

-
-
-
-
-

Redes Recorrentes Supervisionadas

Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática



-
-
-
-
-

Conteúdo

- Histórico
- Teoria
- Redes Recorrentes Supervisionadas
- Treinamento

Histórico

- A rede recorrente por estudos de autômatos (Kleene, 1954).
- Teoria do filtro de Kalman (Rudolf E. Kalman, 1960).
- Controlabilidade e observabilidade (Zadeh & Desoer, 1963) (Kailath, 1980), (Sontag, 1990), (Lewis & Syrmos, 1995).
- O modelo NARX (Leontaritis & Billings 1985).
- O modelo NARX para redes neurais (Chen et al, 1990).
- Arquiteturas de redes recorrentes (Jordan, 1986; Elman, 1990).
- Olin and Giles (1996) mostraram que empregando redes recorrentes de segunda-ordem, a classificação correta de seqüências temporais de tamanho finito é garantida.

Histórico

- O modelo retro-propagação ao longo do tempo (back-propagation through time - BPTT) introduzido em Rumelhart (1986).
- O modelo Back-propagation recorrente foi proposto independentemente por Pineda (1987, 1988, 1989), Almeida (1987, 1988) e Rohwer e Forrest (1987).
- O algoritmo de aprendizagem recorrente em tempo real foi introduzido por Williams e Zipser (1989), de modo similar a identificação de sistemas para sintonia de parâmetros de um sistema dinâmico qualquer (McBride e Narendra, 1965).

Teoria

- Redes recorrentes caracterizam-se por possuírem um ou mais laços de realimentação.
 - Cada laço pode ser de tipo local (de um nodo para ele mesmo) ou global.
 - Uma rede neural totalmente recorrente tem arquitetura na qual uma dada unidade de processamento pode realimentar qualquer outra unidade.
 - Uma rede neural parcialmente recorrente é definida como uma rede formada por maioria de conexões de alimentação direta (predominantes) e conexões de realimentação.

Teoria

- Uma rede recorrente tem resposta que depende do tempo para um sinal externo aplicado, por isto são chamadas de redes recorrentes dinamicamente direcionadas.
- O emprego de realimentação permite que a rede recorrente utilize representações de estados.
 - Tornam-se úteis para aplicações como modelagens e previsões não-lineares, equalização adaptativa de processamento de fala, sistemas de controle, diagnóstico de motores de automóveis.

Teoria

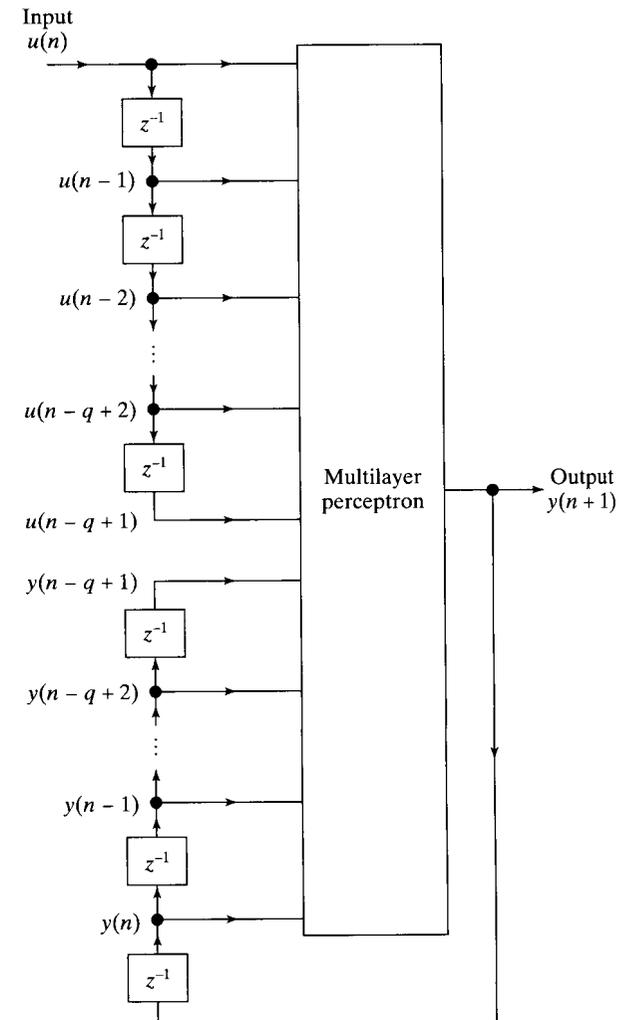
- Modelos com realimentação global caracterizam-se por incorporarem uma MLP ou parte dela e explorarem sua capacidade de construção de mapas não-lineares. Eles são:
 - Modelo recorrente entrada-saída;
 - Modelo por espaço de estados;
 - MLP recorrente;
 - Rede de segunda ordem.
- RNs com realimentação local são globalmente de alimentação direta e localmente recorrentes (Tsoi e Back, 1994):
 - Realimentação local de ativação;
 - Realimentação local de saída;
 - Realimentação local de sinapse.

