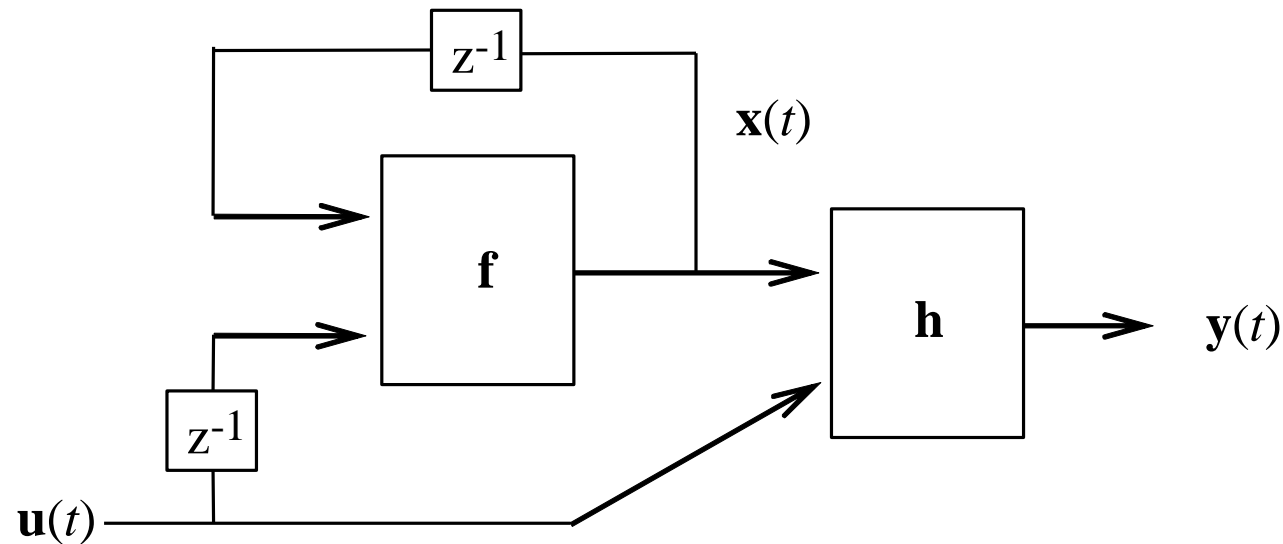


Figura 4 – Representação por espaço de estados de um sistema dinâmico não-linear

$$\begin{cases} \mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

onde $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{y}(t) \in \mathbb{R}^r$, $\mathbf{f}: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^n$ e $\mathbf{h}: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^r$.

9 Excursão por Outras Arquiteturas de Redes Neurais Recorrentes

- Redes neurais recorrentes são estruturas de processamento capazes de representar uma grande variedade de comportamentos dinâmicos (TSOI & BACK, 1994 e TSOI & BACK, 1997).
- A presença de realimentação de informação permite a criação de representações internas e dispositivos de memória capazes de processar e armazenar informações temporais e sinais sequenciais. A presença de conexões recorrentes ou realimentação de informação pode conduzir a comportamentos complexos, mesmo com um número reduzido de parâmetros.
- Como estruturas de processamento de sinais, redes neurais recorrentes se assemelham a filtros não-lineares com resposta ao impulso infinita (NERRAND *et al.*, 1993).

- As Figuras 5 a 7 foram melhor estudadas em DOS SANTOS & VON ZUBEN (2000).

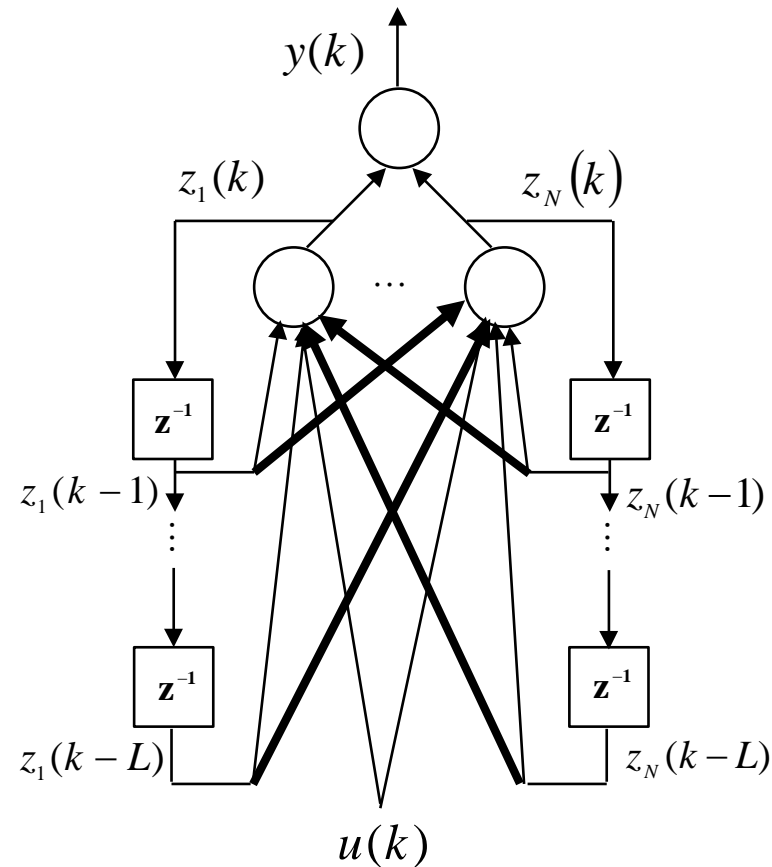


Figura 5 – Globally recurrent neural network (GRNN)

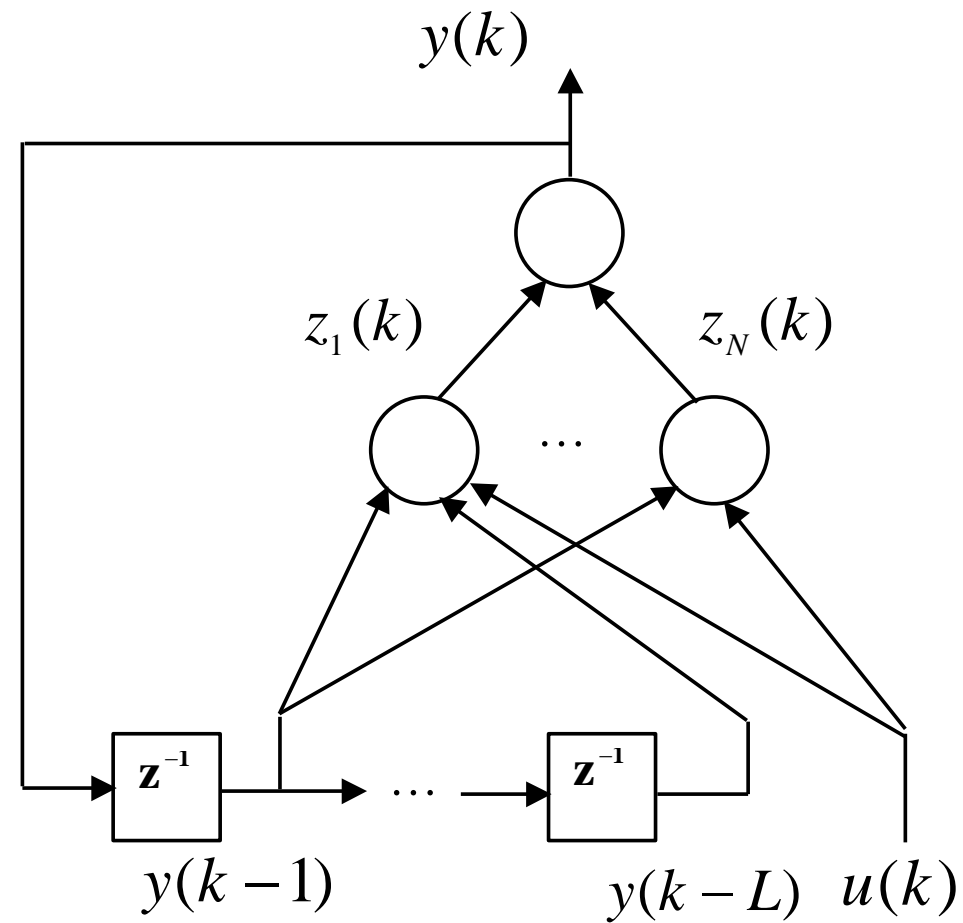


Figura 6 – Output-feedback recurrent neural network (OFRNN)

