Redes Neurais Artificiais – Parte 1

Índice

1. Leituras complementares

- **Páginas WEB:**

ftp://ftp.sas.com/pub/neural/FAQ.html (comp.ai.neural-nets FAQ)

http://nips.djvuzone.org/ (Todos os artigos on-line da conferência "Neural Information Processing Systems (NIPS)")

http://ieeexplore.ieee.org/Xplore (Todas as publicação on-line do IEEE, inclusive de conferências em redes neurais artificiais, como a International Joint Conference on Neural Networks (IJCNN))

- **Periódicos:**

IEEE Transactions on Neural Networks and Learning Systems

Neural Computation (MIT Press)

International Journal of Neural Systems (World Scientific Publishing)

IEEE Transaction on Systems, Man, and Cybernetics (Part B)

Information Sciences (Elsevier)

Learning & Nonlinear Models (SBIC - Brasil)

Neural Networks (Pergamon Press)

Neurocomputing (Elsevier)

Biological Cybernetics (Springer)

Neural Processing Letters (Springer)

Cognitive Science (CSS)

Machine Learning (Springer)

- **Livros:**

1. Arbib, M.A. (ed.) (2002) “The Handbook of Brain Theory and Neural Networks”, The MIT Press, 2nd. edition, ISBN: 0262011972.
2. Bertsekas, D.P. & Tsitsiklis, J.N. (1996) “Neuro-Dynamic Programming”, Athena Scientific, ISBN: 1886529108.
3. Bishop, C.M. (1996) “Neural Networks for Pattern Recognition”, Oxford University Press, ISBN: 0198538642.
4. Bishop, C.M. (2007) “Pattern Recognition and Machine Learning”, Springer, ISBN: 0387310738.
5. Braga, A.P., de Carvalho, A.P.L.F. & Ludermir, T.B. (2007) “Redes Neurais Artificiais – Teoria e Aplicações”, Editora LTC, 2a. edição, ISBN: 9788521615644.
6. Chauvin, Y. & Rumelhart, D.E. (1995) “Backpropagation: Theory, Architectures, and Applications”, Lawrence Erlbaum Associates, ISBN: 080581258X.
7. Cherkassky, V. & Mulier, F. (2007) “Learning from Data: Concepts, Theory, and Methods”, 2nd edition, Wiley-IEEE Press, ISBN: 0471681822.
8. Cristianini N. & Shawe-Taylor, J. (2000) “An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods”, Cambridge University Press, ISBN: 0521780195.
9. da Silva, I.N., Spatti, D.H. & Flauzino, R.A. (2010) “Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas”, Artliber Editora Ltda., ISBN: 9788588098534.

10. Dayan, P. & Abbot, L.F. (2001) “Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems”, The MIT Press, ISBN: 0262041995.

11. Duda, R.O., Hart, P.E. & Stork, D.G. (2000) “Pattern Classification”, 2nd edition, Wiley-Interscience, ISBN: 0471056693.

12. Edelman, G.M. (1988) “Neural Darwinism: The Theory of Neuronal Group Selection”, Basic Books, ISBN: 0465049346.

13. Fausett, L. (2004) “Fundamentals of Neural Networks: Architectures, Algorithms, and Applications”, Dorling Kindersley India, ISBN: 8131700534.

14. Fiesler, E. & Beale, R. (1996) “Handbook of Neural Computation”, Institute of Physics Publishing, ISBN: 0750303123.

15. Gardner, H. (2011) “Frames of Mind: The Theory of Multiple Intelligences”, 3rd edition, BasicBooks, ISBN: 0465024335.

16. Hassoun, M. (2003) “Fundamentals of Artificial Neural Networks”, A Bradford Book, ISBN: 0262514672.

17. Hastie, T., Tibshirani, R. & Friedman, J.H. (2001) “The Elements of Statistical Learning”, Springer, ISBN: 0387952845.

18. Haykin, S. (2008) “Neural Networks and Learning Machines”, 3rd edition, Prentice Hall, ISBN: 0131471392.

19. Hecht-Nielsen, R. (1990) “Neurocomputing”, Addison-Wesley Publishing Co., ISBN: 0201093553.

20. Hertz, J., Krogh, A. & Palmer, R. (1991) “Introduction to the Theory of Neural Computation”, Addison-Wesley, ISBN: 0201515601.

21. Kearns, M.J., Vazirani, U. (1994) “An Introduction to Computational Learning Theory”, The MIT Press, ISBN: 0262111934.

22. Kohonen, T. (1989) “Self-Organization and Associative Memory”, 3rd edition, Springer-Verlag, ISBN: 0387513876. (1st Edition: 1984; 2nd edition: 1988)

23. Kohonen, T. (2000) “Self-Organizing Maps”, 3rd Edition, Springer, ISBN: 3540679219.

24. Luenberger, D.G. (1984) “Linear and Nonlinear Programming”, 2nd edition, Addison-Wesley, ISBN: 0201157942.

25. Mackay, D.J.C. (2003) “Information Theory, Inference and Learning Algorithms”, Cambridge University Press, ISBN: 0521642981.

26. Mardia, K.V., Kent, J.T., Bibby, J.M. (1980) “Multivariate Analysis”. Academic Press, ISBN: 0124712525.

27. Marsland, S. (2009) “Machine Learning: An Algorithmic Perspective”, Chapman and Hall/CRC, ISBN: 1420067184.

28. Masters, T. (1995) “Advanced Algorithms for Neural Networks: A C++ Sourcebook”, John Wiley and Sons, ISBN: 0471105880.

29. Minsky, M.L. (1988) “The Society of Mind”, Simon & Schuster, ISBN: 0671657135.

30. Minsky, M.L. & Papert, S.A. (1988) “Perceptrons: Introduction to Computational Geometry”, Expanded edition, The MIT Press, ISBN: 0262631113. (1st edition: 1969)

31. Mitchell, T.M. (1997) “Machine Learning”, McGraw-Hill, ISBN: 0071154671.

32. Ripley, B.D. (2008) “Pattern Recognition and Neural Networks”, Cambridge University Press, ISBN: 0521717701.

33. Rumelhart, D.E. & McClelland, J.L. (1986) “Parallel Distributed Processing: Explorations in the Microstructure of Cognition”, volumes 1 & 2. The MIT Press, ISBN: 026268053X.

34. Schalkoff, R.J. (1997) “Artificial Neural Networks”, The McGraw-Hill Companies, ISBN: 0071155546.

35. Schölkopf, B. & Smola, A.J. (2001) “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, The MIT Press, ISBN: 0262194759.

36. Sutton, R.S. & Barto, A.G. (1998) “Reinforcement Learning: An Introduction”, The MIT Press, ISBN: 0262193981.

37. Vapnik V.N. (1998) “Statistical Learning Theory”, Wiley-Interscience, ISBN: 0471030031.

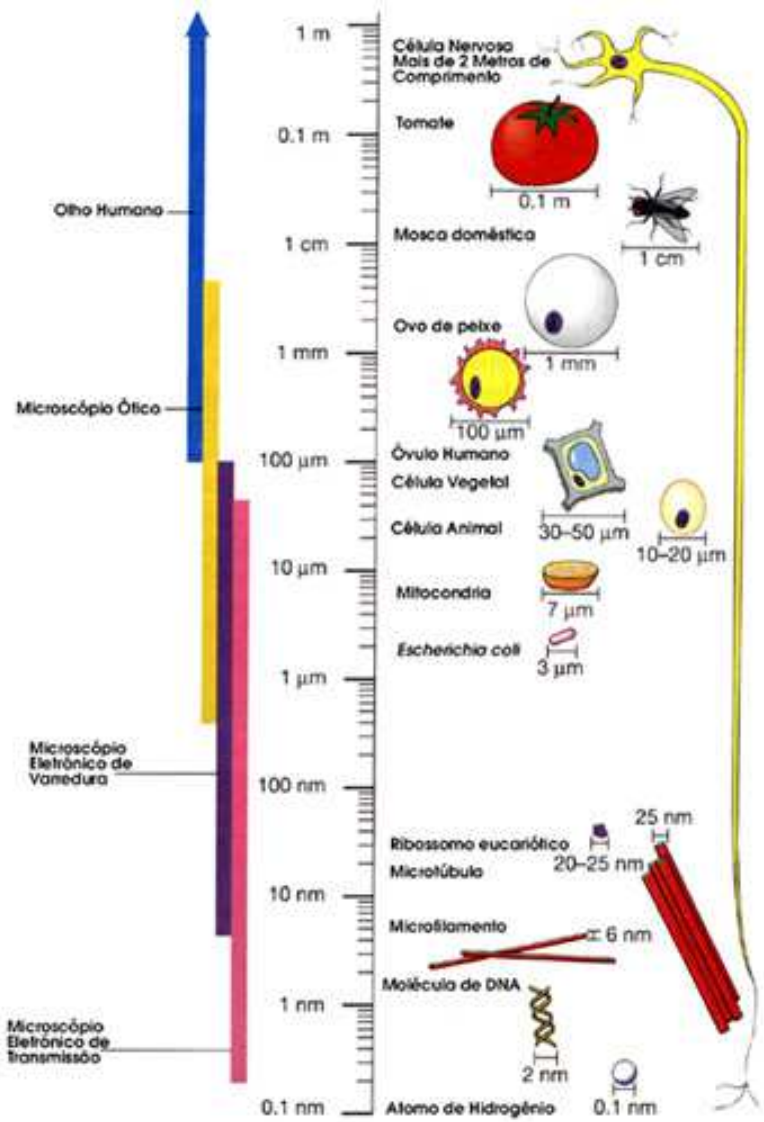
38. Vapnik V.N. (1999) “The Nature of Statistical Learning Theory”, 2nd edition, Springer, ISBN: 0387987800.

39. Weigend, A.S. & Gershenfeld, N.A. (eds.) (1993) “Time Series Prediction: Forecasting the Future and Understanding the Past”, Perseus Press, ISBN: 0201626020.

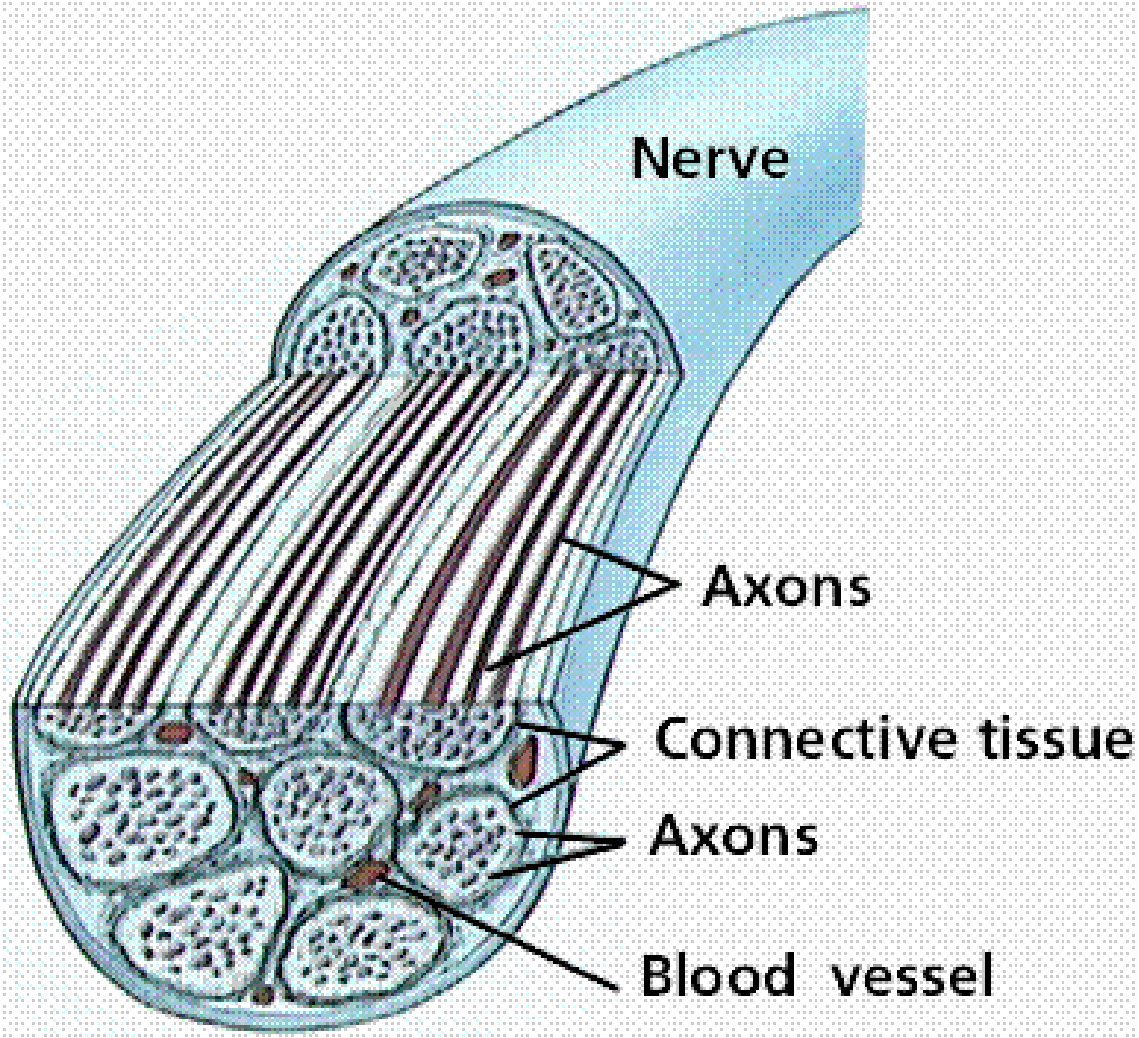
40. Wilson, R.A. & Keil, F.C. (eds.) (2001) “The MIT Encyclopedia of the Cognitive Sciences”, The MIT Press, ISBN: 0262731444.

2. Números e nomenclatura

- Descoberta do microscópio: ~1590
- Descoberta da célula: ~1680
- Célula como unidade constituinte dos seres vivos: ~1830
- Constituintes básicos do cérebro são os neurônios: Ramón y Cajál, ~1909
- O cérebro humano pesa ~1,5 quilos e consome ~20% da energia do corpo;
- 100 gramas de tecido cerebral requerem ~3,5ml de oxigênio por minuto;
- O cérebro humano apresenta ~10^{11} neurônios e ~10^{14} sinapses ou conexões, com uma média de ~1000 conexões por neurônio, podendo chegar a ~10000 conexões.
- Em seres humanos, 70% dos neurônios estão localizados no córtex;
- Tipos de células neurais: horizontal, estrelada, piramidal, granular, fusiforme.
- Classificação de acordo com a função: sensoriais, motoras, intrínsecas.



- O diâmetro do corpo celular de um neurônio mede de ~5µm (célula granular) a ~60µm (célula piramidal);
- Em termos fisiológicos, um neurônio é uma célula com a função específica de receber, processar e enviar informação a outras partes do organismo.
- Um nervo é formado por um feixe de axônios, com cada axônio associado a um único neurônio;
- Os nervos apresentam comprimentos variados, podendo chegar a metros.



Estrutura de um nervo

- A estrutura e as funcionalidades do cérebro são governadas por princípios básicos de alocação de recursos e otimização sujeita a restrições.

LAUGHLIN, S.B. & SEJNOWSKI, T.J. (2003) "Communication in neuronal networks", Science, vol. 301, no. 5641, pp. 1870–1874.

- O ser humano pode reagir simultaneamente a uma quantidade bem limitada de estímulos, o que pode indicar que mecanismos de alocação de recursos (e.g. glicose, oxigênio) baseados em prioridades são implementados no cérebro.

NORMAN, D.A. & BOBROW, D.G. (1975) "On data-limited and resource-limited processes", Cognitive Psychology, vol. 7, pp. 44-64.

- Alguns autores defendem que o córtex humano pode ser modelado na forma de uma rede "mundo pequeno" (BASSETT & BULLMORE, 2006; SPORNS & HONEY, 2006; SPORNS, 2010) ou então uma rede complexa (AMARAL & OTTINO, 2004).

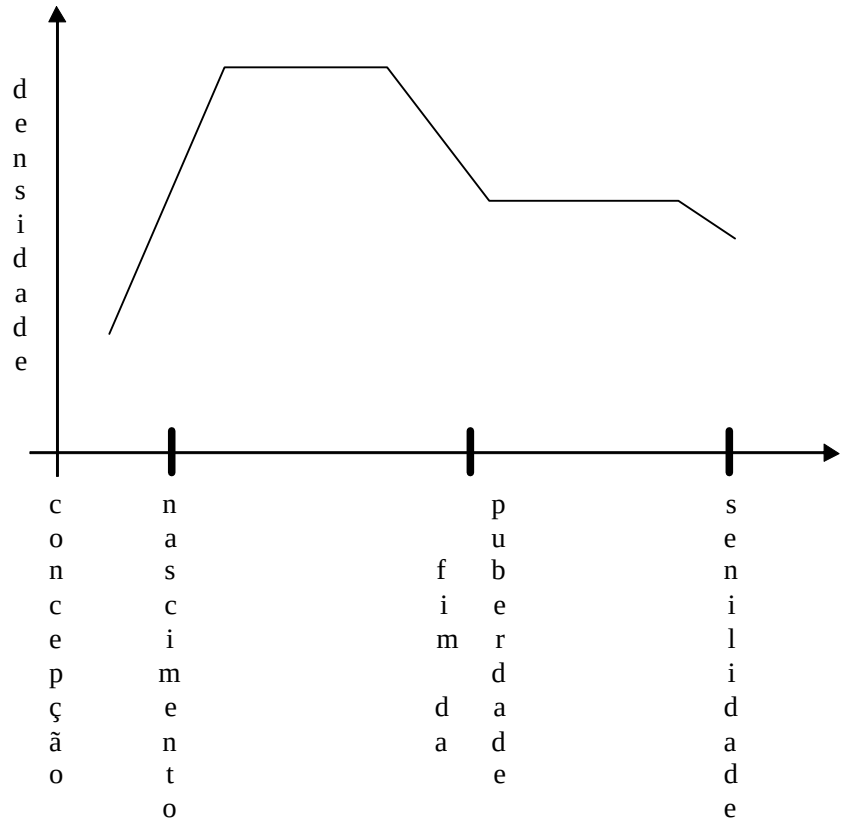
AMARAL, L. & OTTINO, J. (2004) "Complex networks", The European Physical Journal B – Condensed Matter and Complex Systems, vol. 38, no. 2, pp. 147-162.
BASSETT, D.S. & BULLMORE, E. (2006) "Small-world brain networks", Neuroscientist, vol. 12, no. 6, pp. 512-523.
SPORNS, O. & HONEY, C.J. (2006) "Small worlds inside big brains", Proceedings of the National Academy of Science, vol. 103, no. 51, pp. 19219-19220.
SPORNS, O. (2010) "Networks of the Brain", The MIT Press, ISBN: 0262014696.

