IF-705 – Automação Inteligente
Perceptrons de Múltiplas Camadas

Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
aluizioa@cin.ufpe.br

UFPE

Centro de Informática
UFPE

Conteúdo

- Histórico
- Apresentação
- Alguns Tipos de Redes Recorrentes Supervisionadas
- Treinamento
  - BPTT
  - RTRL

Histórico

- A rede recorrente por estudos de autômatos (Kleene, 1954).
- Teoria do filtro de Kalman (Rudolf E. Kalman, 1960).
- 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.

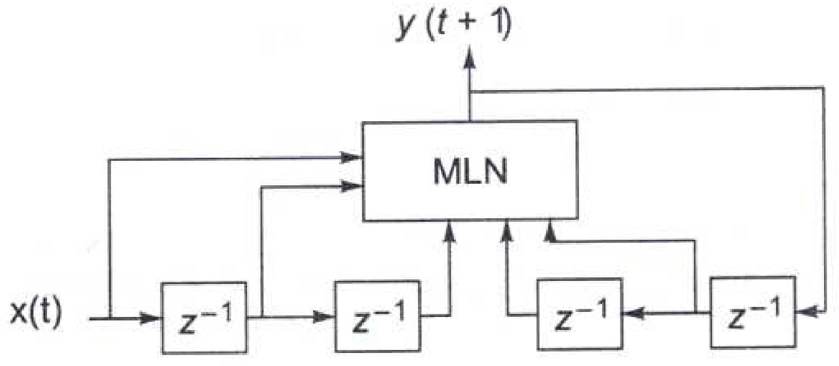
Histórico

- O modelo retro-propagação ao longo do tempo (*back-propagation through time* - BPTT) introduzido em Rumelhart (1986).
- O modelo Retro-propagação 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 Narenda, 1965).

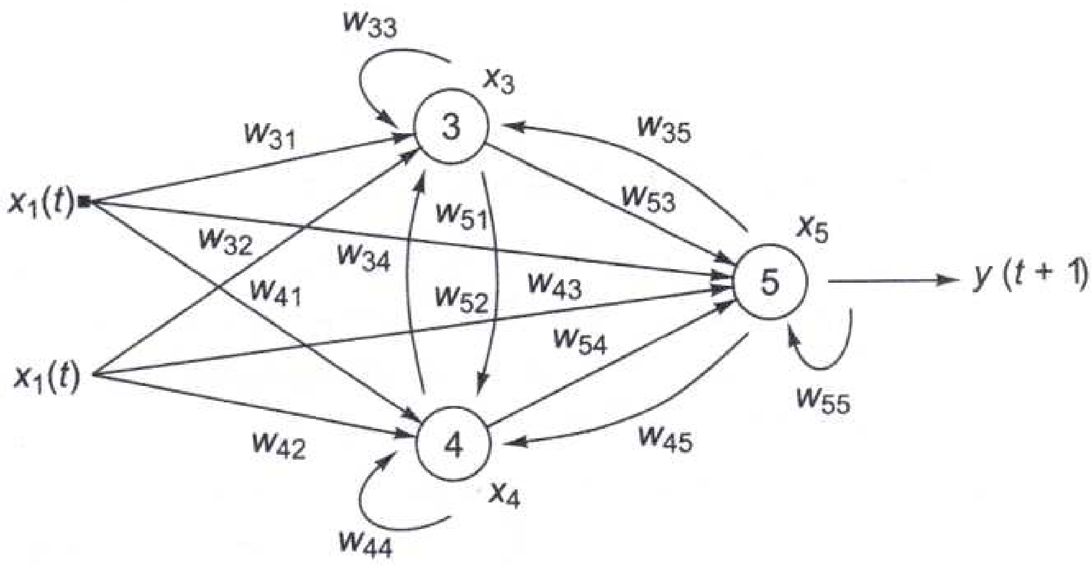
Apresentação

- Nas redes de alimentação direta, depois de ajustados os pesos, a saída dependerá apenas da entrada.
  - Não importa o histórico de ativação das unidades de processamento;
  - Não há dinâmica envolvida.
- Uma rede neural recorrente possui conexões de realimentação:
  - É um sistema dinâmico não-linear;
  - Exibe comportamento dinâmico não-linear: pontos fixos estáveis ou não, ciclos limites, bifurcações e caos, são exemplos.
- 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.