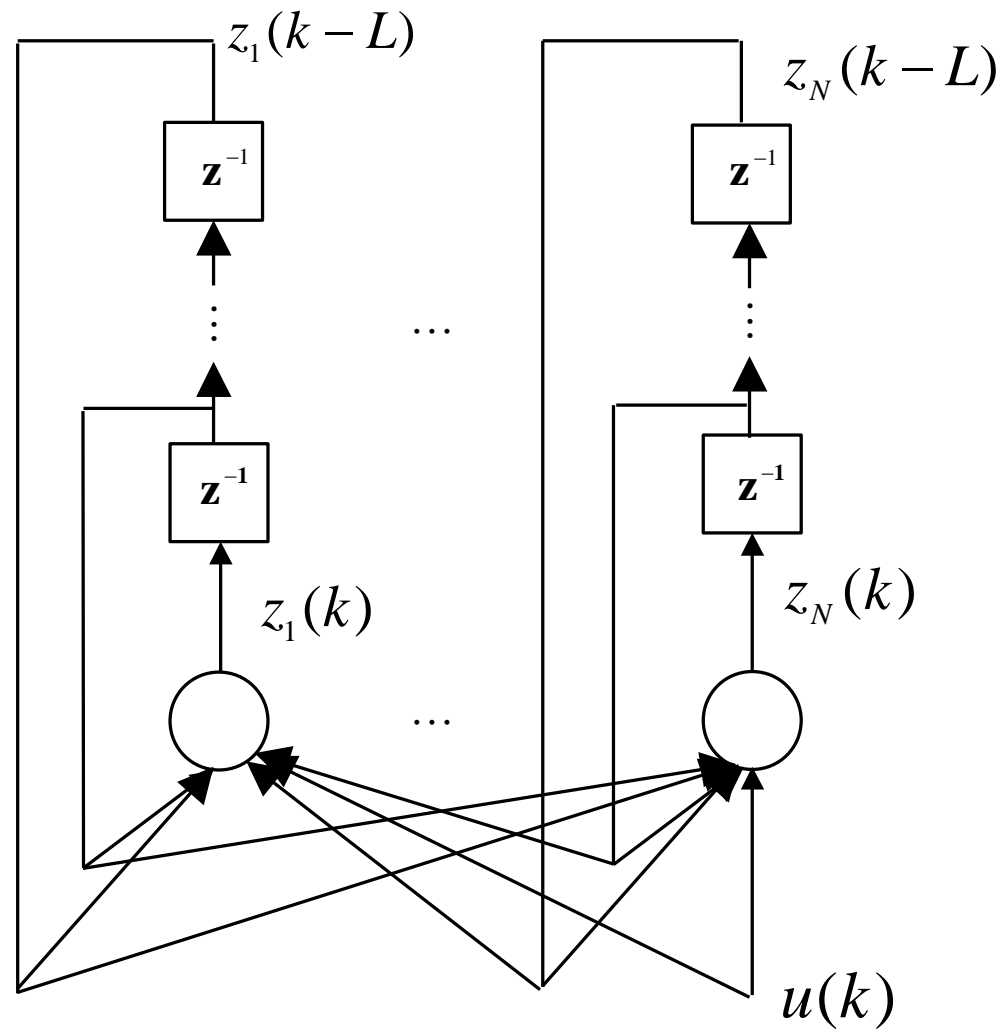


Figura 7 – Fully recurrent neural network (FRNN)

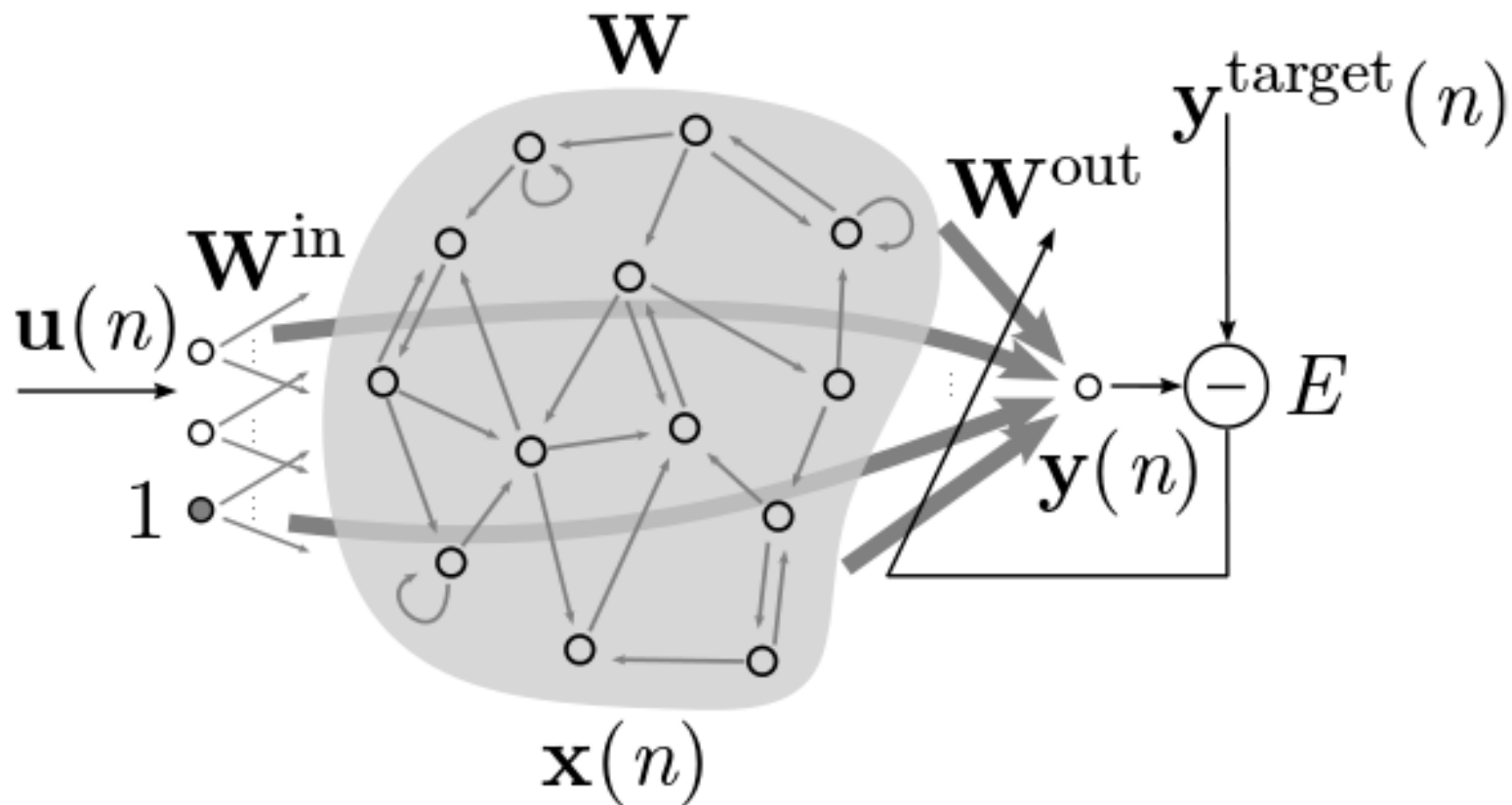


Figura 8 – *Echo state neural network* (JAEGER, 2001; OZTURK *et al.*, 2007)
(pesos da parte dinâmica da rede neural, destacada em cinza, não são ajustáveis)

10 O bloco LSTM (*Long Short-Term Memory*)

- Esta seção está baseada no paper de GREFF et al. (2017).
- Dois importantes desafios associados a boa parte das arquiteturas de redes neurais recorrentes são:
 - (1) Manutenção da flexibilidade do prazo da memória;
 - (2) Disposição de métodos eficientes para o ajuste de pesos sinápticos no caso de memórias de longo prazo.
- Como propor uma estrutura com recorrência, genérica e flexível o suficiente para modelar simultaneamente memórias de longo e de curto prazo?
- Como promover ajustes de pesos eficientes no caso de memórias de longo prazo?
- Atualmente, quem melhor responde a essas duas perguntas é o bloco LSTM (*Long Short-Term Memory*), que corresponde a uma célula de memória dotada de três portas analógicas sintonizáveis, tendo sido originalmente proposta por HOCHREITER & SCHMIDHUBER (1995) e HOCHREITER & SCHMIDHUBER (1997).