NARX

- Modelo Recorrente Entrada-saída → Modelo Não-linear Auto-regressivo com Entradas Exógenas (NARX).

$$y(n+1) = F(y(n), \dots, y(n-q+1), u(n), \dots, u(n-q+1))$$

- O modelo tem uma entrada simples que é aplicada a uma linha de atrasos formando uma memória de q elementos. Ele tem uma saída simples que é realimentada na entrada através de uma outra linha de atraso de q elementos.
- O conteúdo das memórias das linhas de atraso é inserido na camada de entrada da MLP.
- Um vetor de entrada apresentado à rede forma uma janela de dados de valores presentes e passados da entrada (dados exógenos) e valores atrasados da saída (dados regredidos).



NARX

- Seja uma RN com 1 entrada e 1 saída.

$$y(n+q) = \Phi(\mathbf{x}(n), \mathbf{u}_q(n))$$

onde q é a dimensão do espaço de estado, e $\Phi: R^{2q} \rightarrow R$.

- Supondo que a RN é observável, então $\mathbf{x}(n) = \Psi(\mathbf{y}_q(n), \mathbf{u}_{q-1}(n))$ onde $\Psi: R^{2q} \rightarrow R$.
- $y(n+q) = F(\mathbf{y}_q(n), \mathbf{u}_q(n))$

onde $\mathbf{u}_{q-1}(n)$ está contido em $\mathbf{u}_q(n)$ como seus primeiros $(q-1)$ elementos, e o mapa não-linear $F: R^{2q} \rightarrow R$ é válido para ambos Ψ e Φ .

- Em particular, $n=n-q+1$:

$$y(n+1) = F(y(n), \dots, y(n-q+1), u(n), \dots, u(n-q+1))$$

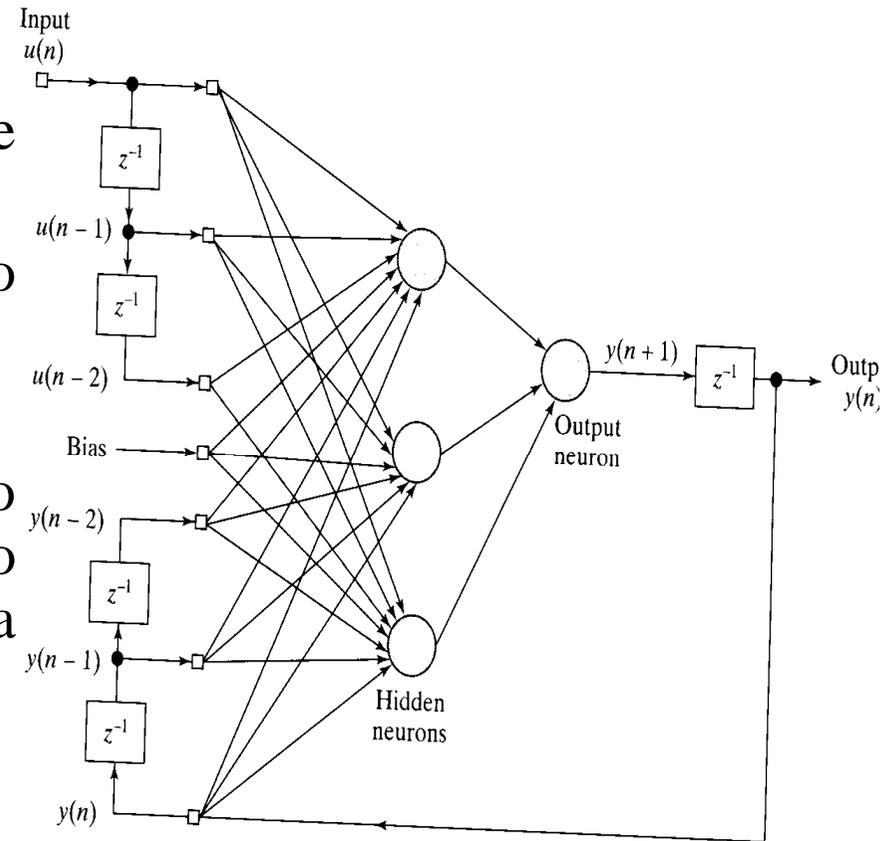


FIGURE 15.7 NARX network with $q = 3$ hidden neurons.