Apresentação



(a)



(b)

RNs recorrentes: (a) parcialmente conectada;

(b) totalmente conectada

Apresentação

$$y(t+1) = x_5(t+1) = f(net_5(t+1)) \therefore$$

$$y(t+1) = f\left(\sum_{i=1}^{5} w_{5i} x_i(t)\right); \text{ onde } f(v) = \frac{1}{1+\exp(-v)}$$

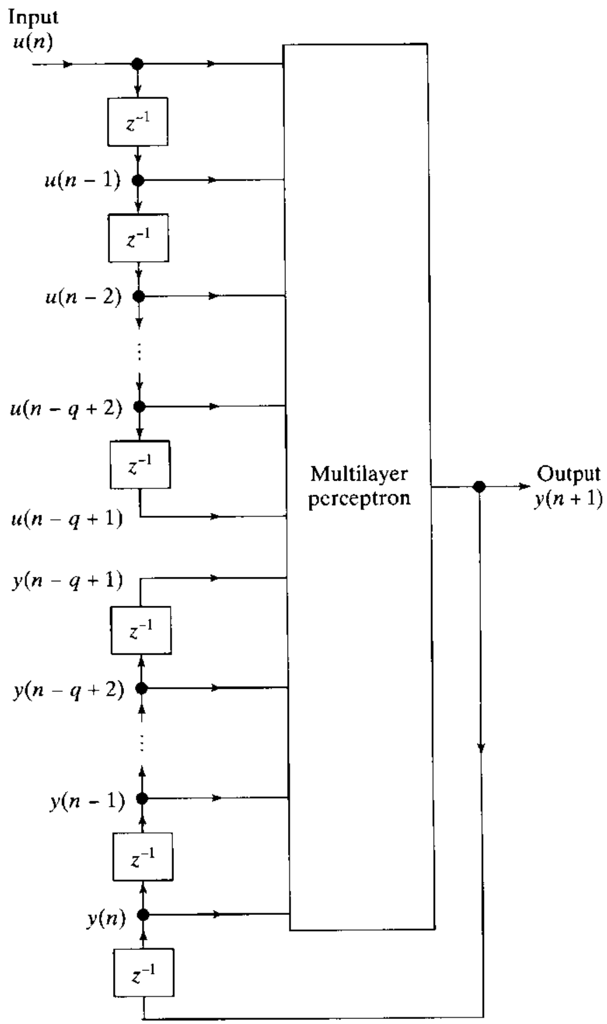
$$x_i(t+1) = f(net_i(t+1)) = f\left(\sum_{j} \omega_{ij} x_j(t)\right)$$

- Qualquer unidade de processamento pode ter sua ativaçã determinada pela equação acima.

Redes Neurais Recorrentes - NARX

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

  $y(n+1) = F(y(n),\ldots,y(n-q+1),u(n),\ldots,u(n-q+1))$
- 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 da 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).



Redes Neurais Recorrentes - 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, $n=n-q+1$:

