

Redes Neurais Artificiais e Máquinas de Aprendizado (Parte 2)

Índice

1.	Rede neural com função de ativação de base radial	3
1.1	Formulação matemática.....	6
1.2	Métodos de treinamento já propostos na literatura	11
1.3	Capacidade de aproximação universal de redes RBF.....	11
1.4	O método dos quadrados mínimos para modelos lineares nos parâmetros	12
1.5	Obtenção da solução do problema de regressão linear	13
1.6	Exemplo	16
1.7	Aproximação usando rede neural RBF	19
1.8	Determinação dos centros e dispersões	21
1.9	Aplicação das propostas de determinação de centros e dispersão.....	22
1.10	Referências para redes neurais RBF	25
1.11	Bibliografia complementar para redes RBF.....	25
2.	Máquinas de aprendizado extremo (ELMs).....	27
2.1	Exemplos de máquinas de aprendizado extremo	32
2.2	Treinamento das ELMs	34
2.3	Como encontrar os pesos sinápticos.....	35
2.4	Como encontrar o coeficiente de ponderação	36
2.5	Referências bibliográficas para ELMs.....	37
3.	O Jogo da Vida.....	38
4.	Exemplos de auto-organização na natureza.....	39
5.	Treinamento não-supervisionado.....	44
6.	Mapas Auto-Organizáveis de Kohonen.....	45
6.1	Arranjo unidimensional	46

6.2	Arranjo bidimensional	47
6.3	Algoritmo de ajuste dos pesos	51
6.4	Um passo de ajuste num arranjo unidimensional	53
6.5	Ajuste de pesos com restrição de vizinhança	54
6.6	Discriminação dos agrupamentos	55
6.7	Ferramentas de visualização e discriminação	57
6.8	Ordenamento de pontos em espaços multidimensionais	61
6.9	Roteamento de veículos (múltiplos mapas auto-organizáveis)	62
6.10	Mapas auto-organizáveis construtivos	63
6.11	Questões a serem melhor investigadas	64
6.12	Extensões	65
6.13	Referências para mapas auto-organizáveis	65
7.	Redes neurais recorrentes	68
7.1	Modelagem de sistemas dinâmicos lineares	73
7.2	Modelagem de sistemas dinâmicos não-lineares	74
7.3	Treinamento supervisionado para redes recorrentes	75
8.	Rede de Hopfield: recorrência e dinâmica não-linear	76
8.1	Princípios básicos de sistemas dinâmicos não-lineares	78
8.2	Exemplos de comportamentos dinâmicos não-lineares	80
8.3	Pontos de equilíbrio como memórias endereçáveis por conteúdo	83
8.4	Recapitulação dos principais conceitos – Rede de Hopfield	88
8.5	Regra de Hebb	89
8.6	Atratores espúrios	90
8.7	Solução de problemas de natureza combinatória	91
8.8	Solução de problemas de programação matemática	95
8.9	Referências bibliográficas – Redes neurais recorrentes	99
9.	Deep learning	104

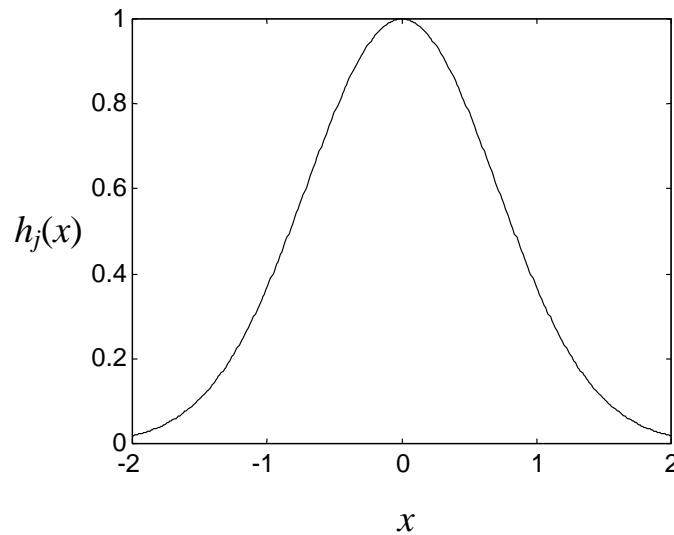
1. Rede neural com função de ativação de base radial

- Uma função de ativação de base radial é caracterizada por apresentar uma resposta que decresce (ou cresce) monotonicamente com a distância a um ponto central.
- O centro e a taxa de decrescimento (ou crescimento) em cada direção são alguns dos parâmetros a serem definidos. Estes parâmetros devem ser constantes caso o modelo de regressão seja tomado como linear nos parâmetros ajustáveis.
- Uma função de base radial monotonicamente decrescente típica é a função gaussiana, dada na forma:

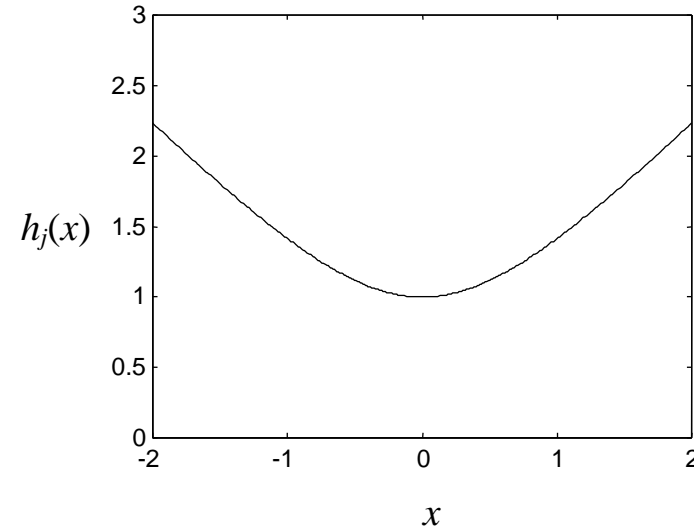
$$\square \quad h_j(x) = \exp\left(-\frac{(x - c_j)^2}{r_j^2}\right), \text{ para o caso escalar (veja Figura 1(a));}$$

- Uma função de base radial monotonicamente crescente típica é a função multiquádrica dada na forma:

□ $h_j(x) = \frac{\sqrt{r_j^2 + (x - c_j)^2}}{r_j}$, para o caso escalar (veja Figura 1(b));



(a)



(b)

Figura 1 – Exemplos de funções de base radial monovariáveis, com $c_j = 0$ e $r_j = 1$

- No caso multidimensional e tomando a função gaussiana, $h_j(\mathbf{x})$ assume a forma:

$$h_j(\mathbf{x}) = \exp\left(-(\mathbf{x} - \mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{x} - \mathbf{c}_j)\right) \quad (1)$$

onde $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ é o vetor de entradas, $\mathbf{c}_j = [c_{j1} \ c_{j2} \ \cdots \ c_{jn}]^T$ é o vetor que define o centro da função de base radial e a matriz Σ_j é definida positiva e diagonal, dada por:

$$\Sigma_j = \begin{bmatrix} \sigma_{j1} & 0 & \cdots & 0 \\ 0 & \sigma_{j2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{jn} \end{bmatrix},$$

de modo que $h_j(\mathbf{x})$ pode ser expandida na forma:

$$h_j(\mathbf{x}) = \exp\left(-\frac{(x_1 - c_{j1})^2}{\sigma_{j1}} - \frac{(x_2 - c_{j2})^2}{\sigma_{j2}} - \cdots - \frac{(x_n - c_{jn})^2}{\sigma_{jn}}\right). \quad (2)$$

- Neste caso, os elementos do vetor $\sigma_j = [\sigma_{j1} \ \sigma_{j2} \ \cdots \ \sigma_{jn}]^T$ são responsáveis pela taxa de decrescimento da gaussiana junto a cada coordenada do espaço de entrada, e o

argumento da função exponencial é uma norma ponderada da diferença entre o vetor de entrada e o centro da função de base radial.

1.1 Formulação matemática

- As funções de base radial (são funções não-lineares) podem ser utilizadas como funções-base em qualquer tipo de modelo de regressão não-linear (linear ou não-linear nos parâmetros) e, particularmente, como função de ativação de qualquer tipo de rede multicamada.
- O fato do modelo de regressão resultante ser linear ou não-linear nos parâmetros se deve à possibilidade ou não de se ajustar os centros e as dispersões das funções.
- As redes neurais com função de ativação de base radial (RBF) apresentam três diferenças principais em relação às redes tipo perceptron multicamadas:
 - Elas sempre apresentam uma única camada intermediária;
 - Neurônios de saída são sempre lineares;
 - Os neurônios da camada intermediária têm apenas uma função de base radial como função de ativação, ao invés de uma função sigmoideal ou outras.

- Como exposto acima, se apenas os pesos da camada de saída formarem o conjunto de parâmetros ajustáveis, então a rede neural é linear nos parâmetros. Caso contrário, ou seja, quando os centros \mathbf{c}_j e as matrizes Σ_j , $j = 1, \dots, n$, também são ajustáveis, a rede neural é não-linear nos parâmetros, admitindo o próprio algoritmo de retro-propagação do erro para o processo de ajuste via treinamento supervisionado, como feito no caso do perceptron multicamadas, embora aqui os mínimos locais tenham uma influência muito maior.
- A arquitetura da rede é apresentada na Figura 2, para o caso de uma única saída, resultando no seguinte mapeamento de entrada-saída:

$$y = \sum_{j=1}^m w_j h_j(\mathbf{x})$$

- Caso \mathbf{c}_j e Σ_j , $j = 1, \dots, n$, sejam ajustáveis, a saída assume a forma:

$$y = \sum_{j=1}^m w_j h_j(\mathbf{c}_j, \Sigma_j, \mathbf{x}).$$

- Substituindo as formas compactas e expandidas de $h_j(\mathbf{x})$, dadas respectivamente pelas equações (1) e (2), resultam:

$$y = \sum_{j=1}^m w_j \exp\left(-(\mathbf{x} - \mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{x} - \mathbf{c}_j)\right)$$

e

$$y = \sum_{j=1}^m w_j \exp\left(-\frac{(x_1 - c_{j1})^2}{\sigma_{j1}} - \frac{(x_2 - c_{j2})^2}{\sigma_{j2}} - \dots - \frac{(x_n - c_{jn})^2}{\sigma_{jn}}\right)$$

- Uma versão para múltiplas saídas é apresentada na Figura 3.
- A consequência imediata do uso de funções de ativação de base radial está na forma como as entradas são processadas pelos neurônios da camada intermediária. Ao invés da ativação interna de cada neurônio da camada intermediária se dar pelo emprego do produto escalar (produto interno) entre o vetor de entradas e o vetor de pesos, como no caso do perceptron, ela é obtida a partir de uma norma ponderada da diferença entre ambos os vetores.

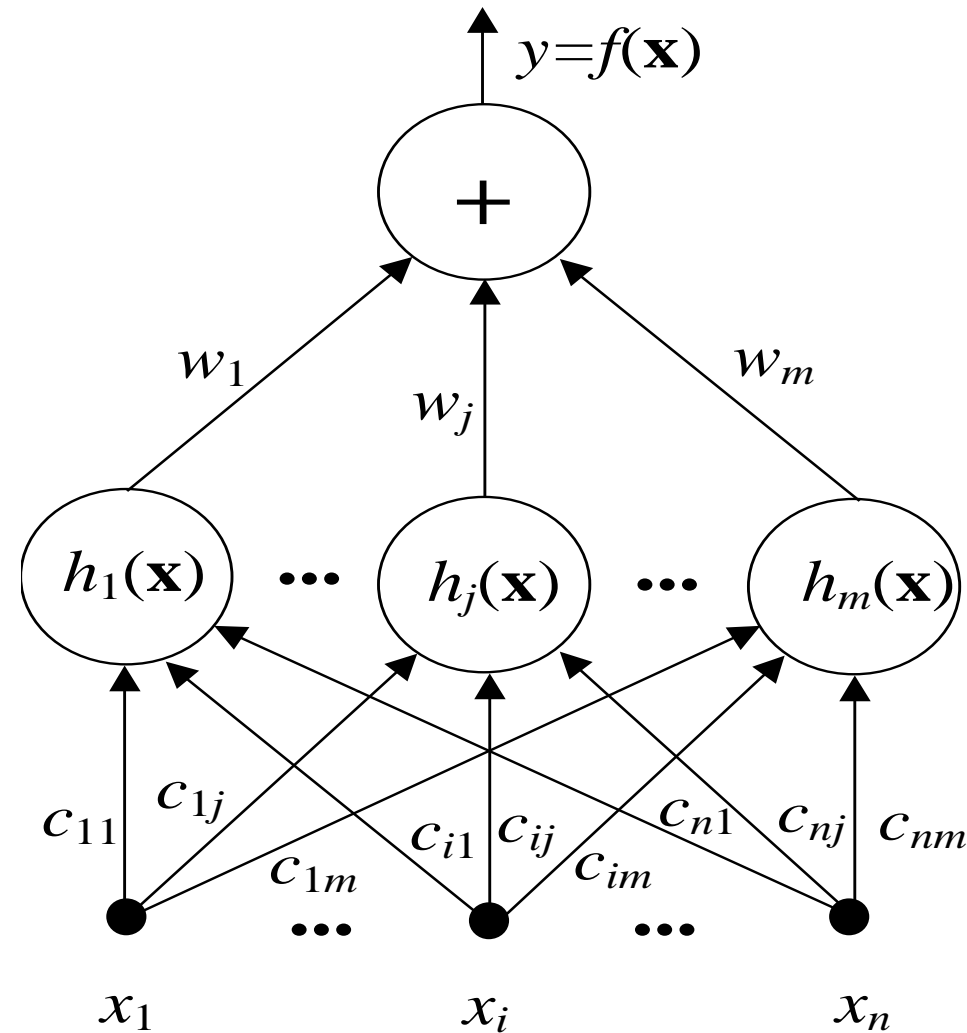


Figura 2 – Rede neural de base radial (BROOMHEAD & LOWE, 1988). Observação: Geralmente é também considerada uma entrada constante para o neurônio de saída.

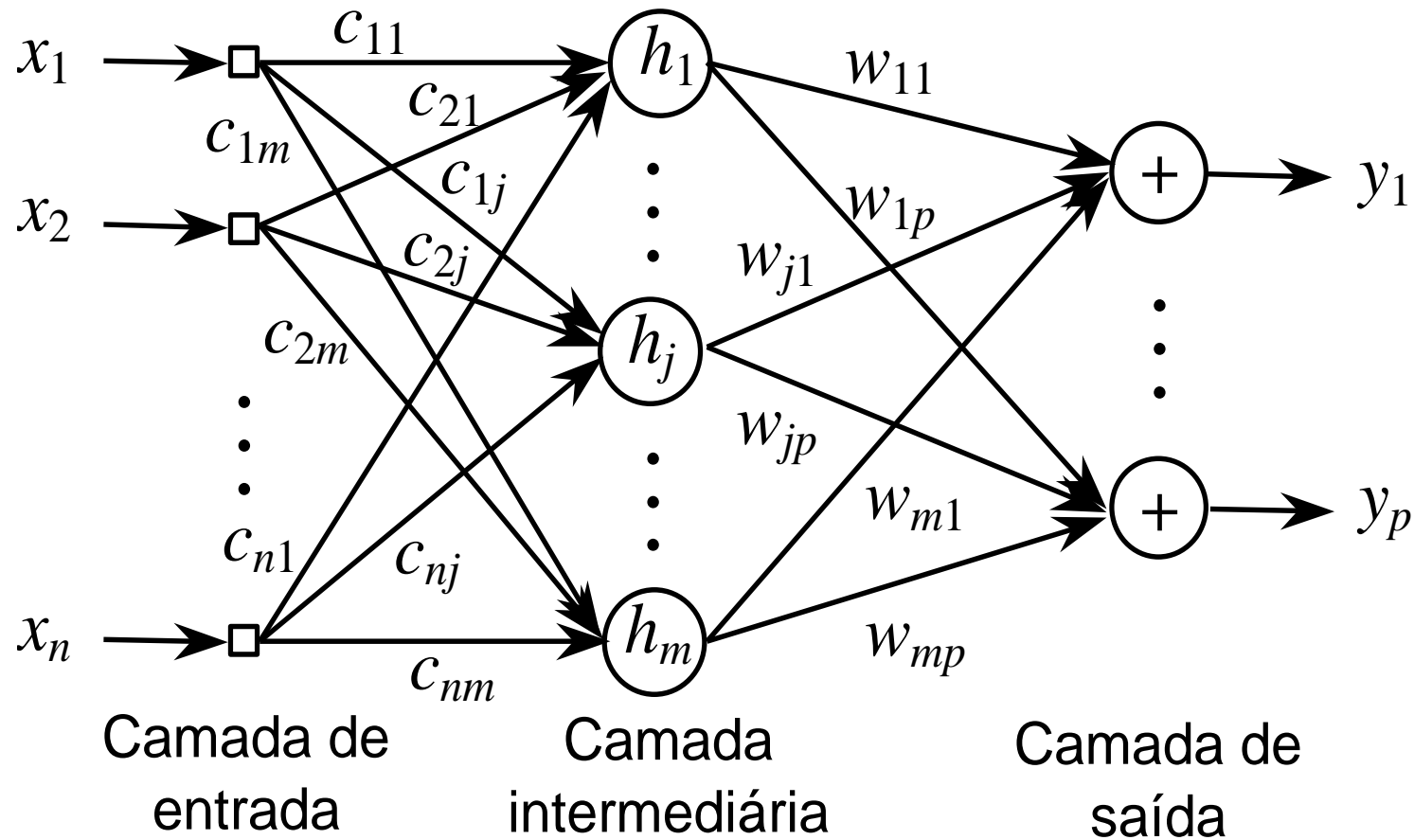


Figura 3 – Rede neural de base radial com múltiplas saídas. Observação: Geralmente é também considerada uma entrada constante para cada neurônio de saída.

1.2 Métodos de treinamento já propostos na literatura

- Várias abordagens para o treinamento de redes neurais com funções de base radial já foram propostas. Geralmente, elas podem ser divididas em duas partes:
 - Definição dos centros, forma e dispersão das funções de base radial, normalmente baseada em treinamento não-supervisionado (quantização vetorial ou algoritmo de treinamento competitivo) ou computação evolutiva;
 - Aprendizado dos pesos da camada de saída, responsáveis pela combinação linear das ativações da camada intermediária, empregando regressão linear.

1.3 Capacidade de aproximação universal de redes RBF

- Dado um número suficiente de neurônios com função de base radial, qualquer função contínua definida numa região compacta pode ser devidamente aproximada usando uma rede RBF (PARK & SANDBERG, 1991).

1.4 O método dos quadrados mínimos para modelos lineares nos parâmetros

- Quando o treinamento supervisionado é aplicado a modelos lineares nos parâmetros, o método dos quadrados mínimos conduz a um problema de otimização que apresenta solução na forma fechada.
- Assim, com um modelo de regressão linear na forma (considerando uma saída):

$$f(\mathbf{x}) = \sum_{j=1}^m w_j h_j(\mathbf{x})$$

e o conjunto de treinamento dado por $\{(\mathbf{x}_i, s_i)\}_{i=1}^N$, o método dos quadrados mínimos se ocupa em minimizar (em relação aos coeficientes da combinação linear) a soma dos quadrados dos erros produzidos a partir de cada um dos N padrões de entrada-saída.

$$\min_{\mathbf{w}} J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^N (s_i - f(\mathbf{x}_i))^2 = \min_{\mathbf{w}} \sum_{i=1}^N \left(s_i - \sum_{j=1}^m w_j h_j(\mathbf{x}_i) \right)^2$$

1.5 Obtenção da solução do problema de regressão linear

- Do Cálculo Elementar, sabe-se que a aplicação da condição de otimalidade (restrições atendidas pelos pontos de máximo e mínimo de uma função diferenciável) permite obter a solução ótima do problema de otimização $\min_{\mathbf{w}} J(\mathbf{w})$, na forma:

1. Diferencie a função em relação aos parâmetros ajustáveis;
2. Iguale estas derivadas parciais a zero;
3. Resolva o sistema de equações resultante.

- No caso em questão, os parâmetros livres são os coeficientes da combinação linear, dados na forma do vetor de pesos $\mathbf{w} = [w_1 \quad \cdots \quad w_j \quad \cdots \quad w_m]^T$.
- O sistema de equações resultante é dado na forma:

$$\frac{\partial J}{\partial w_j} = -2 \sum_{i=1}^N (s_i - f(\mathbf{x}_i)) \frac{\partial f}{\partial w_j} = -2 \sum_{i=1}^N (s_i - f(\mathbf{x}_i)) h_j(\mathbf{x}_i) = 0, \quad j=1, \dots, m.$$

- Separando-se os termos que envolvem $f(\cdot)$, resulta:

$$\sum_{i=1}^N f(\mathbf{x}_i) h_j(\mathbf{x}_i) = \sum_{i=1}^N \left[\sum_{r=1}^m w_r h_r(\mathbf{x}_i) \right] h_j(\mathbf{x}_i) = \sum_{i=1}^N s_i h_j(\mathbf{x}_i), \quad j=1, \dots, m.$$

- Portanto, existem m equações para obter as m incógnitas $\{w_r, r = 1, \dots, m\}$. Desde que os centros não sejam coincidentes, este sistema de equações possui solução única.
- Para encontrar esta solução única do sistema de equações lineares, é interessante recorrer à notação vetorial, fornecida pela álgebra linear, para obter:

$$\mathbf{h}_j^T \mathbf{f} = \mathbf{h}_j^T \mathbf{s}, \quad j=1, \dots, m,$$

onde

$$\mathbf{h}_j = \begin{bmatrix} h_j(\mathbf{x}_1) \\ \vdots \\ h_j(\mathbf{x}_N) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \sum_{r=1}^m w_r h_r(\mathbf{x}_1) \\ \vdots \\ \sum_{r=1}^m w_r h_r(\mathbf{x}_N) \end{bmatrix} \quad \text{e} \quad \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}.$$

- Como existem m equações, resulta:

$$\begin{bmatrix} \mathbf{h}_1^T \mathbf{f} \\ \vdots \\ \mathbf{h}_m^T \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^T \mathbf{s} \\ \vdots \\ \mathbf{h}_m^T \mathbf{s} \end{bmatrix}$$

- Definindo a matriz \mathbf{H} , com sua j -ésima coluna dada por \mathbf{h}_j , temos:

$$\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_m] = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & \cdots & h_m(\mathbf{x}_N) \end{bmatrix}$$

sendo possível reescrever o sistema de equações lineares como segue:

$$\mathbf{H}^T \mathbf{f} = \mathbf{H}^T \mathbf{s}$$

- O i -ésimo componente do vetor \mathbf{f} pode ser apresentado na forma:

$$f_i = f(\mathbf{x}_i) = \sum_{r=1}^m w_r h_r(\mathbf{x}_i) = [h_1(\mathbf{x}_i) \quad h_2(\mathbf{x}_i) \quad \cdots \quad h_m(\mathbf{x}_i)] \mathbf{w}$$

permitindo expressar \mathbf{f} em função da matriz \mathbf{H} , de modo que:

$$\mathbf{f} = \mathbf{H} \mathbf{w}$$

- Substituindo no sistema de equações lineares, resulta a solução ótima para o vetor de coeficientes da combinação linear (que correspondem aos pesos da camada de saída da rede neural de base radial):

$$\mathbf{H}^T \mathbf{H} \mathbf{w} = \mathbf{H}^T \mathbf{s} \Rightarrow \mathbf{w} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{s}$$

- Esta equação de solução do problema dos quadrados mínimos é conhecida como equação normal. Para que exista a inversa de $\mathbf{H}^T \mathbf{H}$, basta que a matriz \mathbf{H} tenha posto completo, já que $m \leq N$.