- Há um expressivo aumento na densidade de conexões sinápticas da vida embrionária até a idade de 2 anos. Quando se atinge a idade de 2 anos, o ser humano apresenta a maior concentração de sinapses, a qual se mantém num nível elevado até o início da puberdade. Até o término da puberdade, há uma queda acentuada no número de sinapses.

- Esse processo de ampliação e redução de sinapses, contudo, não é homogêneo, pois nas regiões sensório-motoras este processo ocorre mais cedo, enquanto que ele é retardado em áreas associadas aos processos cognitivos.

- A redução de sinapses é dramática: o número de sinapses ao término da puberdade pode chegar a 50% do número existente com a idade de 2 anos. Há uma perda de até 100.000 sinapses por segundo na adolescência.

KOLB, B & WHISHAW, I.Q. (2008) "Fundamentals of Human Neuropsychology", Worth Publishers, 6th. edition, ISBN: 0716795868.



Evolução da densidade de sinapses ao longo da vida de um ser humano

- Acredita-se ser impossível que o código genético de um indivíduo seja capaz de conduzir todo o processo de organização topológica do cérebro. Apenas aspectos gerais dos circuitos envolvidos devem estar codificados geneticamente.

- Logo, para explicar as conformações sinápticas, recorre-se a dois mecanismos gerais de perda de sinapses: *experience expectant* e *experience dependent* (KOLB & WHISHAW, 2008).
- Podas baseadas em *experience expectant* estão vinculadas à experiência sensorial para a organização das sinapses. Geralmente, os padrões sinápticos são os mesmos para membros de uma mesma espécie. Exemplo: A formação de sinapses no córtex visual depende da exposição a atributos como linha de orientação, cor e movimento.
- Podas baseadas em *experience dependent* estão vinculadas a experiências pessoais únicas, tal como falar uma língua distinta. Com isso, defende-se que o padrão de conexões do lobo frontal seja formado por podas baseadas em *experience dependent*.
- De fato, a atividade do córtex pré-frontal tende a ser até 4 vezes mais intensa em crianças do que em adultos, o que permite concluir que poda de parte das conexões e fortalecimento de outras contribuem para a maturação cognitiva.

CASEY, B.J., TOTTENHAM, N., LISTON, C. & DURSTON, S. (2005) "Imaging the developing brain: what have we learned about cognitive development?", Trends in Cognitive Science, vol. 9, no. 3, pp. 104-110.

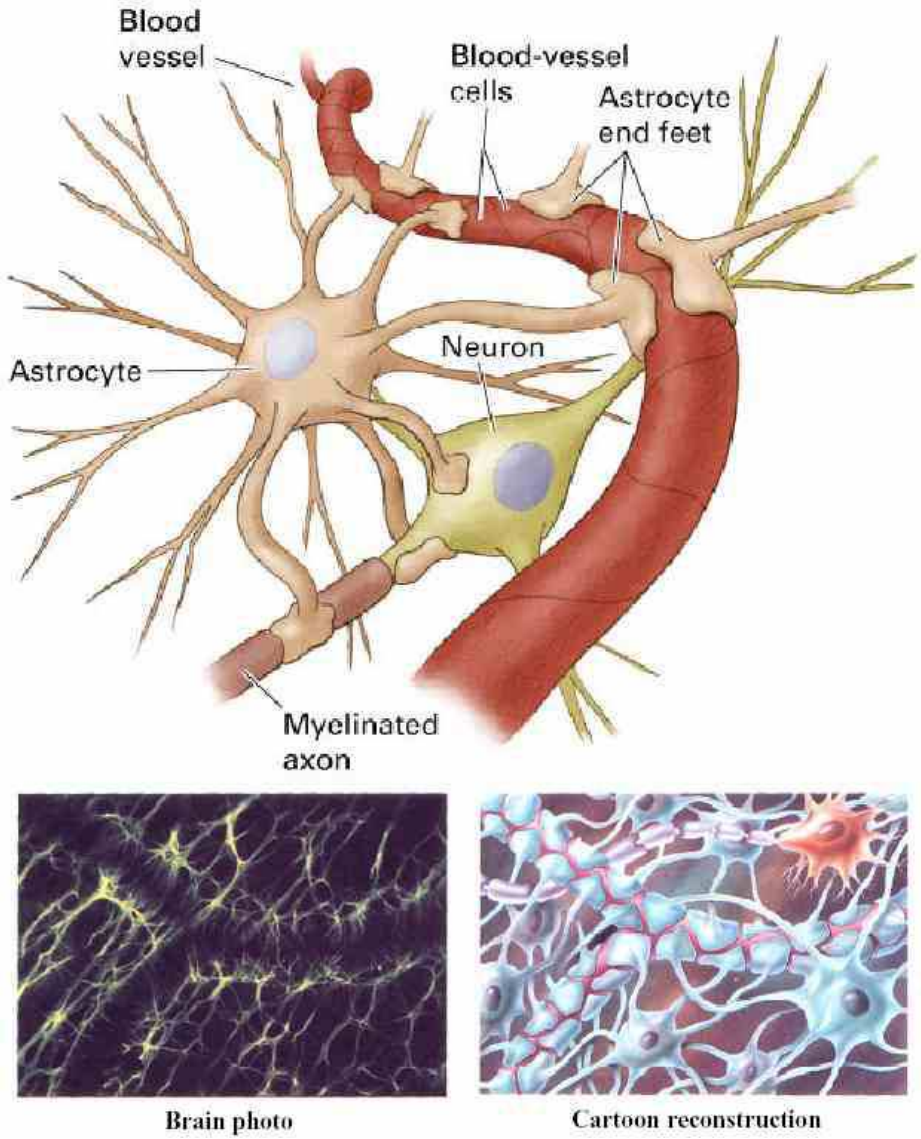
- Em síntese, é possível afirmar que o padrão de conexões no cérebro se inicia sem muita organização e com uma grande densidade de sinapses. Com a experiência de vida, um equilíbrio é atingido. Logo, como o padrão de conexões de um ser humano adulto é obtido a partir da experiência de vida, cada pessoa vai apresentar um padrão de conexões diferente, particularmente nas áreas especializadas em cognição.
- Por outro lado, o sistema sensório-motor em um adulto normal deve apresentar uma conformação similar a de outros adultos normais, visto que a poda nessas áreas é *experience expectant*.

FRANCO, A.R. (2009) "Resource Allocation of the human brain: a competitive equilibrium approach", Ph. D. Thesis, The University of New Mexico, Albuquerque, New Mexico, USA.

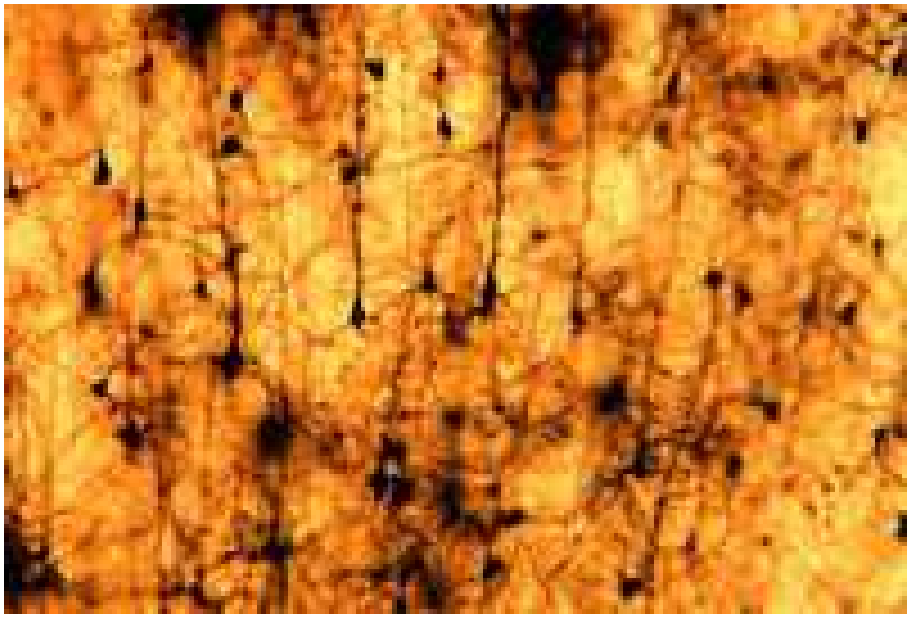
- Voltando agora a atenção para o neurônio biológico, pode-se afirmar que se trata de uma célula especializada em transmitir pulsos elétricos, sendo que as suas principais partes constituintes são:
  - ✓ Membrana celular: é a "pele" da célula;
  - ✓ Citoplasma: tudo que está envolvido pela membrana;

- ✓ Núcleo: contém os cromossomos (DNA);
- ✓ Ribossomos: geram proteínas a partir de mRNAs;
- ✓ Mitocôndria: gera energia para a célula (produz ATP);
- ✓ Soma: corpo celular, excluindo dendritos e axônio;
- ✓ Dendritos: parte do neurônio que recebe informação de outros neurônios;
- ✓ Axônio: parte do neurônio que transmite informação para outros neurônios;
- ✓ Bainha de mielina: revestimento externo lipídico do axônio, responsável por evitar a dispersão dos sinais elétricos, como uma capa isolante;
- ✓ Terminais pré-sinápticos: área do neurônio que armazena neurotransmissores, os quais são liberados por potenciais de ação.

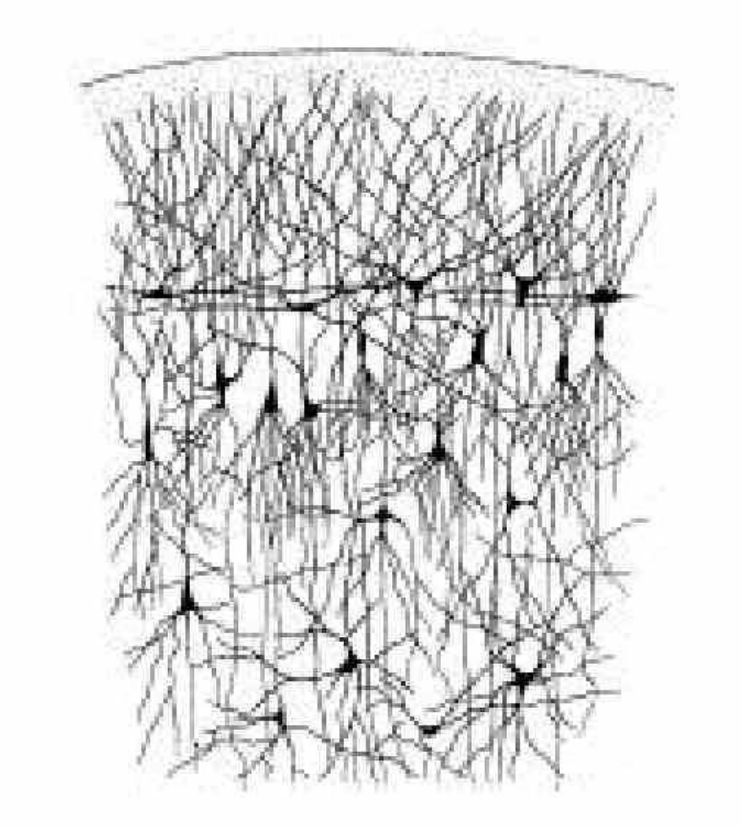
- Os neurônios sensoriais normalmente têm longos dendritos e axônios curtos. Por outro lado, os neurônios motores têm um longo axônio e dendritos curtos (transmitem informação para músculos e glândulas). Já os neurônios intrínsecos ou interneurônios realizam a comunicação neurônio-a-neurônio e compõem o sistema nervoso central.



- Além das células condutoras, o cérebro possui as células não-condutoras, formando a glia (neuróglia).
- Os astrócitos se caracterizam pela riqueza e dimensões de seus prolongamentos cito-plasmáticos, distribuídos em todas as direções. Funções: prover suporte estrutural, nutrientes e regulação química.
- Máxima distância de um neurônio a um vaso sanguíneo: ~50μm.



Rede de neurônios piramidais do córtex de um hamster



Desenho do córtex realizado por RAMÓN Y CAJÁL (1909)

- Hemisfério esquerdo: aprendizado verbal, memória verbal, processamento serial, processamento simbólico, linguagem, inferência, noção de tempo.
- Hemisfério direito: aprendizado espacial, memória espacial, processamento global (imagens), síntese da percepção, pensamento associativo, coordenação motora.
- O cérebro é capaz de perceber regularidades no meio e gerar abstrações que capturam a estrutura destas regularidades, possibilitando a predição de observações futuras e o planejamento de ações visando o atendimento de múltiplos objetivos.

- Organização básica do sistema nervoso (Visão de engenharia)

3. Alguns fatos históricos relevantes

- "Idade da Ilusão"
  - MCCULLOCH & PITTS (1943)
  - WIENER (1948): cibernética
  - MINSKY & PAPPERT (1969): a disputa entre as portas lógicas e os neurônios artificiais para determinar a unidade básica de processamento.
- "Idade das Trevas"
  - Entre 1969 e 1984, houve muito pouca pesquisa científica envolvendo redes neurais artificiais
- "Renascimento"
  - HOPFIELD (1982)
  - RUMELHART & MCCLELLAND (1986)
  - Desenvolvimento da capacidade de processamento e memória dos computadores digitais (simulação computacional / máquina virtual) (anos 80 e 90)

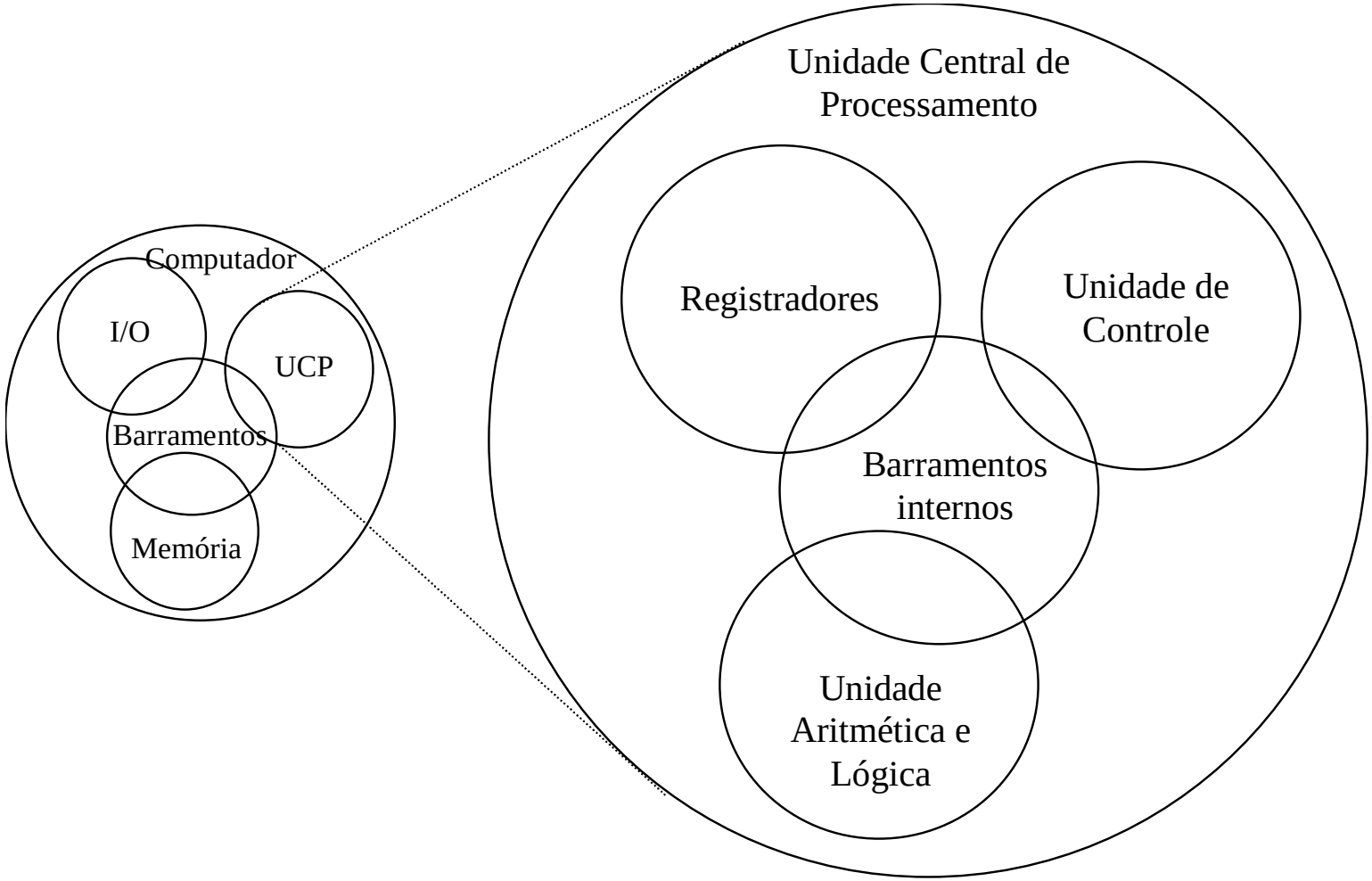
- GARDNER (1983): múltiplas inteligências