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

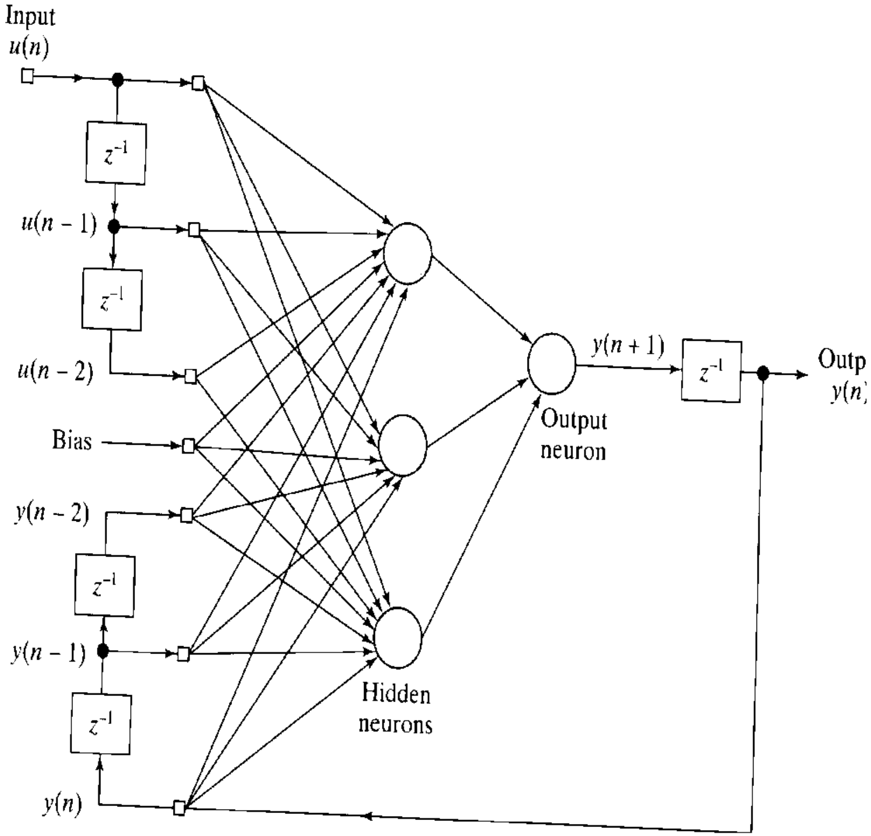


FIGURE 15.7 NARX network with $q = 3$ hidden neurons.

Redes Neurais Recorrentes - 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.

Centro de Informática
UFPE

Redes Neurais Recorrentes - Modelo de Espaço de Estados

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

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

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

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

- 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.

Centro de Informática
UFPE

Redes Neurais Recorrentes - 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.

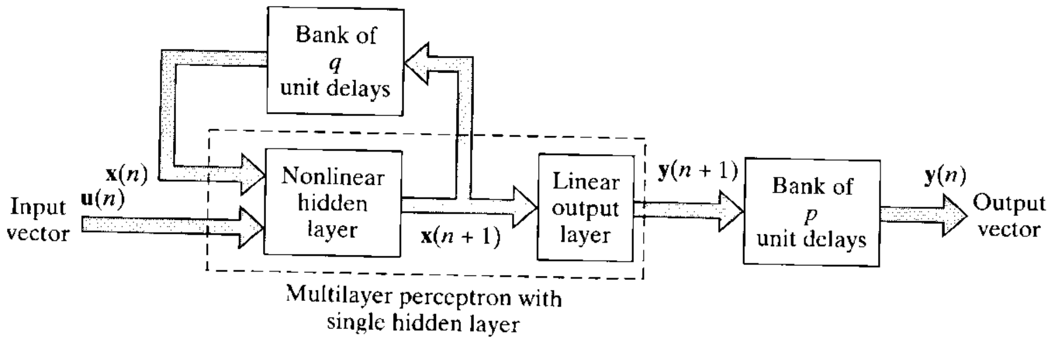


FIGURE 15.2 State-space model.

- 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.

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

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

Centro de Informática UFPE

Redes Neurais Recorrentes - 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.