1.6 Exemplo

- O modelo linear de regressão mais simples é a reta, aplicada nos casos em que a entrada é escalar: $f(x) = w_1 h_1(x) + w_2 h_2(x)$, onde $h_1(x) = 1$ e $h_2(x) = x$.
- Considere que foram amostrados, na presença de ruído, três pontos da curva $y = x$, gerando o conjunto de treinamento: $\{(x_i, s_i)\}_{i=1}^3 = \{(1, 1.1), (2, 1.8), (3, 3.1)\}$.

- Obviamente, não se conhece a equação da curva, mas apenas estes três pontos amostrados. Para estimar w_1 e w_2 , vamos proceder de acordo com os passos do método dos quadrados mínimos.

$$H = \begin{bmatrix} h_1(x_1) & h_2(x_1) \\ h_1(x_2) & h_2(x_2) \\ h_1(x_3) & h_2(x_3) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \mathbf{s} = \begin{bmatrix} 1.1 \\ 1.8 \\ 3.1 \end{bmatrix} \quad \mathbf{w} = (H^T H)^{-1} H^T \mathbf{s} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Para o mesmo conjunto de treinamento, considere agora que

$$f(x) = w_1 h_1(x) + w_2 h_2(x) + w_3 h_3(x),$$

onde $h_1(x) = 1$, $h_2(x) = x$ e $h_3(x) = x^2$. Enquanto no caso anterior tínhamos $m < N$, agora temos $m = N$.

- O efeito da adição da função-base extra $h_3(x)$ representa a adição de uma coluna

$$\mathbf{h}_3 = \begin{bmatrix} h_3(x_1) \\ h_3(x_2) \\ h_3(x_3) \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 9 \end{bmatrix} \text{ junto à matriz } H, \text{ e a solução assume a forma } \mathbf{w} = \begin{bmatrix} 1 \\ -0.2 \\ 0.3 \end{bmatrix}.$$

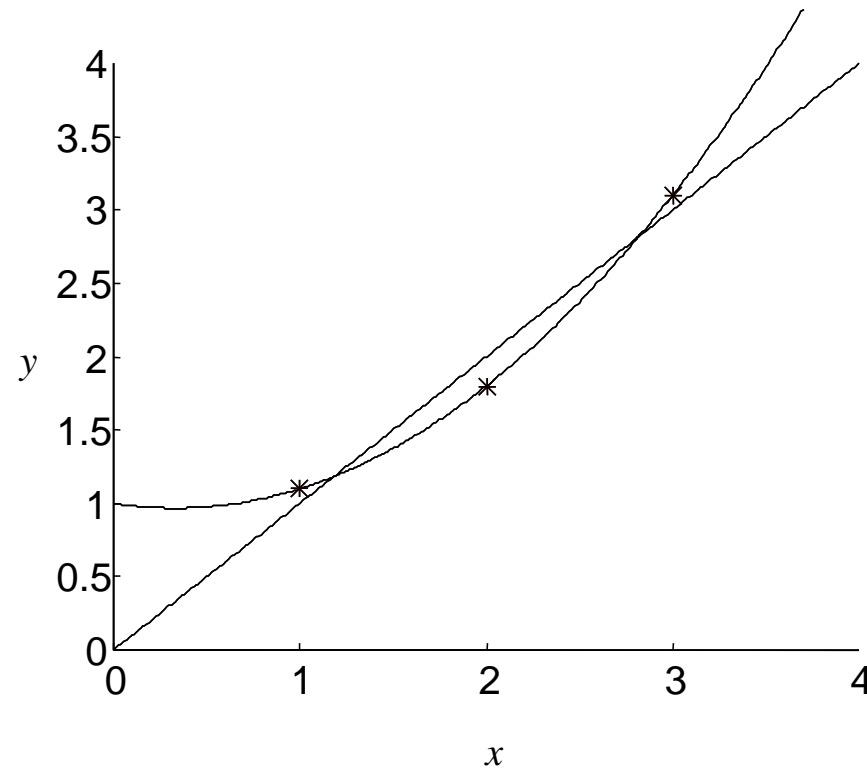
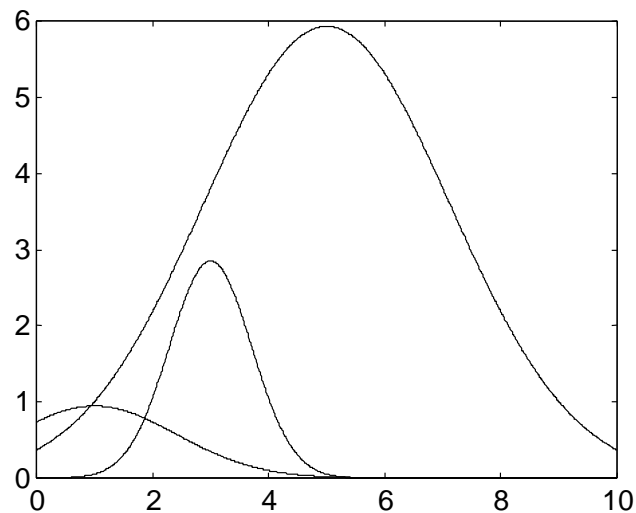
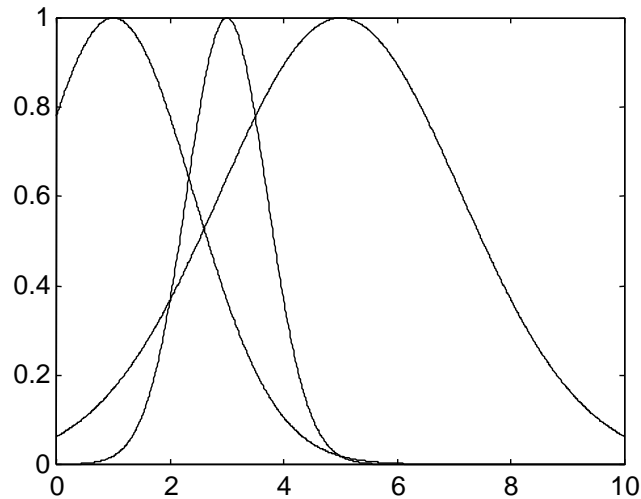


Figura 4 – Modelos de regressão linear (ordem 1 e ordem 2).

- Observe que ambos os modelos são lineares nos parâmetros (daí a denominação de regressão linear), embora para $m = 3$ tenhamos um modelo não-linear.

1.7 Aproximação usando rede neural RBF

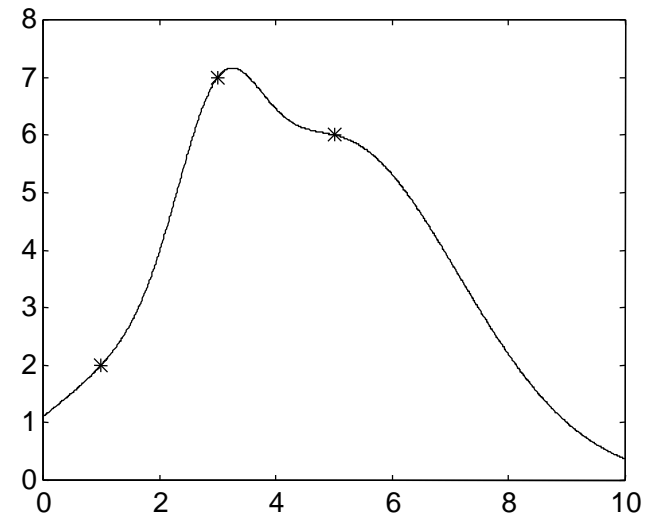


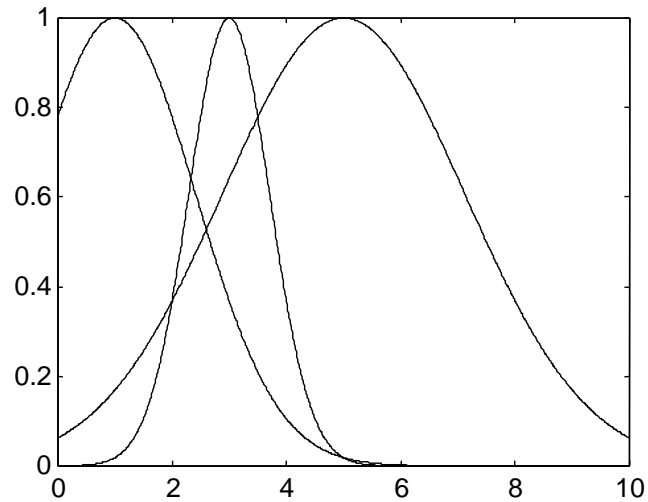
Caso 1: $m = N$

Pontos amostrados: (1,2); (3,7); (5,6)

$$\mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}; \quad \mathbf{r} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} 0.945 \\ 2.850 \\ 5.930 \end{bmatrix}$$

Obs: As funções de base radial têm centros nos valores de x e dispersões arbitrárias.



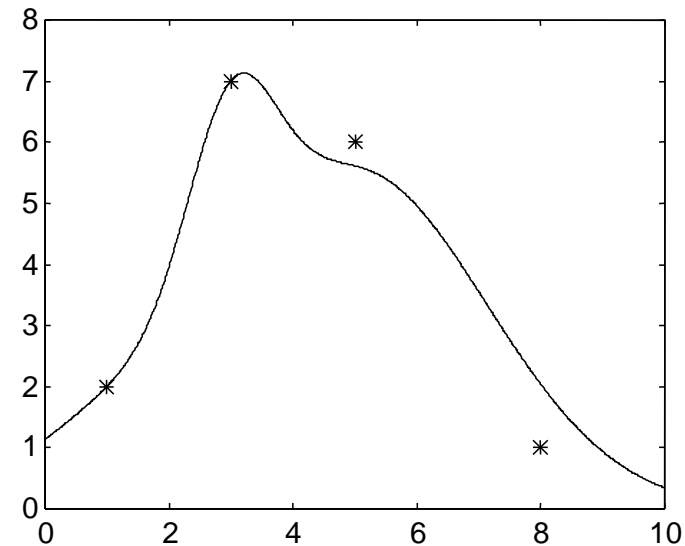
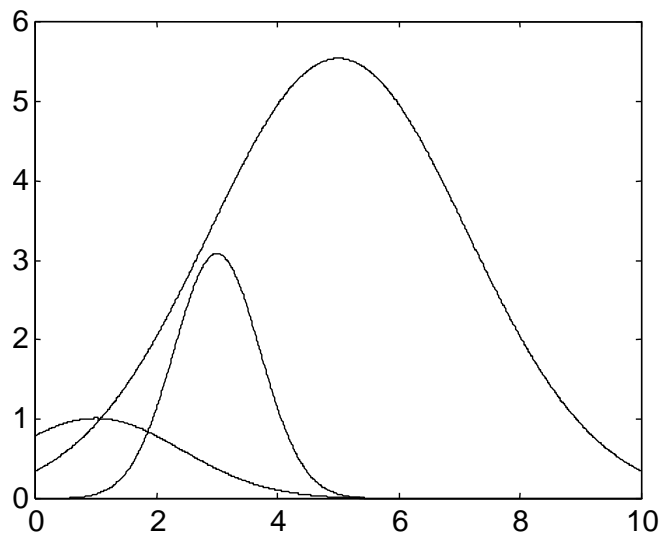


Caso 2: $m < N$

Pontos amostrados: (1,2); (3,7); (5,6); (8,1)

$$\mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}; \quad \mathbf{r} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} 1.012 \\ 3.084 \\ 5.538 \end{bmatrix}$$

Obs: As funções de base radial são as mesmas do Caso 1.



1.8 Determinação dos centros e dispersões

- No caso de algoritmos que se ocupam apenas com o ajuste dos pesos da camada de saída de uma rede RBF (modelos lineares nos parâmetros), é necessário estabelecer algum critério para fixação dos centros.
- Existem critérios para o caso de número variável de centros (redes construtivas, por exemplo), mas serão mencionados aqui apenas aqueles geralmente empregados para o caso de um número fixo e previamente especificado de centros.
- Existem basicamente 3 alternativas:
 1. Espalhar os centros uniformemente ao longo da região em que se encontram os dados;
 2. Escolher aleatoriamente, ou segundo algum critério específico, um subconjunto de padrões de entrada como centros;
 3. Auto-organizar os centros, de acordo com a distribuição dos dados de entrada.
Exemplo: Empregando *k-means*.
- Quanto às dispersões das funções de base radial, usualmente se adota uma única dispersão para todos os centros, na forma (HAYKIN, 2008):

$$\sigma = \frac{d_{\max}}{\sqrt{2m}}$$

onde m é o número de centros, e d_{\max} é a distância máxima entre os centros.

1.9 Aplicação das propostas de determinação de centros e dispersão

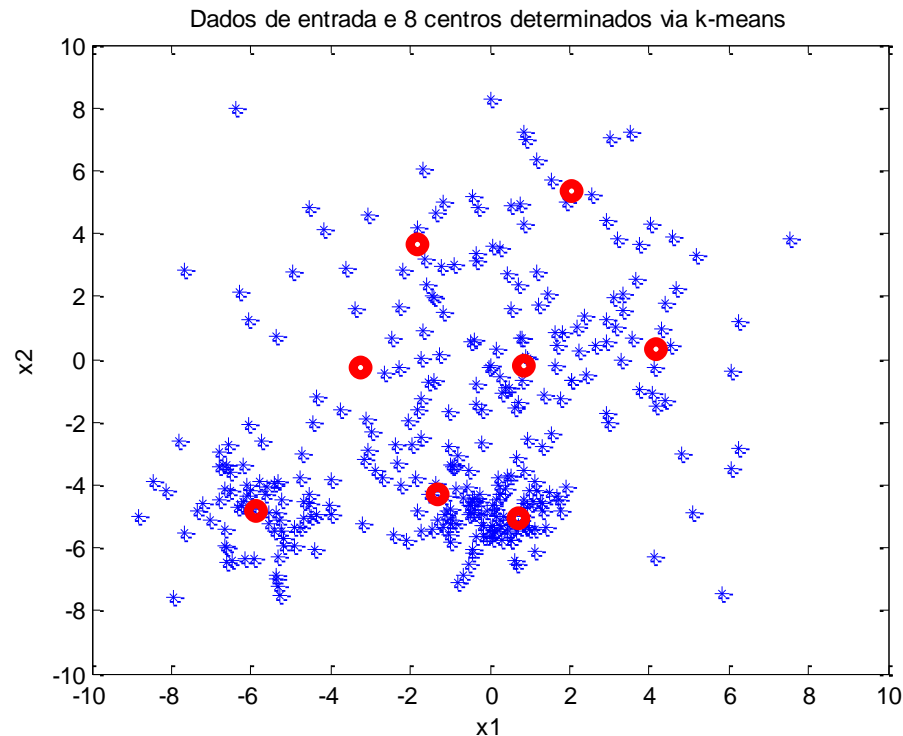


Figura 5 – Proposta de posicionamento dos centros das funções de base radial para uma rede neural RBF com 8 neurônios na camada intermediária, empregando o algoritmo *k-means*.

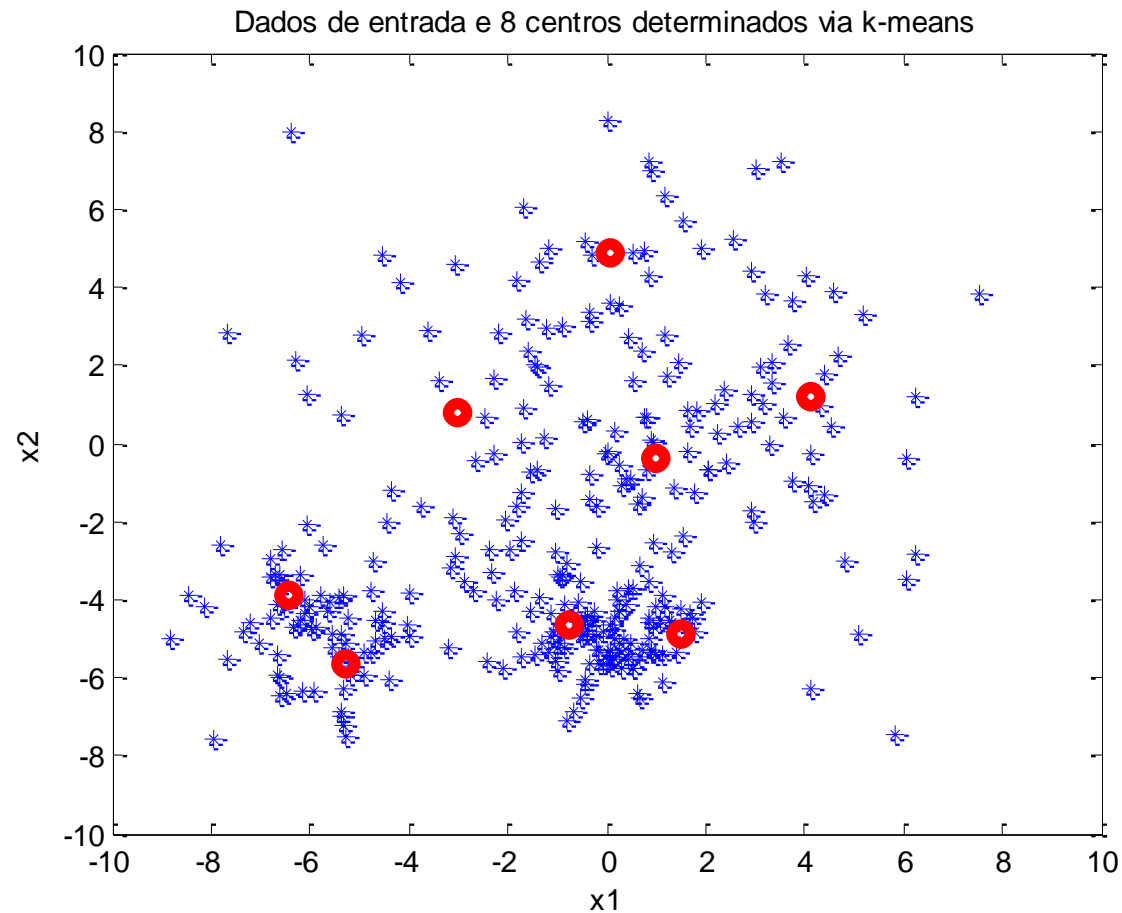


Figura 6 – Outra proposta de posicionamento dos centros para os mesmos dados, produzida por uma segunda execução do algoritmo *k-means*.

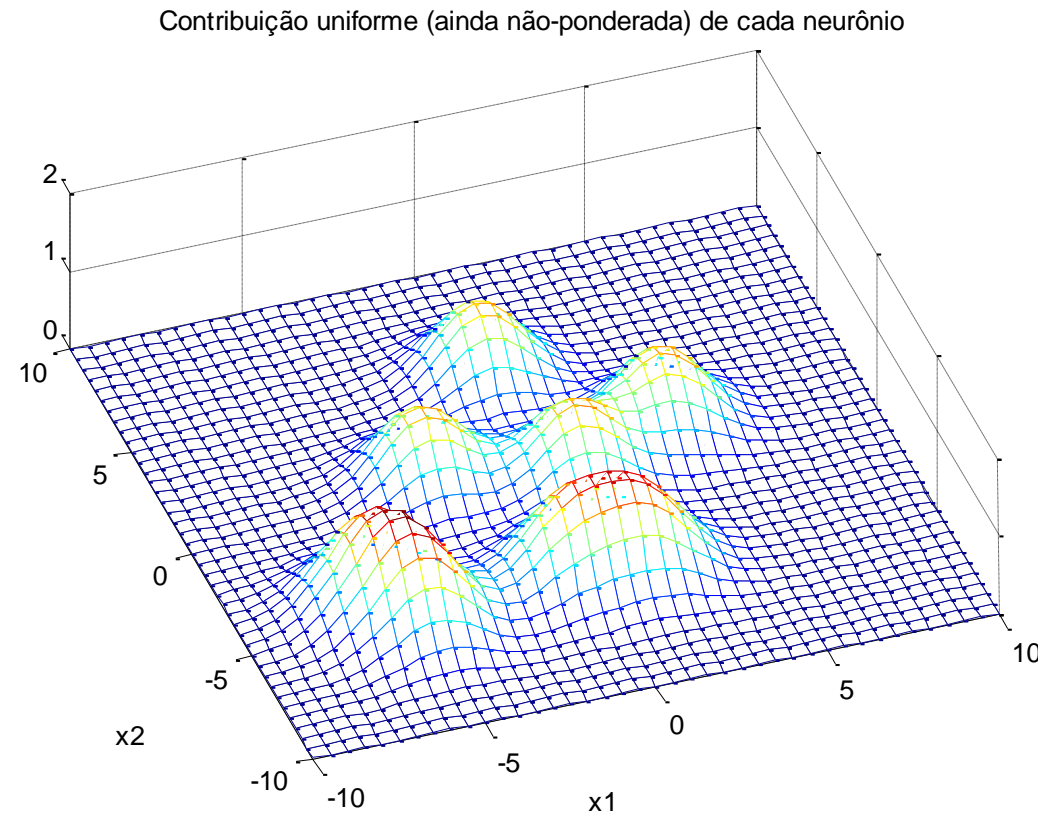


Figura 7 – Ativação dos neurônios da rede neural RBF com os centros da Figura 6, considerando todos os pesos de saída iguais a 1 e ausência de peso de bias. A dispersão é a mesma para todas as funções de ativação, dada pela fórmula da pg. 22.

- Com o critério de dispersão da pg. 22, evita-se que as funções de base radial sejam excessivamente pontiagudas, ou então com uma base demasiadamente extensa.

1.10 Referências para redes neurais RBF

- BROOMHEAD, D.S. & LOWE, D. “Multivariate functional interpolation and adaptive networks”, *Complex Systems*, 2: 321-355, 1988.
- HAYKIN, S. “Neural Networks and Learning Machines”, 3rd edition, Prentice Hall, 2008.
- PARK, J. & SANDBERG, I.W. “Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2): 246-257, 1991.