- Redes neurais recorrentes com blocos LSTM são estado-da-arte em aplicações envolvendo reconhecimento de escrita à mão (DOETSCH et al., 2014; GRAVES et al., 2009), geração de escrita à mão (GRAVES, 2014), tradução de linguagem (LUONG et al., 2015), modelagem acústica da fala (SAK et al., 2014), síntese de fala (FAN et al., 2014), análise de áudio (MARCHI et al., 2014), análise de vídeo (DONAHUE et al., 2017) e predição de estrutura secundária de proteínas (SONDERBY & WINTHER, 2015).
- A Figura 9 apresenta o bloco LSTM e o contrasta com um neurônio recorrente simples. Ele não corresponde à versão original de HOCHREITER & SCHMIDHUBER (1995), mas sim a uma versão estendida proposta 10 anos depois, em GRAVES & SCHMIDHUBER (2005). A Figura 10 contém aspectos fundamentais da notação envolvida, sendo que ambas essas figuras foram extraídas de GREFF et al. (2017).
- O treinamento do bloco LSTM se dá por técnicas como *backpropagation through time* (BPTT) (ver seção 13) e algoritmos evolutivos (SCHMIDHUBER et al., 2007).

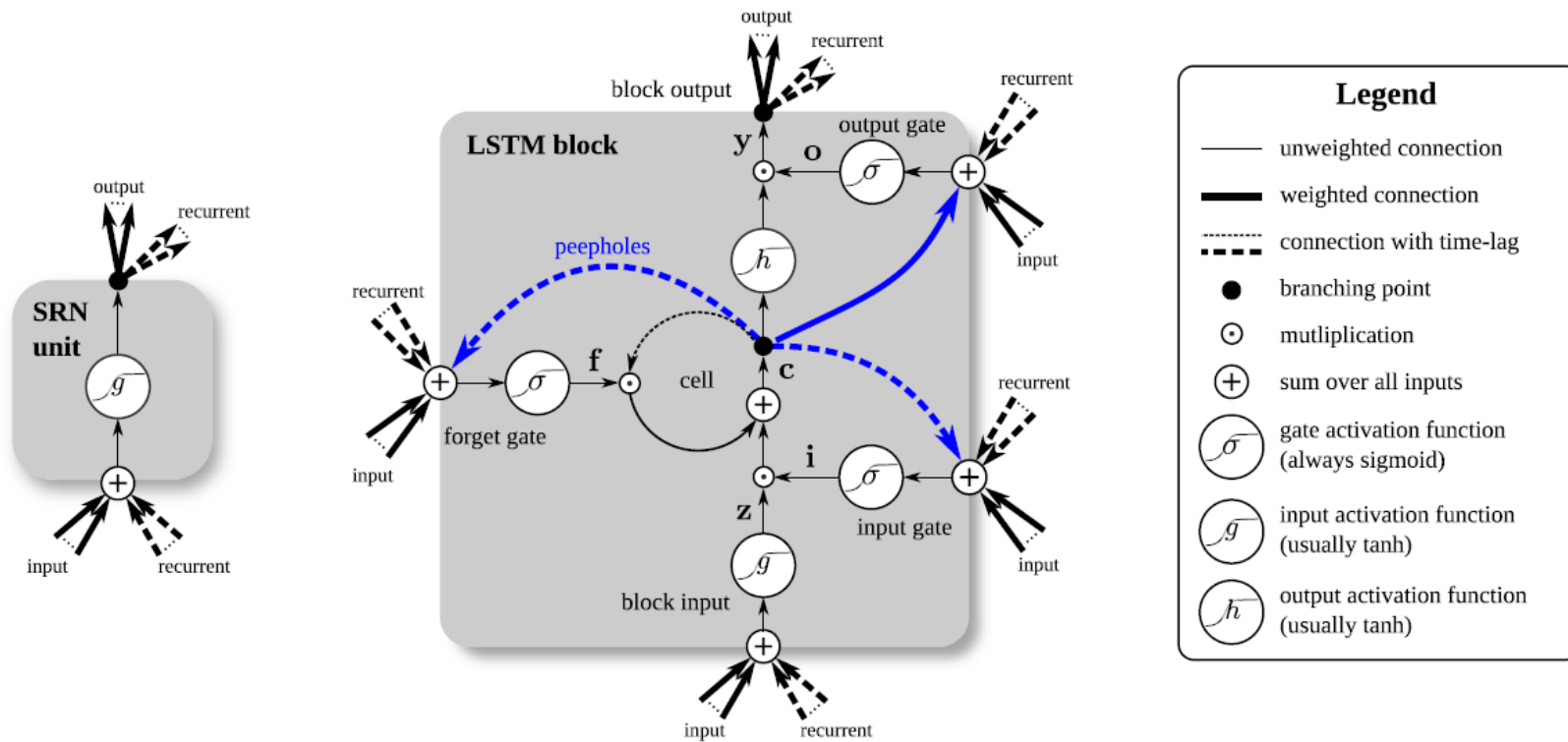


Figura 9 – O bloco LSTM e um neurônio recorrente (SRN)

- 1) *Input Weights:* $W_z, W_s, W_f, W_o \in \mathbb{R}^{N \times M}$.
- 2) *Recurrent Weights:* $R_z, R_s, R_f, R_o \in \mathbb{R}^{N \times N}$.
- 3) *Peephole Weights:* $p_s, p_f, p_o \in \mathbb{R}^N$.
- 4) *Bias Weights:* $b_z, b_s, b_f, b_o \in \mathbb{R}^N$.

Then the vector formulas for a vanilla LSTM layer forward pass can be written as

$$\begin{aligned}
 \bar{z}^t &= W_z x^t + R_z y^{t-1} + b_z \\
 z^t &= g(\bar{z}^t) && \text{block input} \\
 \bar{i}^t &= W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i \\
 i^t &= \sigma(\bar{i}^t) && \text{input gate} \\
 \bar{f}^t &= W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f \\
 f^t &= \sigma(\bar{f}^t) && \text{forget gate} \\
 c^t &= z^t \odot i^t + c^{t-1} \odot f^t && \text{cell} \\
 \bar{o}^t &= W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o \\
 o^t &= \sigma(\bar{o}^t) && \text{output gate} \\
 y^t &= h(c^t) \odot o^t && \text{block output}
 \end{aligned}$$

where σ , g , and h are pointwise nonlinear activation functions. The logistic sigmoid ($\sigma(x) = (1/1 + e^{-x})$) is used as the gate activation function and the hyperbolic tangent ($g(x) = h(x) = \tanh(x)$) is usually used as the block input and output activation function. Pointwise multiplication of two vectors is denoted by \odot .

Figura 10 – Dimensões e equacionamento de um passo adiante para o bloco LSTM

11 O Efeito da Recorrência no Processo de Adaptação

- Redes neurais não-recorrentes podem ser interpretadas como poderosos operadores de transformação de representação, mas não são capazes de reutilizar a informação transformada, produzindo apenas mapeamentos estáticos.
- Esta é a razão para que este tipo de rede neural encontre dificuldade em representar comportamento dinâmico, já que o vetor de saída da rede neural, denominado $\hat{\mathbf{s}}(t)$, depende apenas do vetor de entrada definido no mesmo instante, denominado $\mathbf{x}(t)$. Isto conduz a mapeamentos do tipo:

$$\hat{\mathbf{s}}(t) = RN(\mathbf{x}(t), \theta(t)). \quad (1)$$

onde $\theta(t)$ denota o vetor de parâmetros no instante t .