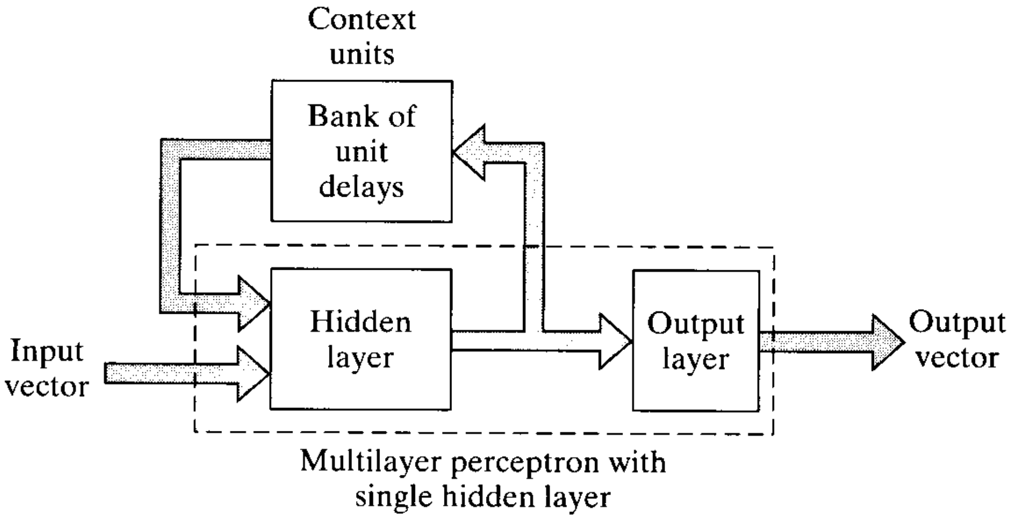


FIGURE 15.3 Simple Recurrent network (SRN).

Treinamento

- Entre os treinamentos mais empregados estão:
  - Abordagem *offline*: Retropropagação ao longo do tempo (*Back Propagation Through Time* - BPTT);
  - Abordagem *online*: Aprendizagem Recorrente em Tempo Real (*Real Time Recurrent Learning* - RTRL).
- Nos dois métodos, transforma-se a rede recorrente em uma versão de rede de alimentação direta. Deste modo, o treinamento ocorre como se fora uma rede instantânea:

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.