1.11 Bibliografia complementar para redes RBF

- BISHOP, C.M. “Improving the generalisation properties of radial basis function neural networks”, *Neural Networks*, 3(4): 579-588, 1991.
- BISHOP, C.M. “Neural Networks for Pattern Recognition”, Clarendon Press, 1995.
- CHEN, C.-L., CHEN, W.-C. & CHANG, F.-Y. “Hybrid learning algorithm for Gaussian potential function networks”, *IEE Proceedings D*, 140(6): 442-448, 1993.
- CHEN, S., CHNG, E.S. & ALKADHIMI, K. “Regularized Orthogonal Least Squares Algorithm for Constructing Radial Basis Function Networks”, *International Journal of Control*, 64(5): 829-837, 1996.
- CHEN, S., COWAN, C.F.N. & GRANT, P.M. “Orthogonal Least Squares Algorithm for Radial Basis Function Networks”, *IEEE Transactions on Neural Networks*, 2(2): 302-309, 1991.
- DE CASTRO, L.N. & VON ZUBEN, F.J. Automatic Determination of Radial Basis Functions: An Immunity-Based Approach. *International Journal of Neural Systems*, vol. 11, no. 6, pp. 523-535, 2001.
- FREEMAN, J.A.S. & SAAD, D. “Learning and Generalization in Radial Basis Function Networks”, *Neural Computation*, 7: 1000-1020, 1995.
- FRITZKE, B. “Fast learning with incremental RBF Networks”, *Neural Processing Letters*, 1(1): 2-5, 1994.
- GOMM, J.B. & YU, D.L. “Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training”, *IEEE Transactions on Neural Networks*, 11(2):306-314, 2000.
- HWANG, Y.-S. & BANG, S.-Y. “An Efficient Method to Construct a Radial Basis Function Neural Network Classifier”, *Neural Networks*, 10(8): 1495-1503, 1997.

- KARAYIANNIS, N.B. “Gradient Descent Learning of Radial Basis Neural Networks”, *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1815-1820, 1997.
- KARAYIANNIS, N.B. & MI, G.W. “Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques”, *IEEE Transactions on Neural Networks*, 8(6): 1492-1506, 1997.
- KUBAT, M. “Decision trees can initialize radial-basis function networks”, *IEEE Transactions on Neural Networks*, 9(5): 813-821, 1998.
- LIPPMANN, R.P. “Pattern Classification Using Neural Networks”, *IEEE Communications Magazine*, November, pp. 47-63, 1989.
- MICCHELLI, C.A. “Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions”, *Constructive Approximation*, 2: 11-22, 1986.
- MOODY, J. & DARKEN, C. “Fast Learning in Networks of Locally-Tuned Processing Units”, *Neural Computation*, 1: 281-294, 1989.
- MULGREW, B. “Applying Radial Basis Functions”, *IEEE Signal Processing Magazine*, pp. 50-66, March 1996.
- ORR, M.J.L. “Introduction to Radial Basis Function Networks”, *Technical Report*, Centre for Cognitive Science, University of Edinburgh, Scotland, 1996. (<http://www.anc.ed.ac.uk/~mjo/papers/intro.ps>)
- ORR, M.J.L. “Recent Advances in Radial Basis Function Networks”, *Technical Report*, Institute for Adaptive and Neural Computation, University of Edinburgh, Scotland, 1999. (<http://www.anc.ed.ac.uk/~mjo/papers/recad.ps>)
- ORR, M.J.L. “Regularisation in the Selection of Radial Basis Function Centres”, *Neural Computation*, 7(3): 606-623, 1995.
- POGGIO, T. & GIROSI, F. “Networks for Approximation and Learning”, *Proceedings of the IEEE*, 78(9): 1481-1497, 1990.
- SUTANTO, E.L., MASON, J.D. & WARWICK, K. “Mean-tracking clustering algorithm for radial basis function centre selection. *International Journal of Control*, 67(6): 961-977, 1997.
- WANG, Z. & ZHU, T. “An Efficient Learning Algorithm for Improving Generalization Performance of Radial Basis Function Neural Networks”, *Neural Networks*, 13(4-5): 545-553, 2000.
- WETTSCHERECK, D. & DIETTERICH, T. “Improving the Performance of Radial Basis Function Networks by Learning Center Locations”, *Advances in Neural Information Processing Systems*, 4:1133-1140, 1992.
- WHITEHEAD, B.A. & CHOATE, T.D. “Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction”, *IEEE Transactions on Neural Networks*, 7(4): 869-880, 1996.
- WHITEHEAD, B.A. & CHOATE, T.D. “Evolving Space-Filling Curves to Distribute Radial Basis Functions Over an Input Space”, *IEEE Transactions on Neural Networks*, 5(1): 15-23, 1994.
- YINGWEI, L., SUNDARARAJAN, N. & SARATCHANDRAN, P. “A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function Neural Networks”, *Neural Computation*, pp. 461-478, 1996.

2. Máquinas de aprendizado extremo (ELMs)

- Todas as propostas de redes neurais não-recorrentes (*feedforward*) já apresentadas no curso, como o perceptron de múltiplas camadas (MLP) e a rede neural com funções de ativação de base radial (RBF), produzem a sua saída (podendo ser múltiplas saídas) como uma combinação linear das ativações dos neurônios da camada anterior.
- Tomando uma única camada intermediária, pode-se afirmar, portanto, que redes neurais MLP e RBF sintetizam mapeamentos multidimensionais de entrada-saída por meio de uma composição aditiva de funções-base, na forma:

$$\hat{s}_{kl} = \sum_{j=1}^n w_{kj} f(\mathbf{v}_j, b_j, \mathbf{x}_l) + w_{k0}$$

onde

- \hat{s}_{kl} é a k -ésima saída da rede neural para o l -ésimo padrão de entrada \mathbf{x}_l ;
- $f(\mathbf{v}_j, b_j, \bullet)$ é a j -ésima função da base de funções-base.

- No caso da rede neural MLP, as funções-base são funções de expansão ortogonal (*ridge functions*), enquanto que, no caso da rede neural RBF, as funções-base têm um comportamento radial em relação a um centro de ativação máxima.
- Nos dois casos, como em outros casos de composição aditiva de funções-base, há demonstração teórica da capacidade de aproximação universal. A capacidade de aproximação universal é uma **propriedade existencial**. Ela afirma que existe um número n finito de neurônios e uma certa configuração de pesos sinápticos que permitem obter um erro de aproximação arbitrariamente baixo para os dados de treinamento, supondo que se considera uma região compacta do espaço de entrada e que o mapeamento original, que é amostrado para produzir os dados de treinamento, é contínuo.
- É intuitivo concluir, também, que quanto maior o número n de neurônios na camada intermediária, maior é a flexibilidade do modelo matemático resultante, ou seja, maiores são as “possibilidades de contorção” do mapeamento a ser sintetizado.

- Por outro lado, é sabido também que há o risco de sobre-ajuste aos dados, produzindo modelos que generalizam mal frente a novos dados de entrada-saída. A máxima capacidade de generalização está associada a modelos otimamente regularizados, ou seja, que se contorcem na medida certa, de acordo com as demandas de cada aplicação.
- Com isso, uma definição adequada do número de neurônios e dos pesos sinápticos é fundamental para garantir uma boa capacidade de generalização.
- Um resultado fundamental da literatura, restrito a problemas de classificação de padrões, foi apresentado por BARTLETT (1997; 1998). Nesses trabalhos, como o próprio título indica, conclui-se que controlar a norma dos pesos sinápticos é mais relevante para a capacidade de generalização do que controlar o tamanho da rede neural, ou seja, o número n de neurônios na camada intermediária.
- De fato, pode-se introduzir o conceito de \langle número efetivo de neurônios na camada intermediária \rangle , o qual é determinado pela configuração dos pesos da camada de saída da rede neural.

- As máquinas de aprendizado extremo exploram este resultado “de forma extrema”, ou seja, jogam toda a responsabilidade por garantir uma boa capacidade de generalização aos pesos da camada de saída, permitindo que os pesos da camada intermediária, responsáveis por definir as funções-base, sejam definidos de modo aleatório.
- Por serem definidos de modo aleatório, portanto desvinculados das demandas da aplicação, deve-se considerar um valor elevado para n , podendo inclusive ultrapassar o valor de N , que representa o número de amostras para treinamento.
- Por mais que pareça estranho trabalhar com valores de n elevados e até maiores que N , as máquinas de aprendizado extremo se sustentam em três argumentos muito poderosos:
 - ✓ O problema de treinamento passa a ser linear nos parâmetros ajustáveis, o que representa uma enorme economia de recursos computacionais para se realizar o treinamento supervisionado;

- ✓ A capacidade de generalização pode ser maximizada controlando-se a norma dos pesos na camada de saída, não dependendo de forma significativa do número n de neurônios na camada intermediária;
- ✓ Há recursos computacionais disponíveis para implementar redes neurais sobredimensionadas.
- E já que as funções-base podem ser definidas aleatoriamente, então não há razão também para que elas tenham formas sigmoidais ou tenham base radial. Logo, o elenco de funções-base pode ser também arbitrário, embora as demonstrações de capacidade de aproximação universal para ELMs restrinjam ainda as alternativas de funções-base.
- Por outro lado, são incluídas funções trigonométricas e até a função sinal.

2.1 Exemplos de máquinas de aprendizado extremo

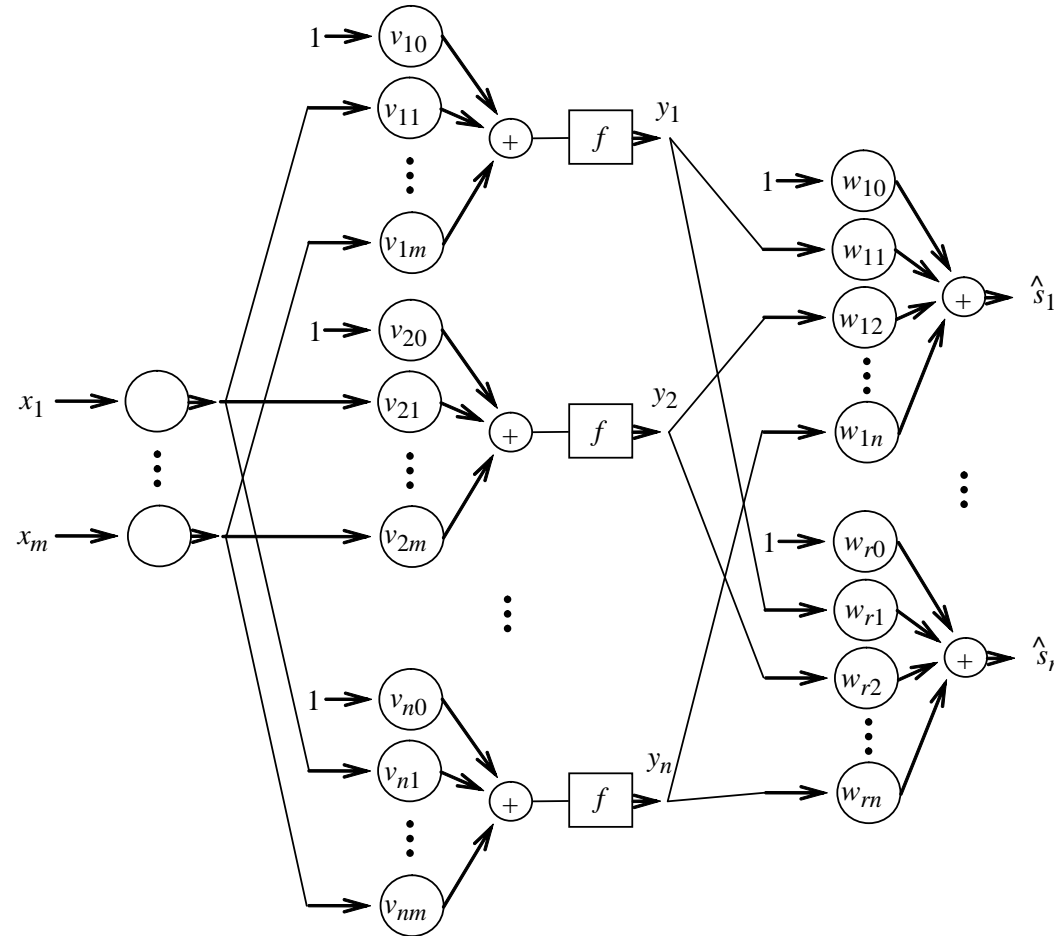


Figura 8 – Rede neural perceptron com uma camada intermediária

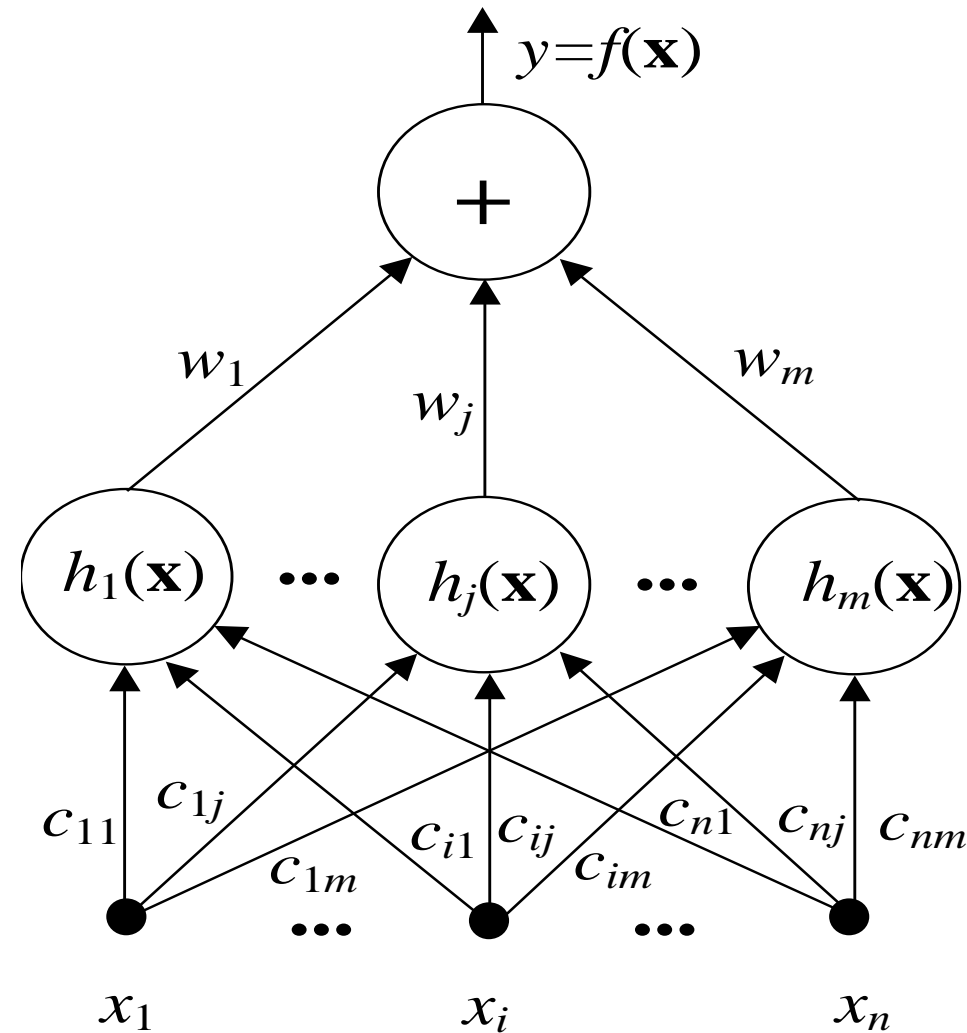


Figura 9 – Rede neural com funções de ativação de base radial. Observação: Geralmente é também considerada uma entrada constante para o neurônio de saída.

2.2 Treinamento das ELMs

- Treinar uma máquina de aprendizado extremo é equivalente a resolver o seguinte problema de otimização para cada uma das saídas da rede neural:

$$\mathbf{w}_k^* = \arg \min_{\mathbf{w}_k \in \mathcal{R}^{n+1}} J(\mathbf{w}_k) + C_k \times \|\mathbf{w}_k\|^2$$

onde

1. k é o índice da saída;
2. n é o número de neurônios na camada intermediária;
3. $\|\cdot\|^2$ é a norma euclidiana;
4. C_k é um coeficiente de ponderação, a ser determinado, por exemplo, por métodos de busca unidimensional;

$$5. J(\mathbf{w}_k) = \frac{1}{2} \sum_{l=1}^N \left[\sum_{j=1}^n w_{kj} f(\mathbf{v}_j, b_j, \mathbf{x}_l) + w_{k0} - s_{kl} \right]^2 ;$$

6. N é o número de amostras disponíveis para treinamento.

2.3 Como encontrar os pesos sinápticos

- Uma vez fornecido o coeficiente de ponderação C_k , para a k -ésima saída da rede neural, o vetor de pesos sinápticos é obtido como segue:

1. Monta-se a matriz H_{inicial} de dimensão $N \times n$, com as ativações de todos os neurônios para todos os padrões de entrada, produzindo:

$$H_{\text{inicial}} = \begin{bmatrix} f(\mathbf{v}_1, b_1, \mathbf{x}_1) & f(\mathbf{v}_2, b_2, \mathbf{x}_1) & \cdots & f(\mathbf{v}_n, b_n, \mathbf{x}_1) \\ f(\mathbf{v}_1, b_1, \mathbf{x}_2) & \ddots & & \vdots \\ \vdots & & & \\ f(\mathbf{v}_1, b_1, \mathbf{x}_N) & \cdots & & f(\mathbf{v}_n, b_n, \mathbf{x}_N) \end{bmatrix}$$

2. Acrescenta-se uma coluna de 1's à matriz H_{inicial} , produzindo a matriz H :

$$H = \begin{bmatrix} f(\mathbf{v}_1, b_1, \mathbf{x}_1) & f(\mathbf{v}_2, b_2, \mathbf{x}_1) & \cdots & f(\mathbf{v}_n, b_n, \mathbf{x}_1) & 1 \\ f(\mathbf{v}_1, b_1, \mathbf{x}_2) & \ddots & & \vdots & 1 \\ \vdots & & & \vdots & \vdots \\ f(\mathbf{v}_1, b_1, \mathbf{x}_N) & \cdots & & f(\mathbf{v}_n, b_n, \mathbf{x}_N) & 1 \end{bmatrix}$$

3. Monta-se o vetor \mathbf{s}_k , contendo todos os padrões de saída, na forma:

$$\mathbf{s}_k = [s_{k1} \quad s_{k2} \quad \cdots \quad s_{kN}]^T$$

4. Considerando que a matriz H tenha posto completo, o vetor w_k é obtido como segue:

4.1. Se $(n+1) \leq N$, $\mathbf{w}_k = (H^T H + C_k I)^{-1} H^T \mathbf{s}_k$;

4.2. Se $(n+1) > N$, $\mathbf{w}_k = H^T (H H^T + C_k I)^{-1} \mathbf{s}_k$.

2.4 Como encontrar o coeficiente de ponderação

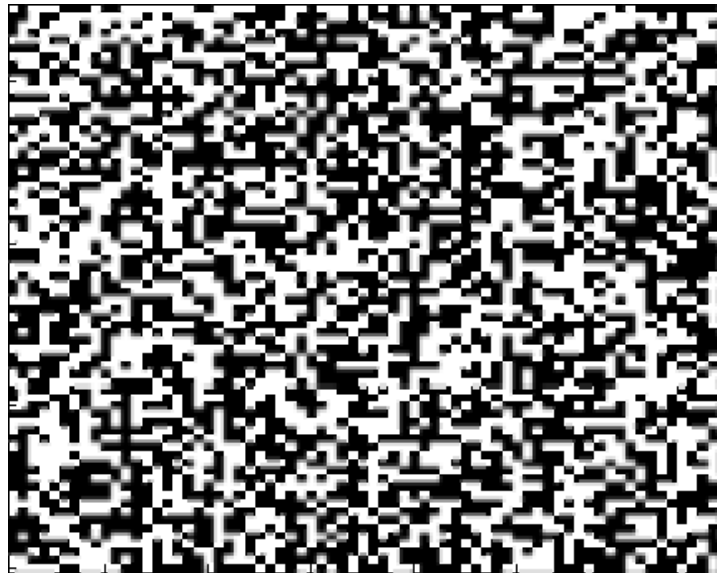
- A maximização da capacidade de generalização requer a definição de um valor adequado para o coeficiente de ponderação C_k , associado à saída k .
- Sugere-se aqui o uso de uma busca unidimensional empregando um conjunto de validação. O valor ótimo de C_k é aquele que minimiza o erro junto ao conjunto de validação.

2.5 Referências bibliográficas para ELMs

- BARTLETT, P.L. For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems*, volume 9, pp. 134-140, 1997.
- BARTLETT, P.L. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525-536, 1998.
- HUANG, G.-B., CHEN, L., SIEW, C.-K. Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes. *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879-892, 2006.
- HUANG, G.-B., WANG, D.H., LAN, Y. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, vol. 2, pp. 107-122, 2011.
- HUANG, G.-B., ZHOU, H., DING, X., ZHANG, R. Extreme Learning Machines for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012.
- HUANG, G.-B., ZHU, Q.-Y., SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. *Proceedings of the International Joint Conference on Neural Networks (IJCNN'2004)*, vol. 2, pp. 985-990, 2004.
- HUANG, G.-B., ZHU, Q.-Y., SIEW, C.-K. Extreme learning machine: theory and applications. *Neurocomputing*, vol. 70, pp. 489-501, 2006.

3. O Jogo da Vida

- Concebido por John Horton Conway, matemático britânico, em 1970. Procure assistir aos vídeos em: $\langle \text{https://www.youtube.com/watch?v=C2vgICfQawE} \rangle$ e em $\langle \text{https://www.youtube.com/watch?v=xP5-ileKXE8} \rangle$.



- Célula morta e com três vizinhos vivos ressuscita;
- Célula viva e com dois ou três vizinhos vivos permanece viva;
- Em todos os outros casos, a célula morre ou permanece morta.

4. Exemplos de auto-organização na natureza

Conchas



Processos Físicos

- ✓ Dunas
- ✓ Reagentes químicos
- ✓ Rachaduras na lama
- ✓ Células de convecção de Bénard
- ✓ Rachadura em tinta
- ✓ Rugas em verniz



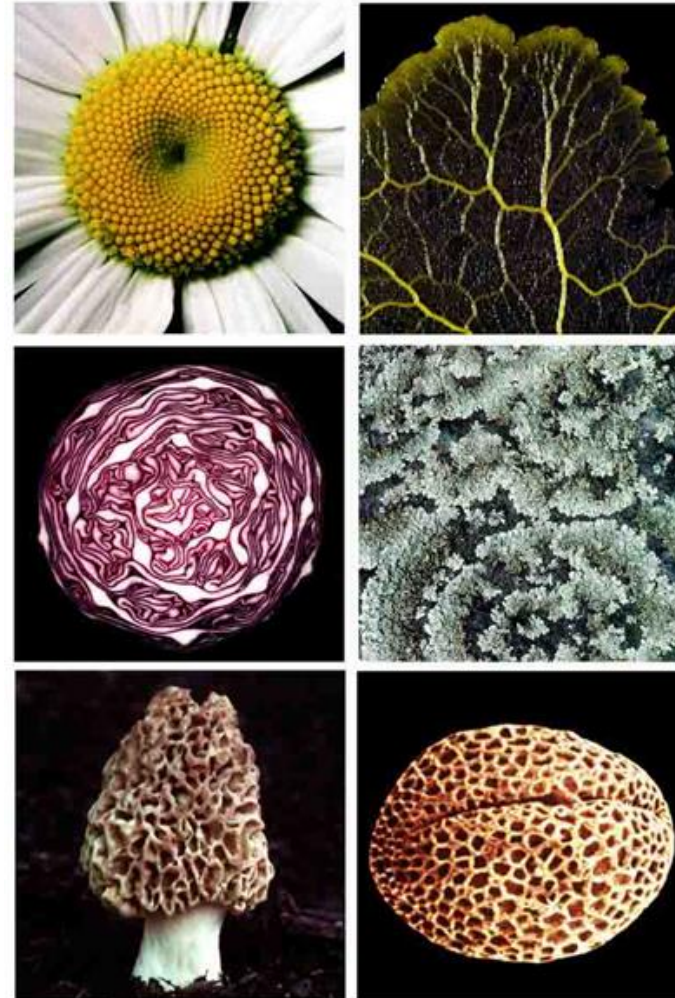
Em animais

- ✓ Listras da zebra
- ✓ Manchas da girafa
- ✓ Listras do tigre
- ✓ Anfíbios
- ✓ Lagartos



Em plantas

- ✓ Margarida
- ✓ *Slime mold*
- ✓ Repolho vermelho
- ✓ Líquen
- ✓ Cogumelo
- ✓ Grão de pólen



Em insetos

- ✓ Asas de borboletas
- ✓ Exoesqueletos de besouros



5. Treinamento não-supervisionado

- Como aprender a representar padrões de entrada de modo a refletir a estrutura estatística de toda a coleção de dados de entrada? Que aspectos da entrada devem ser reproduzidos na saída?
- Em contraposição ao treinamento supervisionado, não há aqui nenhuma saída desejada explícita ou avaliação externa da saída produzida para cada dado de entrada.
- O treinamento não-supervisionado é predominante no cérebro humano. É sabido que as propriedades estruturais e fisiológicas das sinapses no córtex cerebral são influenciadas pelos padrões de atividade que ocorrem nos neurônios sensoriais. No entanto, em essência, nenhuma informação prévia acerca do conteúdo ou significado do fenômeno sensorial está disponível.
- Sendo assim, a implementação de modelos computacionais para ajuste de pesos sinápticos via treinamento não-supervisionado deve recorrer apenas aos dados de

entrada, tomados como amostras independentes de uma distribuição de probabilidade desconhecida.