- Por outro lado, em redes neurais recorrentes o tempo é representado pelo seu efeito real no processamento. Considerando apenas a existência de recorrência externa, ou seja, realimentação apenas da informação de saída da rede neural, resultam modelos do tipo:

$$\hat{\mathbf{s}}(t) = RN_{rec}(\mathbf{x}(t), \hat{\mathbf{s}}(t-1), \theta(t)). \quad (2)$$

- Por substituição recursiva desta equação nela mesma, fica claro que o vetor de saída da rede neural depende da história do vetor de entrada e do vetor de parâmetros.
- Resulta, então, uma rede neural paramétrica com recorrência externa, apresentada na Figura 11, a seguir. Repare que \mathbf{x} e $\hat{\mathbf{s}}$ da equação acima passam a ser denominados de \mathbf{u} e \mathbf{x} , respectivamente.
- O processo de treinamento vai envolver **duas dinâmicas acopladas**: a dinâmica da rede neural e a dinâmica do ajuste de pesos.
- A disponibilidade de redes neurais recorrentes de importância prática está associada à existência de algoritmos de otimização eficientes para o ajuste dos pesos sinápticos. A configuração da Figura 11 é uma generalização do modelo que considera um único neurônio, apresentado na Figura 9.

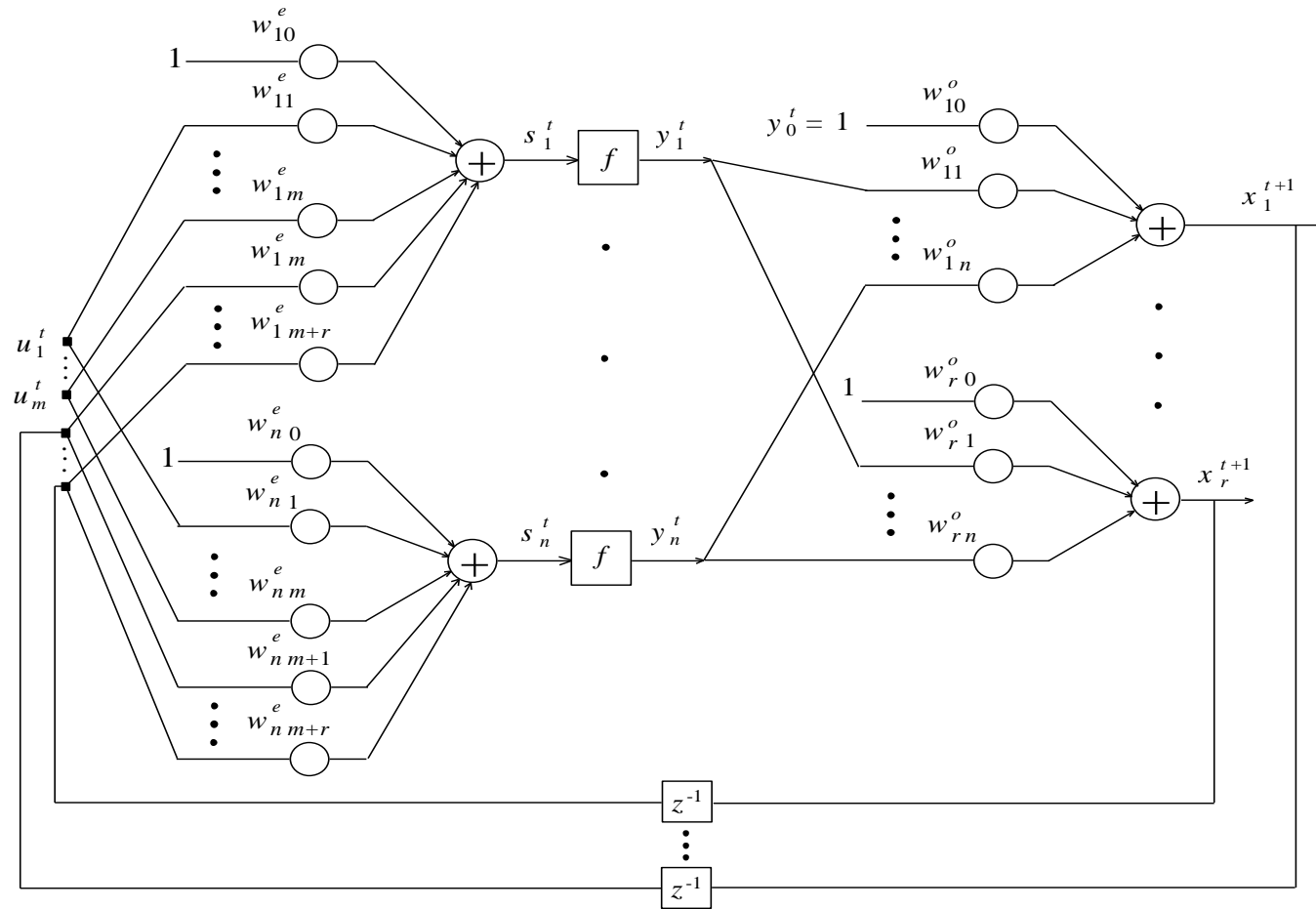


Figura 11 – Rede neural com
recorrência externa

$$x_k^{t+1} = \sum_{j=0}^n w_{kj}^o y_j^t = \sum_{j=1}^n \left[w_{kj}^o f(s_j^t) \right] + w_{k0}^o = \sum_{j=1}^n \left[w_{kj}^o f \left(\sum_{i=1}^m w_{ji}^e u_i^t + w_{j0}^e + \underbrace{\sum_{i=m+1}^{m+r} w_{ji}^e x_{i-m}^t}_{\text{termo adicional}} \right) \right] + w_{k0}^o$$

12 Treinamento Supervisionado para Redes Recorrentes

- Observe que uma rede neural do tipo perceptron com uma camada intermediária é um caso particular da rede recorrente apresentada na Figura 11.
- Caso o vetor de parâmetros seja constante ao longo do tempo, é possível substituir $\theta(t)$ por θ nas equações (1) e (2). Com isso, fica claro que, ao contrário do modelo de rede neural não-recorrente, o modelo de rede neural recorrente é uma função composta de θ . Logo, a análise variacional dos modelos com e sem recorrência produz os seguintes resultados:

- Rede neural não-recorrente: $\hat{\mathbf{s}}(t) = RN(\mathbf{x}(t), \theta)$

$$\frac{\partial \hat{\mathbf{s}}(t)}{\partial \theta} = \frac{\partial RN}{\partial \theta}$$

- Rede neural recorrente: $\hat{\mathbf{s}}(t) = RN_{rec}(\mathbf{x}(t), \hat{\mathbf{s}}(t-1), \theta)$

$$\frac{\partial \hat{\mathbf{s}}(t)}{\partial \theta} = \frac{\partial RN_{rec}}{\partial \theta} = \frac{\partial RN_{rec}}{\partial \theta} + \underbrace{\frac{\partial RN_{rec}}{\partial \hat{\mathbf{s}}(t-1)} \frac{\partial \hat{\mathbf{s}}(t-1)}{\partial \theta}}_{\text{termo adicional}}$$

- O problema de modelagem de sistemas dinâmicos de tempo discreto utilizando redes neurais recorrentes pode ser colocado na forma:
 - ◇ Seja $\{(\mathbf{x}(t), \mathbf{s}(t))\}_{t=1}^N \in X \times \mathbb{R}^r$, onde $X \subset \mathbb{R}^m$, um conjunto de dados amostrados de um sistema dinâmico de tempo discreto a ser identificado;
 - ◇ Seja $RN_{rec}(\mathbf{x}(t), \hat{\mathbf{s}}(t-1), \theta): X \times \mathbb{R}^r \times \mathbb{R}^P \rightarrow \mathbb{R}^r$ o mapeamento realizado pela rede neural recorrente, onde $\theta \in \mathbb{R}^P$ é um vetor contendo todos os P parâmetros da rede neural, ordenados de forma arbitrária, mas fixa;
 - ◇ Resolva o seguinte problema de otimização:

$$\min_{\theta} J(\theta) = \min_{\theta} \frac{1}{2} \sum_{t=1}^N \|RN_{rec}(\mathbf{x}(t), \hat{\mathbf{s}}(t-1), \theta) - \mathbf{s}(t)\|^2.$$

- A solução deste problema de otimização não-linear só pode ser obtida via processos numéricos iterativos, já que soluções analíticas não estão disponíveis, principalmente devido à natureza não-linear do modelo de aproximação $RN_{rec}(\mathbf{x}(t), \hat{\mathbf{s}}(t-1), \theta)$.