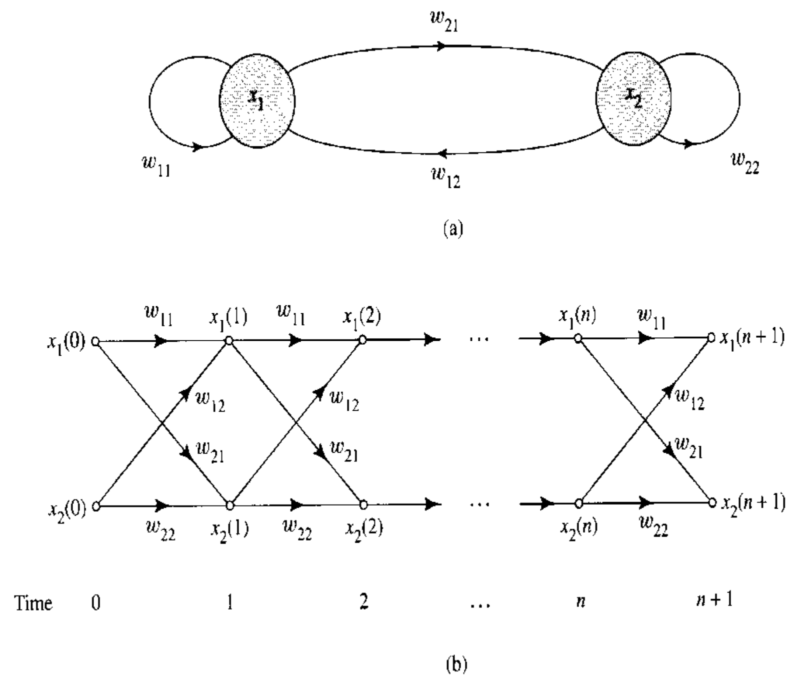
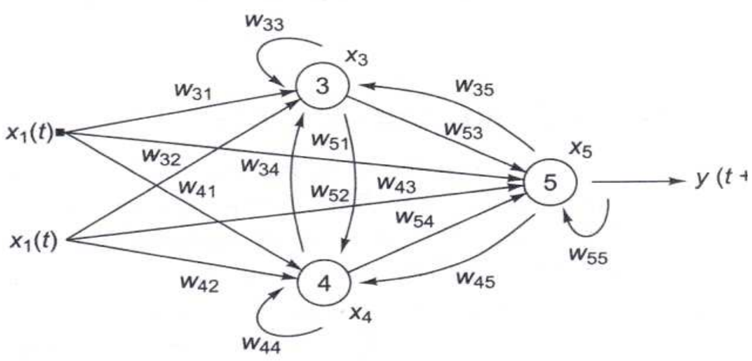
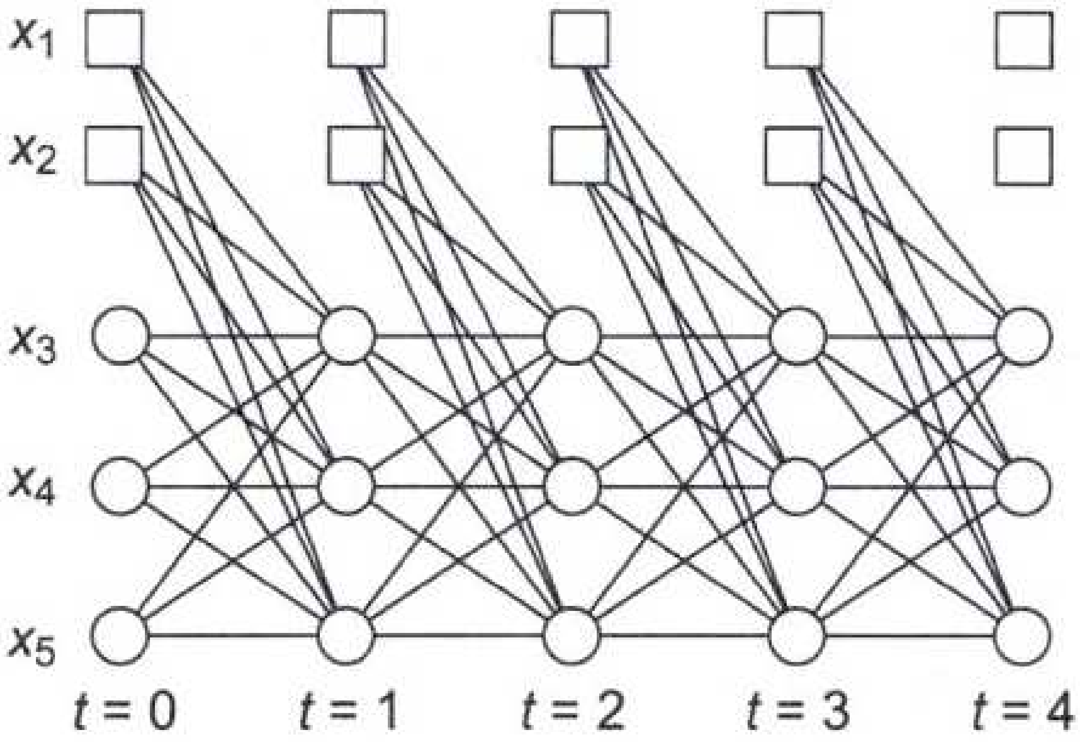


FIGURE 15.10 (a) Architectural graph of a two-neuron recurrent network $\mathcal{N}$. (b) Signal-flow graph of the network $\mathcal{N}$ unfolded in time.