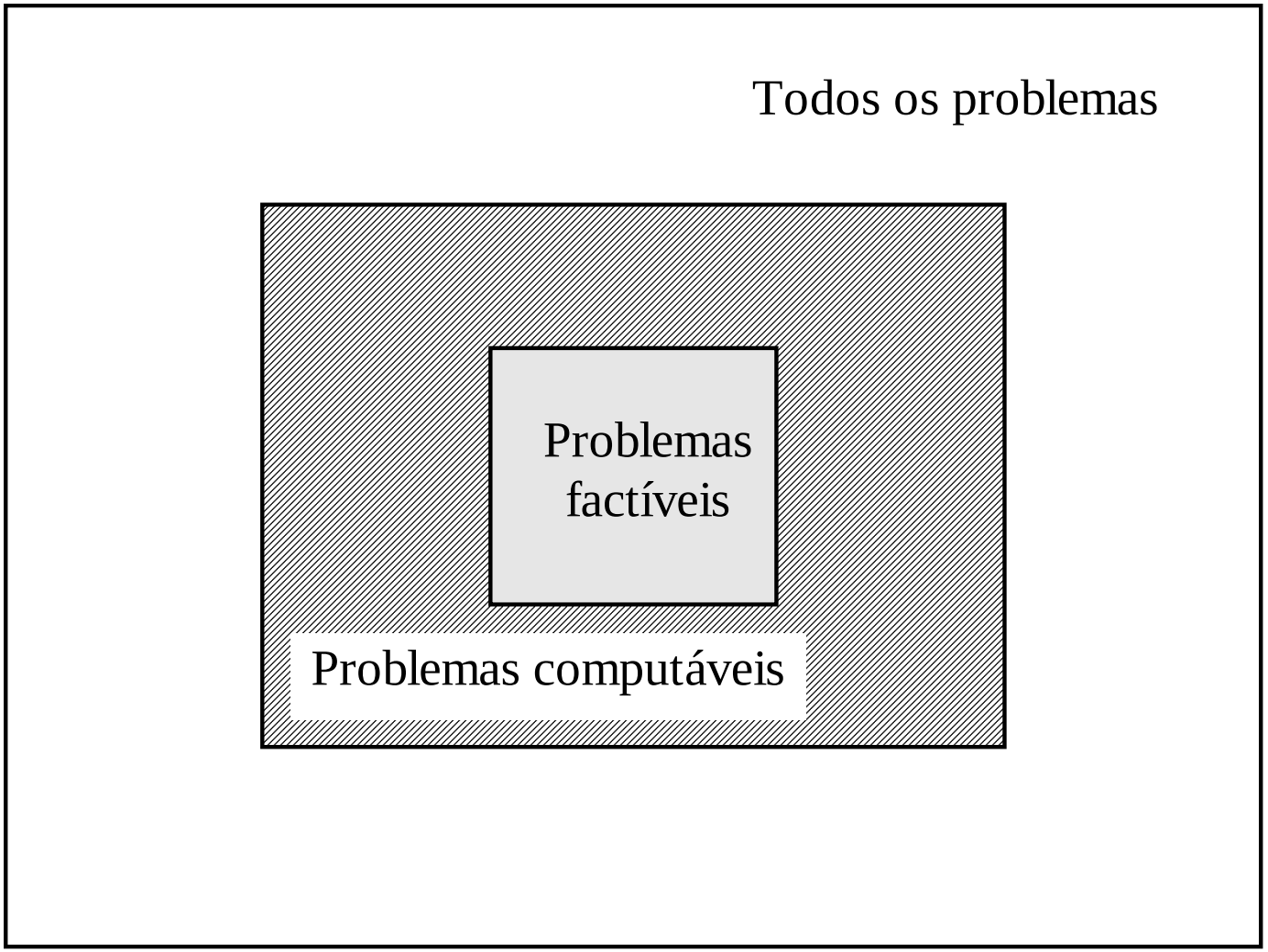
  1. Vivacidade verbal
  2. Vivacidade matemático-lógica
  3. Aptidão espacial
  4. Gênio cinestésico
  5. Dons musicais
  6. Aptidão interpessoal (liderança e ação cooperativa)
  7. Aptidão intrapsíquica (modelo preciso de si mesmo)

4. Computação digital × Neurocomputação (Conexionismo)

- É muito comum a associação do conceito de computação com aquele predominante no caso dos computadores com arquitetura do tipo Von Neumann: algoritmos são elaborados e em seguida implementados na forma de programas de computador a serem executados. No entanto, a computação realizada pelo cérebro requer um outro tipo de definição, que contemple processamento paralelo e distribuído, além de aprendizado.

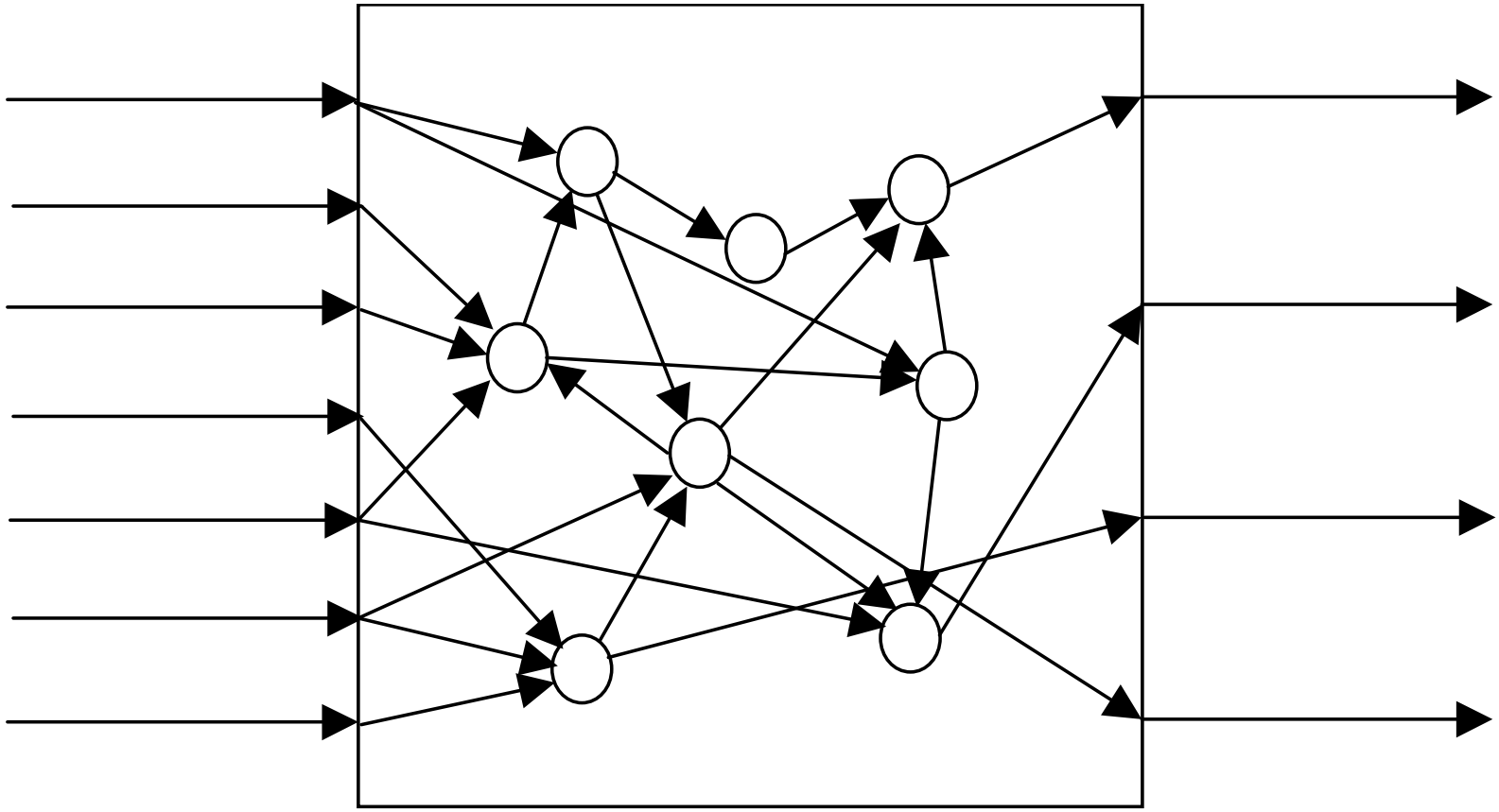
- Uma arquitetura neurocomputacional é baseada na interconexão de unidades de processamento simples e similares, denominadas neurônios artificiais e dotadas de grande poder de adaptação.
- Há uma diferença de paradigmas entre computadores com arquitetura do tipo Von Neumann e redes neurais artificiais (RNAs): os primeiros realizam processamento e armazenagem de dados em dispositivos fisicamente distintos, enquanto RNAs usam o mesmo dispositivo físico para tal.
- A motivação que está por trás deste paradigma alternativo de processamento computacional é a possibilidade de elaborar mecanismos distintos de solução para problemas intratáveis ou ainda não-resolvidos com base na computação convencional, além de criar condições para reproduzir habilidades cognitivas e de processamento de informação muito desejadas em aplicações de engenharia.
- Paradigmas *bottom-up* (Ex: Inteligência computacional) × Paradigmas *top-down* (Ex: Sistemas especialistas)


Computador
I/O
UCP
Barramentos
Memória
Unidade Central de Processamento
Registradores
Unidade de Controle
Barramentos internos
Unidade Aritmética e Lógica



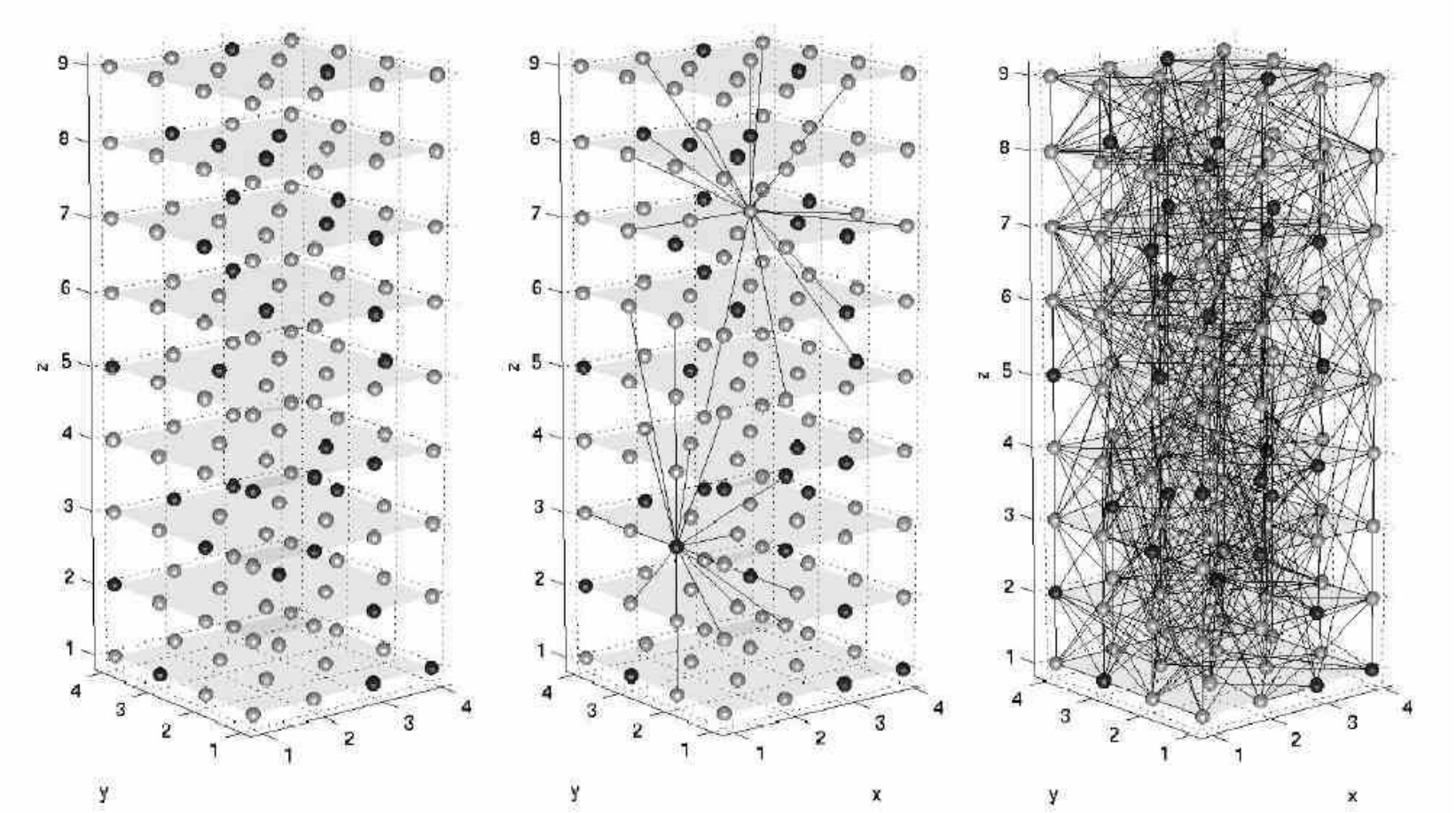
Cenário desafiador para a computação digital

Como abordar os problemas na região hachurada?



Exemplo genérico de um neurocomputador

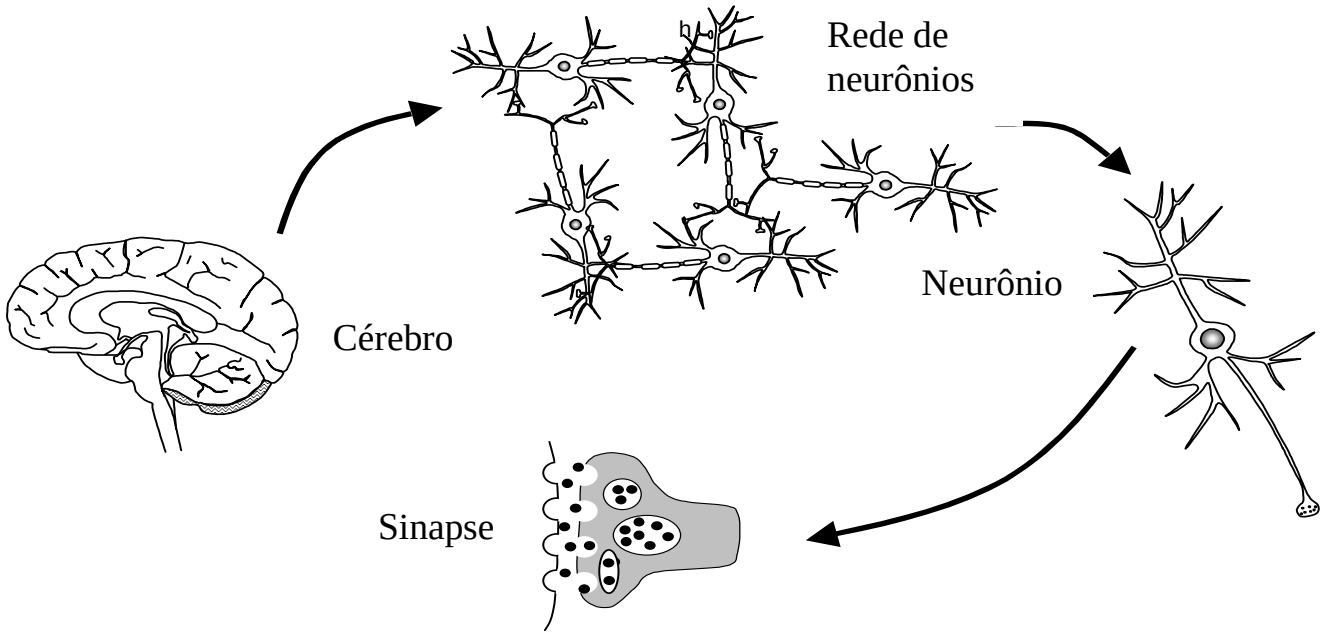
- Rede neural × Rede neuronal × Rede neurônica × Rede neuronial



Rede = nós + conexões (paradigma conexionista)

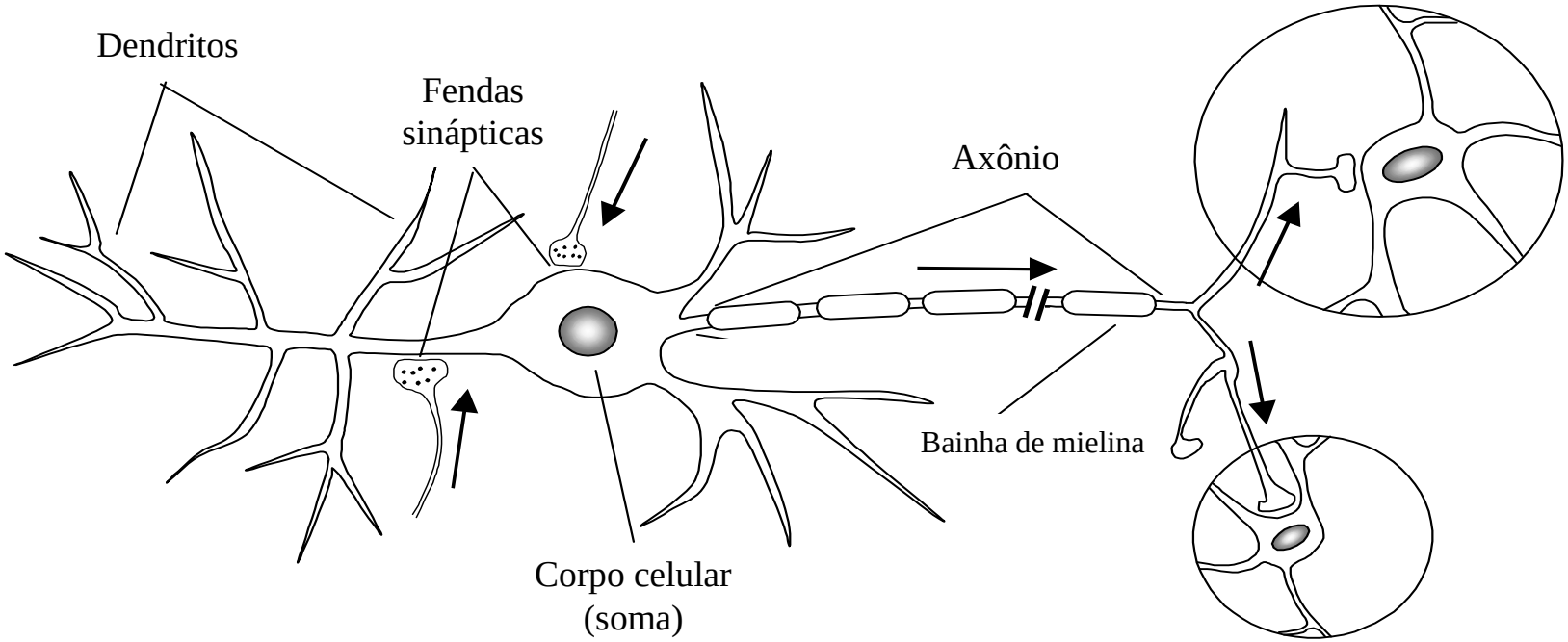
5. Níveis de Organização no Sistema Nervoso

- O sistema nervoso pode ser organizado em diferentes níveis: *moléculas*, *sinapses*, *neurônios*, *camadas*, *mapas* e *sistemas*.
- Uma estrutura facilmente identificável no sistema nervoso é o neurônio, especialista em processamento de sinais.
- Dependendo das condições de operação, os neurônios são capazes de gerar um *sinal*, mais especificamente um *potencial elétrico*, que é utilizado para transmitir informação a outras células.