- A ideia básica deste processo numérico é realizar ajustes iterativos no vetor de parâmetros $\theta \in \mathbb{R}^P$ sempre em direções em que a função-objetivo $J(\theta)$ decresça.
- Observe que $J(\theta)$ define uma superfície no espaço \mathbb{R}^{P+1} , ou seja, a dimensão do espaço de busca é igual ao número de parâmetros a serem ajustados, havendo uma dimensão a mais que indica o valor da função-objetivo para cada θ .
- Como o vetor de parâmetros $\theta \in \mathbb{R}^P$ não vai ser constante durante o treinamento, aproximações para o cálculo do vetor gradiente devem ser adotadas.

13 *Backpropagation Through Time* (BPTT)

- Para redes neurais recorrentes com configurações similares às aquelas apresentadas nas Figuras 3 e 4, é possível implementar uma técnica baseada em gradiente para ajuste dos pesos sinápticos.
- A ideia é “desdobrar” a rede neural recorrente ao longo do tempo, produzindo arquiteturas como na Figura 12 a seguir.

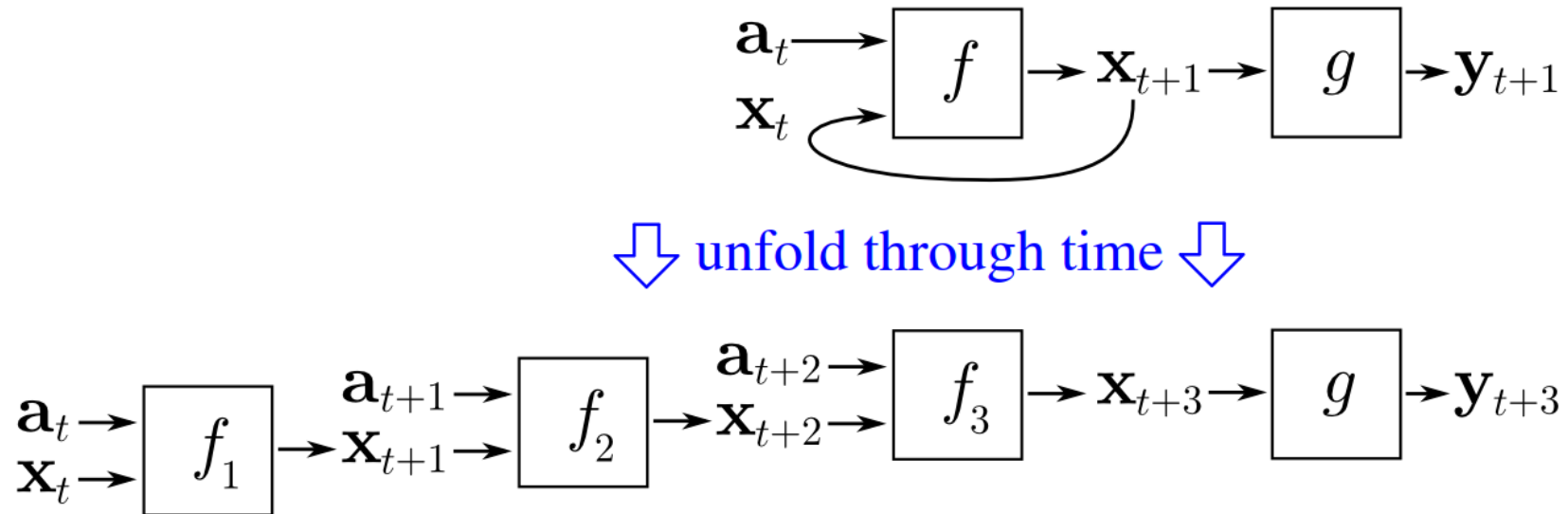


Figura 12 – Representação do desdobramento no tempo, para posterior aplicação
do *backpropagation through time* (BPTT) (Fonte:

http://en.wikipedia.org/wiki/Backpropagation_through_time).

- Geralmente, uma etapa de BPTT é aplicada a cada apresentação de uma amostra de treinamento, ou seja, o treinamento é *on-line*.

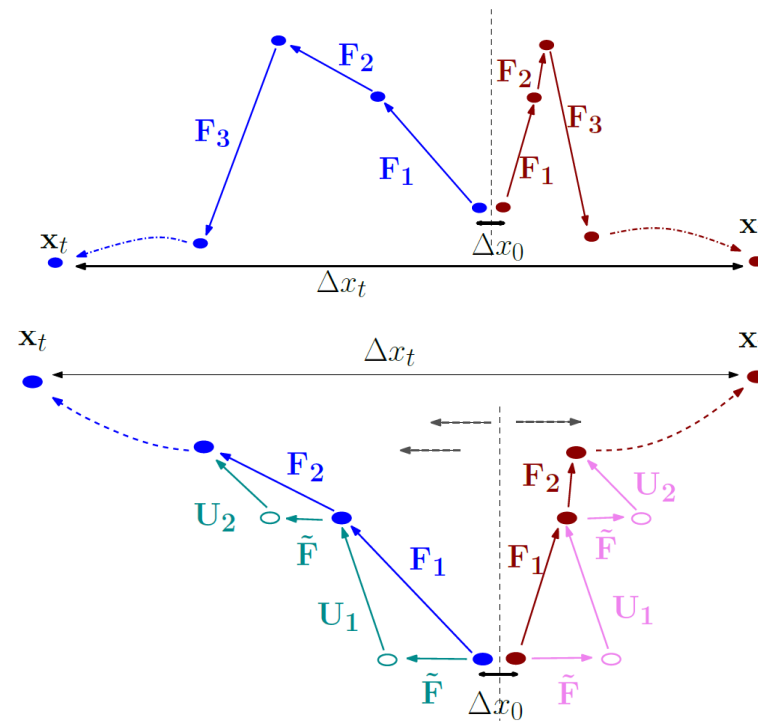
- Os pesos associados ao mapeamento f são atualizados como a média dos pesos após a etapa de BPTT, tomando as k instâncias de f , sendo que no caso da Figura 12 tem-se $k = 3$.
- BPTT tende a ser mais eficiente para ajuste de pesos de redes neurais recorrentes do que buscas heurísticas, como técnicas evolutivas (MCDONNELL & WAGEN, 1994; SJÖBERG *et al.*, 1995). No entanto, os desafios associados à existência de mínimos locais são bem mais acentuados do que no caso de redes neurais não-recorrentes.
- Para mais detalhes referentes ao formalismo envolvido, consultar NERRAND *et al.* (1994), PEARLMUTTER (1995), PINEDA (1987), PINEDA (1989), WERBOS (1990) e WILLIAMS & ZIPSER (1989).
- BPTT é também empregado no caso de LSTM, assim como algoritmos evolutivos (SCHMIDHUBER *et al.*, 2007), conforme já antecipado na seção 10.