6. Mapas Auto-Organizáveis de Kohonen

- Um mapa de Kohonen é um arranjo de neurônios, geralmente restrito a espaços de dimensão 1 ou 2, que procura estabelecer e preservar noções de vizinhança (preservação topológica).
- Se estes mapas apresentarem propriedades de auto-organização, então eles podem ser aplicados a problemas de clusterização e ordenação espacial de dados.
- Neste caso, vai existir um mapeamento do espaço original (em que os dados se encontram) para o espaço em que está definido o arranjo de neurônios.
- Como geralmente o arranjo de neurônios ocorre em espaços de dimensão reduzida (1 ou 2), vai existir uma redução de dimensão sempre que o espaço original (em que os dados se encontram) apresentar uma dimensão mais elevada.

6.1 Arranjo unidimensional

- Um mapa de Kohonen unidimensional é uma sequência ordenada de neurônios lineares, onde o número de pesos de cada neurônio é igual ao número de entradas.
- Há uma relação de vizinhança entre os neurônios (no espaço unidimensional vinculado ao arranjo), mas há também uma relação entre os pesos dos neurônios no espaço de dimensão igual ao número de entradas. Para entender a funcionalidade dos mapas de Kohonen, é necessário considerar ambas as relações.

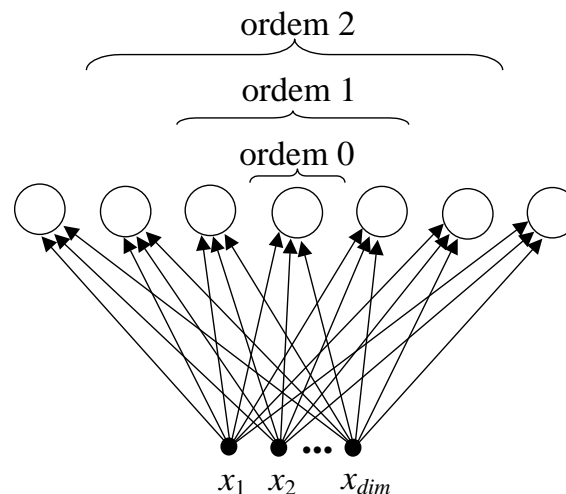


Figura 10 – Rede de Kohonen em arranjo unidimensional: ênfase na vizinhança

6.2 Arranjo bidimensional

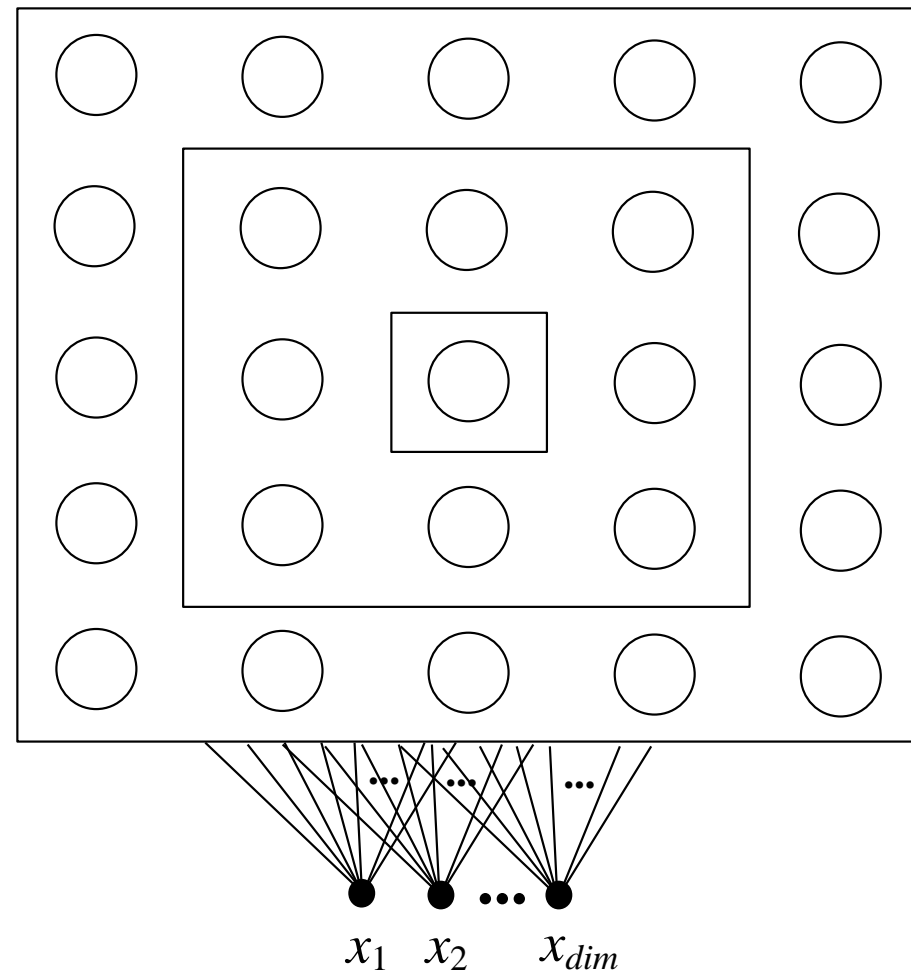


Figura 11 – Rede de Kohonen em arranjo bidimensional: ênfase na vizinhança

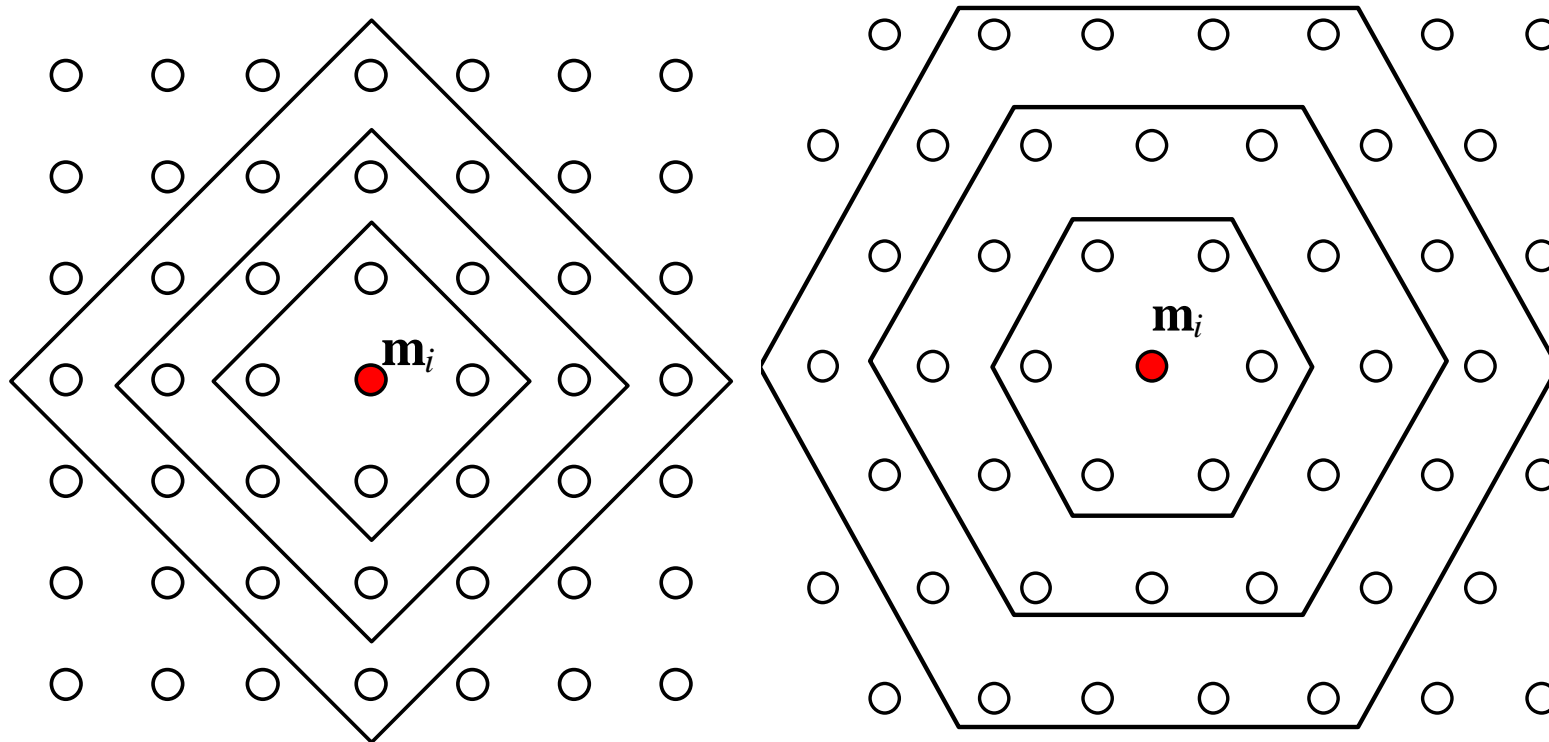


Figura 12 – Outras configurações de mapas e de vizinhança (figuras extraídas de ZUCHINI, 2003)

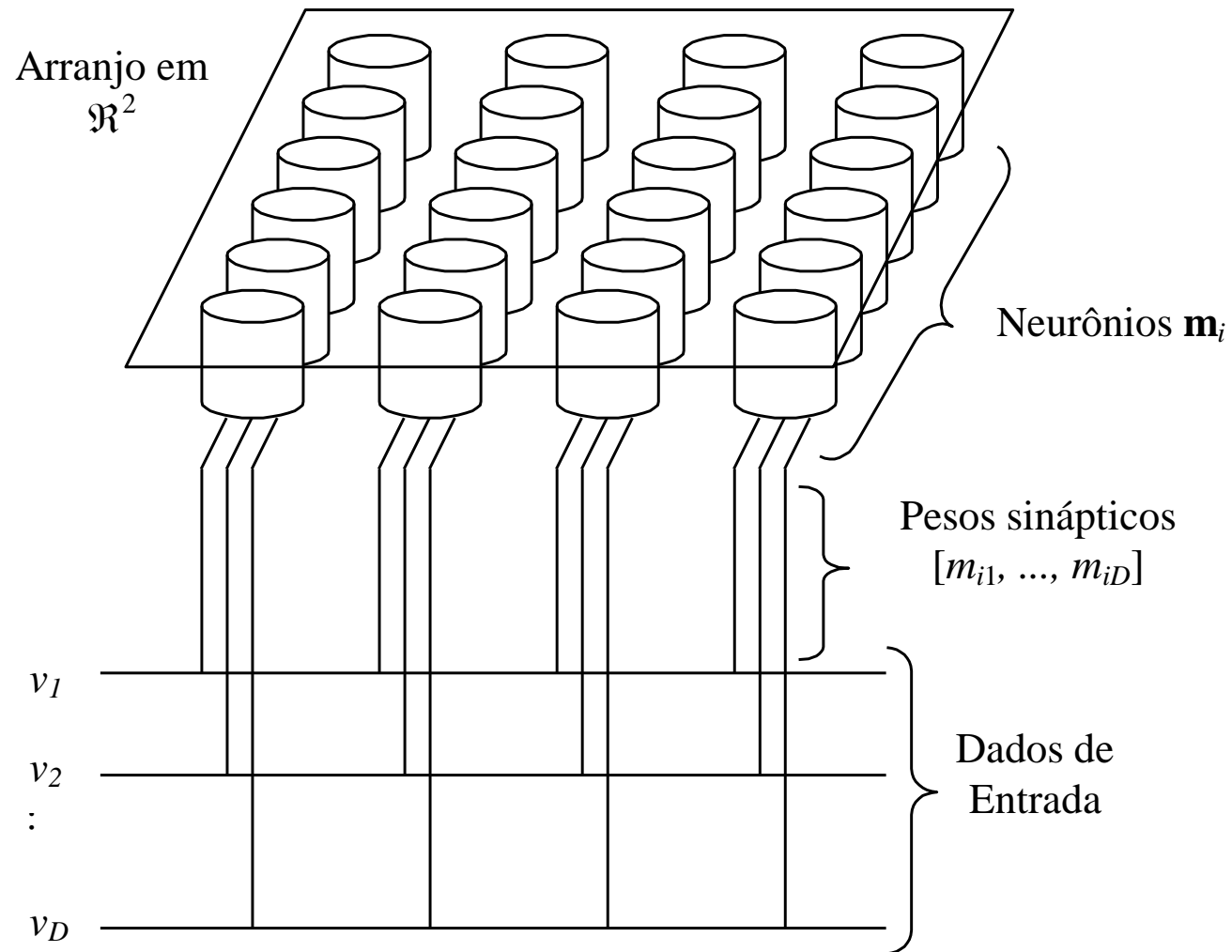


Figura 13 – Outra perspectiva para arranjo 2D (figura extraída de ZUCHINI, 2003)

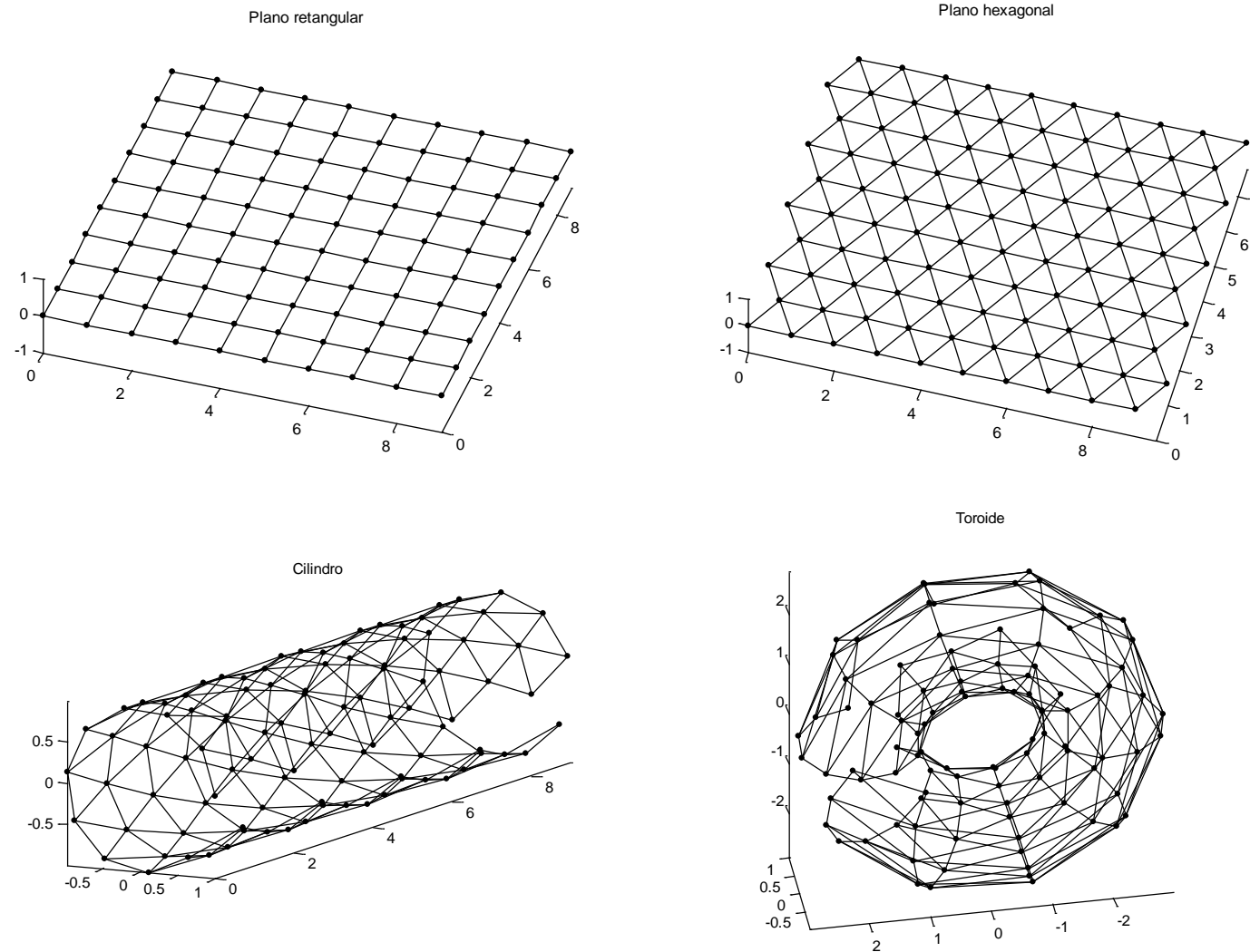
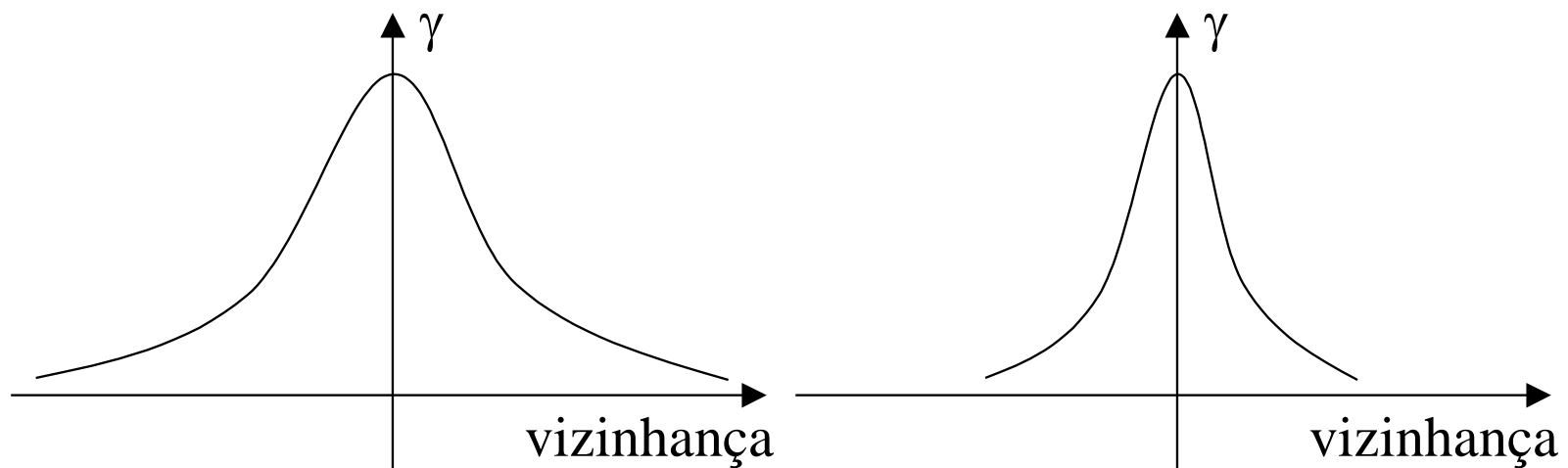
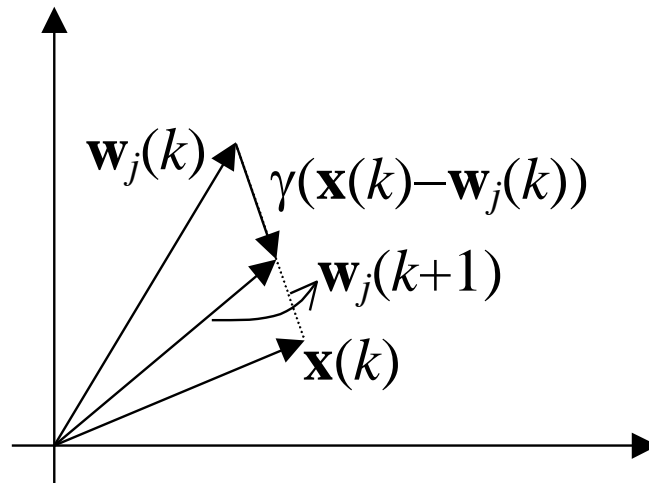


Figura 14 – Arranjos com e sem vizinhança nos extremos (figuras extraídas de ZUCHINI, 2003)

6.3 Algoritmo de ajuste dos pesos

```
while <condição de parada> é falso,  
    Ordene aleatoriamente os  $N$  padrões de entrada;  
    for  $i=1$  até  $N$ ,  
         $j = \arg \min_j \|\mathbf{x}_i - \mathbf{w}_j\|$   
         $\forall J \in \text{Viz}(j)$  do:  
             $\mathbf{w}_J = \mathbf{w}_J + \gamma(\text{dist}(j, J))(\mathbf{x}_i - \mathbf{w}_J);$   
        end do  
    end for  
    Atualize a taxa de aprendizado  $\gamma$ ;  
    Atualize a vizinhança;  
    Avalie a condição de parada;  
end while
```