- Os neurônios utilizam uma variedade de mecanismos bioquímicos para o processamento e transmissão de informação, incluindo os *canais iônicos*.
- Os canais iônicos são ativados por neurotransmissores e permitem um fluxo de entrada e saída de *íons*, gerando um *potencial de ação* (*spike*).
- O processo de transmissão de sinais entre neurônios é fundamental para a capacidade de processamento de informação do cérebro.
- Uma das descobertas mais relevantes em neurociência foi a de que a *efetividade* da transmissão de sinais pode ser modulada, permitindo que o cérebro se adapte a diferentes situações.
- A *plasticidade sináptica*, ou seja, a capacidade das sinapses sofrerem modificações, é o ingrediente chave para o aprendizado da maioria das RNAs.
- Os neurônios podem receber e enviar sinais de/para vários outros neurônios.

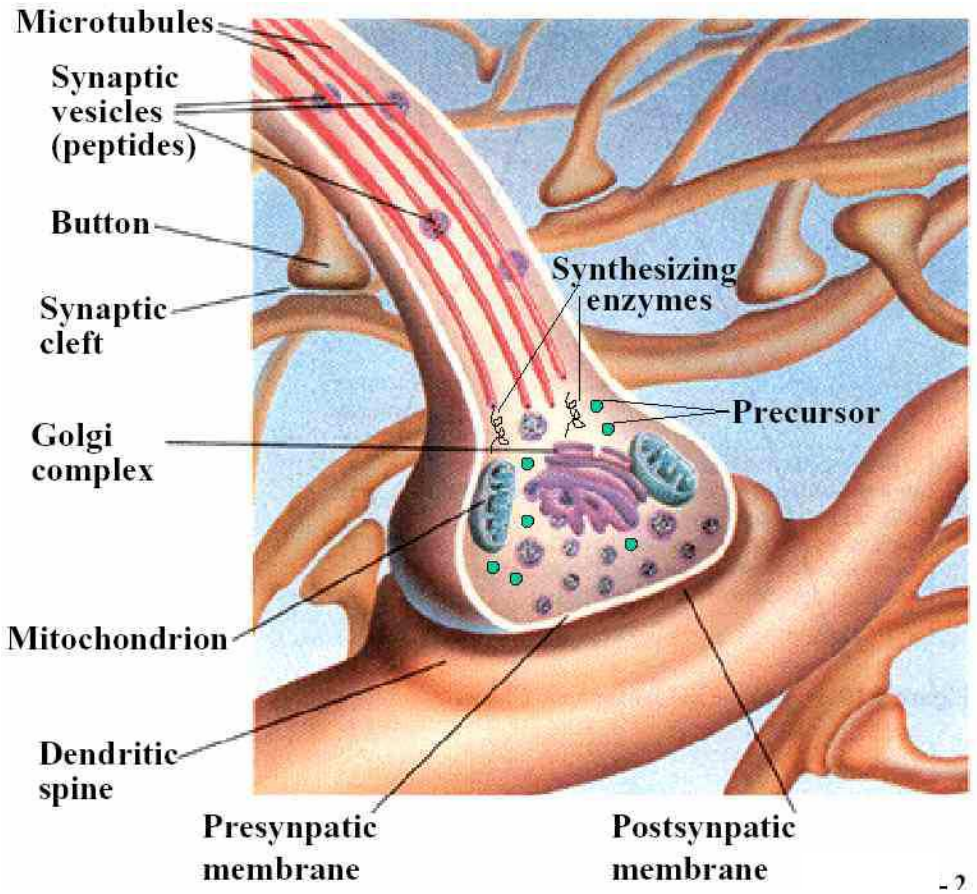
- Os neurônios que enviam sinais, chamados de neurônios *pré-sinápticos* ou "*enviadores*", fazem contato com os neurônios *receptores* ou *pós-sinápticos* em regiões especializadas, denominadas de *sinapses*.



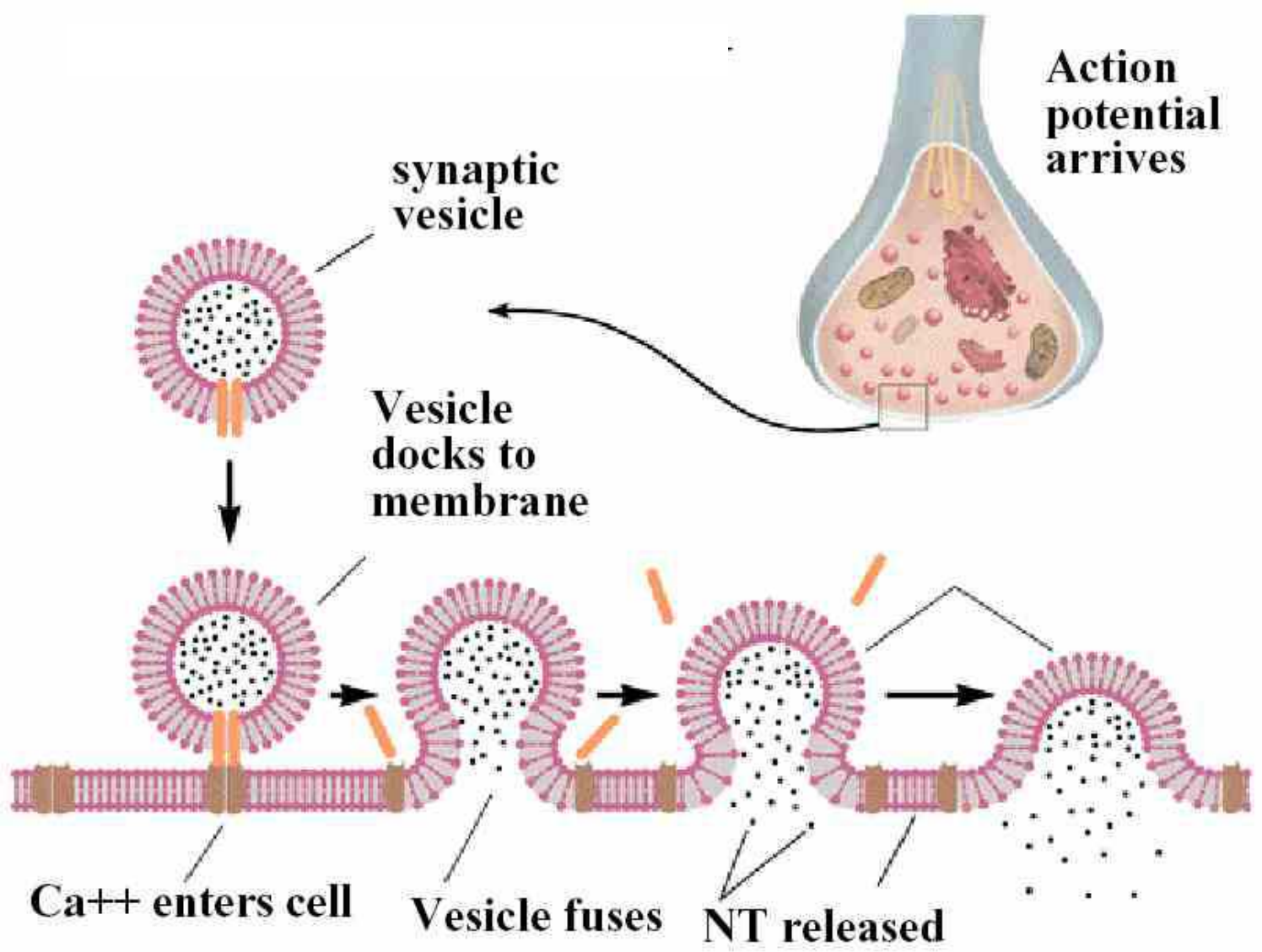
- A sinapse é, portanto, a junção entre o axônio de um neurônio pré-sináptico e o dendrito ou corpo celular de um neurônio pós-sináptico (ver figura acima).

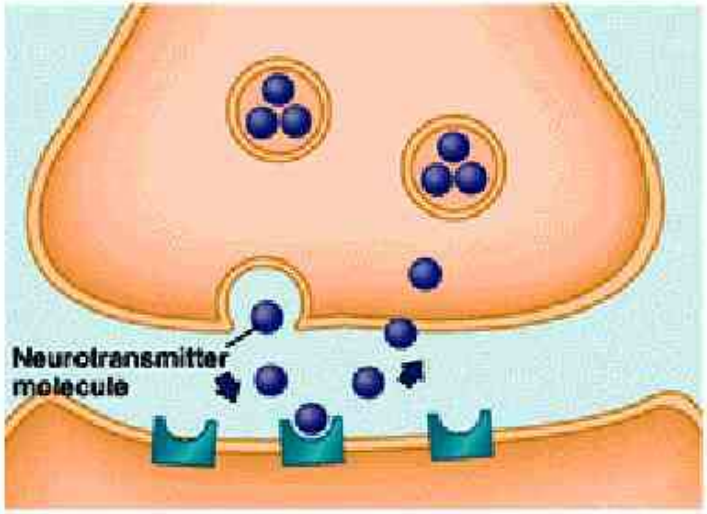
- A capacidade de processamento de informação das sinapses permite que elas alterem o estado de um neurônio pós-sináptico, eventualmente gerando um pulso elétrico, denominado *potencial de ação*, no neurônio pós-sináptico.
- Logo, um neurônio pode ser visto como um dispositivo capaz de receber estímulos (de entrada) de diversos outros neurônios e propagar sua única saída, função dos estímulos recebidos e do estado interno, a vários outros neurônios.
- Existem diversos mecanismos envolvidos na transmissão de informação (sinais) entre neurônios. Como os neurônios são células encapsuladas por membranas, pequenas aberturas nestas membranas (*canais*) permitem a transferência de informação entre eles.
- Os mecanismos básicos de processamento de informação são baseados no movimento de átomos carregados, ou *íons*:
  - ✓ Os neurônios habitam um ambiente líquido contendo uma certa concentração de íons, que podem entrar ou sair do neurônio através dos canais.

- ✓ Um neurônio é capaz de alterar o potencial elétrico de outros neurônios, denominado de *potencial de membrana*, que é dado pela diferença do potencial elétrico dentro e fora do neurônio.
- ✓ Quando um potencial de ação chega ao final do axônio, ele promove a liberação de neurotransmissores (substâncias químicas) na fenda sináptica, os quais se difundem e se ligam a receptores no neurônio pós-sináptico.
- ✓ Essa ligação entre neurotransmissores e receptores conduz à abertura dos canais iônicos, permitindo a entrada de íons na célula. A diferença de potencial resultante apresenta a forma de um pulso elétrico (*spike*).
- ✓ Esses pulsos elétricos se propagam pelo neurônio pós-sináptico até o corpo celular, onde são *integrados*. A ativação do neurônio pós-sináptico irá se dar apenas no caso do efeito resultante destes pulsos elétricos integrados ultrapassar um dado *limiar*.
- ✓ Alguns neurotransmissores possuem a capacidade de *ativar* um neurônio enquanto outros possuem a capacidade de *inibir* a ativação do neurônio.

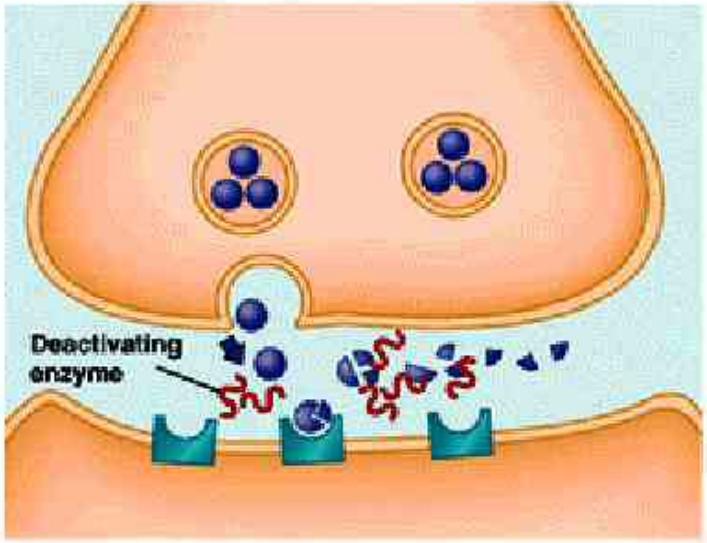


- A sinapse é uma fenda entre os terminais pré-sináptico e pós-sináptico, medindo ~20 nm.

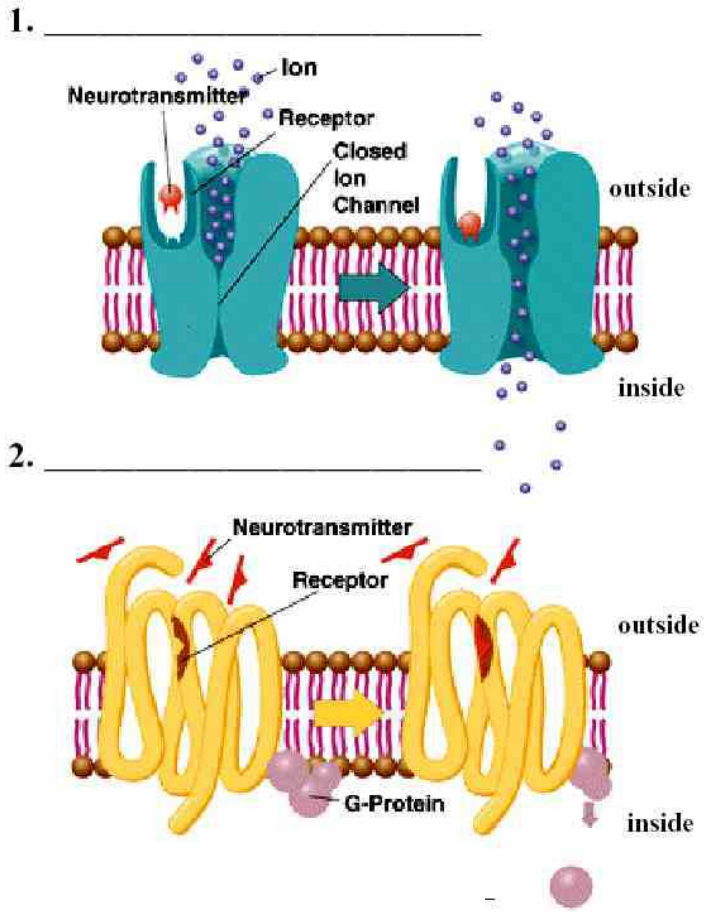

Action potential arrives
synaptic vesicle
Vesicle docks to membrane
Ca++ enters cell
Vesicle fuses
NT released