Treinamento

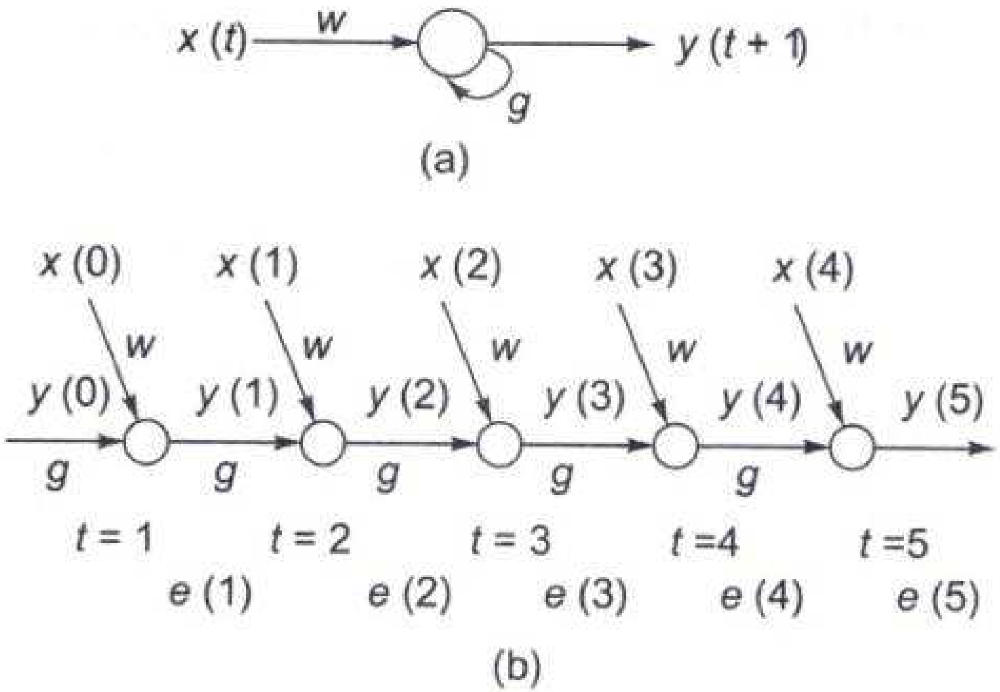
- Exemplo: Desdobramento da rede .





Treinamento

- Exemplo: Figura 2.17



(a) Original network with a single neuron and (b) unfolded network

Treinamento

Example 2.6 Enumerate steps for training the unfolded network of the simple recurrent network given in Figure 2.17.

Solution

The steps to proceed further are as follows:

1. Compute the response of the sequence $y(1)$ to $y(5)$, given the sequence $x(0)$ to $x(4)$, and $y(0)$.
2. Compute the error $e(5) = y^d(5) - y(5)$
   $\Delta w^5 = \eta \delta_5 x(4)$
   $\Delta g^5 = \eta \delta_5 y(4)$, where $\delta_5 = y(5)(1 - y(5))e(5)$
3. Compute $e(4)$, $e(3)$, $e(2)$, and $e(1)$ and proceed the same way until the first node $t = 1$
   $\Delta w^4 = \eta \delta_4 x(3)$
   $\Delta g^4 = \eta \delta_4 y(3)$
   where $\delta_4 = y(4)(1 - y(4))[\delta_5 g + e(4)]$

Treinamento

- De modo similar:

$$\Delta w^3 = \eta \delta_3 x(2)$$

$$\Delta g^3 = \eta \delta_3 y(2) \quad \delta_3 = y(3)(1 - y(3))[\delta_4 g + e(3)]$$

$$\Delta w^2 = \eta \delta_2 x(1)$$

$$\Delta g^2 = \eta \delta_2 y(1) \quad \delta_2 = y(2)(1 - y(2))[\delta_3 g + e(2)]$$

$$\Delta w^1 = \eta \delta_1 x(0)$$

$$\Delta g^1 = \eta \delta_1 y(0) \quad \delta_1 = y(1)(1 - y(1))[\delta_2 g + e(1)]$$

- Regra de atualização:

$$w^{\text{new}} = w^{\text{old}} + \sum_{i=1}^{5} \Delta w^i$$

$$g^{\text{new}} = g^{\text{old}} + \sum_{i=1}^{5} \Delta g^i$$

Treinamento

- Aprendizagem recorrente em tempo real (*real-time recurrent learning* - RTRL)
  - A rede é tratada no tempo de modo análogo ao tratamento no espaço;
  - A regra delata generalizada é aplicada para ajustes dos pesos;
  - Camada concatenada entrada-realimentação;
  - Camada de processamento de nodos que computam.
  - A função de custo a ser minimizada será:
  - $E(t+1)=(1/2)|\boldsymbol{y}^d(t+1)-\boldsymbol{y}(t+1)|^2$