14 O problema dos gradientes explosivos e que se anulam

- A aplicação de métodos baseados em gradiente para o treinamento de redes neurais recorrentes pode se defrontar com um de dois fenômenos que inviabilizam o ajuste de pesos: gradientes explosivos e gradientes que se anulam (do inglês *exploding and vanishing gradients*).
- Os gradientes explosivos podem promover ajustes exagerados nos vetores de pesos, saturando neurônios e destruindo progressos anteriores obtidos pelo ajuste iterativo vinculado ao treinamento supervisionado.
- Os gradientes que se anulam impedem que pesos sinápticos continuem a sofrer ajuste durante o treinamento supervisionado.
- Um estudo mais aprofundado desses dois fenômenos e de como lidar com eles foi apresentado em PASCANU *et al.* (2013).
- Entende-se que os gradientes explosivos ocorrem quando a operação do sistema dinâmico, representado pela rede neural recorrente sob treinamento, cruza a

fronteira entre duas bacias de atração. Logo, a solução proposta é evitar cruzar esta fronteira ou definir heurísticas capazes de lidar com o processo de ajuste durante este processo de cruzamento de fronteiras entre bacias de atração.

- As figuras a seguir, extraídas de PASCANU *et al.* (2013), mostram cenários de gradiente explosivo para sistemas dinâmicos com e sem entrada externa.



- Por outro lado, entende-se que os gradientes que se anulam são produzidos quando o sistema dinâmico encontra-se próximo de um ponto fixo, ou seja, próximo da região de convergência de uma bacia de atração. Logo, a solução proposta é permanecer próximo a fronteiras entre bacias de atração, sem cruzá-las.

15 Predição de Séries Temporais: Um Passo ou Múltiplos Passos

- Não-recorrente: Chave em A para treinamento e chave em B para operação (não está traçada a linha de derivação de atraso);
- Recorrente: Chave em B tanto para treinamento como para operação.

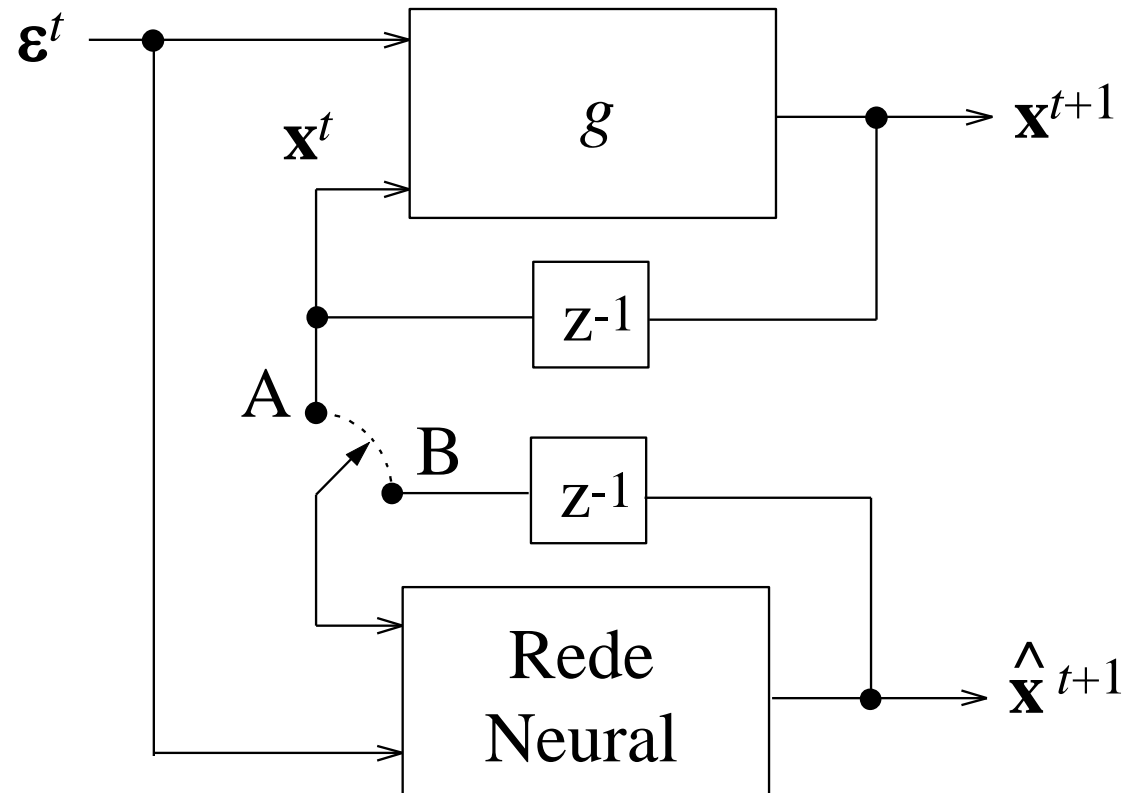


Figura 13 – Predição de séries temporais utilizando uma rede neural

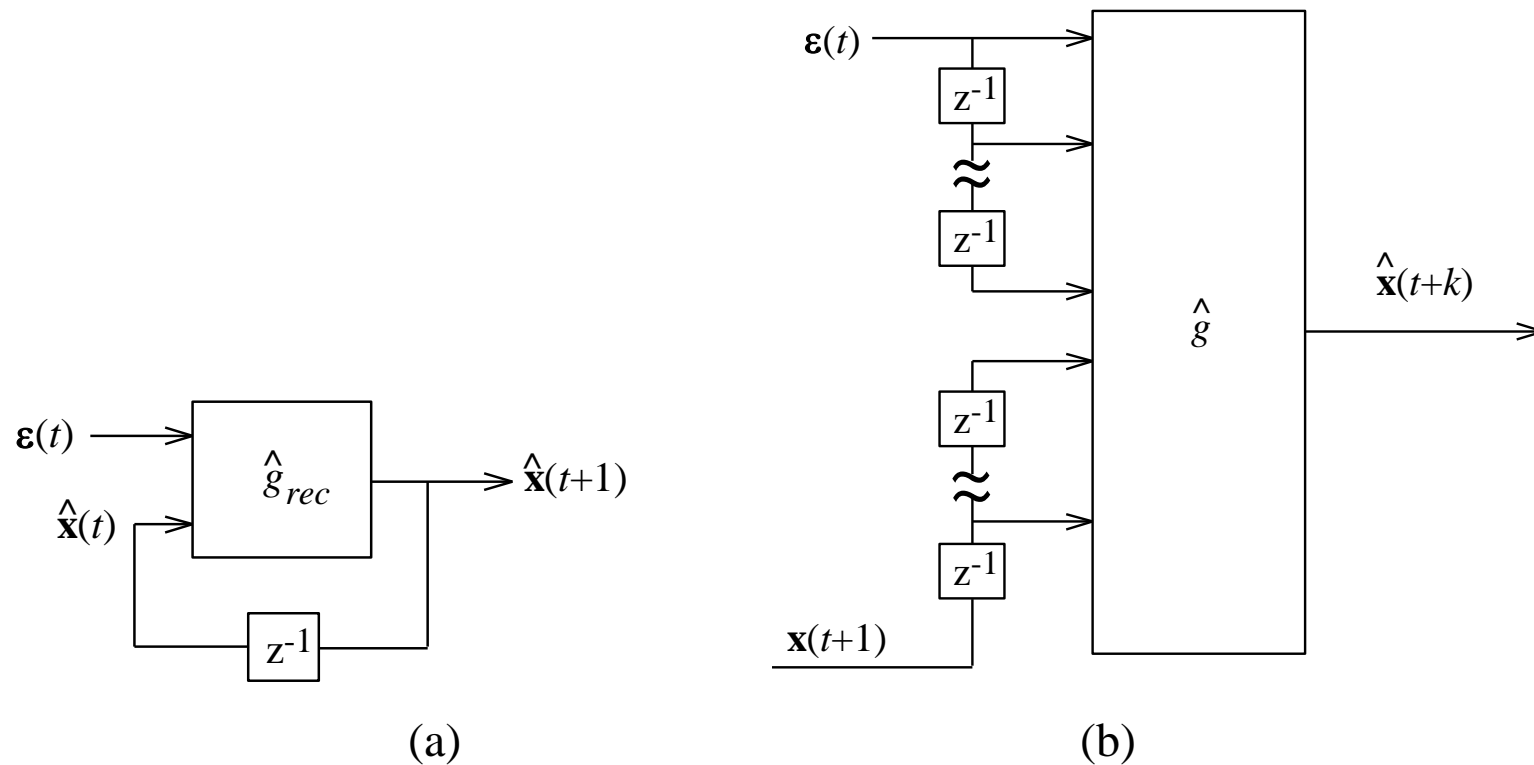


Figura 14 – Modelos para previsão de séries temporais

- (a)** Predição de múltiplos passos à frente **(b)** Predição de k passos à frente (k fixo, mas arbitrário)
Uso de linha de derivação de atraso (*tapped-delay line*)

- Para uma discussão mais formal, consultar BACK & TSOI (1991) e CONNOR *et al.* (1994).