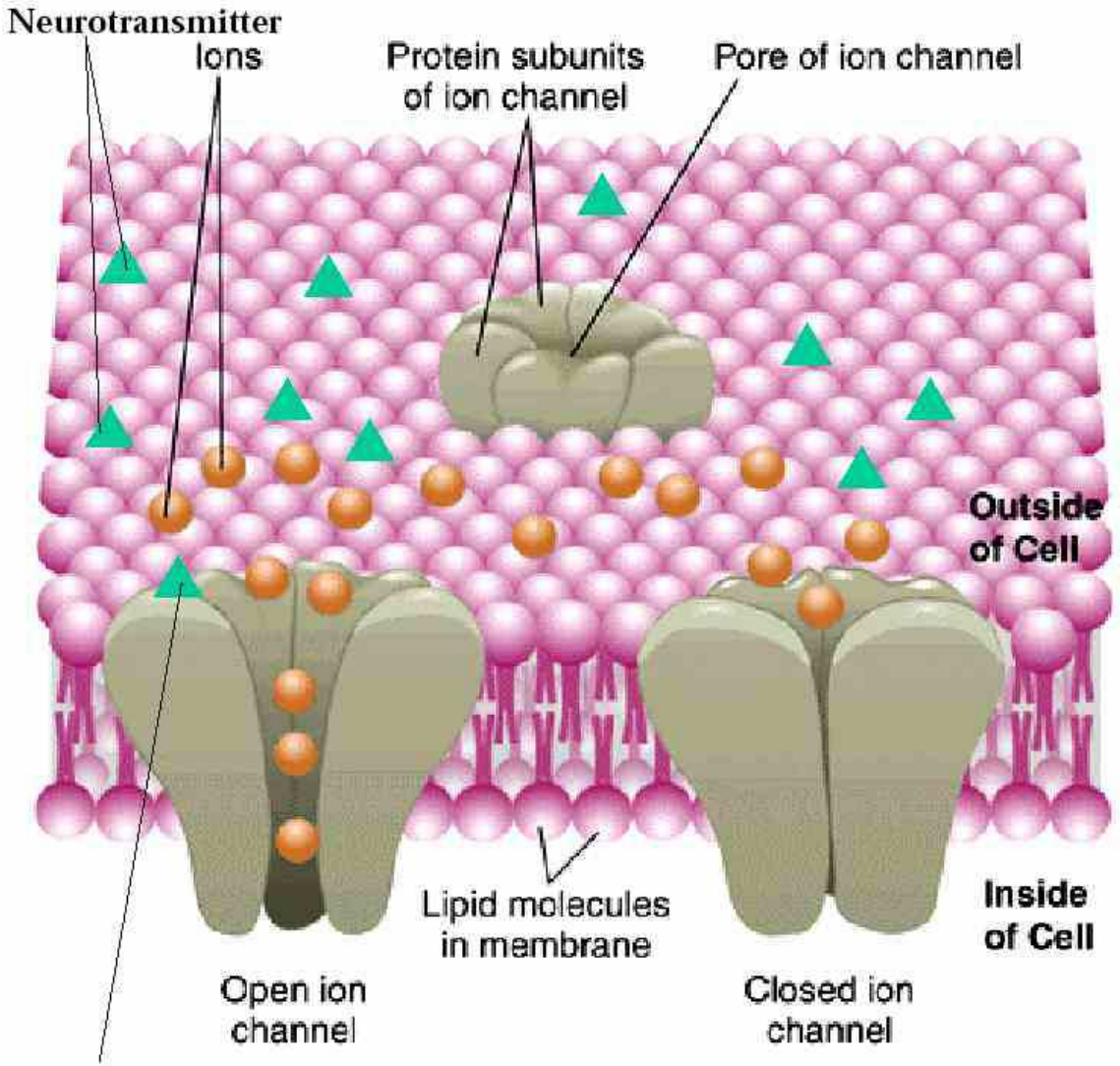
Reuptake



Deactivating Enzymes

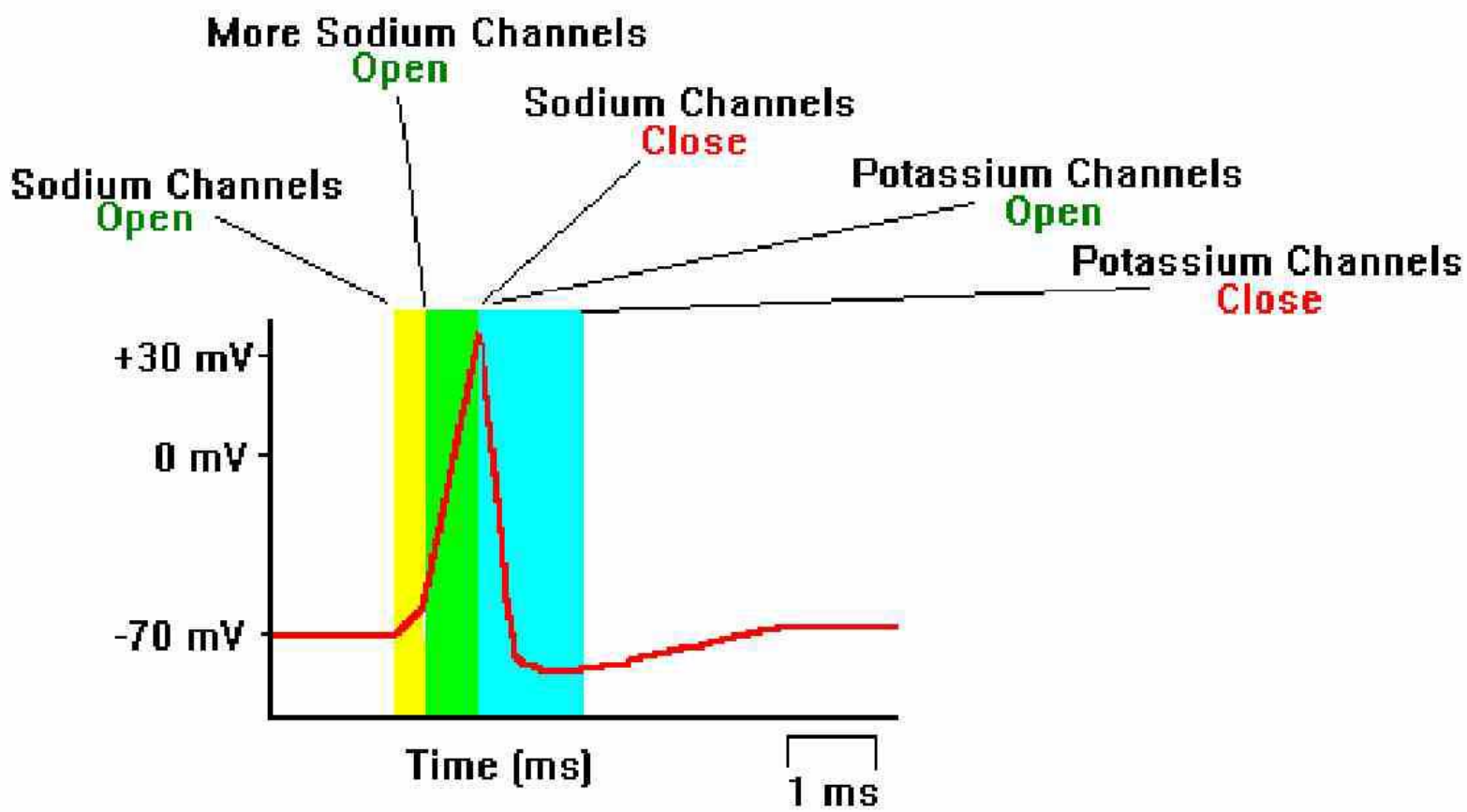


- Neurotransmissores putativos: serotonina, endorfina, dopamina, etc. Ao todo, são mais de 30 compostos orgânicos.
- O mal de Parkinson, por exemplo, é atribuído a uma deficiência de dopamina.



Binding of Neurotransmitter to ion channel

- A ativação de um neurônio é também denominada de *spiking*, *firing*, ou *disparo de um potencial de ação* (*triggering of an action potential*).

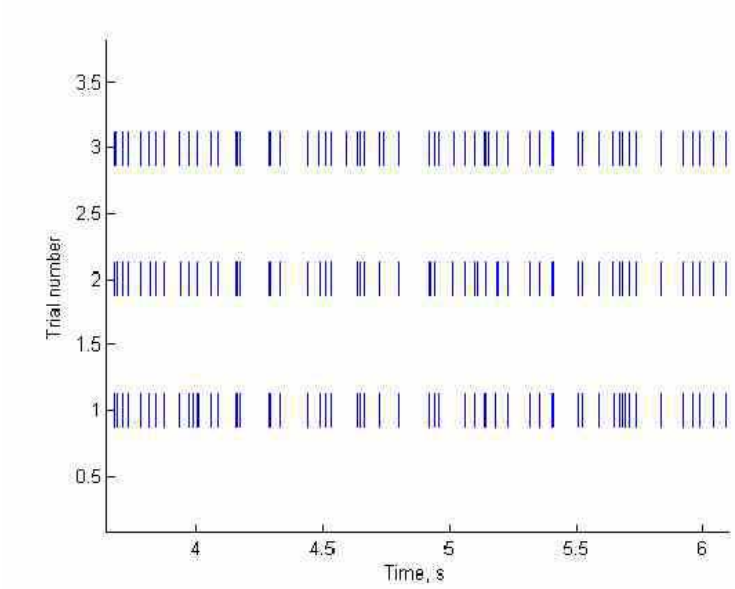


- Passos envolvidos no estabelecimento e extinção do potencial de ação:

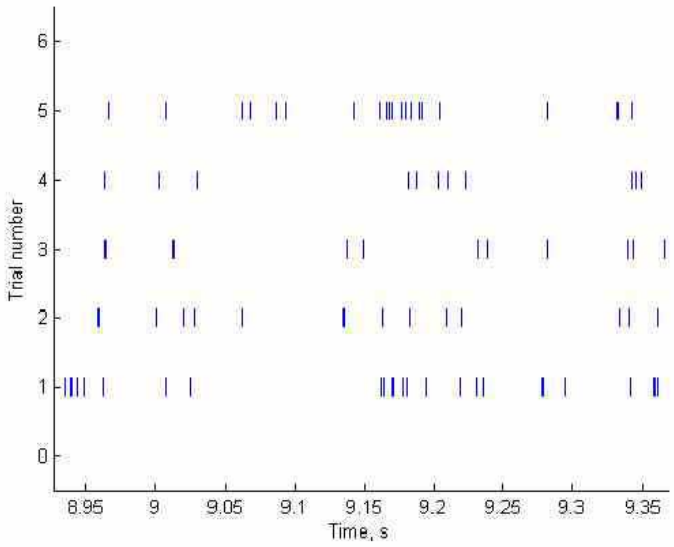
    1. Em uma célula em repouso, a parte externa da membrana é mais positiva que a parte interna, havendo mais íons de potássio dentro da célula e mais íons de sódio fora da célula.
    2. Pela ação dos neurotransmissores na sinapse, íons de sódio se movem para dentro da célula, causando uma diferença de potencial denominada potencial de ação. Com esta entrada de íons de sódio, o interior da célula passa a ser mais positivo que o exterior.
    3. Em seguida, íons de potássio fluem para fora da célula, restaurando a condição de interior mais negativo que exterior.
    4. Com as bombas de sódio-potássio, é restaurada finalmente a condição de maior concentração de íons de potássio dentro da célula e maior concentração de íons de sódio fora da célula.

- Segue-se um período refratário, durante o qual a membrana não pode ser estimulada, evitando assim a retropropagação do estímulo.

- Bombas de sódio e potássio: os íons de sódio que haviam entrado no neurônio durante a despolarização, são rebombeados para fora do neurônio mediante o funcionamento das bombas de sódio e potássio, que exigem gasto de energia.
- Para cada molécula de ATP empregada no bombeamento, 3 íons de sódio são bombeados para fora e dois íons de potássio são bombeados para dentro da célula. Esta etapa ocorre após a faixa azul da figura anterior.

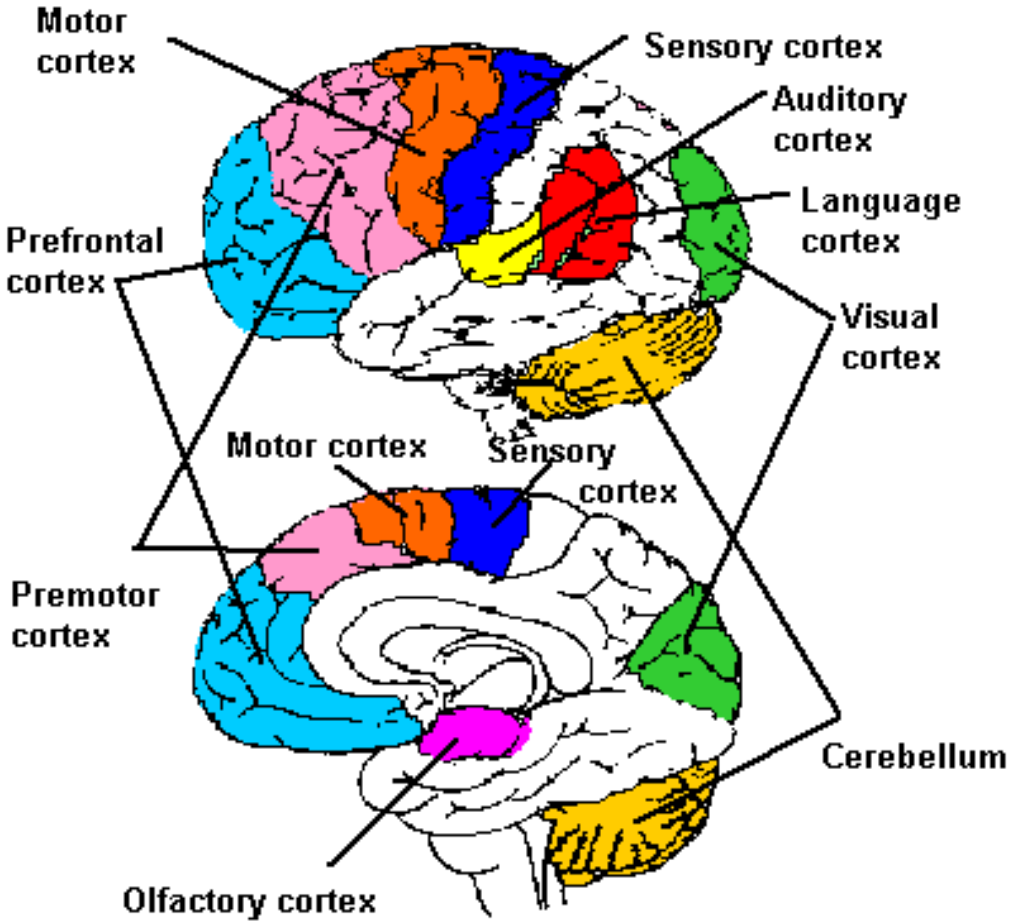


Neurônio periférico



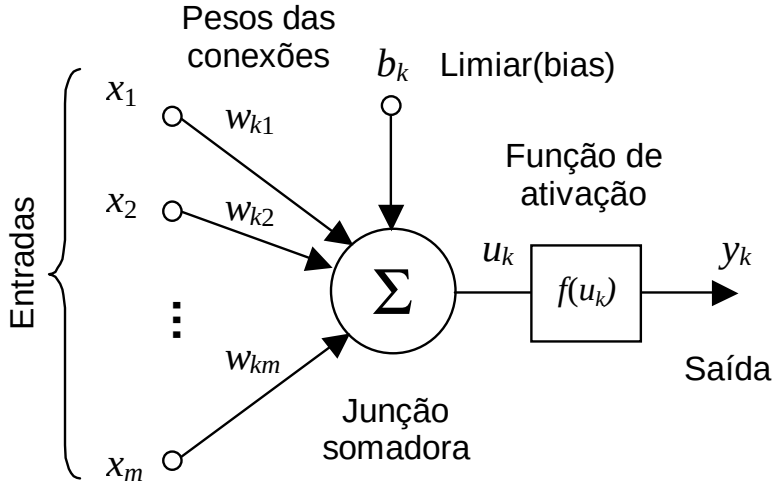
Neurônio do córtex

- O córtex corresponde à superfície externa do cérebro: uma estrutura bidimensional com vários dobramentos, fissuras e elevações.
- Diferentes partes do córtex possuem diferentes funções (ver figura abaixo).



6. Neurônio artificial

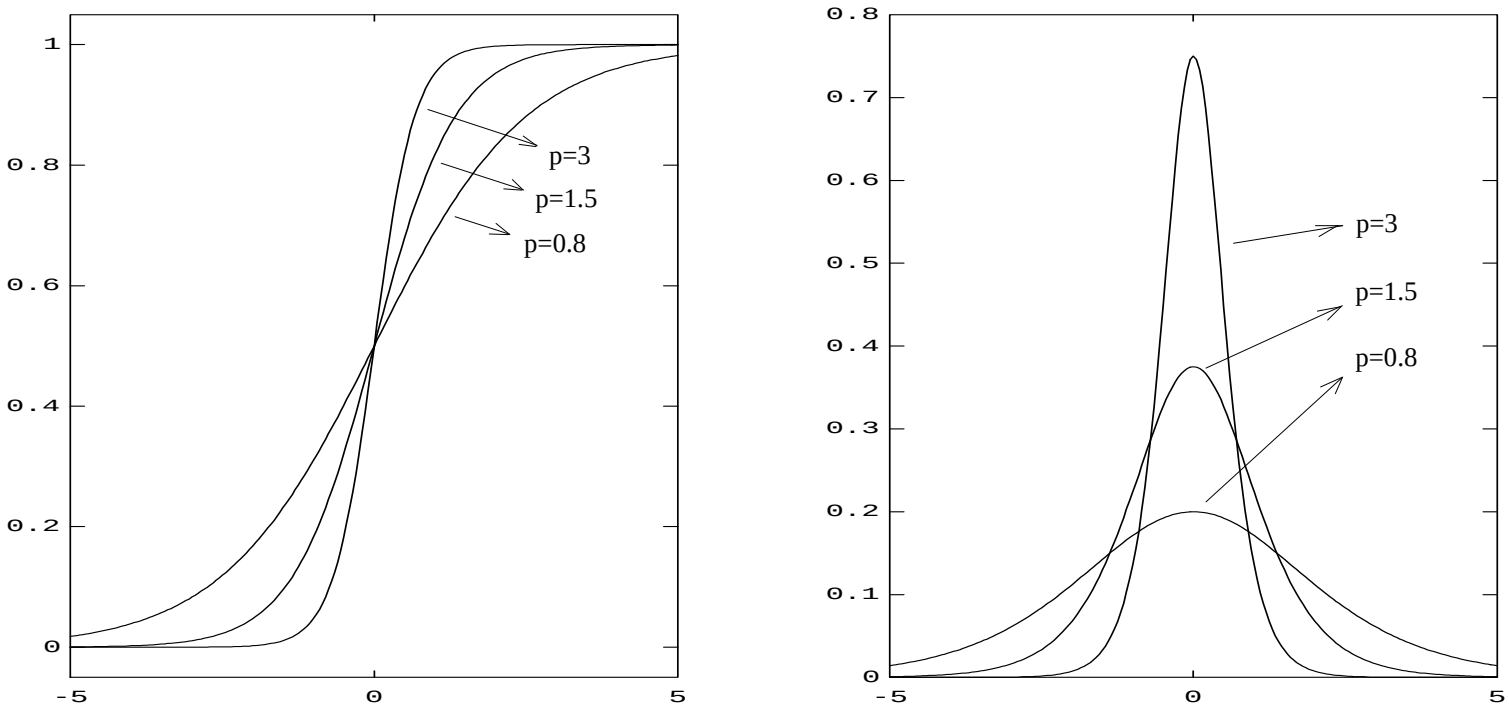
- Modelo matemático: Simplificações da realidade com o propósito de representar aspectos relevantes de um sistema em estudo, sendo que detalhes de menor significância são descartados para viabilizar a modelagem.