Modelo de Espaço de Estados

- O estado de um sistema dinâmico é definido como um conjunto de variáveis que descrevem a situação atual deste sistema. O estado resume a história do sistema e determina, em conjunto com a entrada o comportamento futuro do sistema.
- Seja um vetor q -dimensional $\mathbf{x}(n)$ que denota o estado de um sistema discreto no tempo e não linear. Seja um vetor m -dimensional $\mathbf{u}(n)$ que denota a entrada do sistema e um vetor p -dimensional $\mathbf{y}(n)$ que denota a saída do sistema.

Modelo de Espaço de Estados

- O comportamento dinâmico deste sistema (livre de ruído) é definido como:

$$\mathbf{x}(n+1) = \boldsymbol{\varphi}(\mathbf{W}_a \mathbf{x}(n) + \mathbf{W}_b \mathbf{u}(n)) \text{ (equação de estado)}$$

$$\mathbf{y}(n) = \mathbf{C} \mathbf{x}(n) \text{ (equação de saída)}$$

onde \mathbf{W}_a é uma matriz $q \times q$; \mathbf{W}_b é uma matriz $q \times (m+1)$; \mathbf{C} é uma matriz $p \times q$; e $\boldsymbol{\varphi}: R^q \rightarrow R^q$ é um mapa diagonal descrito por $\boldsymbol{\varphi}: [x_1, x_2, \dots, x_q]^T \rightarrow [\varphi(x_1), \varphi(x_2), \dots, \varphi(x_q)]^T$ para algum componente não-linear sem memória $\varphi: R \rightarrow R$.

- Os espaços de dimensão R^m , R^q e R^p são chamados de espaços de entrada, estados e saída, levando a um modelo recorrente de ordem q com m entradas e p saídas.

Modelo de Espaço de Estados

- Os nodos escondidos definem o estado da rede neural. A saída da camada escondida é realimentada para a camada de entrada através de um banco de unidades de atrasos.

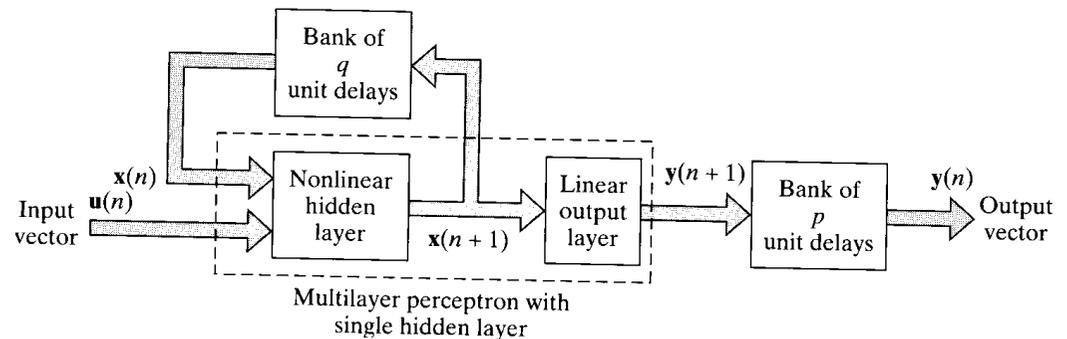


FIGURE 15.2 State-space model.

- Logo, a camada de entrada passa a ser a concatenação dos nodos de realimentação e dos nodos fontes. A rede é conectada ao ambiente externo. A ordem do modelo é determinada pelo número de unidades de atrasos empregadas na realimentação.

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n))$$

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n)$$

Modelo de Espaço de Estados

- A rede recorrente simples (SRN) se difere do modelo anterior pois tem uma camada de saída não-linear e não possui o banco de unidades de atrasos na saída.