6.4 Um passo de ajuste num arranjo unidimensional

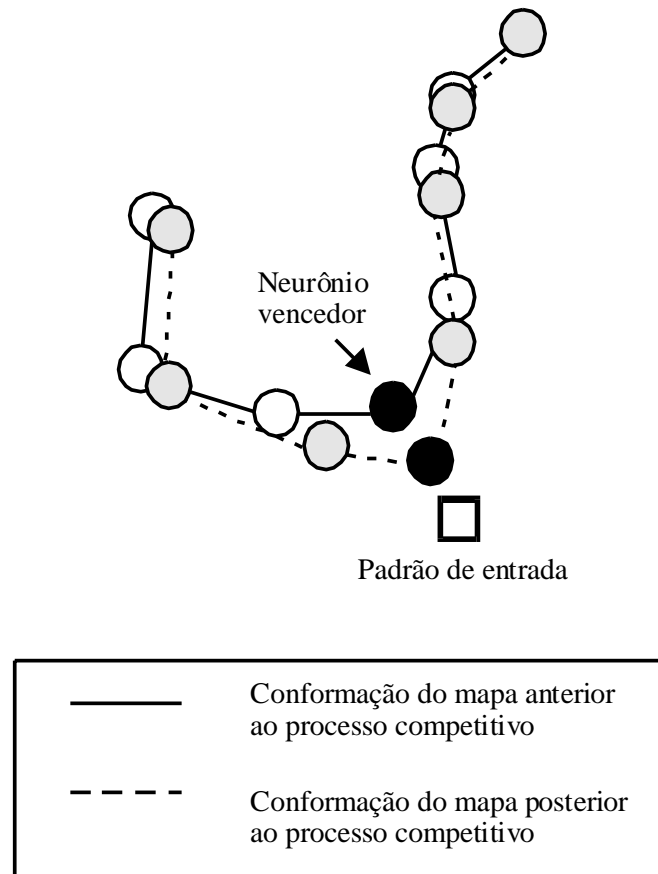


Figura 15 – Ajuste do neurônio vencedor e de seus vizinhos mais próximos

6.5 Ajuste de pesos com restrição de vizinhança

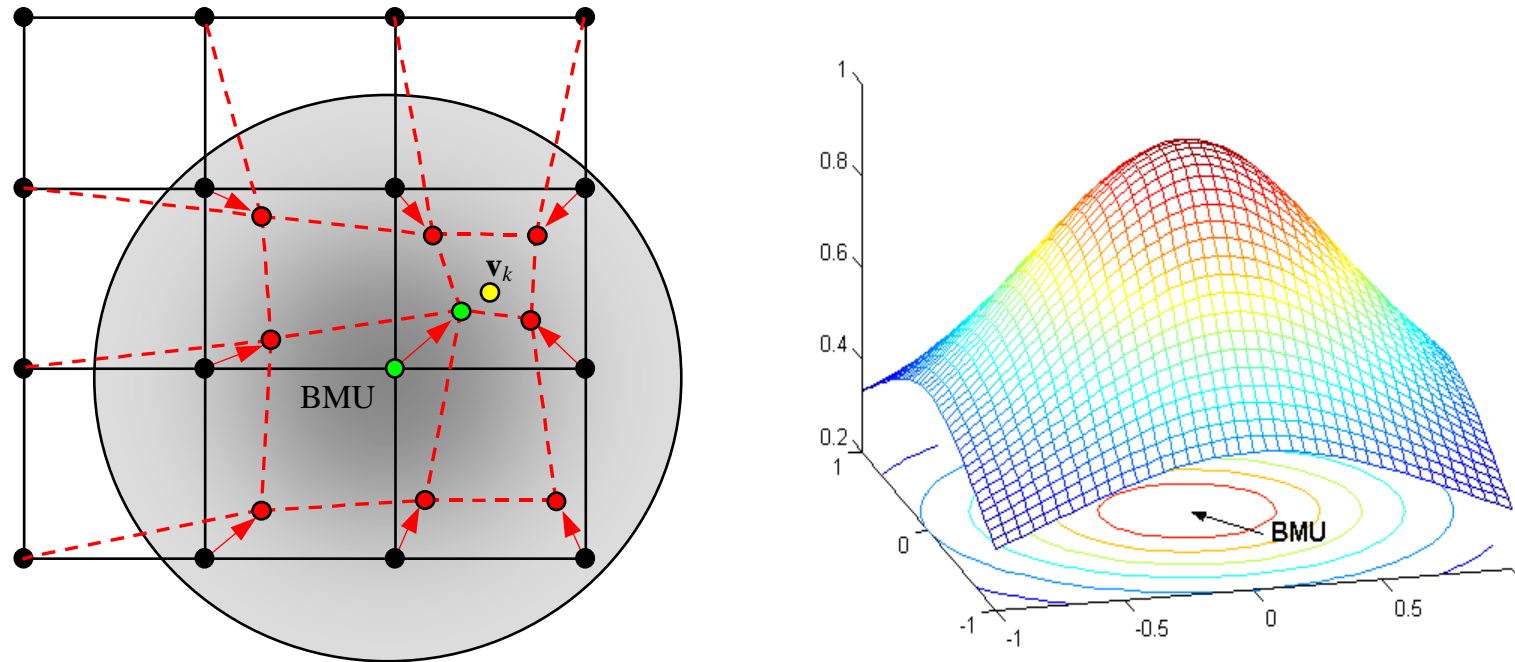


Figura 16 – BMU (*Best Matching Unit*) e seus vizinhos (figuras extraídas de ZUCHINI, 2003)

- O neurônio que venceu para uma dada amostra é o que sofre o maior ajuste. No entanto, dentro de uma vizinhança, todos os neurônios vizinhos também sofrerão um ajuste de pesos, embora de menor intensidade.

6.6 Discriminação dos agrupamentos

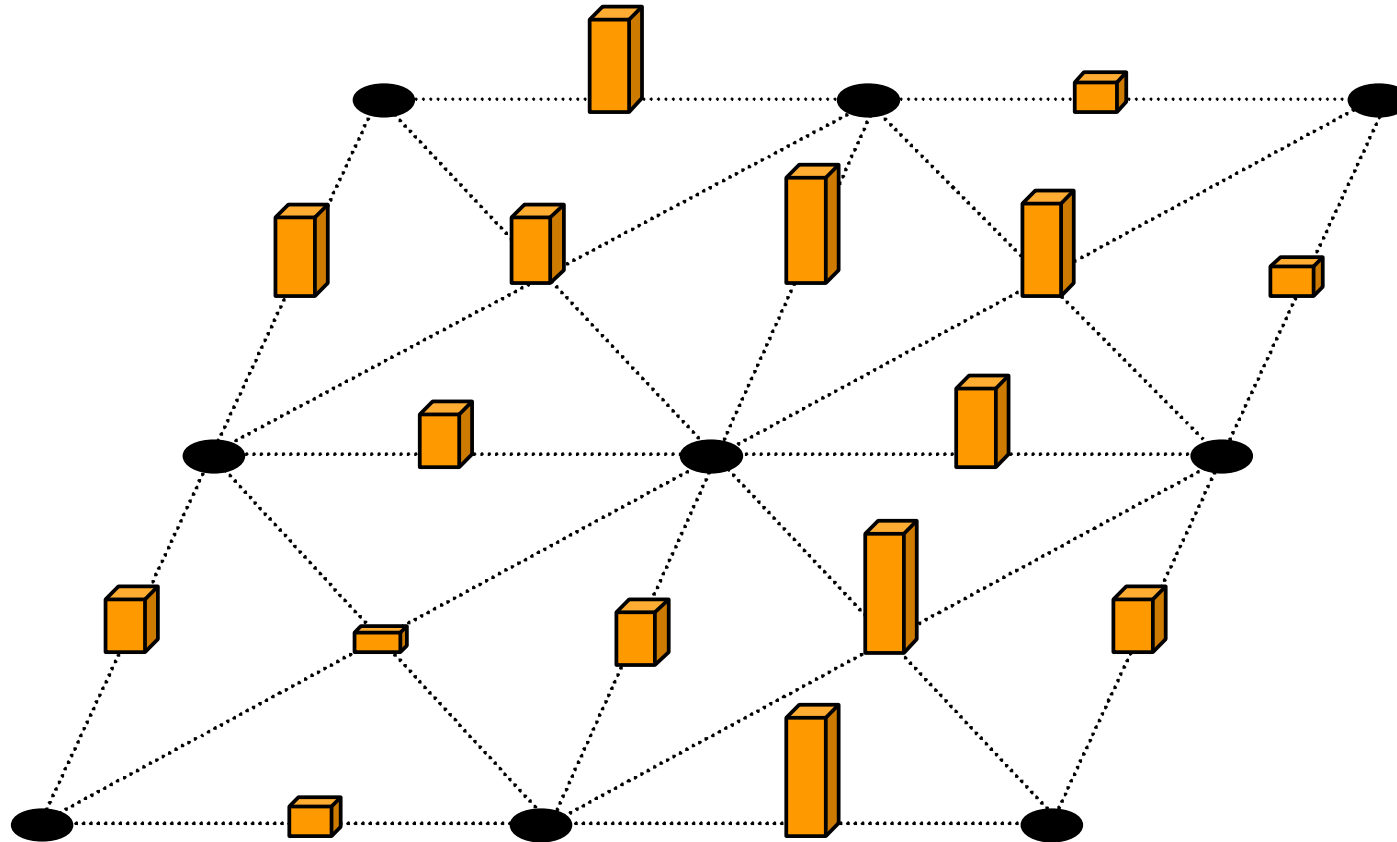


Figura 17 – Exemplo de matriz-U para arranjo retangular (figura extraída de ZUCHINI, 2003)

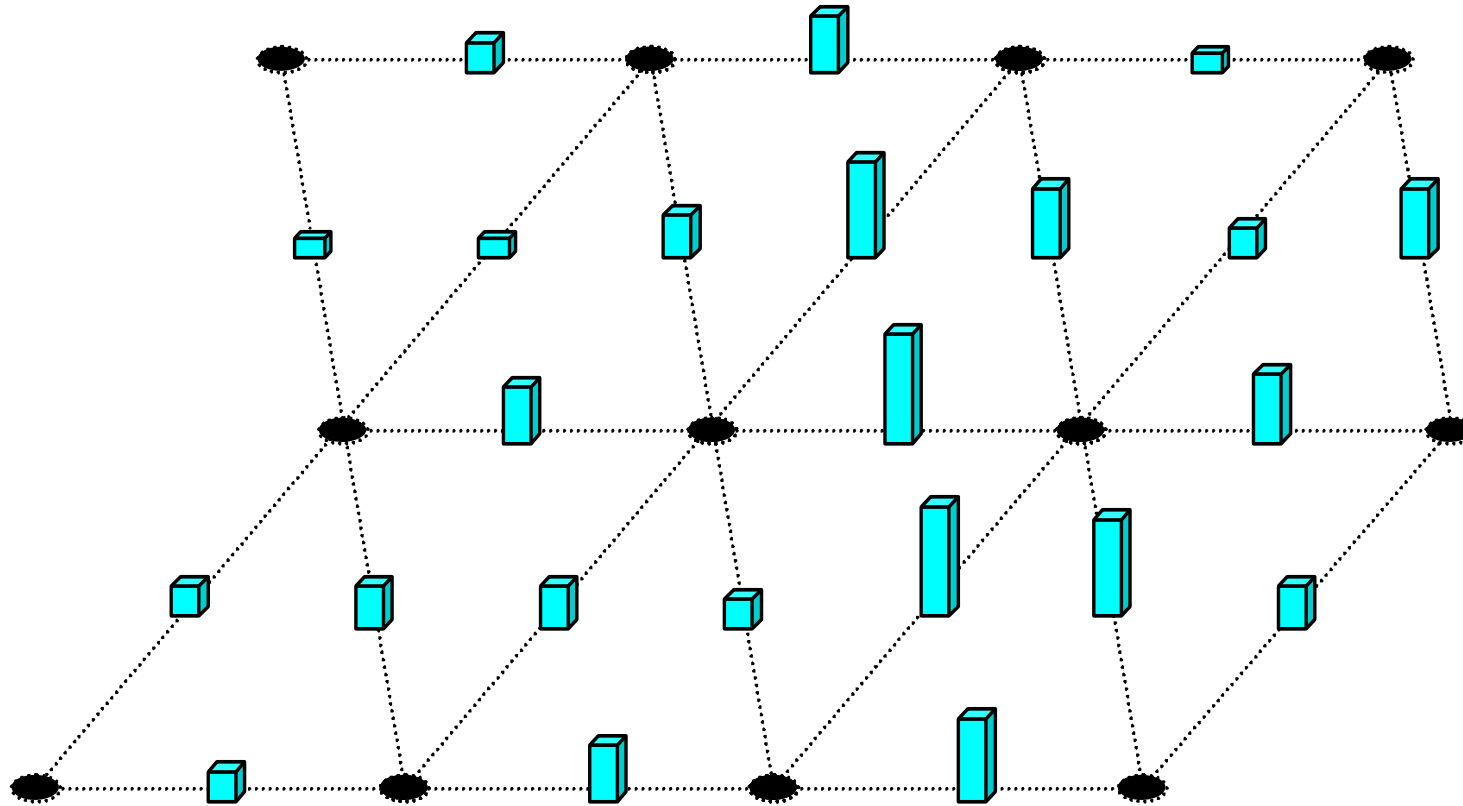


Figura 18 – Exemplo de matriz-U para arranjo hexagonal (figura extraída de ZUCHINI, 2003)

6.7 Ferramentas de visualização e discriminação

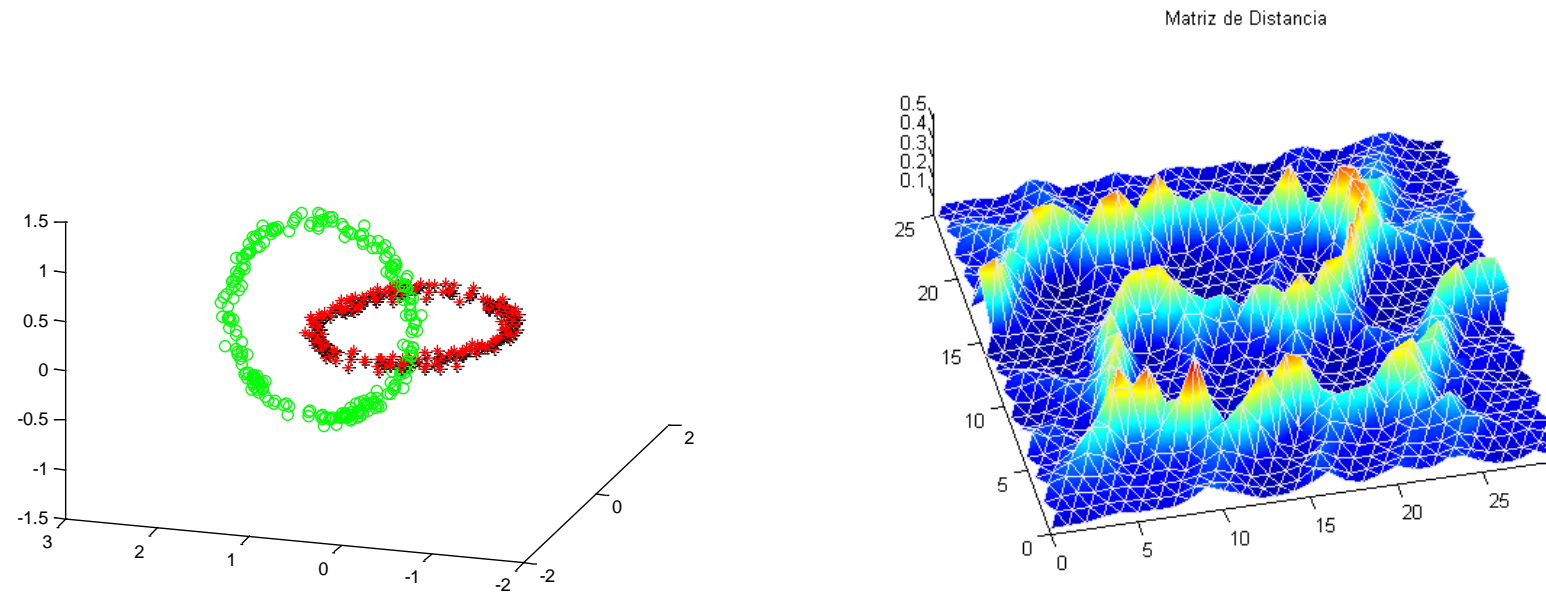


Figura 19 – Matriz-U para grid hexagonal (figuras extraídas de ZUCHINI, 2003)

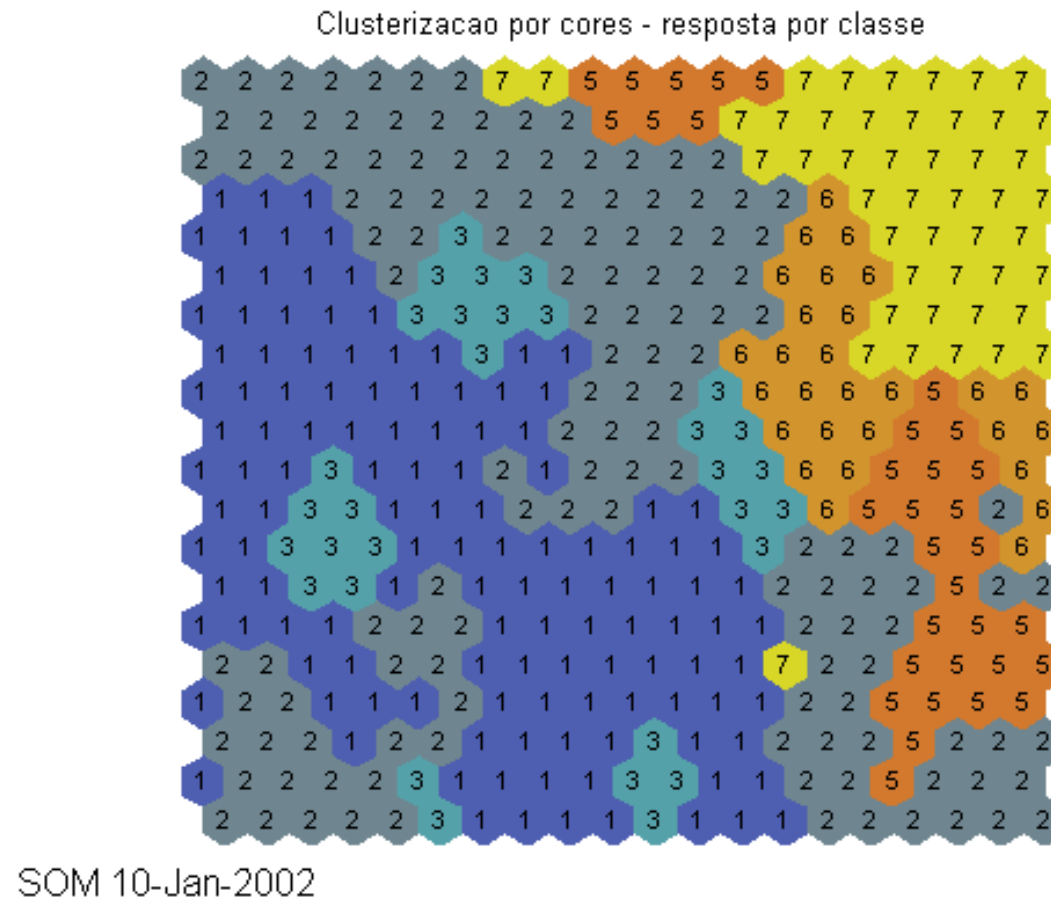
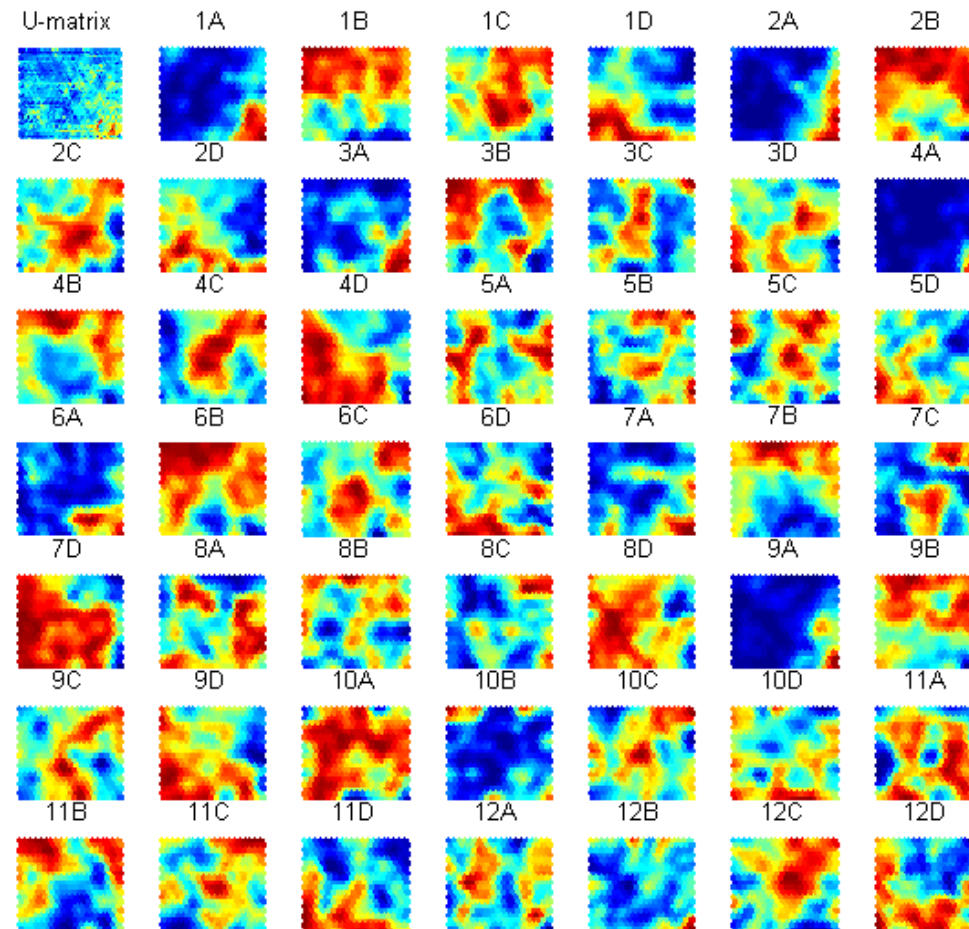


Figura 20 – Interpretação do mapa após auto-organização (figura extraída de ZUCHINI, 2003)



SOM 04-Apr-2002

Figura 21 – Busca por correlações (matriz-U por atributo de entrada do mapa) após a auto-organização (figura extraída de ZUCHINI, 2003) (VESANTO & AHOLA, 1999)