16 Referências Bibliográficas

- ATTUX, R.R.F. “Sobre dinâmica caótica e convergência em algoritmos de equalização autodidata”, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 2001.
- BACK, A. D. & TSOI, A. C. “FIR and IIR synapses, a new neural network architecture for time series modeling”, *Neural Computation*, vol. 3, pp. 375-385, 1991.
- BINDER, P.M. & JENSEN, R.V. “Simulating Chaotic Behavior with Finite State Machines”, *Physical Review A*, vol. 34, pp. 4460-4463, 1986.
- CONNOR, J.T.; MARTIN, R.D. & ATLAS, L.E. “Recurrent neural networks and robust time series prediction”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240, 1994.
- DOETSCH, P.; KOZIELSKI, M. & NEY, H. “Fast and robust training of recurrent neural networks for offline handwriting recognition”, *Proceedings of the 14th International Conference on Frontiers of Handwriting Recognition*, pp. 279-284, 2014.
- DONAHUE, J.; HENDRICKS, L.A.; ROHRBACH, M.; VENUGOPALAN, S.; GUADARRAMA, S.; SAENKO, K. & DARRELL, T. “Long-term recurrent convolutional networks for visual recognition and description”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677-691, 2017.
- DOS SANTOS, E.P. & VON ZUBEN, F.J. “Efficient Second-Order Learning Algorithms for Discrete-Time Recurrent Neural Networks”, in L.R. Medsker and L.C. Jain (eds.) *Recurrent Neural Networks: Design and Applications*, CRC Press, pp. 47-75, 2000.
- ECKMANN, J.-P.; KAMPHORST, S.O.; RUELLE, D. & CILIBERTO, S. “Liapunov Exponents From Time Series”, *Physical Review A*, vol. 34, pp. 4971-4979, 1986.
- ELMAN, J.L. “Finding structure in time”, *Cognitive Science*, vol. 14, pp. 179-211, 1990.

- FAN, Y.; QIAN, Y.; XIE, F. & SOONG, F.K. “TTS synthesis with bidirectional LSTM based recurrent neural networks”, Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH’2014), pp. 1964-1968, 2014.
- FRANÇOIS, P. & HAKIM, V. “Design of genetic networks with specified functions by evolution in silico”, Proceedings of the National Academy of Science, vol. 101, no. 2, pp. 580-585, 2004.
- FREEMAN, W.J. “Mass action in the nervous system”, Academic Press, 1975.
- GOLDBETER, A. “Biochemical Oscillations and Cellular Rhythms”, Cambridge University Press, 1996.
- GRAVES, A. “Generating sequences with recurrent neural networks”, arXiv:1308.0850, 2014.
- GRAVES, A.; LIWICKI, M.; FERNÁNDEZ, S.; BERTOLAMI, R.; BUNKE, H. & SCHMIDHUBER, J. “A novel connectionist system for unconstrained handwriting recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, 2009.
- GRAVES, A. & SCHMIDHUBER, J. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”, Neural Networks, vol. 18, nos. 5-6, pp. 602-610, 2005.
- GREFF, K.; SRIVASTAVA, R.K.; KOUTNÍK, J.; STEUNEBRINK, B.R. & SCHMIDHUBER, J. “LSTM: A Search Space Odyssey”, IEEE Transactions on Neural Networks and learning Systems, to appear, 2017.
- HOCHREITER, S. & SCHMIDHUBER, J. “Long short-term memory”, Fakultät für Informatik, Technische Universität München, Technical Report FKI-207-95, Munich, Germany, 1995.
- HOCHREITER, S. & SCHMIDHUBER, J. “Long short-term memory”, Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
- JAEGER, H. “The Echo State Approach to Analyzing and Training Neural Networks”, Technical Report no. 148, Bremen: German National Research Center for Information Technology, 2001.
- KATZ, G.; BARRETT, C.; DILL, D.; JULIAN, K. & KOCHENDERFER, M. “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks”, arXiv:1702.01135v1, Feb 3rd 2017.
- KELSO, J.A.S. “Dynamic Patterns: The Self-organization of Brain and Behavior”, The MIT Press, 1995.

- KOLEN, J.F. “Exploring the Computational Capabilities of Recurrent Neural Networks”, Ph.D. Thesis, The Ohio State University, 1994.
- KOZMA, R.; AGHAZARIAN, H.; HUNTSHERGER, T.; TUNSTEL, E. & FREEMAN, W.J. “Computational Aspects of Cognition and Consciousness in Intelligent Devices”, IEEE Computational Intelligence Magazine, vol. 2, no. 3, pp. 53-64, 2007.
- KOZMA, R. “Intentional systems: Review of neurodynamics, modeling, and robotics implementation”, Physics of Life Reviews, vol. 5, no. 1, pp 1-21, 2008.
- LUONG, M.-T.; SUTSKEVER, I.; LE, Q.V.; VINYALS, O. & ZAREMBA, W. “Addressing the rare word problem in neural machine translation”, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 11–19, 2015.
- MAASS, W. “Networks of Spiking Neurons: The Third Generation of Neural Network Models”, *Neural Networks*, vol. 10, no. 9, pp. 1659-1671, 1997.
- MARCHI, E.; FERRONI, G.; EYBEN, F.; GABRIELLI, L.; SQUARTINI, S. & SCHULLER, B. “Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks”, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2014), pp. 2164-2168, 2014.
- MCCAFFREY, D.F.; ELLNER, S.; GALLANT, A.R. & NYCHKA, D.W. “Estimating the Lyapunov Exponent of a Chaotic System With Nonparametric Regression”, Journal of the American Statistical Association, vol. 87, no. 419, pp. 682-695, 1992.
- MCCULLOCH, W.S. & PITTS, W. “A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.
- MCDONNELL, J.R. & WAGEN, D. “Evolving Recurrent Perceptrons for Time-Series Modeling”, IEEE Transactions Neural Networks, vol. 5, no. 1, pp. 24-38, 1994.

- MISRA, J. & SAHA, I. “Artificial neural networks in hardware: A survey of two decades of progress”, *Neurocomputing*, vol. 74, no. 1-3, pp. 239-255, 2010.
- NERRAND, O.; ROUSSEL-RAGOT, P.; PERSONNAZ, L. & DREYFUS, G. “Neural Networks and Nonlinear Adaptive Filtering: Unifying Concepts and New Algorithms”, *Neural Computation*, vol. 5, no. 2, pp. 165-199, 1993.
- NERRAND, O.; ROUSSEL-GAGOT, P.; URBANI, D.; PERSONNAZ, L. & DREYFUS, G. “Training recurrent neural networks: Why and how? An illustration in dynamical process modeling”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 178-184, 1994.
- OTT, E. “Chaos in Dynamical Systems”, Cambridge University Press, 1993.
- OZTURK, M.; XU, D. & PRÍNCIPE, J. “Analysis and Design of Echo State Networks”, *Neural Computation*, vol. 19, no. 1, pp. 111-138, 2007.
- PASCANU, R.; MIKOLOV, T. & BENGIO, Y. “On the difficulty of training recurrent neural networks”, *Proceedings of the 30th International Conference on Machine Learning (ICML’2013)*, vol. 3, pp. 1310-1318, 2013.
- PEARLMUTTER, B.A. “Gradient calculations for dynamic recurrent neural networks: a survey”, *IEEE Transactions on Neural Networks*, vol. 6, no. 5, pp. 1212-1228, 1995.
- PERETTO, P. “An Introduction to the Modeling of Neural Networks”, Cambridge University Press, 1992.
- PINEDA, F.J. “Generalization of back-propagation to recurrent neural networks”, *Physical Review Letters*, vol. 59, no. 19, pp. 2229-2232, 1987.
- PINEDA, F.J. “Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation”, *Neural Computation*, vol. 1, no. 2, pp. 161-172, 1989.
- SAK, H.; SENIOR, A. & BEAUFAYS, F. “Long short-term memory recurrent neural network architectures for large scale acoustic modeling”, *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH’2014)*, pp. 338-342, 2014.