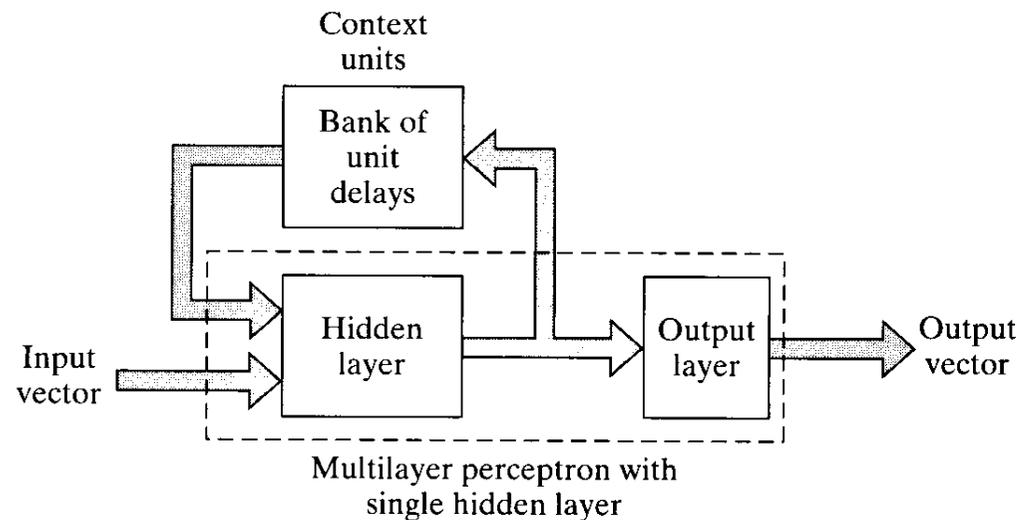


FIGURE 15.3 Simple Recurrent network (SRN).

MLP Recorrente (RMLP)

- Esta rede tem ao menos uma camada escondida. Cada camada com processamento (escondidas e saída) de uma RMLP tem uma realimentação para ela própria:

$$\mathbf{x}_I(n+1) = \boldsymbol{\varphi}_I(\mathbf{x}_I(n), \mathbf{u}(n))$$

$$\mathbf{x}_{II}(n+1) = \boldsymbol{\varphi}_{II}(\mathbf{x}_{II}(n), \mathbf{x}_I(n+1)), \dots,$$

$$\mathbf{x}_O(n+1) = \boldsymbol{\varphi}_O(\mathbf{x}_O(n), \mathbf{x}_K(n))$$

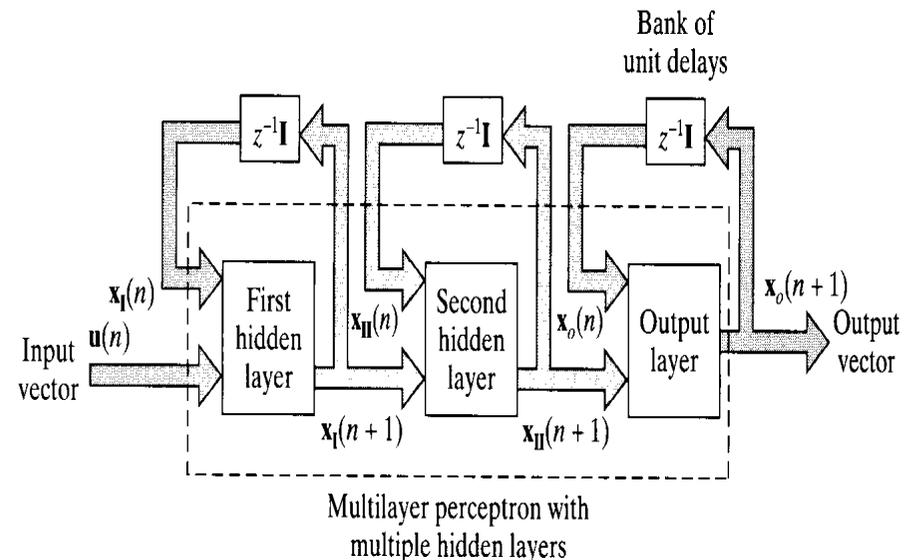


FIGURE 15.4 Recurrent multilayer perceptron.

Rede de Segunda Ordem

- Quando o campo local induzido v_k é combinado usando multiplicações, este nodo é dito ser de segunda ordem.
- Uma rede recorrente de segunda ordem é definida como

$$v_k(n) = b_k + \sum_i \sum_j w_{kij} x_i(n) u_j(n)$$

$$x_k(n+1) = \phi(v_k(n)) = 1 / (1 + \exp(-v_k(n)))$$
- O produto $x_i(n)u_j(n)$ representa o par {estado, entrada} e uma peso w_{kij} positivo representa a transição de estado {estado, entrada} \rightarrow {próximo estado}, enquanto que um peso negativo representa a ausência de uma transição. A transição de estado é descrita por $\delta(x_i, u_j) = x_k$.
- Redes de segunda ordem são usadas para representar e aprender autômatos determinísticos de estados finitos (DFA).