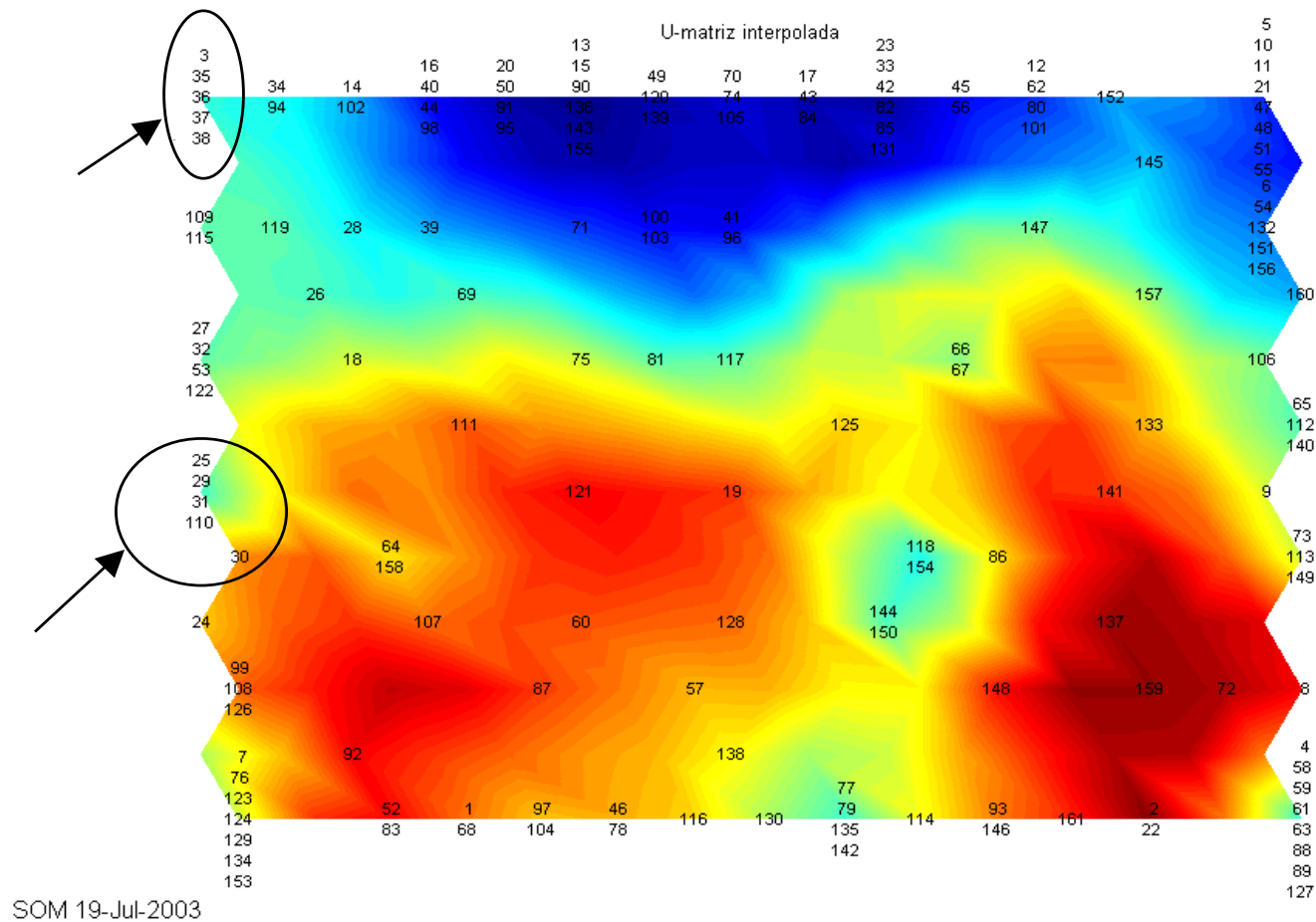


Figura 22 – Interpretação do mapa após auto-organização (figura extraída de ZUCHINI, 2003)

6.8 Ordenamento de pontos em espaços multidimensionais

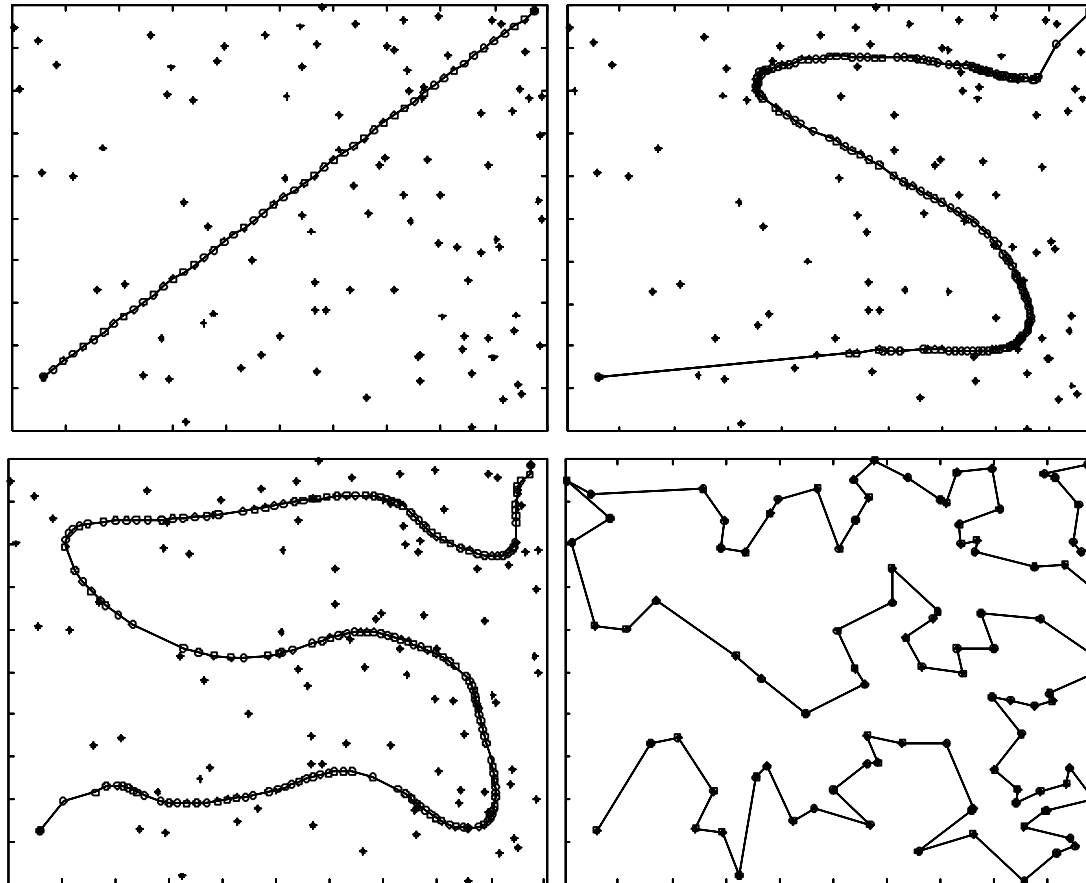


Figura 23 – Várias etapas do processo de auto-organização (GOMES *et al.*, 2004)

6.9 Roteamento de veículos (múltiplos mapas auto-organizáveis)

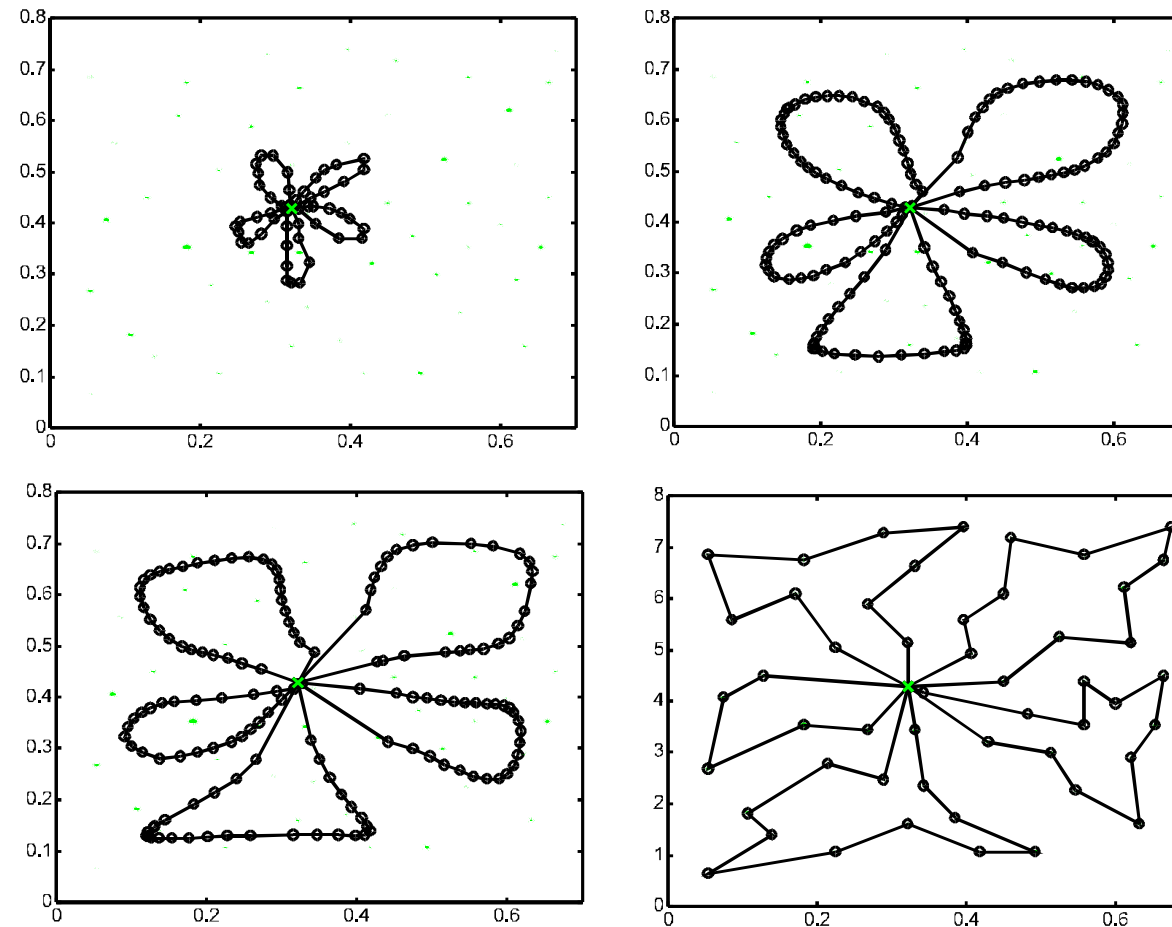


Figura 24 – Várias etapas do processo de auto-organização (GOMES & VON ZUBEN, 2002)

6.10 Mapas auto-organizáveis construtivos

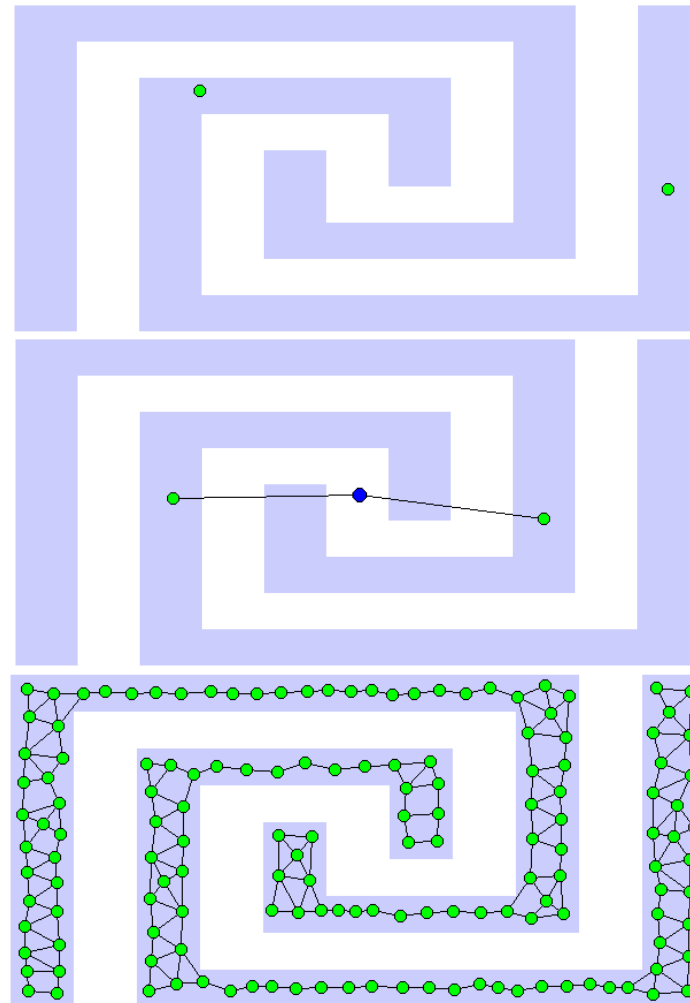


Figura 25 – *Growing Neural Gas* (FRITZKE, 1995)

6.11 Questões a serem melhor investigadas

- Sintonia de parâmetros
- Neurônios que não vencem nunca (devem ser podados para aumentar eficiência)
- Neurônios que vencem sempre
- Dimensão do arranjo para uma dada aplicação
- Número de neurônios, uma vez definido o arranjo
- Inicialização dos pesos
- Apresentação dos dados à rede (padrão-a-padrão ou em batelada?)
- Interpretação do mapa resultante (análise discriminante)
- Métodos construtivos e de poda
- Outras aplicações e múltiplos mapeamentos simultâneos
- Comparações com ferramentas similares

6.12 Extensões

- Learning Vector Quantization (LVQ)

$$\Delta \mathbf{w}_j = \begin{cases} \gamma(\mathbf{x}_i - \mathbf{w}_j) & \text{se a classe for correta} \\ -\gamma(\mathbf{x}_i - \mathbf{w}_j) & \text{se a classe for incorreta} \end{cases}$$

6.13 Referências para mapas auto-organizáveis

- ANGÉNIOL, B., VAUBOIS, C. & LE TEXIER, J.Y. “Self-organizing feature maps and the travelling salesman problem”, *Neural Networks*, 1, 289-293, 1988.
- ARAS, N., OOMMEN, B.J. & ALTINEL, I.K. “The Kohonen Network Incorporating Explicit Statistics and its Application to The Travelling Salesman Problem”, *Neural Networks*, 12, 1273-1284, 1999.
- BARLOW, H. B. “Unsupervised learning”, *Neural Computation*, 1: 295-311, 1989.
- BECKER, S. & PLUMBLEY, M. “Unsupervised neural network learning procedures for feature extraction and classification”, *International Journal of Applied Intelligence*, 6: 185-203, 1996.
- COSTA, J.A.F. “Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis”, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação (FEEC/Unicamp), Dezembro 1999.
- EVERITT, B. “Cluster Analysis”, 3rd. edition, John Wiley, 1993.
- FAQ: The self-organized systems (<http://www.calresco.org/sos/sosfaq.htm>)
- FAVATA, F. & WALKER, R. “A Study of the Application of Kohonen-Type Neural Networks to the Traveling Salesman Problem”, *Biological Cybernetics* 64, 463-468, 1991.

- FORT, J.C. “Solving a Combinatorial Problem via Self-Organizing Maps”, *Biological Cybernetics*, 59, 33-40, 1988.
- FRITZKE, B. “A Growing Neural Gas Network Learns Topologies”, in Tesauro, G., Touretzky, D.S., and Leen, T.K. (eds.). *Advances in Neural Information Processing Systems 7*, The MIT Press, pp. 625-632, 1995.
- GOMES, L.C.T. & VON ZUBEN, F.J. A Neuro-Fuzzy Approach to the Capacitated Vehicle Routing Problem. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'2002)*, vol. 2, pp. 1930-1935, Honolulu, Hawaii, May 12-17, 2002.
- GOMES, L.C.T., VON ZUBEN, F.J. & MOSCATO, P.A. “A Proposal for Direct-Ordering Gene Expression Data by Self-Organising Maps”, *International Journal of Applied Soft Computing*, vol. 5, pp. 11-21, 2004.
- JAIN, A.K., MURTY, M.N. & FLYNN, P.J. “Data Clustering: A Review”, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.
- KASKI, S. “Data Exploration Using Self-Organizing Maps”, Ph.D. Thesis, Helsinki University of Technology, Neural Networks Research Centre, 1997.
- KAUFFMAN, S. “The Origins of Order: Self-Organization and Selection in Evolution”, Oxford University Press, 1993.
- KOHONEN, T. “Self-organized formation of topologically correct feature maps”, *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.
- KOHONEN, T. “Self-Organization and Associative Memory”, 3rd. edition, Springer, 1989 (1st. edition, 1984).
- KOHONEN, T. “The Self-Organizing Map”, *Proceedings of the IEEE*, 78:1464-1480, 1990.
- KOHONEN, T., OJA, E., SIMULA, O., VISA, A. & KANGAS, J. “Engineering applications of the self-organizing map”, *Proceedings of the IEEE*, 84:1358-1384, 1996.
- KOHONEN, T. “Self-Organizing Maps”, 2nd. edition, Springer, 1997.

- MATSUYAMA, Y. “Self-Organization via Competition, Cooperation and Categorization Applied to Extended Vehicle Routing Problems”, *Proc. International Joint Conference on Neural Networks*, 1, 385-390, 1991.
- MODARES, A., SOMHOM, S. & ENKAWA, T. “A Self-Organizing Neural Network Approach for Multiple Traveling Salesman and Vehicle Routing Problems”, *Int. Transactions in Operational Research*, 6, 591-606, 1999.
- POTVIN, J.-I. & ROBILLARD, C. “Clustering for Vehicle Routing with a Competitive Neural Network”, *Neurocomputing*, 8, 125-139, 1995.
- SMITH, K.A. “Neural Networks for Combinatorial Optimization: A Review of More than a Decade of Research”, *INFORMS Journal on Computing*, 11, 1, 15-34, 1999.
- TOOLBOX: <http://www.cis.hut.fi/projects/somtoolbox/>
- ULTSCH, A. “Knowledge Extraction from Self-Organizing Neural Networks”, in O. Opitz *et al.* (eds.) *Information and Classification*, Springer, pp. 301-306, 1993.
- VAKHUTINSKY, A. I. & GOLDEN, B. L. “Solving Vehicle Routing Problems Using Elastic Nets”, *Proc. IEEE International Conference on Neural Networks*, 7, 4535-4540, 1994.
- VESANTO, J. & AHOLA, J. “Hunting for Correlations in Data Using the Self-Organizing Map”, in International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99), ICSC Academic Press, pp. 279-285, 1999.
- ZUCHINI, M.H. “Aplicações de Mapas Auto-Organizáveis em Mineração de Dados e Recuperação de Informação”, Tese de Mestrado, Faculdade de Engenharia Elétrica e de Computação (FEEC/Unicamp), Setembro 2003.

7. Redes neurais recorrentes

- Redes neurais recorrentes são estruturas de processamento capazes de representar uma grande variedade de comportamentos dinâmicos.
- 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).
- Repare que o processo de treinamento vai envolver duas dinâmicas acopladas: a dinâmica da rede neural e a dinâmica do ajuste de pesos.
- Seguem alguns exemplos de arquiteturas de redes neurais recorrentes (DOS SANTOS & VON ZUBEN, 2000).