- A saída do neurônio *k* pode ser descrita por: $y_k = f(u_k) = f\left(\sum_{j=1}^{m} w_{kj} x_j + b_k\right)$
- É possível simplificar a notação acima de forma a incluir o bias simplesmente definindo um sinal de entrada de valor $x_0 = 1$ com peso associado $w_{k0} = b_k$:

$$y_k = f(u_k) = f\left(\sum_{j=0}^{m} w_{kj} x_j\right) = f\left(\mathbf{w}^T \mathbf{x}\right)$$

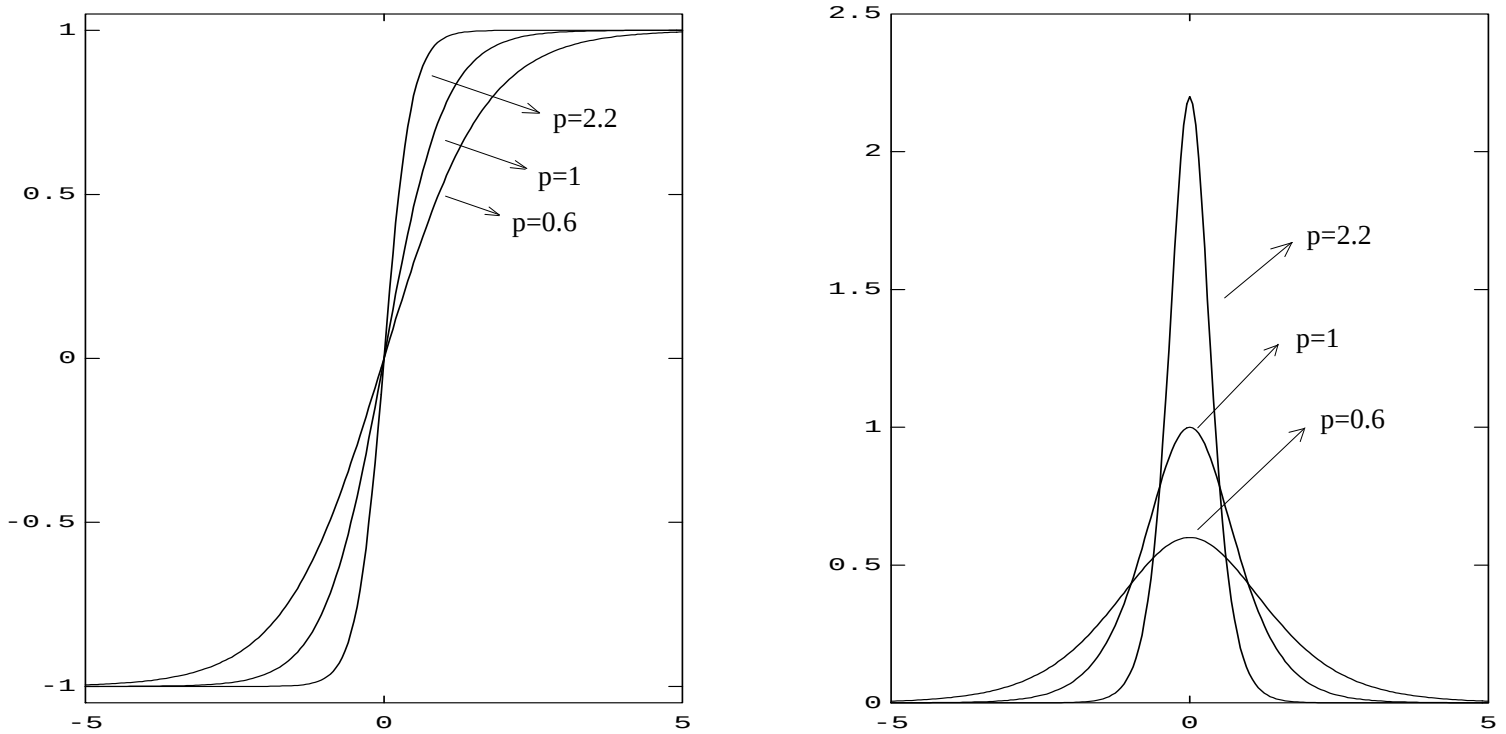
$$y = f(\mathbf{u}_k) = \frac{e^{p\mathbf{u}_k}}{e^{p\mathbf{u}_k} + 1} = \frac{1}{1 + e^{-p\mathbf{u}_k}} \qquad \frac{\partial y}{\partial \mathbf{u}_k} = p\mathbf{u}_k(1 - \mathbf{u}_k) > 0$$



a) **b)**

Função logística (a) e sua derivada em relação à entrada interna (b)

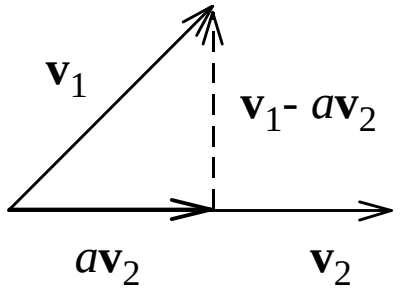
$$y = f(\mathbf{u}_k) = \tanh(p\mathbf{u}_k) = \frac{e^{p\mathbf{u}_k} - e^{-p\mathbf{u}_k}}{e^{p\mathbf{u}_k} + e^{-p\mathbf{u}_k}} \qquad \frac{\partial y}{\partial \mathbf{u}_k} = p\left(1 - \mathbf{u}_k^2\right) > 0$$



a) **b)**

Função tangente hiperbólica (a) e sua derivada em relação à entrada interna (b)

7. Produto interno, projeção e expansão ortogonal



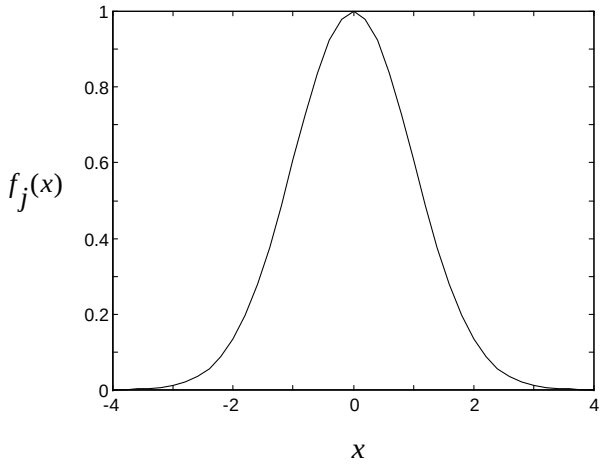
Projeção realizada pelo produto interno no $\Re^2$

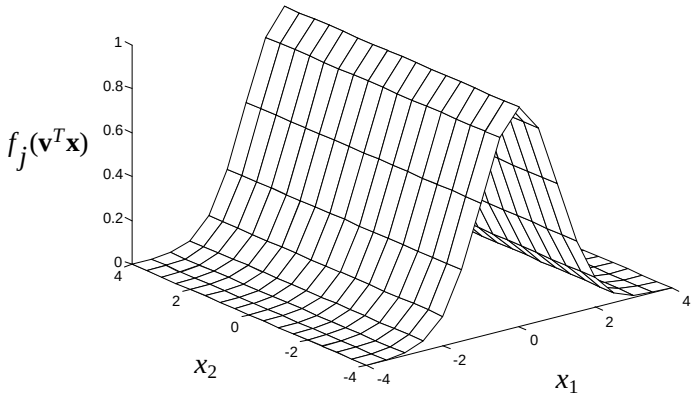
Sejam $\mathbf{v}_1, \mathbf{v}_2 \in \Re^2$ elementos não-nulos. Considere um escalar a tal que $a\mathbf{v}_2$ corresponda à projeção de $\mathbf{v}_1$ na direção de $\mathbf{v}_2$. Então, pode-se afirmar que

$$a\mathbf{v}_2 \perp \mathbf{v}_1 - a\mathbf{v}_2,$$

conduzindo a $\langle a\mathbf{v}_2, \mathbf{v}_1 - a\mathbf{v}_2 \rangle = 0$. Logo, $a\langle \mathbf{v}_2, \mathbf{v}_1 \rangle - a^2 \langle \mathbf{v}_2, \mathbf{v}_2 \rangle = 0$, permitindo obter a na forma: $a = \dfrac{\langle \mathbf{v}_2, \mathbf{v}_1 \rangle}{\langle \mathbf{v}_2, \mathbf{v}_2 \rangle}$. A projeção de $\mathbf{v}_1$ na direção de $\mathbf{v}_2$ ($\mathbf{v}_2 \neq \mathbf{0}$) assume a forma:

$$\text{proj}_{\mathbf{v}_2}(\mathbf{v}_1) = \frac{\langle \mathbf{v}_2, \mathbf{v}_1 \rangle}{\langle \mathbf{v}_2, \mathbf{v}_2 \rangle} \mathbf{v}_2$$





(a) $f_j(x) = e^{-0,5 \cdot x^2}$ (b) $f_j(\mathbf{v}^T\mathbf{x}) = e^{-0,5 \cdot \left([1\ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^2}$

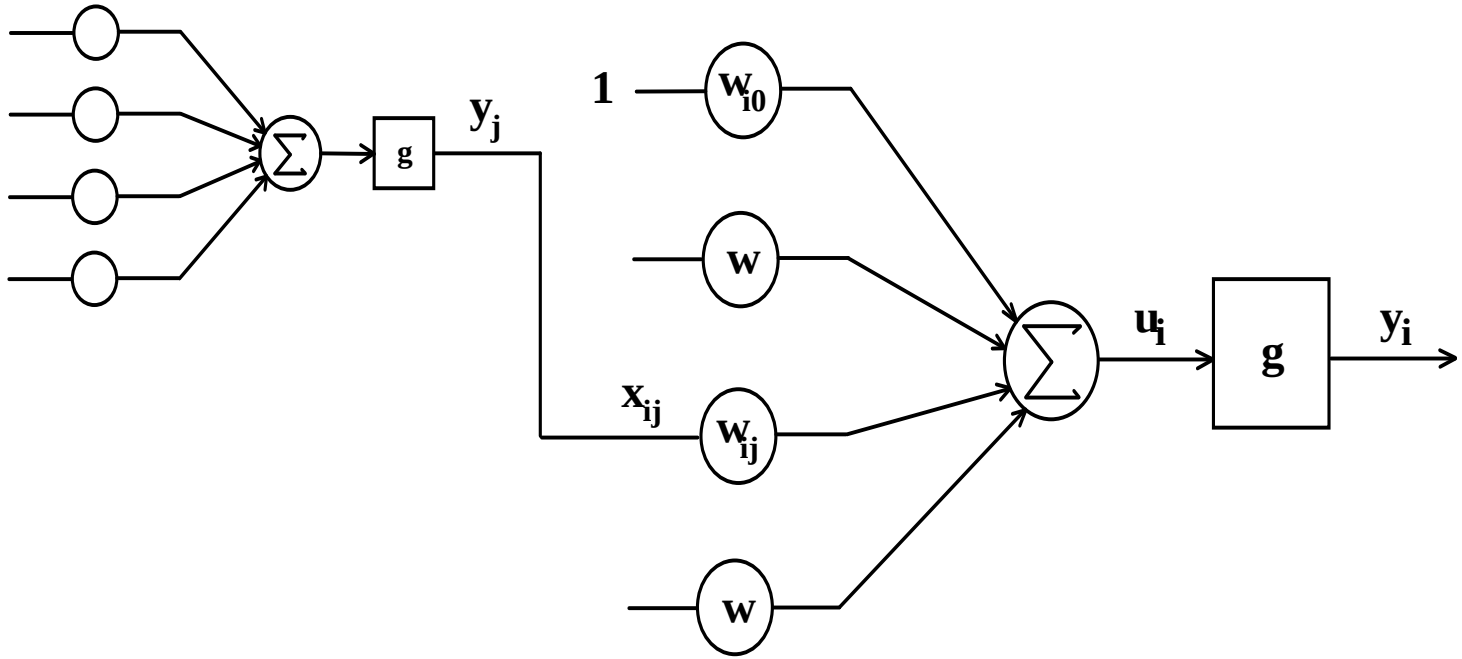
A função em (b) é uma expansão ortogonal da função em (a), em que

$$\mathbf{v} = [1\ 0]^T \text{ e } \mathbf{x} = [x_1\ \ x_2]^T$$

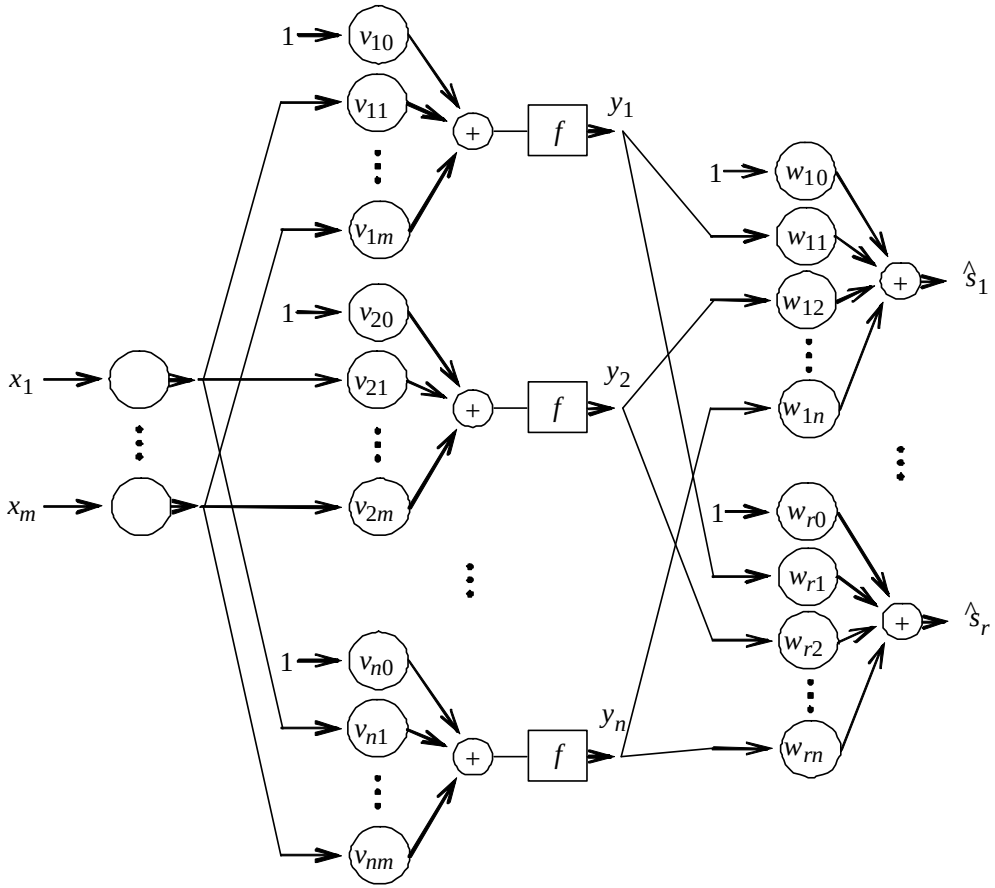
Funções como em (b) são também conhecidas como *ridge functions*.

8. Redes neurais e perceptron com uma camada intermediária

- O processo de conexão entre neurônios artificiais leva à geração de sinapses e à construção de redes neurais artificiais.



- As estruturas mais conhecidas são em camadas, onde a saída de cada neurônio de uma camada precedente é entrada para todos os neurônios da camada seguinte.



Rede neural perceptron com uma camada intermediária
Do inglês *Multilayer Perceptron* (MLP)

$$\hat{s}_k = \sum_{j=0}^{n} w_{kj}\, f\left(\sum_{i=0}^{m} v_{ji} x_i\right) = \sum_{j=0}^{n} w_{kj}\, f\left(\mathbf{v}_j^T \mathbf{x}\right) = \hat{g}_k\left(\mathbf{x}, \theta\right),\ k = 1,...,r$$

9. Superfície de erro

- Seja X uma região compacta do $\Re^m$ e seja g: $X \subset \Re^m \rightarrow \Re$ a função a ser aproximada (formulação para uma única saída, $r = 1$);
- O conjunto de dados de aproximação $\{(\mathbf{x}_l, s_l) \in \Re^m \times \Re\}_{l=1}^N$ é gerado considerando-se que os vetores de entrada $\mathbf{x}_l$ estão distribuídos na região compacta $X \subset \Re^m$ de acordo com uma função densidade de probabilidade fixa d_P: $X \subset \Re^m \rightarrow [0,1]$ e que os vetores de saída s_l são produzidos pelo mapeamento definido pela função g na forma:

$$s_l = g(\mathbf{x}_l) + \varepsilon_l, \quad l = 1,\ldots,N,$$

  onde $\varepsilon_l \in \Re$ é uma variável aleatória de média zero e variância fixa.
- A função g que associa a cada vetor de entrada $\mathbf{x} \in X$ uma saída escalar $s \in \Re$ pode ser aproximada com base no conjunto de dados de aproximação $\{(\mathbf{x}_l, s_l) \in \Re^m \times \Re\}_{l=1}^N$ por uma composição aditiva de funções de expansão ortogonal na forma:

$$\hat{s}_l = \hat{g}(\mathbf{x}_l, \theta) = \sum_{j=0}^{n} w_j \, f\left(\sum_{i=0}^{m} v_{ji} x_{li} \right) = \sum_{j=0}^{n} w_j \, f\left(\mathbf{v}_j^T \mathbf{x}_l \right)$$

onde θ é o vetor contendo todos os pesos da rede neural.