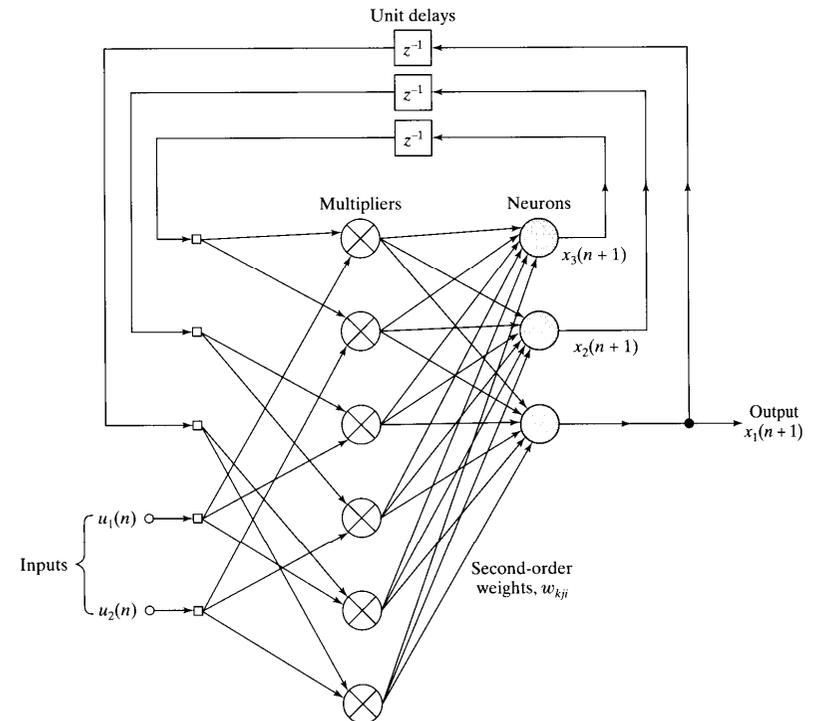
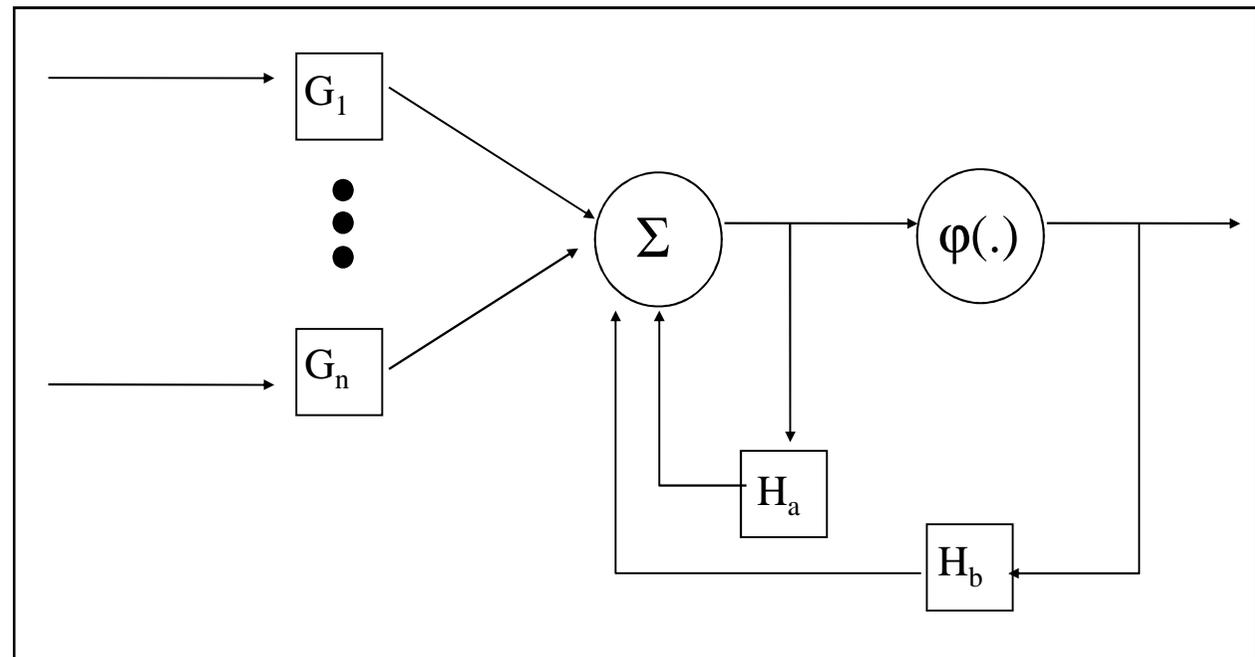


FIGURE 15.5 Second-order recurrent network; bias connections to the neurons are omitted to simplify the presentation. The network has 2 inputs and 3 state neurons, hence the need for $3 \times 2 = 6$ multipliers.