$$E(t+1) = \frac{1}{2}(y^d(t+1) - y(t+1))^2$$

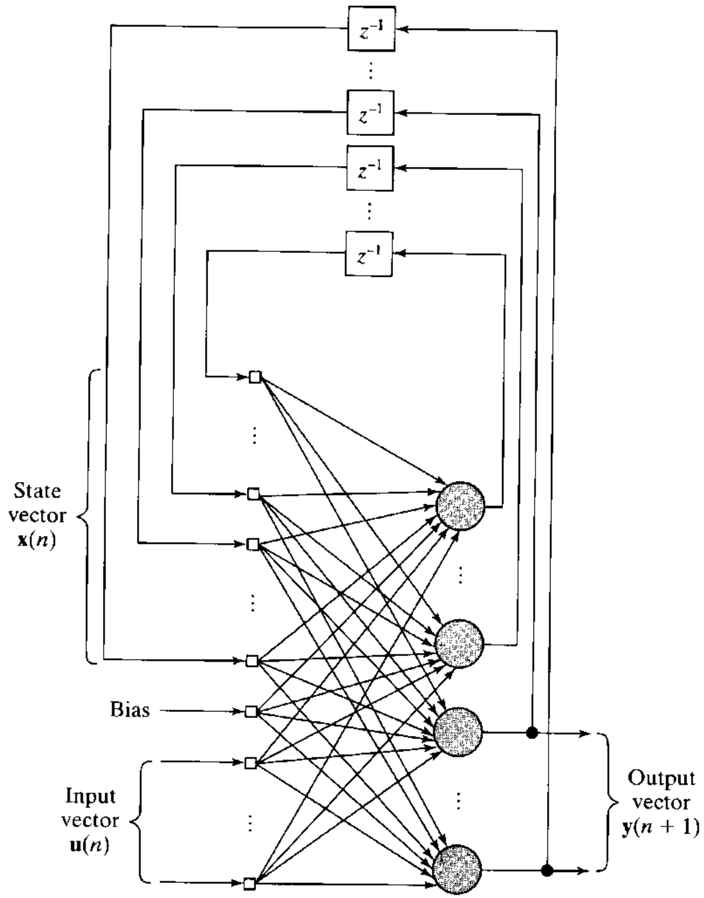


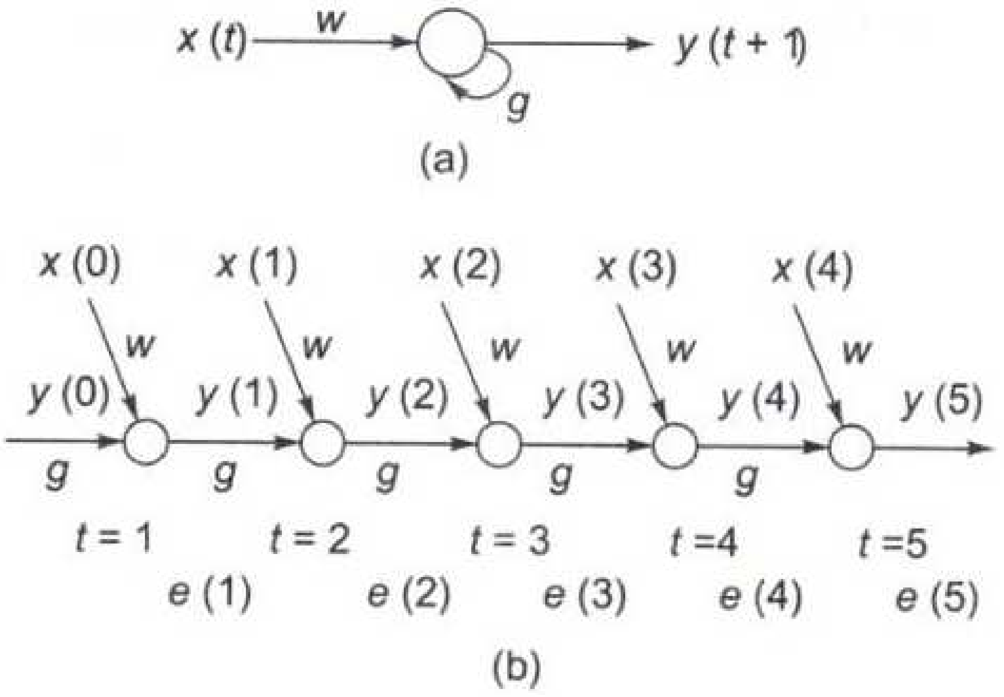
FIGURE 15.11 Fully connected recurrent network for formulation of the RTRL algorithm.