- SCHMIDHUBER, J.; WIERSTRA, D.; GAGLIOLO, M. & GOMEZ, F. “Training recurrent neural networks by Evolino”, *Neural Computation*, vol. 19, no. 3, pp. 757-779, 2007.
- SIEGELMANN, H.T. “Neural Networks and Analog Computation – Beyond the Turing Limit”, Birkhäuser, 1999.
- SJÖBERG, J.; ZHANG, Q.; LJUNG, L.; BENVENISTE, A.; DELYON, B.; GLORENNEC, P.; HJALMARSSON, H. & JUDITSKY, A. “Nonlinear Black-box Modelling in System Identification: a Unified Overview”, *Automatica*, vol. 31, no. 12, pp. 1691-1724, 1995.
- SONDERBY, S.K. & WINTHER, O. “Protein Secondary Structure prediction with long short term memory networks”, arXiv:1412.7828v2, 2015.
- TSOI, A.C. & BACK, A.D. “Discrete time recurrent neural network architectures: a unifying review”, *Neurocomputing*, vol. 15, nos. 3-4, pp. 183-223, 1997.
- TSOI, A.C. & BACK, A.D. “Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 229-239, 1994.
- WERBOS, P.J. “Backpropagation Through Time: What It Does and How to Do It”, *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.
- WILLIAMS, R.J. & ZIPSER, D.A. “Learning Algorithm for Continually Running Fully Recurrent Neural Networks”, *Neural Computation*, vol. 1, no. 2, pp. 270-280, 1989.
- WOLF, A.; SWIFT, J.B.; SWINNEY, H.L. & VASTANO, J.A. “Determining Lyapunov Exponents From a Time Series”, *Physica D*, vol. 16, pp. 285-315, 1985.

17 Referências Adicionais e links relevantes

- AMIT, D.J. “Modeling Brain Function: The World of Attractor Neural Networks”, Cambridge University Press, 1989.
- CHANG, W.-F. & MAK, M.-W. “A conjugate gradient learning algorithm for recurrent neural networks”, *Neurocomputing*, vol. 24, nos. 1-3, pp. 173-189, 1999.
- CLEEREMANS, A.; SERVAN-SCHREIBER, D. & MCCLELLAND, J. “Finite state automata and simple recurrent networks”, *Neural Computation*, vol. 1, no. 3, pp. 372-381, 1989.
- DOS SANTOS, E.P. & VON ZUBEN, F.J. “Improved Second-Order Training Algorithms for Globally and Partially Recurrent Neural Networks”, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'99)*, vol. 3, pp. 1501-1506, July 1999.
- GILES, C.L.; MILLER, C.B.; CHEN, D.; CHEN, H.H.; SUN, G.Z. & LEE, Y.C. “Learning and extracting finite state automata with second-order recurrent neural networks”, *Neural Computation*, vol. 4, no. 3, pp. 393-405, 1992.
- HAYKIN, S. “Neural Networks and Learning Machines”, 3rd edition, Prentice Hall, 2008.
- HUNT, K.J.; SBARBARO, D.; ZBIKOWSKI, R. & GAWTHROP, P.J. “Neural Networks for Control Systems – A Survey”, *Automatica*, vol. 28, no. 6, pp. 1083-1112, 1992.
- KHALIL, H.K. “Nonlinear Systems”, 2nd. edition, Prentice Hall, 1996.
- KILIAN, J. & SIEGELMANN, H.T. “The Dynamic Universality of Sigmoidal Neural Networks”, *Information and Computation*, vol. 128, pp. 48-56, 1996.
- LEVIN, A.V. & NARENDRA, K.S. “Control of nonlinear dynamical systems using neural networks – controllability and stabilization”, *IEEE Transactions on Neural Networks*, vol. 4, no. 2, pp. 192-206, 1993.

- LEVIN, A.V. & NARENDRA, K.S. “Control of nonlinear dynamical systems using neural networks – Part II: observability, identification, and control”, IEEE Transactions on Neural Networks, vol. 7, no. 1, pp. 30-42, 1996.
- MEDSKER, L.R. & JAIN, L.C. (eds.) “Recurrent Neural Networks: Design and Applications”, CRC Press, 2000.
- MEISS, J.D. “Frequently Asked Questions about Nonlinear Science”, Department of Applied Mathematics, University of Colorado at Boulder, <http://amath.colorado.edu/faculty/jdm/faq-Contents.html>.
- MILLER, W.T.; SUTTON, R.S. & WERBOS, P.J. “Neural Networks for Control”, MIT Press. 1990.
- NARENDRA, K.S. & PARTHASARATHY, K. “Identification and control of dynamical systems using neural networks”, IEEE Transactions on Neural Networks, vol. 1, no. 1, pp. 4-27, 1990.
- OLAH, C. “Neural Networks (General)”, <http://colah.github.io/>.
- PHAM, D.T. & LIU, X. “Dynamic system modelling using partially recurrent neural networks”, Journal of Systems Engineering, pp. 134-141, 1992.
- SIEGELMANN, H.T., HORNE, B.G. & GILES, C.L. “Computational capabilities of recurrent NARX neural networks”, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 27, pp. 208-215, 1997.
- SIEGELMANN, H.T. & SONTAG, E.D. “Analog computation via neural networks”, Theoretical Computer Science, vol. 131, pp. 331-360, 1994.
- SIEGELMANN, H.T. & SONTAG, E.D. “On the Computational Power of Neural Nets”, Journal of Computer and System Sciences, vol. 50, pp. 132-150, 1995.
- SIEGELMANN, H.T. & SONTAG, E.D. “Turing Computability with Neural Nets”, Applied Mathematics Letters, vol. 4, no. 6, pp. 77-80, 1991.
- SLOTINE, J.-J. & LI, W. “Applied Nonlinear Control”, Prentice Hall, 1991.

- SONTAG, E. “Some Topics in Neural Networks and Control”, Technical Report LS93-02, Department of Mathematics, Rutgers University, 1993.
- VERGIS, A.; STEIGLITZ, K. & DICKINSON, B. “The complexity of analog computation”, Mathematics and Computers in Simulation, vol. 28, pp. 91-113, 1986.
- VIDYASAGAR, M. “Nonlinear Systems Analysis”, 2nd. edition, Prentice Hall, 1993.
- VON ZUBEN, F.J. “Modelos paramétricos e não-paramétricos de redes neurais artificiais e aplicações”, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, 1996.
- VON ZUBEN, F.J. & NETTO, M.L.A. “Exploring the Nonlinear Dynamic Behavior of Artificial Neural Networks”, Proceedings of the IEEE International Conference on Neural Networks, vol. II, pp. 1000-1005, 1994.
- VON ZUBEN, F.J. & NETTO, M.L.A. “Second-order training for recurrent neural networks without teacher-forcing”, Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), vol. 2, pp. 801-806, 1995.
- ZBIKOWSKI, R.W. “Recurrent Neural Networks: Some Control Aspects”, Ph.D. Thesis, Faculty of Engineering, Glasgow University, 1994.

Links para softwares:

- <https://www.tensorflow.org/tutorials/recurrent>
- http://deeplearning.net/software_links/
- <https://neuroscience.stanford.edu/mbct/home>
- <https://www.mathworks.com/help/nnet/ref/layrecnet.html>