Redes Globalmente de Alimentação Direta e Localmente Recorrentes (LRGF)

- Realimentação local de ativação (*local activation feedback*) permite combinações de versões atrasadas do somatório ponderado de cada unidade de processamento.
- Realimentação local de saída (*local output feedback*) consiste da entrada em uma unidade de versões atrasadas de sua própria saída.
- Realimentação local de sinapse (*local synapse feedback*) compreende uma estrutura de realimentação em cada sinapse.



Controlabilidade e Observabilidade

- Um sistema LTID é *controlável* se um dado estado pode ser dirigido para qualquer estado desejado em um número finito de passos de tempo.
- Um sistema LTID é *observável* se seu estado pode ser determinado a partir de um conjunto finito de medidas de pares entrada-saída.
- Um estado \underline{x} é um estado de equilíbrio se para uma entrada \underline{u} ele satisfizer a condição $\underline{x} = \varphi(\underline{A}\underline{x} + \underline{B}\underline{u})$, sem perda de generalidade faz-se $\underline{x} = \mathbf{0}$ and $\underline{u} = \mathbf{0} \rightarrow \mathbf{0} = \varphi(\mathbf{0})$. A origem representa o ponto de equilíbrio.

Controlabilidade e Observabilidade

- Lineariza-se $\underline{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{A}\underline{\mathbf{x}} + \mathbf{B}\underline{\mathbf{u}})$ através da expansão da expressão por série de Taylor em torno de $\underline{\mathbf{x}} = \mathbf{0}$ e $\underline{\mathbf{u}} = \mathbf{0}$ na qual se considera os termos de primeira ordem:

$\delta \mathbf{x}(n+1) = \boldsymbol{\varphi}'(\mathbf{0}) \mathbf{W}_a \delta \mathbf{x}(n) + \boldsymbol{\varphi}'(\mathbf{0}) \mathbf{w}_b \delta u(n)$ onde $\delta \mathbf{x}(n)$ e $\delta u(n)$ são pequenos deslocamentos e a matriz $q \times q$ $\boldsymbol{\varphi}'(\mathbf{0})$ é o jacobiano de $\boldsymbol{\varphi}(\mathbf{v})$ para o argumento \mathbf{v} .

- O sistema linearizado compreende as equações de estado e de saída:

$$\delta \mathbf{x}(n+1) = \mathbf{A} \delta \mathbf{x}(n) + \mathbf{b} \delta u(n)$$

$$\delta y(n) = \mathbf{c}^T \delta \mathbf{x}(n)$$

Controlabilidade e Observabilidade

- O sistema linearizado, $\delta\mathbf{x}(n+1) = \mathbf{A} \delta\mathbf{x}(n) + \mathbf{b} \delta\mathbf{u}(n)$, é controlável se:
 - A matriz $\mathbf{M}_c = [\mathbf{A}^{q-1} \mathbf{b}, \dots, \mathbf{A} \mathbf{b}, \mathbf{b}]$ tem posto (*rank*) q , pois assim a equação do processo linearizado tem solução única.
 - A matriz \mathbf{M}_c é chamada de matriz de controlabilidade do sistema linearizado.
- De modo análogo, pode-se linearizar a expressão da saída:
 $\delta y(n) = \mathbf{c}^T \delta \mathbf{x}(n) \rightarrow \mathbf{M}_o = [\mathbf{c}, \mathbf{c} \mathbf{A}^T, \dots, \mathbf{c} (\mathbf{A}^T)^{q-1}]$
- O sistema linearizado, $\delta\mathbf{x}(n+1) = \mathbf{A} \delta\mathbf{x}(n) + \mathbf{b} \delta\mathbf{u}(n)$ e $\delta y(n) = \mathbf{c}^T \delta \mathbf{x}(n)$ é observável se:
 - A matriz $\mathbf{M}_o = [\mathbf{c}, \mathbf{c} \mathbf{A}^T, \dots, \mathbf{c} (\mathbf{A}^T)^{q-1}]$ tem posto q .
 - A matriz \mathbf{M}_o é chamada de matriz de observabilidade do sistema linearizado.