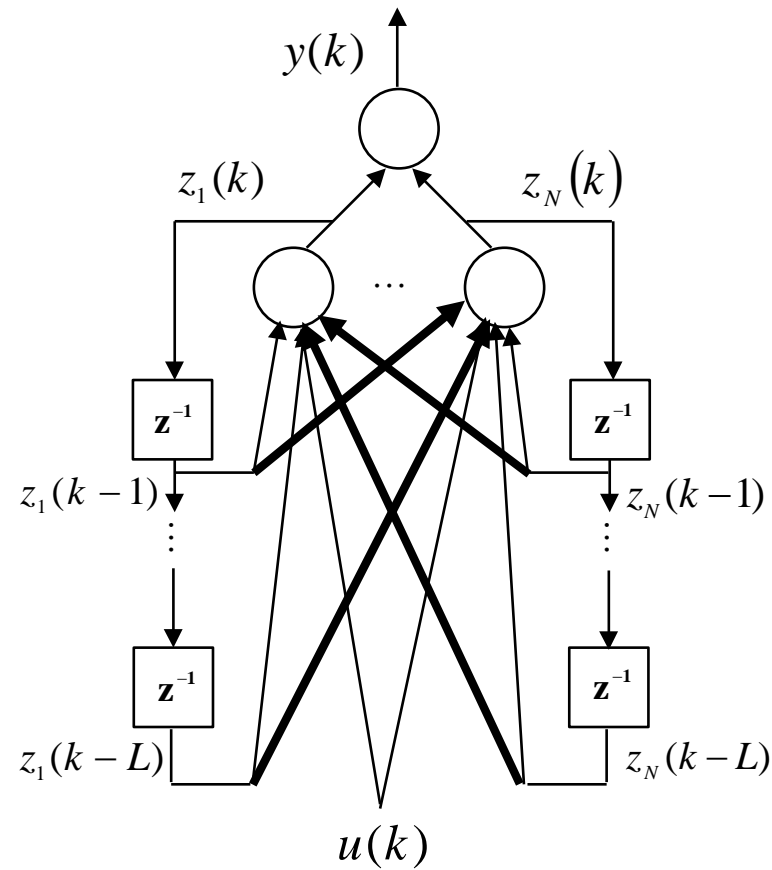


Figura 26 – *Globally recurrent neural network (GRNN)*

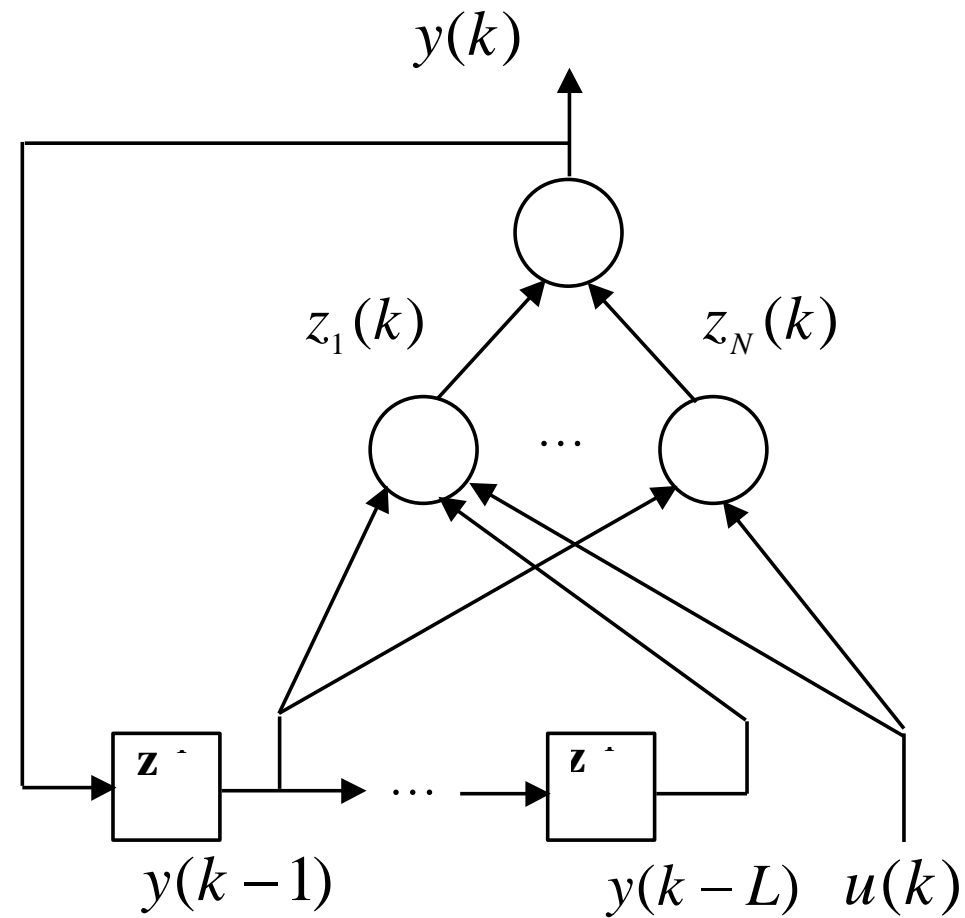


Figura 27 – *Output-feedback recurrent neural network (OFRNN)*

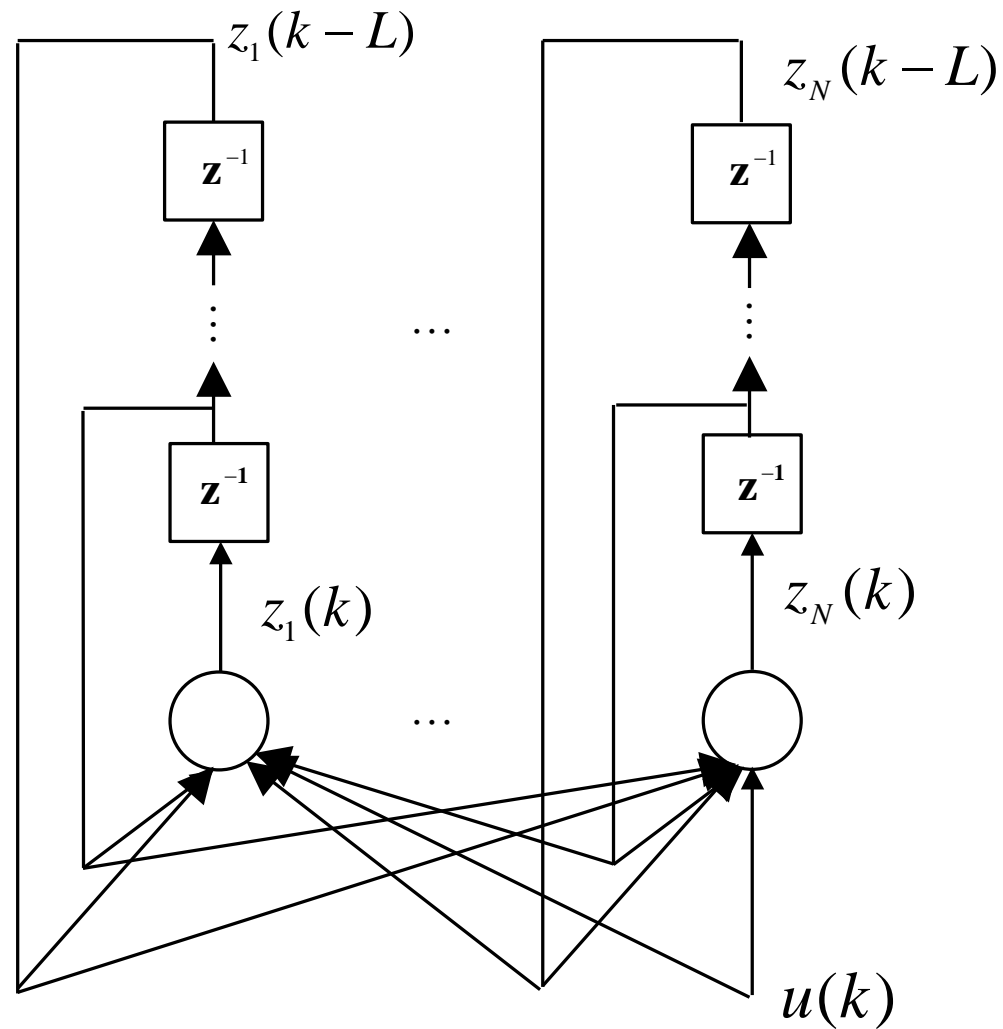


Figura 28 – *Fully recurrent neural network (FRNN)*

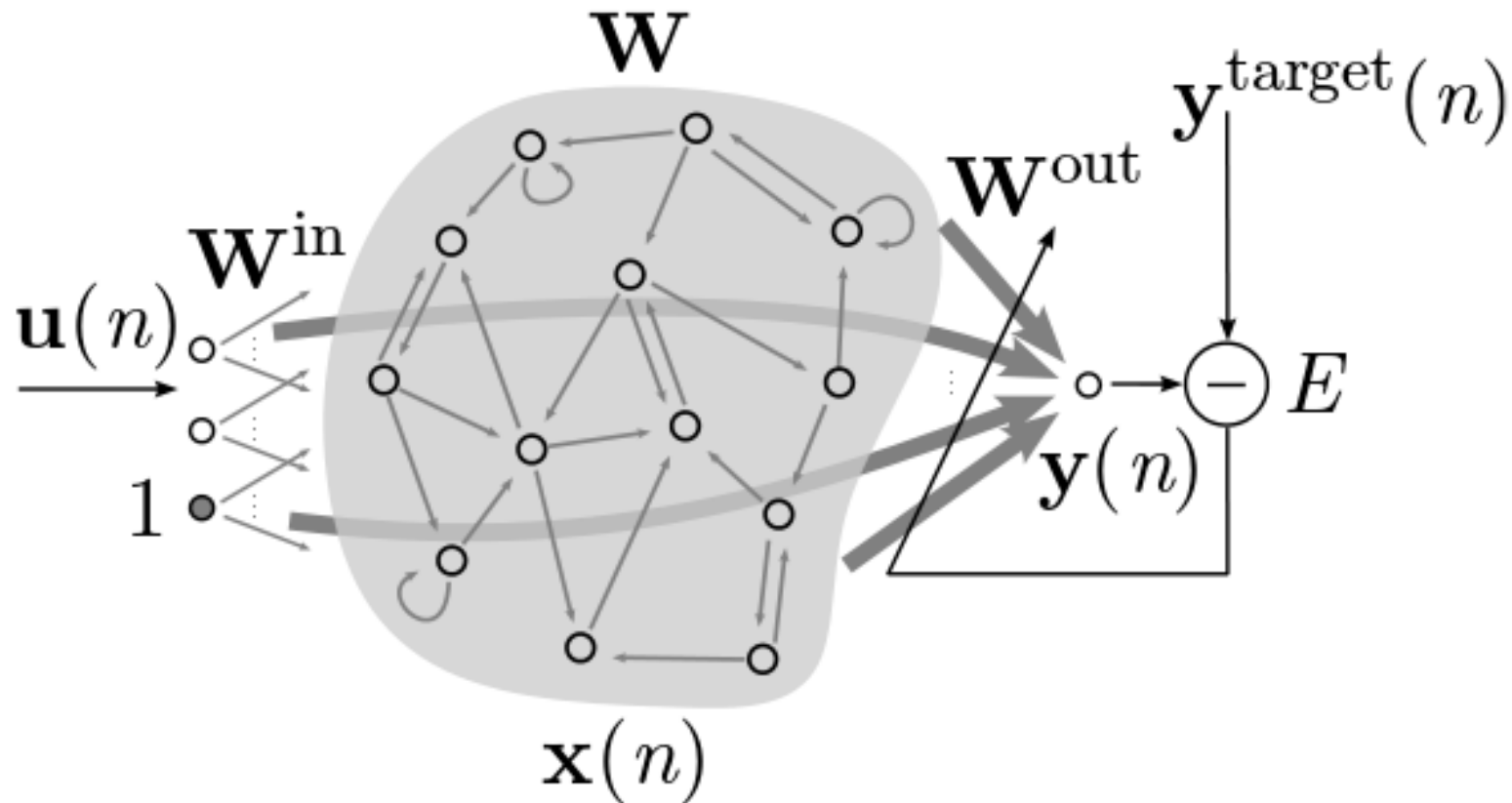


Figura 29 – *Echo state neural network* (JAEGER, 2001; OZTURK *et al.*, 2007)
(pesos da parte dinâmica da rede neural não são ajustáveis)

7.1 Modelagem de sistemas dinâmicos lineares

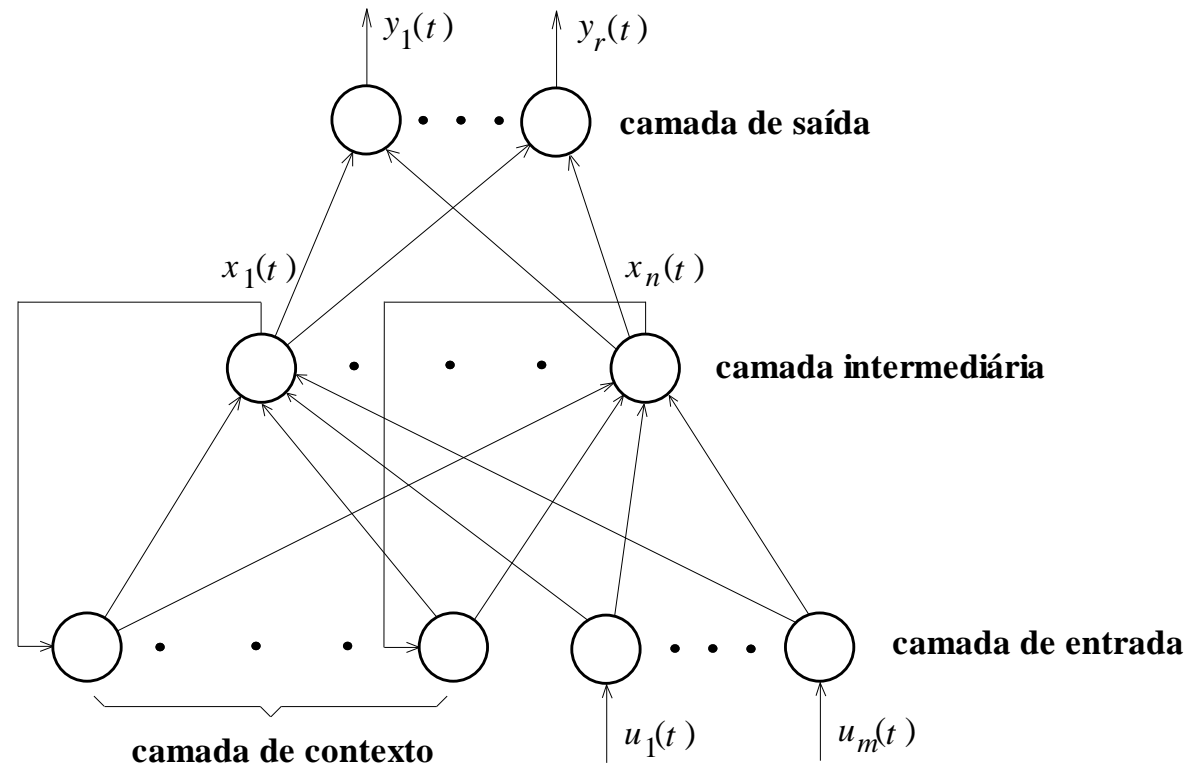


Figura 30 – Estrutura detalhada da Rede de Elman (ELMAN, 1990)

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

7.2 Modelagem de sistemas dinâmicos não-lineares

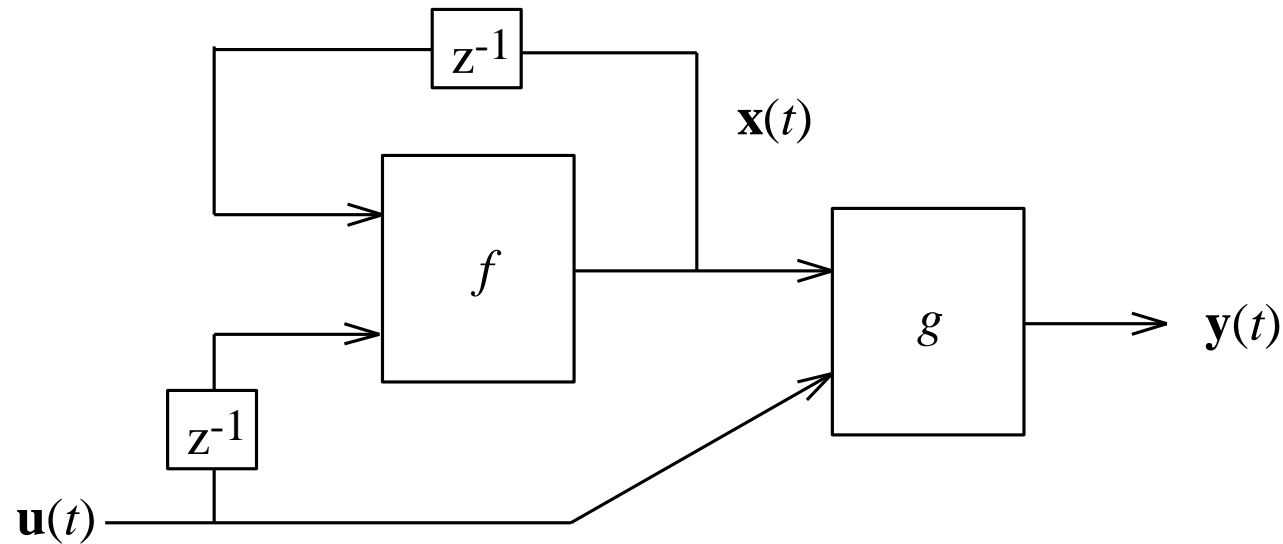


Figura 31 – Representação por espaço de estados de um sistema dinâmico não-linear

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

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

7.3 Treinamento supervisionado para redes recorrentes

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

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

- A disponibilidade de redes neurais recorrentes de importância prática está associada à existência de algoritmos de otimização eficientes para o ajuste dos pesos sinápticos.

8. Rede de Hopfield: recorrência e dinâmica não-linear

- Inspirada em conceitos de física estatística e dinâmica não-linear;
- Principais características: Unidades computacionais não-lineares
Simetria nas conexões sinápticas
Totalmente realimentada (exceto auto-realimentação)

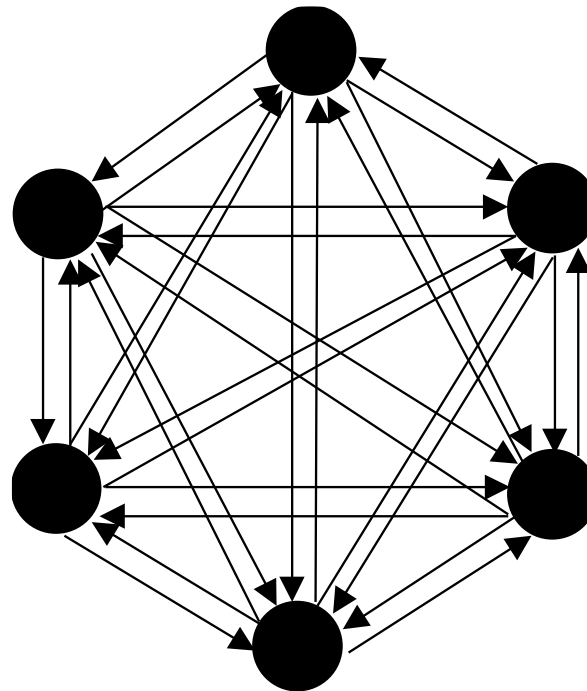


Figura 32 – Rede Neural de Hopfield: ênfase nas conexões

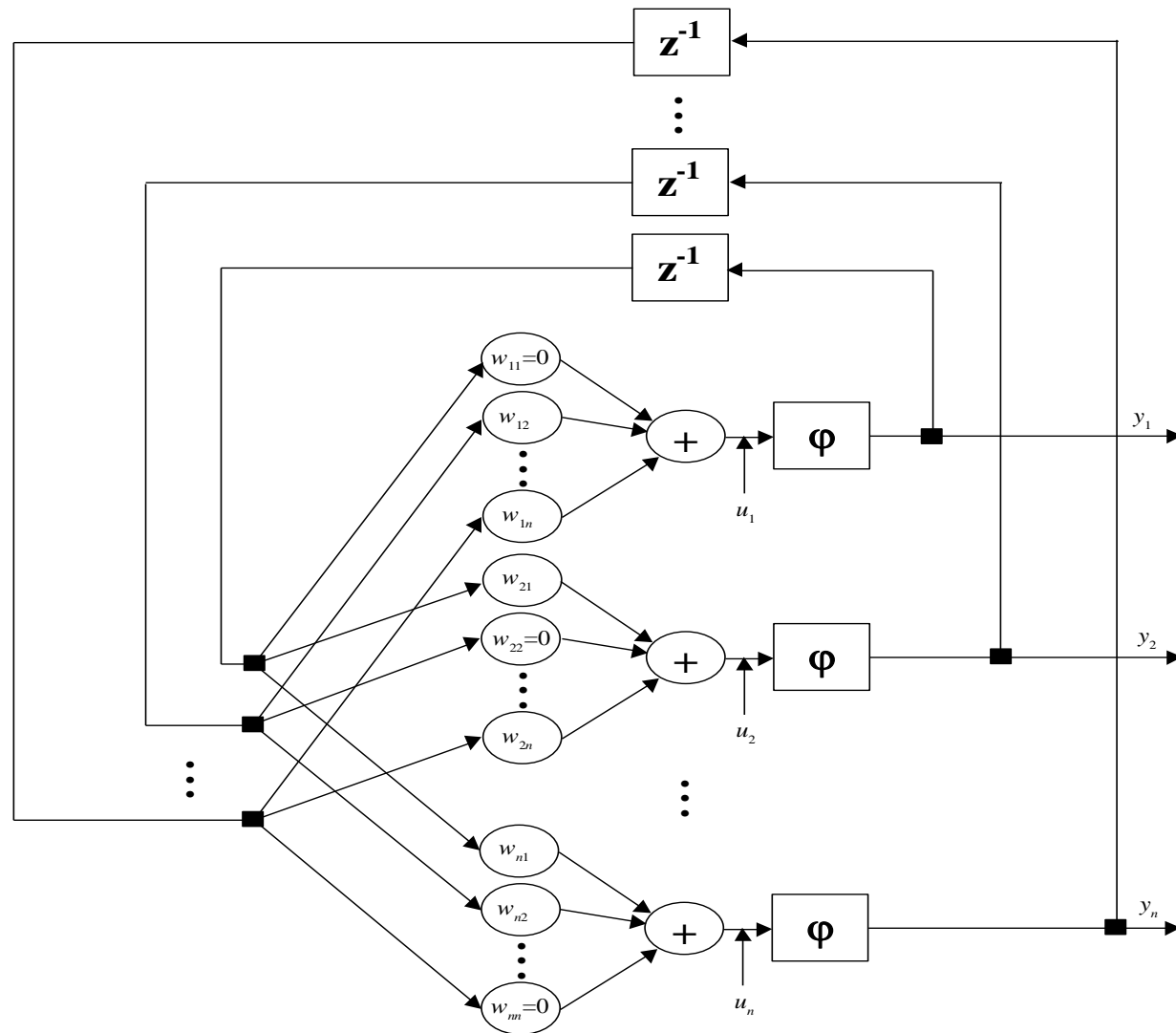


Figura 33 – Rede Neural de Hopfield: ênfase no processamento dinâmico (caso discreto)

8.1 Princípios básicos de sistemas dinâmicos não-lineares

- A teoria de sistemas dinâmicos se ocupa em descrever matematicamente sistemas em movimento, permitindo classificar e prever seu comportamento no tempo.
- 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.
- Neste tópico do curso nos restringiremos à análise e síntese de equações de estado não-forçadas e invariantes no tempo (sistema autônomo):

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

8.2 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. Já no caso não-linear, o estado pode ir para infinito em tempo finito.
- **Múltiplos pontos de equilíbrio isolados**: um sistema linear pode 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, e assim *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 devem 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.

- **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.

8.3 Pontos de equilíbrio como memórias endereçáveis por conteúdo

- Um sistema dinâmico não linear autônomo pode exibir 4 comportamentos em estado estacionário: ponto de equilíbrio, ciclo limite, quase-periodicidade e caos.
- HOPFIELD (1982) propôs uma metodologia para sintetizar sistemas dinâmicos autônomos que só exibem pontos de equilíbrio. Mais ainda, a localização de cada um dos pontos de equilíbrio, no espaço de estados, pode ser especificada pelo projetista do sistema dinâmico não-linear, no caso, uma rede neural recorrente.

- Incorporação de um princípio físico fundamental: armazenagem de informação em uma configuração dinamicamente estável (requer um tempo para se acomodar em uma condição de equilíbrio \rightarrow dinâmica de relaxação \rightarrow estado estacionário).
- Cada padrão a ser armazenado fica localizado em um vale da superfície de energia. Como a dinâmica não-linear da rede é estabelecida de modo a minimizar a energia, os vales representam pontos de equilíbrio estável (cada qual com a sua base de atração).

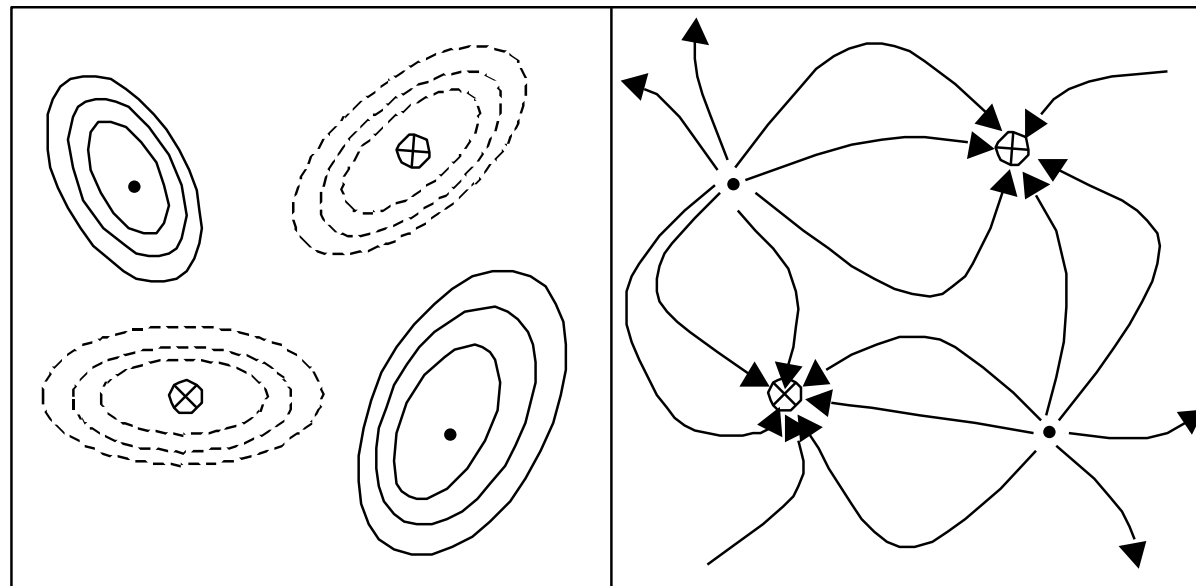


Figura 34 – Superfície de energia: pontos de equilíbrio e bases de atração

- Memória \leftrightarrow Ponto de equilíbrio estável: embora outros pesquisadores já viessem buscando a implementação de tal conceito, HOPFIELD (1982) foi o primeiro a formulá-lo em termos precisos.
- Este tipo de sistema dinâmico pode operar como:
 - 1) Memória associativa (endereçável por conteúdo);
 - 2) Dispositivo computacional para resolver problemas de otimização de natureza combinatória.
- HOPFIELD (1982) considerou um espaço de estados discreto, o que foi generalizado depois para espaços de estados contínuos.
- Como veremos a seguir, a rede de Hopfield pode ser vista como uma memória associativa não-linear, ou uma memória endereçável por conteúdo, cuja principal função é restaurar um padrão binário armazenado (item de memória), em resposta à apresentação de uma versão incompleta (papel restaurador) ou ruidosa (papel de corretor de erro) deste padrão.



memórias

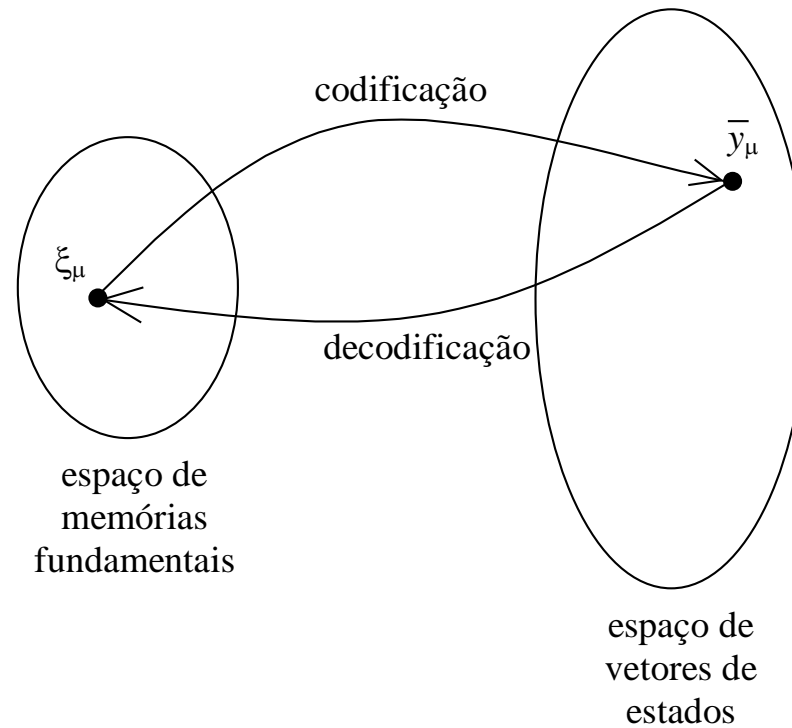


entradas



padrões restaurados

- Como memorizar?
 - Como restaurar?
- Portanto, a recuperação do padrão armazenado na memória se dá a partir de um subconjunto das informações contidas no padrão.
 - A essência da memória endereçável por conteúdo é mapear uma memória fundamental ξ_μ em um ponto fixo estável \bar{y}_μ do sistema dinâmico representado pela rede recorrente.