Centro de Informática UFPE

U F P E

Treinamento

- Exemplo: $y(t+1) = f\big(net(t+1)\big) = f(wx(t) + gy(t))$



(a) Original network with a single neuron and (b) unfolded network

Treinamento

A função de custo a ser minimizada é $E(t+1) = \frac{1}{2}\left(y^d(t+1) - y(t+1)\right)^2$

Logo, os ajustes dos parâmetros livres são

$$w(t+1) = w(t) + \eta \frac{\partial E(t+1)}{\partial w} = w(t) - [y^d(t+1) - y(t+1)] \frac{\partial y(t+1)}{\partial w}$$

$$g(t+1) = g(t) + \eta \frac{\partial E(t+1)}{\partial g} = g(t) - [y^d(t+1) - y(t+1)] \frac{\partial y(t+1)}{\partial g}$$

Determinação das derivadas parciais:

$$\frac{\partial y(t+1)}{\partial w} = \frac{\partial y(t+1)}{\partial net(t+1)} \times \frac{\partial net(t+1)}{\partial w}$$

$$\frac{\partial y(t+1)}{\partial net(t+1)} = y(t+1)(1 - y(t+1))$$

$$\frac{\partial net(t+1)}{\partial w} = \frac{\partial [wx(t) + gy(t)]}{\partial w} = x(t) + g\frac{\partial y(t)}{\partial w} = x(t) + gP_w(t)$$

Centro de Informática UFPE

Treinamento

Portanto,

$$\frac{\partial y(t+1)}{\partial w} = y(t+1)\big(1 - y(t+1)\big)[x(t) + gP_w(t)]$$

De modo análogo.

$$\frac{\partial y(t+1)}{\partial g} = y(t+1)\big(1 - y(t+1)\big)\big[y(t) + gP_g(t)\big]$$

onde se define

$$P_w(t) = \frac{\partial y(t)}{\partial w}, \qquad P_w(0) = 0$$

$$P_g(t) = \frac{\partial y(t)}{\partial g}, \qquad P_g(0) = 0$$

Centro de Informática
UFPE

Treinamento

Logo,

$$\frac{\partial E(t+1)}{\partial w} = -[y^d(t+1) - y(t+1)]y(t+1)\big(1 - y(t+1)\big)[x(t) + gP_w(t)]$$

$$\frac{\partial E(t+1)}{\partial g} = -[y^d(t+1) - y(t+1)]y(t+1)\big(1 - y(t+1)\big)\big[y(t) + gP_g(t)\big]$$

Finalmente, os ajustes dos pesos podem ser re-escritos

$$w(t+1) = w(t) + \eta\frac{\partial E(t+1)}{\partial w} = w(t) + \eta[y^d(t+1) - y(t+1)]P_w(t+1)$$

$$g(t+1) = g(t) + \eta\frac{\partial E(t+1)}{\partial g} = g(t) + \eta[y^d(t+1) - y(t+1)]P_G(t+1)$$

Referências

- Behera, L. & Kar, I. (2010). *Intelligent Systems and Control: Principles and Applications*. Oxford University Press. ISBM13: 978-0-19-806315-5.
- Haykin, S. (2008). *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall.

Centro de Informática UFPE

U F P E