- Logo, o erro quadrático médio produzido na saída da rede neural, considerando as *N* amostras, assume a forma:

$$J(\theta) = \frac{1}{N} \sum_{l=1}^{N} (\hat{s}_l - s_l)^2 = \frac{1}{N} \sum_{l=1}^{N} (\hat{g}(\mathbf{x}_l, \theta) - s_l)^2 =$$

$$= \frac{1}{N} \sum_{l=1}^{N} \left(\sum_{j=0}^{n} w_j \, f\left(\sum_{i=0}^{m} v_{ji} x_{li} \right) - s_l \right)^2 = \frac{1}{N} \sum_{l=1}^{N} \left(\sum_{j=0}^{n} w_j \, f\left(\mathbf{v}_j^T \mathbf{x}_l \right) - s_l \right)^2$$

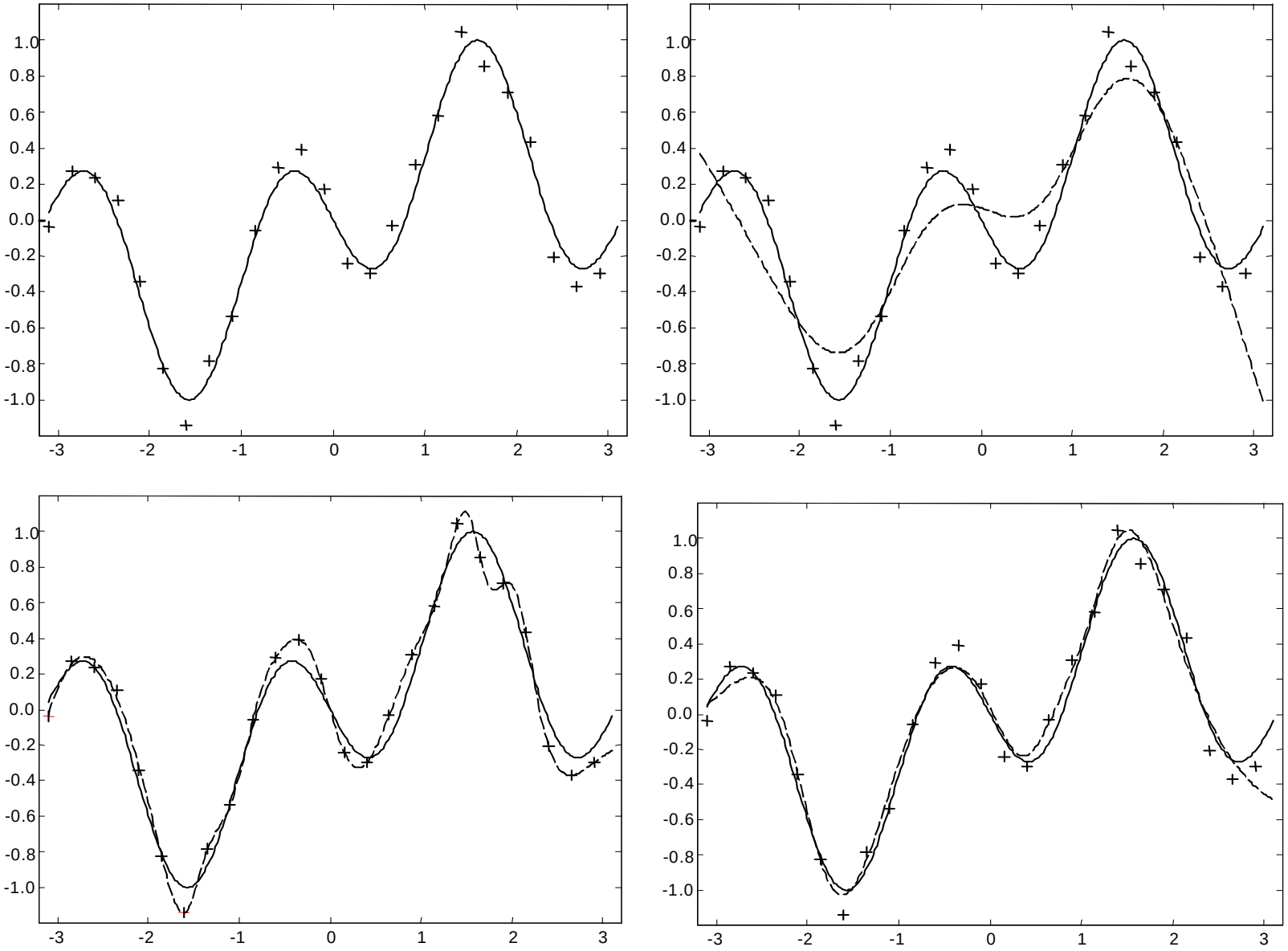
- Sendo *P* a dimensão do vetor θ, então tem-se que: $J : \Re^P \rightarrow \Re^1$.
- A superfície de erro definida por $J(\theta)$ reside no espaço $\Re^{P+1}$, sendo que deve-se buscar em $\Re^P$ um ponto que minimiza $J(\theta)$, supondo que se queira minimizar o erro entre a saída produzida pelo rede neural e a saída desejada.

10. Otimização não-linear irrestrita e capacidade de generalização

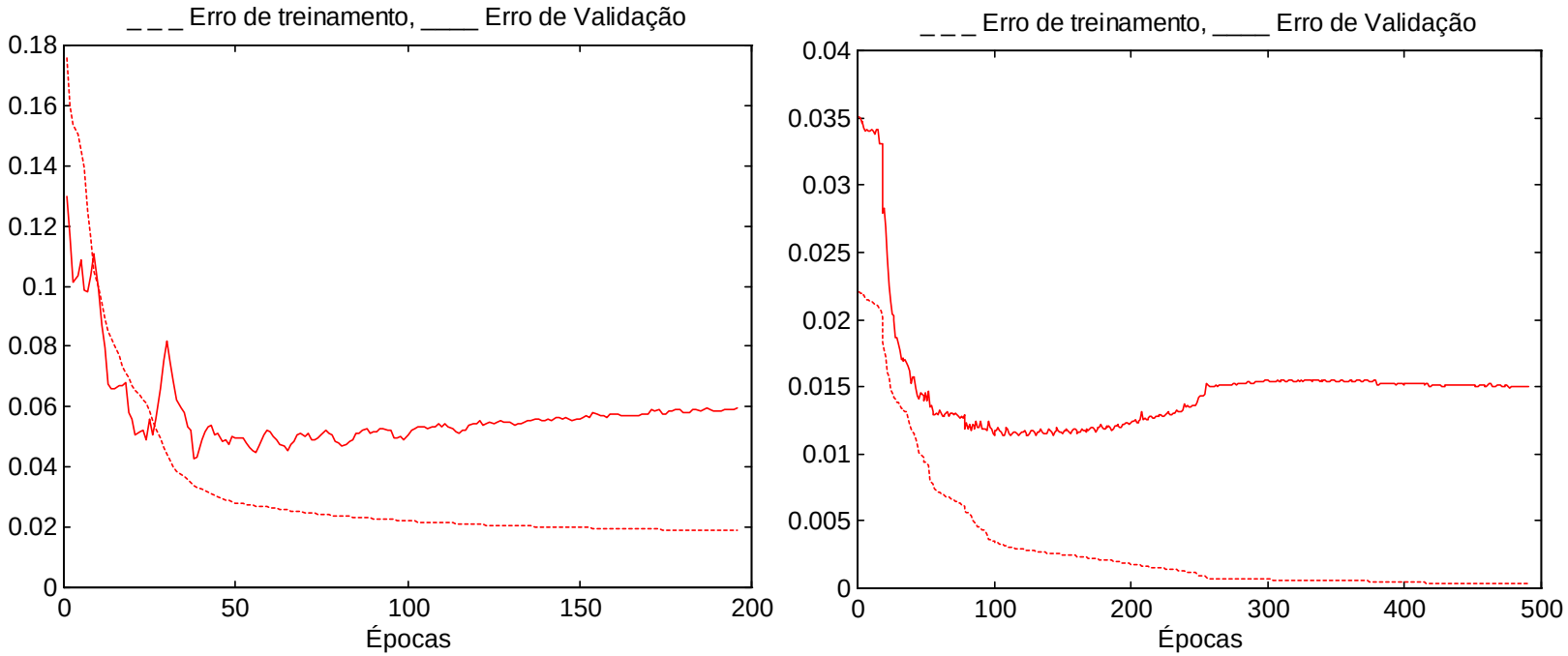
- Diversos tipos de parâmetros da rede neural poderiam ser submetidos a processos de ajuste durante o treinamento, como (*i*) pesos sinápticos; (*ii*) parâmetros da função de ativação de cada neurônio; (*iii*) número de neurônios na camada intermediária; (*iv*) número de camadas intermediárias.
- Iremos nos restringir aqui ao ajuste dos pesos sinápticos. Neste caso, o processo de treinamento supervisionado de redes neurais artificiais multicamadas é equivalente a um problema de otimização não-linear irrestrita, em que a superfície de erro é minimizada a partir do ajuste dos pesos sinápticos.
- Iremos nos restringir também a perceptrons com uma única camada intermediária, visto que com apenas uma camada intermediária a rede neural já apresenta **capacidade de aproximação universal** (CYBENKO, 1989; HORNIK *et al.*, 1989; HORNIK *et al.*, 1990; HORNIK *et al.*, 1994).

- Um problema comum a todos os modelos de aproximação de funções que possuem capacidade de aproximação universal, não apenas redes neurais artificiais do tipo MLP, é a necessidade de controlar adequadamente o seu grau de flexibilidade.
- Como o conjunto de amostras disponível para treinamento supervisionado é finito, infinitos mapeamentos podem produzir o mesmo desempenho de aproximação, independente do critério de desempenho adotado. Esses mapeamentos alternativos vão diferir justamente onde não há amostras disponíveis para diferenciá-los.
- Visando maximizar a **capacidade de generalização** do modelo de aproximação (no caso, uma rede neural MLP), ou seja, buscando encontrar o grau de flexibilidade adequado para o modelo de aproximação (dada a demanda da aplicação), um procedimento recomendado é dividir o conjunto de amostras disponível para treinamento em dois: um conjunto que será efetivamente empregado no ajuste dos pesos (conjunto de treinamento) e um conjunto que será empregado para definir o momento de interromper o treinamento (conjunto de validação).

- Deve-se assegurar que ambos os conjuntos sejam suficientemente representativos do mapeamento que se pretende aproximar. Assim, minimizar o erro junto ao conjunto de validação implica em maximizar a capacidade de generalização. Logo, espera-se que a rede neural que minimiza o erro junto ao conjunto de validação (não usado para o ajuste dos pesos) tenha o melhor desempenho possível junto a novas amostras.
- A figura alto/esquerda a seguir mostra um mapeamento unidimensional a ser aproximado (desconhecido pela rede neural) e amostras sujeitas a ruído de média zero (única informação disponível para o treinamento da rede neural). A figura alto/direita mostra o resultado da aproximação produzida por uma rede neural com muito poucos neurônios, a qual foi incapaz de realizar a aproximação (tem baixa flexibilidade). Já as figuras baixo/esquerda e baixo/direita mostram o resultado de uma mesma rede neural (com número suficiente de neurônios), mas à esquerda ocorreu sobre-treinamento, enquanto que à direita o treinamento foi interrompido quando minimizou-se o erro junto a dados de validação (não apresentados).


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-3
-2
-1
0
1
2
3

- Curvas típicas de erro de treinamento e validação são apresentadas a seguir.



- No entanto, nem sempre o erro de validação apresenta este comportamento, e cada caso deve ser analisado isoladamente. Como a curva do erro de validação oscila bastante e esboça um comportamento pouco previsível, não é indicado desenvolver detectores automáticos de mínimos e encerrar o treinamento ali. O mais indicado é sobre-treinar a rede e armazenar os pesos associados ao mínimo do erro de validação.

- Não existe um consenso sobre como produzir a melhor partição do conjunto de dados, ou seja, sobre como dividi-lo de forma que possamos encontrar uma rede com a melhor capacidade de generalização em todos os casos. Uma sugestão de partida pode ser 80% das amostras para treinamento e 20% para validação.
- Quando as amostras correspondem a dados rotulados em problemas de classificação de padrões, procure respeitar a distribuição junto a cada classe.

10.1 Gradiente, hessiana e algoritmos de otimização

- Considere uma função contínua e diferenciável até 2a. ordem em todos os pontos do domínio de interesse, tal que: f: $\Re^n \rightarrow \Re$ e $\mathbf{x} \in \Re^n$
- Expansão em série de Taylor em torno do ponto $\mathbf{x}^* \in \Re^n$:

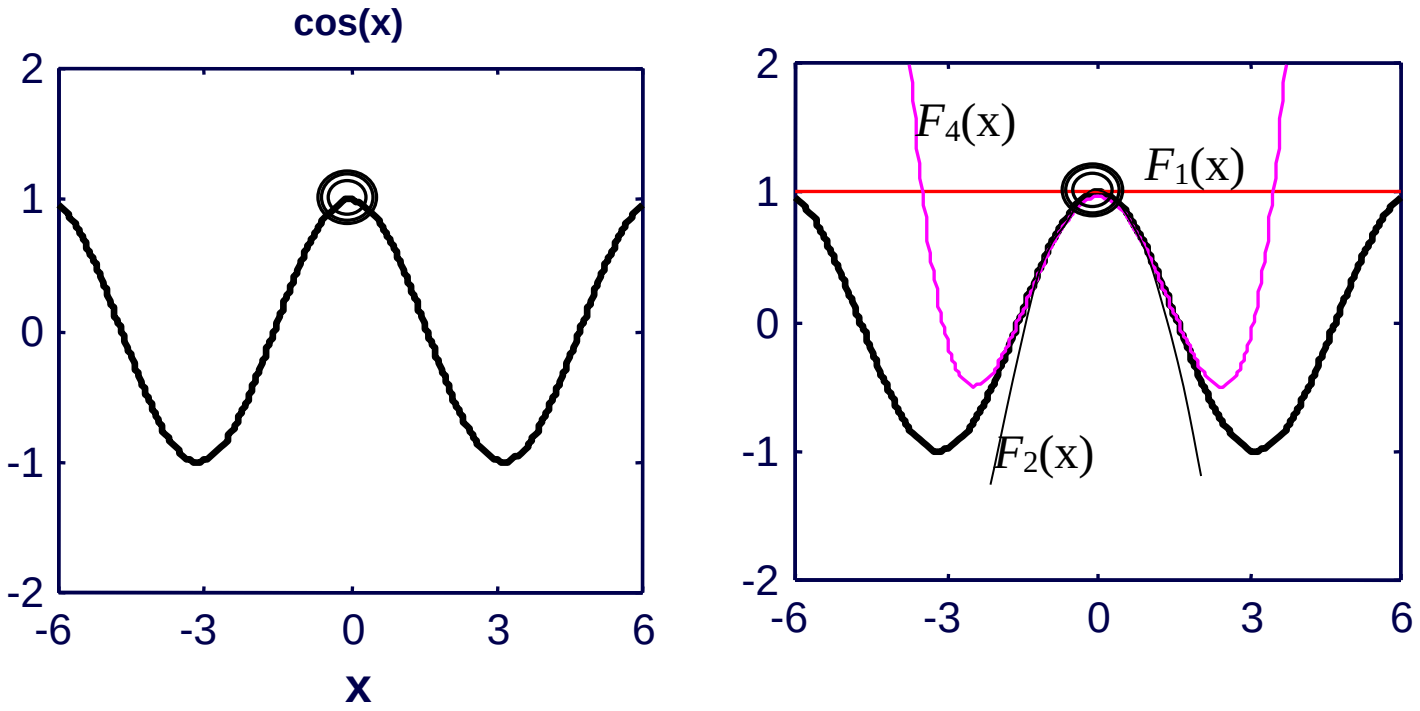
$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + O(3)$$

$$\nabla f(x^*) = \begin{bmatrix} \dfrac{\partial f(\mathbf{x}^*)}{\partial x_1} \\ \dfrac{\partial f(\mathbf{x}^*)}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\mathbf{x}^*)}{\partial x_n} \end{bmatrix} \qquad \nabla^2 f(\mathbf{x}^*) = \begin{bmatrix} \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_1^2} & \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_2^2} & & \vdots \\ & & \ddots & \\ \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_n \partial x_1} & \cdots & & \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x_n^2} \end{bmatrix}$$

Vetor gradiente **Matriz hessiana**

- O algoritmo de retropopagação do erro (do inglês *backpropagation*) é empregado para obter o vetor gradiente, onde cada elemento do vetor gradiente está associado a um peso da rede neural e indica o quanto a saída é influenciada por uma variação incremental neste peso.
- Existem várias técnicas para obter exatamente ou aproximadamente a informação de 2a. ordem em redes neurais MLP (BATTITI, 1992; BISHOP, 1992).

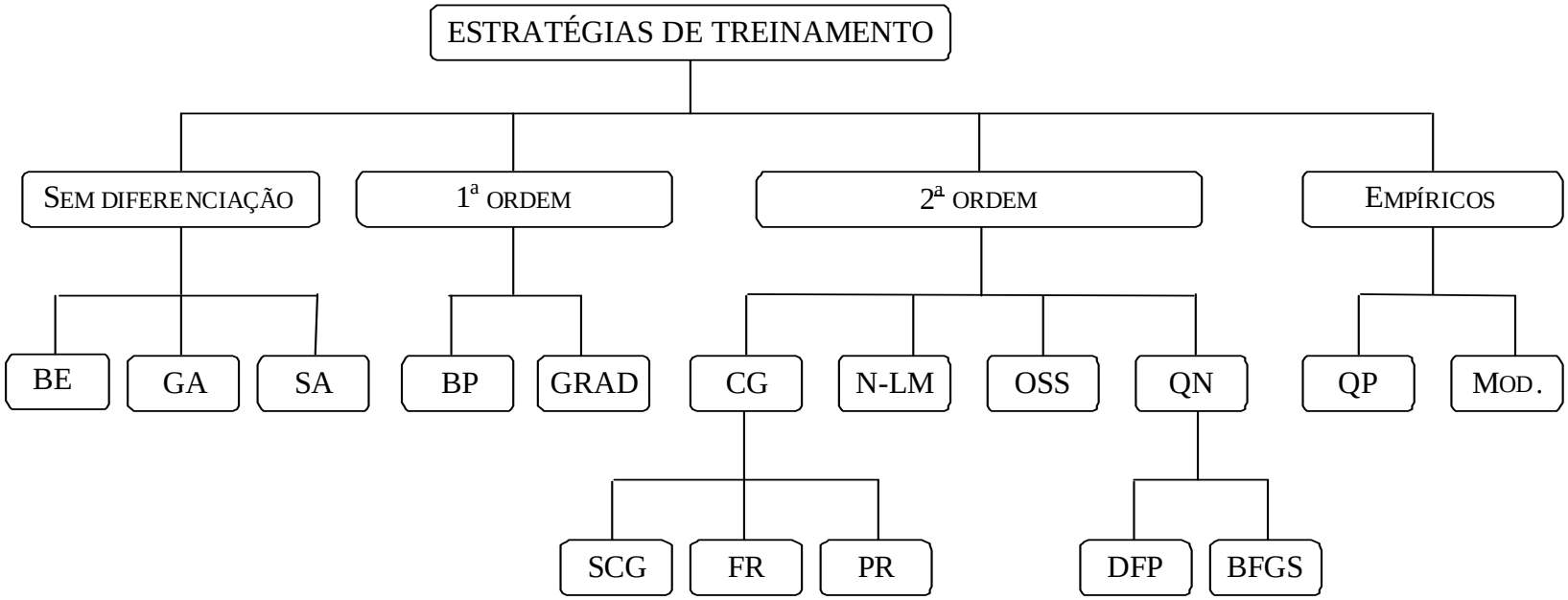
- Na função cos(*x*) a seguir, observam-se as aproximações de primeira, segunda e quarta ordem em torno do ponto *x* = 0.