- Os pesos da rede neural de Hopfield não são definidos via algoritmos iterativos de treinamento, e sim via técnicas de síntese de dinâmicas não-lineares. Para maiores detalhes, consultar a literatura pertinente.

8.4 Recapitulação dos principais conceitos – Rede de Hopfield

- Não-linearidade é condição necessária para produzir múltiplos atratores no espaço de estados de sistemas dinâmicos.
- Hopfield resolveu (parcialmente) o seguinte problema: *Dado um conjunto de estados específicos que devem estar associados a memórias fundamentais, como gerar um sistema dinâmico não-linear que apresente pontos de equilíbrio estável justamente nestes estados específicos?*
- Se este sistema dinâmico não-linear puder ser sintetizado, então vai existir uma superfície de energia com mínimos locais nos referidos estados específicos, sendo que a dinâmica do sistema vai atuar no sentido de conduzir o estado inicial do sistema a um dos mínimos locais da superfície de energia (particularmente àquele em cuja base de atração se encontra a condição inicial).

8.5 Regra de Hebb

- A regra de aprendizado de Hebb é a mais antiga e mais famosa regra de aprendizado, podendo também ser apresentada em duas partes, na forma:
 1. Se os dois neurônios localizados um em cada lado de uma conexão sináptica são ativados simultaneamente (de modo síncrono), então a intensidade da conexão é aumentada.
 2. Se os dois neurônios localizados um em cada lado de uma conexão sináptica são ativados de modo assíncrono, então a intensidade da conexão é reduzida.
- A 2ª parte da regra de Hebb não fazia parte de sua versão original, tendo sido introduzida posteriormente.
- A regra de Hebb pode ser interpretada como um mecanismo (interativo, local e dependente do tempo) de aumentar a eficiência sináptica em função da correlação existente entre as atividades pré- e pós-sináptica.
- Na literatura, existem outras propostas de regras de aprendizado.

8.6 Atratores espúrios

- Quando a rede neural de Hopfield armazena K memórias fundamentais através do ajuste de seus pesos pela regra de Hebb generalizada, os estados estáveis presentes na superfície de energia não vão se restringir aos estados associados às memórias fundamentais armazenadas. Todos os estados estáveis não associados às memórias fundamentais armazenadas são denominados atratores espúrios.
- Os atratores espúrios existem em virtude dos seguintes fatores:
 1. A função de energia E é simétrica, no sentido de que os estados correspondentes ao reverso das memórias fundamentais armazenadas também são estados estáveis;
 2. Toda combinação linear de um número ímpar de estados estáveis também vai ser um estado estável (AMIT, 1989).
 3. Para um grande número K de memórias fundamentais, a função de energia vai produzir pontos de equilíbrio que não estão correlacionados com nenhuma das memórias fundamentais armazenadas na rede (inflexibilidade da superfície de energia).

8.7 Solução de problemas de natureza combinatória

- Os problemas de natureza combinatória são aqueles que se enquadram entre aqueles de mais difícil solução com base nas ferramentas matemáticas e computacionais hoje disponíveis.
- Exemplo: Problema do caixeiro viajante (TSP) → Dadas as localizações de um número específico N de cidades (distribuídas em um plano), o problema é encontrar o menor percurso que se inicia e termina numa mesma cidade, tendo passado uma única vez por todas as outras cidades. É um problema de fácil formulação, mas para o qual não se conhece nenhum método que garanta a obtenção da solução ótima, além do método exaustivo de testar todas as possibilidades e optar pela que produz o menor percurso. Em virtude da explosão (crescimento fatorial em N) de percursos possíveis com o aumento no número de cidades, o método exaustivo torna-se computacionalmente intratável, mesmo para problemas com um número reduzido de cidades (Ex: Para 100 cidades, o número de percursos possíveis é da ordem de 10^{156}).

- A aplicação pioneira de redes de Hopfield no tratamento do problema do caixeiro viajante (uma abordagem extensível a outros problemas de natureza combinatória) se deu com o trabalho de HOPFIELD & TANK (1985).
- Ao mesmo tempo em que era necessário minimizar a função-objetivo, a qual avalia a distância total do percurso, também existiam restrições a serem atendidas, como passar ao menos uma vez em cada cidade.
- Como a violação de uma única restrição torna a correspondente solução inválida, é necessário incorporar junto à função-objetivo termos que penalizam a violação de cada restrição. Além disso, esta função-objetivo estendida deve corresponder à superfície de energia da rede de Hopfield, de tal forma que a aplicação da dinâmica da rede conduza o estado sempre para pontos de menor energia. Com isso, uma possível representação da função de energia assume a forma:

$$E = E^{obj} + c_1 E_1^{restr} + \dots + c_m E_m^{restr}$$

- A formulação original empregada por HOPFIELD & TANK (1985) é apresentada a seguir:

$$E = \frac{A}{2} \left(\sum_X \sum_i \sum_{j \neq i} u_{Xi} u_{Xj} \right) + \frac{B}{2} \left(\sum_i \sum_X \sum_{Y \neq X} u_{Xi} u_{Yi} \right) + \frac{C}{2} \left(\sum_X \sum_i u_{Xi} - N \right)^2 + \frac{D}{2} \left(\sum_X \sum_{Y \neq X} \sum_i d_{XY} u_{Xi} (u_{Y(i+1)} + u_{Y(i-1)}) \right)$$

onde N é o número de cidades e A , B , C , e D são coeficientes de ponderação a serem devidamente definidos.

- Os primeiro e segundo termos penalizam mais de um neurônio ativo por linha e por coluna. Já o terceiro termo penaliza um número de neurônios ativos que difira de N . Por fim, o quarto termo mede o comprimento do percurso realizado pelo caixeiro viajante, sendo que o que se busca é justamente minimizar este comprimento de percurso.

- Infelizmente, não é uma tarefa elementar obter os coeficientes A , B , C , e D , sendo que uma escolha inadequada pode fazer com que soluções de boa qualidade, incluindo a solução ótima, deixem de ser mínimos locais da função de energia.

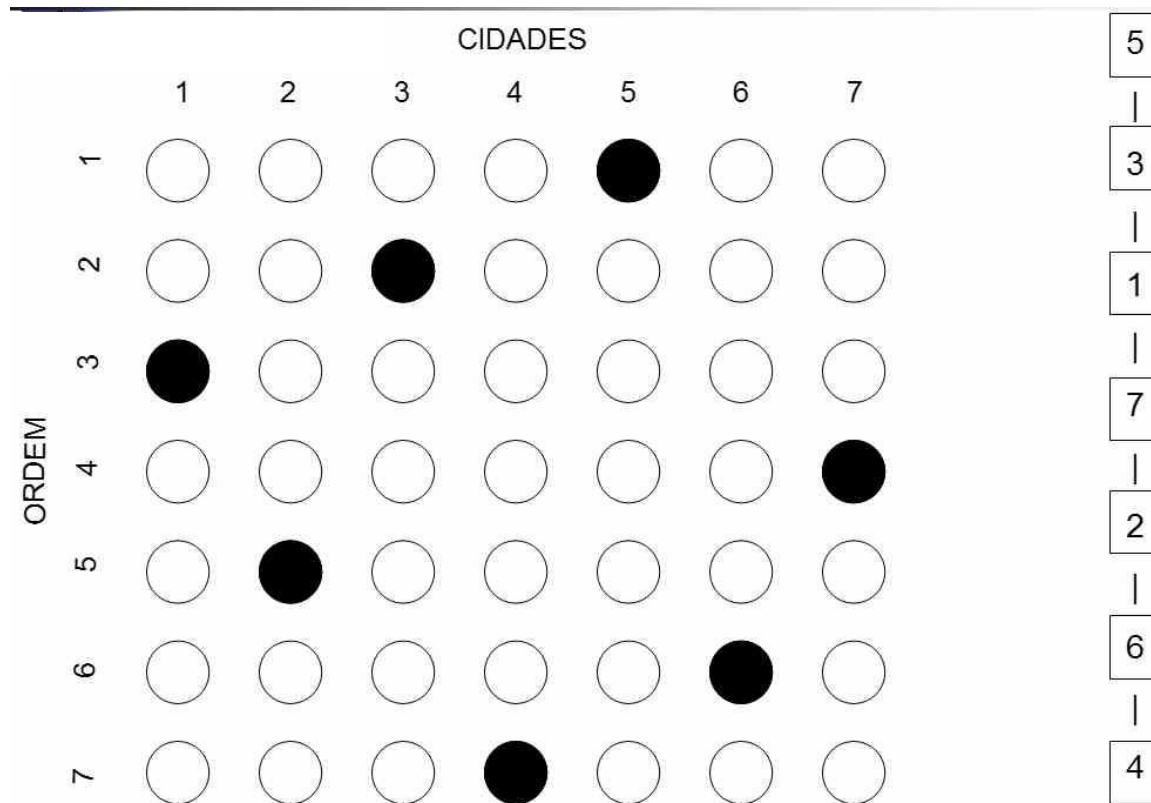


Figura 1 – Interpretação do ponto de equilíbrio como uma solução para o problema do caixeiro viajante (repare que há um e somente um neurônio ativo por linha e por coluna)

8.8 Solução de problemas de programação matemática

- Como as redes neurais recorrentes podem ser implementadas em hardware específico, inerentemente paralelo e distribuído, a sua aplicação junto a problemas de otimização de larga escala, possivelmente envolvendo restrições de tempo real, apresenta um grande potencial.
- A primeira iniciativa de se empregar um computador analógico para se resolver um problema de programação linear pode ser atribuída a PYNE (1956). E após HOPFIELD & TANK (1985), houve muitas outras iniciativas voltadas para o tratamento de problemas de otimização a partir da especificação de funções de energia a serem minimizadas.
- KENNEDY & CHUA (1988) propuseram redes neurais recorrentes para a solução de problemas de programação não-linear, que inclui a formulação de HOPFIELD & TANK (1985) como um caso particular. No entanto, a presença de termos de penalidade permitiam apenas a obtenção de soluções aproximadas.

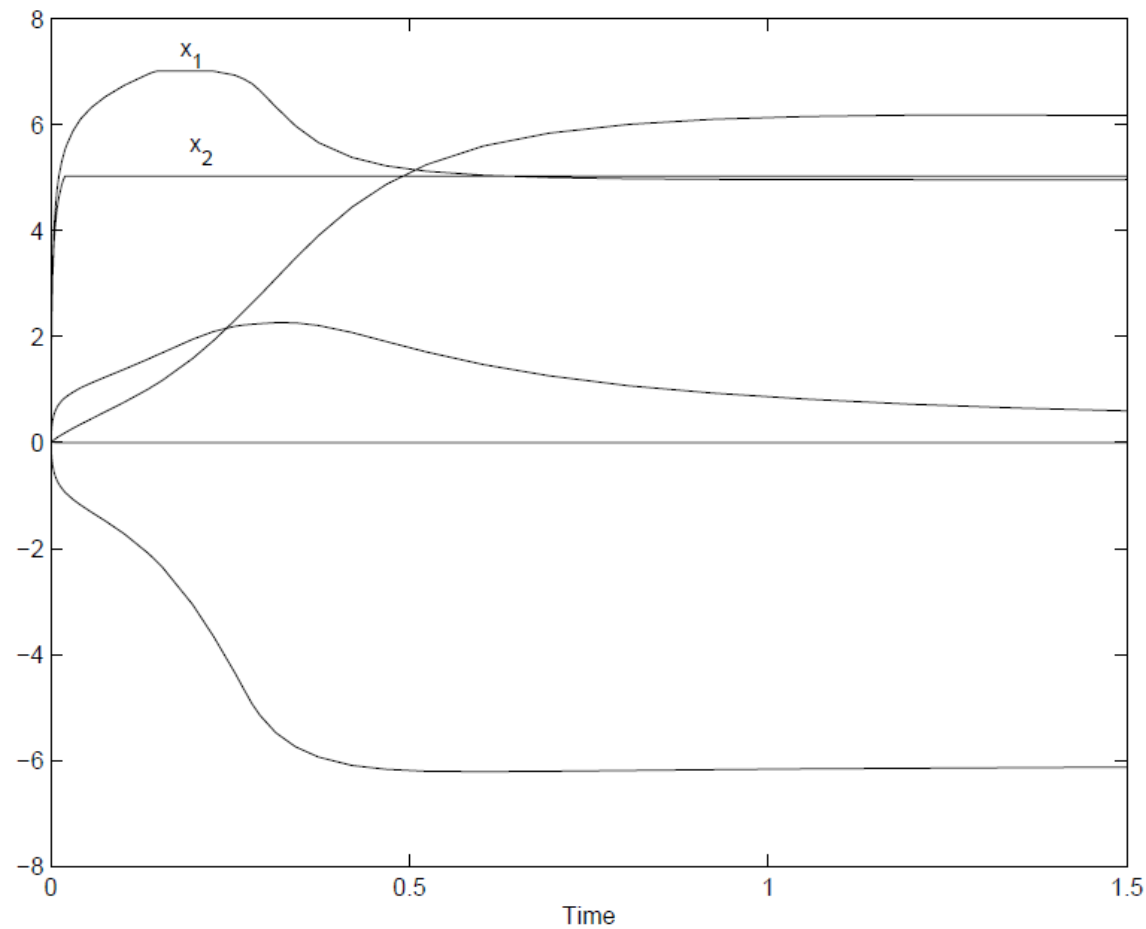
- Outras abordagens se sucederam, geralmente fundamentadas em termos de penalidade ou formulações lagrangeanas. Consulte CICHOCKI & UNBEHAUEN (1993) e LILLO *et al.* (1993) para uma revisão do estado-da-arte à época.
- Fazendo uso de formulações primais-duais e métodos de projeção, XIA & WANG (2001) apresentam uma revisão do uso bem sucedido de redes neurais recorrentes para a solução de problemas de programação quadrática e de programação linear.
- Como um exemplo (XIA & WANG, 2001), considere o problema quadrático a seguir:

$$\begin{array}{ll}\text{Minimize} & x_1^2 + x_2^2 + x_1x_2 - 30x_1 - 30x_2 \\ \text{subject to} & \frac{5}{12}x_1 - x_2 \leq \frac{35}{12}, \\ & \frac{5}{2}x_1 + x_2 \leq \frac{35}{2}, \\ & -x_1 \leq 5, \\ & x_2 \leq 5.\end{array}$$

que pode ser reformulado como segue:

$$\begin{array}{ll}\text{Minimize} & x_1^2 + x_2^2 + x_1 x_2 - 30x_1 - 30x_2 \\ \text{subject to} & \frac{5}{12}x_1 - x_2 + x_3 = \frac{35}{12}, \\ & \frac{5}{2}x_1 + x_2 + x_4 = \frac{35}{2}, \\ & -5 \leq x_1 \leq 7, -5 \leq x_2 \leq 5, 0 \leq x_3 \leq 10, 0 \leq x_4 \leq 35.\end{array}$$

- Este problema tem solução única $(x_1, x_2) = (5, 5)$, e sua solução empregando uma rede neural recorrente produz como resultado o gráfico a seguir, na formulação primal-dual. Repare que, após um transitório, ocorre uma convergência para a solução desejada.



- Como trabalhos mais recentes na área, tem-se DA SILVA *et al.* (2006), LEUNG *et al.* (2003) e WEN *et al.* (2009).

8.9 Referências bibliográficas – Redes neurais recorrentes

- AMIT, D.J. *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press, 1989.
- 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.
- BHAYA, A., KASZKUREWICZ, E. & KOZYAKIN, V.S. Existence and Stability of a Unique Equilibrium in Continuous-Valued Discrete-Time Asynchronous Hopfield Neural Networks. *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 620-628, 1996.
- BRUCK, J. On the convergence properties of the Hopfield model. *Proc. of the IEEE*, vol. 78, pp. 1579-1585, 1990.
- CICHOCKI, A. & UNBEHAUEN, R. *Neural Networks for Optimization and Signal Processing*, John Wiley, 1993.
- CLEEREMANS, A., SERVAN-SCHREIBER, D., MCCLELLAND, J. Finite state automata and simple recurrent networks. *Neural Computation*, vol. 1, no. 3, pp. 372-381, 1989.
- COHEN, M.A. & GROSSBERG, S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 815-826, 1983.
- 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.
- DA SILVA, I.N., AMARAL, W.C., ARRUDA, L.V.R. Neural approach for solving several types of optimization problems, *Journal of Optimization Theory and Applications*, vol. 128, pp. 563-580, 2006.
- 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.

- 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.
- ELMAN, J. L. Finding structure in time. *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- FUNAHASHI, K.-I., NAKAMURA, Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, vol. 6, no. 5, pp. 801-806, 1993.
- GILES, C., MILLER, C., CHEN, D., CHEN, H., SUN, G., LEE, Y. 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.
- HOPFIELD, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 79, pp. 2554-2558, 1982.
- HOPFIELD, J.J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 81, pp. 3088-3092, 1984.
- HOPFIELD, J.J. & TANK, D.W. ‘Neural’ computation of decisions in optimization problems. *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- HOPFIELD, J.J. & TANK, D.W. Computing with neural circuits: A model. *Science*, vol. 233, pp. 625-633, 1986.
- 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.
- 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.
- KENNEDY, M.D. & CHUA, L.N. Neural networks for nonlinear programming, *IEEE Transactions on Circuits and Systems*, vol. 35, no. 5, pp. 554-562, 1988.
- KHALIL, H.K. *Nonlinear Systems*. 2nd. edition, Prentice Hall, 1996.

- KOLEN, J.F. *Exploring the Computational Capabilities of Recurrent Neural Networks*. Ph.D. Thesis, The Ohio State University, 1994.
- KOZMA, R. Intentional systems: Review of neurodynamics, modeling, and robotics implementation. *Physics of Life Reviews*, vol. 5, no. 1, pp 1-21, 2008.
- LEUNG, Y., CHEN, K. & GAO, X. A high-performance feedback neural network for solving convex nonlinear programming problems, *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1469–1477, 2003.
- 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.
- LILLO, W.E., LOH, M.H., HUI, S. & ZAK, S.H. On solving constrained optimization problems with neural networks: A penalty method approach, *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 931-940, 1993.
- MAASS, W. Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Networks*, vol. 10, no. 9, pp. 1659-1671, 1997.
- 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.
- 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.
- MORITA, M. Associative memory with nonmonotonic dynamics. *Neural Networks*, vol. 6, pp. 115-126, 1993.
- 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.

- 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.
- 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.
- PHAM, D. T., LIU, X. Dynamic system modelling using partially recurrent neural networks. *Journal of Systems Engineering*, pp. 134-141, 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.
- PYNE, I.B. Linear programming on an electronic analogue computer. *Transactions of the American Institute of Electrical Engineering*, vol. 75, pp. 139, 1956.
- 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.
- 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.
- 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.
- 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.
- 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, June 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.
- WEN, U.-P., LAN, K.-M. & SHIH, H.-S. A review of Hopfield neural networks for solving mathematical programming problems, *European Journal of Operational Research*, vol. 198, pp. 675-687, 2009.
- 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.
- XIA, Y. & WANG, J. Recurrent neural networks for optimization: The state of the art, in Medsker, L.R. & Jain, L.C. (eds.) *Recurrent Neural Networks: Design and Applications*, Chapter 2, pp. 23-55, CRC Press, 2001.
- ZBIKOWSKI, R.W. *Recurrent Neural Networks: Some Control Aspects*. Ph.D. Thesis, Faculty of Engineering, Glasgow University, 1994.

9. Deep learning

- A possibilidade de lidar com cada vez mais dados no treinamento de redes neurais, visando obter ganhos continuados de desempenho em tarefas de classificação e regressão, levou à proposição e ao estabelecimento de redes neurais com muitas camadas intermediárias (mais que 5 camadas) e com milhões ou até bilhões de conexões sinápticas.
- Devido à existência de muitas camadas intermediárias, essas redes neurais são ditas serem de arquitetura profunda e o seu treinamento, portanto, é denominado *deep learning*.
- Existem camadas totalmente conectadas, como no caso de MLPs, mas como grande parte das aplicações envolve tratamento de imagens digitais, também é bastante difundido o emprego de camadas convolucionais. Também existem redes neurais recorrentes com arquiteturas profundas. Os neurônios também podem diferir em sua

função de ativação e operação, com as versões ReLU (*retified linear unit*) e LSTM (*long short-term memory*), essa última para redes recorrentes.

- Outros conceitos importantes são os de auto-encoders, da técnica de regularização denominada *dropout*, máquina de Boltzmann e o conceito de *manifold*.
- A mais relevante motivação prática para o emprego de *deep learning* é o fato de que os atributos essenciais dos estímulos de entrada são automaticamente determinados durante o treinamento, sendo que as camadas mais próximas da entrada respondem por atributos de baixo nível e as mais distantes por atributos de alto nível, contendo um maior grau de abstração e de relacionamento entre atributos de baixo nível.
- Para uma visão geral da área, sugere-se uma leitura dos seguintes artigos de revisão:
 - ✓ BENGIO, Y. “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
 - ✓ LECUN, Y.; BENGIO, Y. & HINTON, G. “Deep learning”, *Nature*, vol. 521, pp. 436-444, 2015.
 - ✓ SCHMIDHUBER, J. “Deep learning in neural networks: An overview”, *Neural Networks*, vol. 61, pp. 85-117, 2015.