Controlabilidade e Observabilidade

- Teorema da controlabilidade local: Seja uma rede neural recorrente e sua versão linearizada em torno de sua origem. Se o sistema linearizado for controlável, então a rede recorrente é localmente controlável em torno da origem.
- Teorema da observabilidade local: Seja uma rede recorrente e sua versão linearizada em torno da origem. Se o sistema linearizado for observável, então a rede recorrente é localmente observável em torno da origem.

Poder Computacional de Redes Recorrentes

- Teorema I (Siegelmann e Sontag, 1991): Toda máquina de Turing pode ser simulada por uma rede recorrente completamente conectada construída sobre neurônios com função de ativação sigmoideal.
- A máquina de Turing é um dispositivo de computação abstrato formado por blocos funcionais:

- Unidade de Controle: assume um estado entre número finito de opções;
- Fita linear: fita infinita em ambas direções que armazena um símbolo (de um conjunto deles) por componente (quadrado na fita).
- Cabeça ler-escrever: Transmite dados de e para a unidade de controle quando se move sobre a fita.

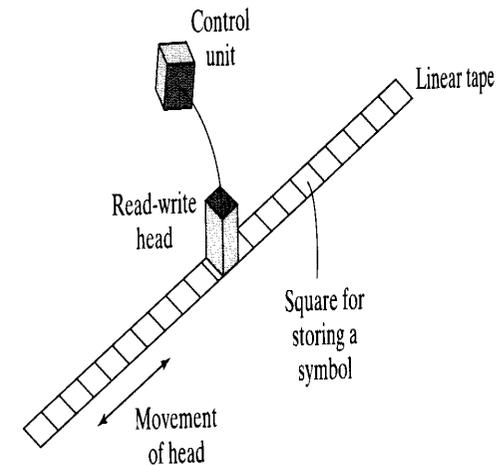


FIGURE 15.8 Turing Machine.

Poder Computacional de Redes Recorrentes

- Teorema II (Siegelmann et al., 1997): Redes NARX com uma camada escondida cujos neurônios limitados com funções de ativação saturadas em um lado e neurônio linear de saída pode simular redes recorrentes totalmente conectadas com funções de ativação saturadas em um lado e limitadas, exceto para um atraso linear.

- Uma RN recorrente totalmente conectada com N neurônios tem atraso linear se computa uma tarefa no tempo T , e o tempo total para uma NARX equivalente é $(N+1)T$.
- Função de ativação limitada e saturada em um lado (*bounded, one-sided saturated activation functions - BOSS*):
 1. $a \leq \varphi(x) \leq b$, $a \neq b$, para todo $x \in \mathbb{R}$
 2. Existe s e S , $\varphi(x) = S$ para todo $a \leq x$.
 3. $\varphi(x_1) \neq \varphi(x_2)$ para alguns x_1 e x_2 .

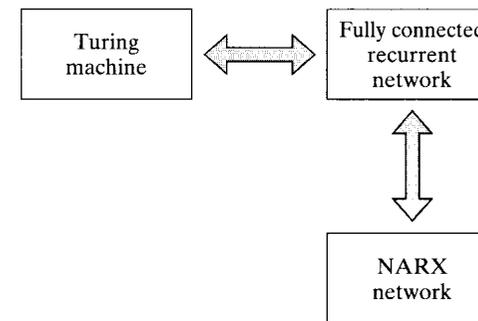


FIGURE 15.9 Illustration of Theorems I and II, and corollary to them.

- Corolário dos Teo. I e II (Giles, 1996): Redes NARX com uma camada escondida de nodos, funções de ativação BOSS e nodos lineares de saída são equivalentes em Turing.

Treinamento

- Treinamento por época: Para uma dada época, a rede recorrente inicia execução de um estado inicial até atingir um novo estado, ponto no qual o treinamento é parado e a rede é ressetada para um estado para a próxima época.
- Treinamento contínuo é adequado para situações onde não estão disponíveis estados para ressertar e/ou aprendizagem for requerida. A rede aprende enquanto a rede executa o processamento do sinal.

Treinamento

- O algoritmo Retro-propagação ao longo do tempo (BPTT) é uma extensão do algoritmo MLP-BP padrão. Ele pode ser obtido por desdobramento da operação temporal da rede para uma rede de camadas com alimentação para frente. A topologia desta rede tem número de camadas igual ao número de passos de tempo considerados.

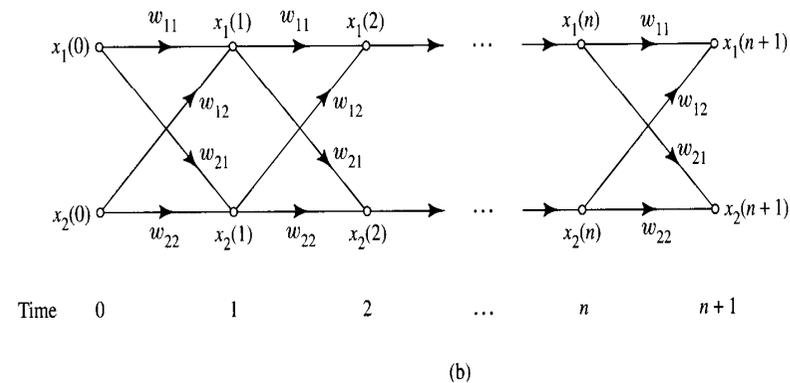
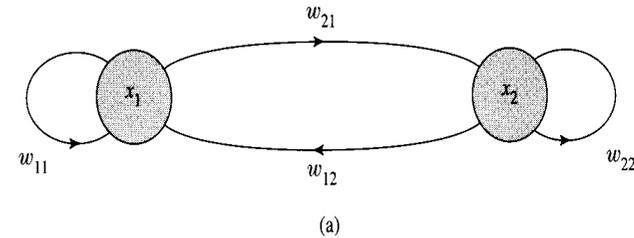


FIGURE 15.10 (a) Architectural graph of a two-neuron recurrent network \mathcal{N} . (b) Signal-flow graph of the network \mathcal{N} unfolded in time.

Treinamento

- Aprendizagem recorrente em tempo real (*real-time recurrent learning* - RTRL)
 - Camada concatenada entrada-realimentação;
 - Camada de processamento de nodos que computam.

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$$

$$E(n) = (1/2) \mathbf{e}^T(n) \mathbf{e}(n)$$

$$E_{\text{Total}} = \sum_n E(n)$$

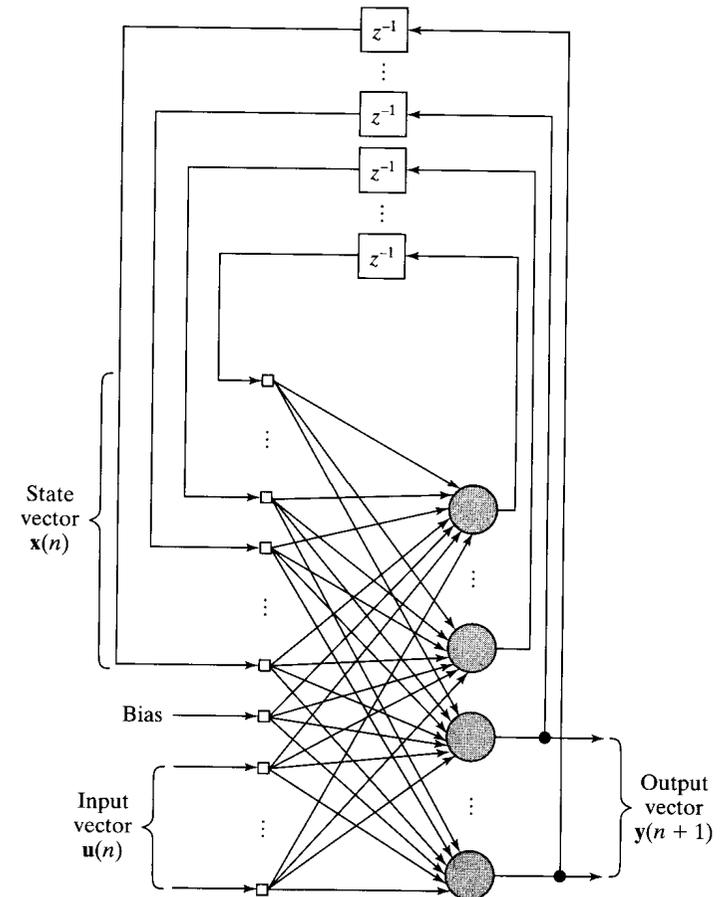


FIGURE 15.11 Fully connected recurrent network for formulation of the RTRL algorithm.