- O processo de otimização não-linear envolvido no ajuste de pesos de uma rede neural vai realizar aproximações locais de primeira ordem ou de primeira e segunda ordem junto à superfície de erro e realizar ajustes incrementais e recursivos na forma:

$$\theta_{k+1} = \theta_k + \text{passo} * \text{direção}$$

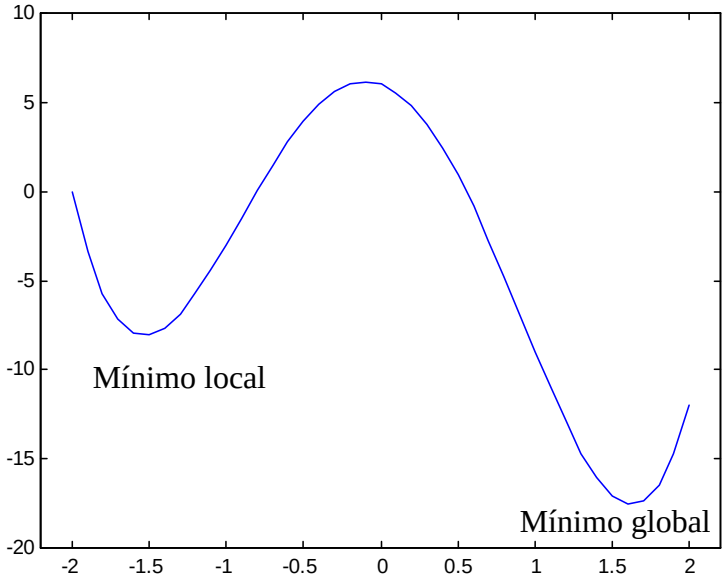
- Parte-se de uma condição inicial θ_0 e aplica-se iterativamente a fórmula acima, sendo que a direção depende da informação local de primeira e segunda ordem. Cada proposta de algoritmo de otimização vai diferir na forma de computar o *passo* e a *direção* de ajuste, a cada iteração.
- A figura a seguir apresenta uma classificação dos principais algoritmos empregados para o treinamento supervisionado de redes neurais artificiais.
- Aquele que é utilizado no toolbox fornecido pelo professor é o gradiente conjungado escalonado (do inglês *Scaled Conjugate Gradient* – SCG). Uma vantagem deste algoritmo é que ele apresenta um custo computacional (memória e processamento por iteração) linear com o número de pesos e não quadrático, como a maioria dos algoritmos de 2a. ordem.



10.2 Mínimos locais

- Como o processo de ajuste é iterativo e baseado apenas em informações locais, os algoritmos de otimização geralmente convergem para o mínimo local mais próximo, que pode representar uma solução inadequada (com nível de erro acima do aceitável).

- Repare que algoritmos de 2a. ordem tendem a convergir mais rápido para os mínimos locais, mas não se pode afirmar que eles convergem para mínimos de melhor qualidade que aqueles produzidos pelos algoritmos de primeira ordem.



10.3 Condição inicial para os pesos da rede neural

- Embora existam técnicas mais elaboradas, os pesos da rede neural podem ser inicializados com valores pequenos e aleatoriamente distribuídos em torno de zero.

11. Rede neural com função de ativação de base radial

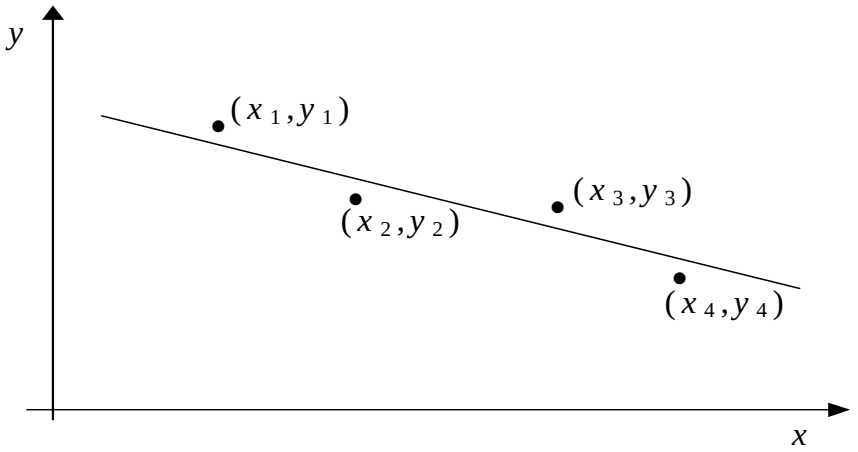
11.1 Regressão paramétrica e não-paramétrica

- Em problemas paramétricos, os parâmetros livres, bem como as variáveis dependentes e independentes, geralmente têm uma interpretação física.
- Exemplo: ajuste de uma reta a uma distribuição de pontos

$f(x) = y = ax + b$

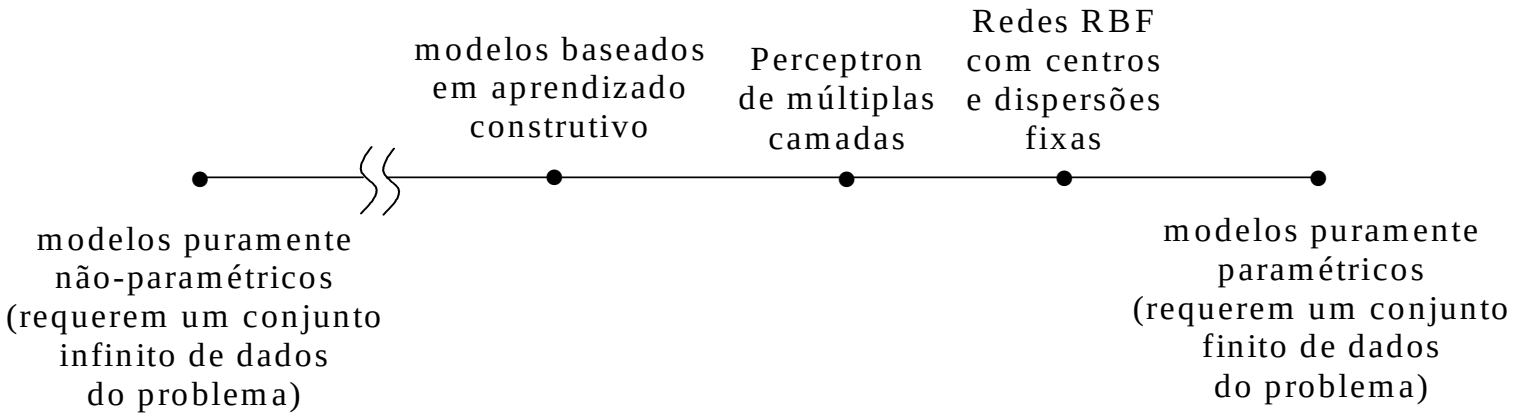
a,b desconhecidos

y: sujeito a ruído



- Regressão não-paramétrica: sua característica distintiva é a ausência (completa ou quase completa) de conhecimento a priori a respeito da forma da função que está sendo estimada. Sendo assim, mesmo que a função continue a ser estimada a partir do

ajuste de parâmetros livres, o conjunto de "formas" que a função pode assumir (classe de funções que o modelo do estimador pode prever) é muito amplo.



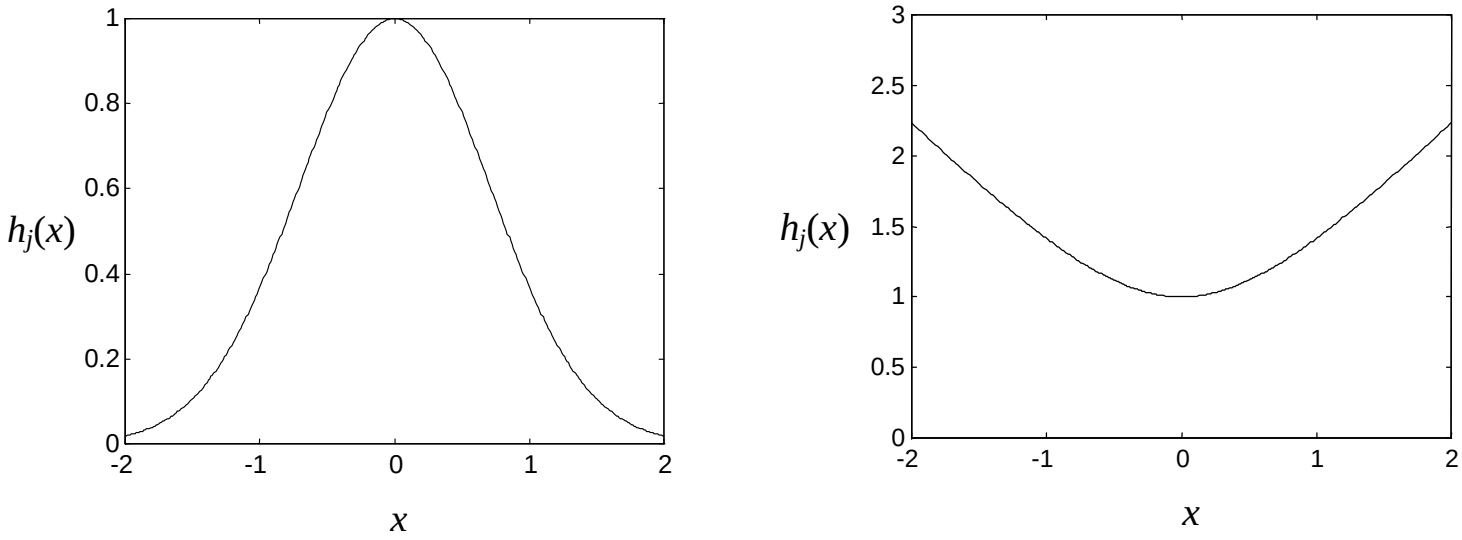
- Com base no exposto acima, fica evidente que redes neurais artificiais para treinamento supervisionado pertencem à classe de modelos de regressão não (puramente) paramétricos. Sendo assim, os pesos não apresentam um significado físico particular em relação ao problema de aplicação. Mas certamente existem modelos de redes neurais "mais não-paramétricos" que outros.

- Além disso, estimar os parâmetros de um modelo não-paramétrico (por exemplo, pesos de uma rede neural artificial) não é o objetivo primário do aprendizado supervisionado. O objetivo primário é estimar a "forma" da função em uma região compacta do espaço de aproximação (ou ao menos a saída para certos valores desejados de entrada).
- Por outro lado, em regressão paramétrica, o objetivo primário é estimar o valor dos parâmetros, por dois motivos:
    1. A "forma" da função já é conhecida;
    2. Os parâmetros admitem uma interpretação física.

11.2 Funções 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:
  - $h_j(x) = \exp\left(-\frac{(x-c_j)^2}{r_j^2}\right)$, para o caso escalar (veja figura 1(a));
- Uma função de base radial monotonicamente crescente típica é a função multiquadrática 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:

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

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

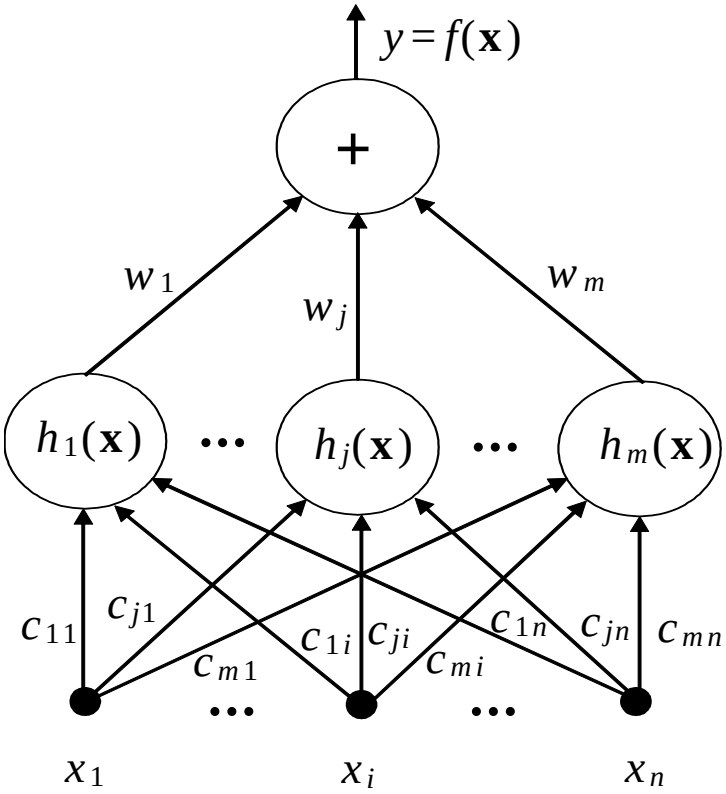
$$\Sigma_j = \begin{bmatrix} \sigma_{j1} & 0 & \cdots & 0 \\ 0 & \sigma_{j2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{jn} \end{bmatrix},$$

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

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

- Neste caso, os elementos do vetor $\sigma_j = \begin{bmatrix} \sigma_{j1} & \sigma_{j2} & \cdots & \sigma_{jn} \end{bmatrix}^T$ 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.

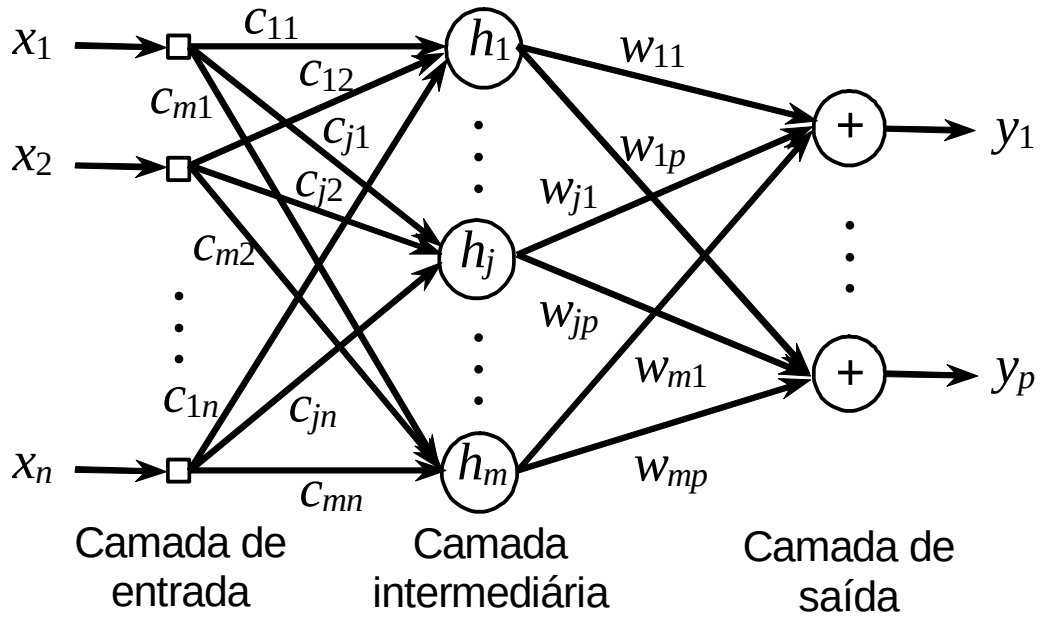
11.3 Rede Neural RBF (*Radial Basis Function Neural Network*)



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

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

- Uma versão para múltiplas saídas é apresentada abaixo.
- 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 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.



- 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* (HARTMAN *et al.*, 1990; PARK & SANDBERG, 1991).
- As redes *RBF* são redes de aprendizado local, de modo que é possível chegar a uma boa aproximação desde que um número suficiente de dados para treinamento seja fornecido na região de interesse.
- Em contraste, *perceptrons* multicamadas são redes de aprendizado "global" (em virtude da natureza das funções de ativação) que fazem aproximações de efeito global, em regiões compactas do espaço de aproximação.
- Redes neurais com capacidade de aproximação local são muito eficientes quando a dimensão do vetor de entradas é reduzida. Entretanto, quando o número de entradas não é pequeno, as redes *MLP* apresentam uma maior capacidade de generalização.
- Isto ocorre porque o número de funções de base radial deve aumentar exponencialmente com o aumento da dimensão da entrada.

11.4 O problema dos quadrados mínimos para aproximação de funções

- Considere o problema de encontrar os coeficientes de um polinômio de grau *n* que interpole *n*+1 pontos:

$$c_0 + c_1 x_j + \cdots + c_n x_j^n = y_j, \quad j = 0,1,\ldots,n,$$

- Estas *n*+1 equações com *n*+1 incógnitas podem ser colocadas na forma matricial:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow \mathbf{X} \cdot \mathbf{c} = \mathbf{y}$$

- Os coeficientes que compões o vetor **c** são obtidos simplesmente fazendo $\mathbf{c} = \mathbf{X}^{-1}\mathbf{y}$. Para a existência da inversa, basta que todas as colunas de **X** sejam LI, ou seja, que **X** tenha posto completo.

- Um problema de interpolação equivalente é o de encontrar os n+1 coeficientes de uma composição aditiva de funções-base que interpole n+1 pontos:

$$\sum_{j=0}^{n} c_j f_j(x_i) = y_i, \quad i = 0,1,\ldots,n$$

$$\begin{bmatrix} f_0(x_0) & f_1(x_0) & \cdots & f_n(x_0) \\ f_0(x_1) & \ddots & & \vdots \\ \vdots & & & \\ f_0(x_n) & \cdots & & f_n(x_n) \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow \mathbf{F} \cdot \mathbf{c} = \mathbf{y} \; .$$

- Particularmente no caso de problemas de aproximação de funções em que os parâmetros ajustáveis influenciam linearmente na saída do modelo, como no caso de redes neurais RBF com ajuste apenas dos pesos da camada de saída, resultam problemas parecidos com os dois problemas de interpolação acima, mas agora o número de amostras tende a ser bem maior que o número de parâmetros a determinar. Assim, o equivalente às matrizes **X** e **F** terá bem mais linhas que colunas.

- A razão para lidarmos agora com problemas de aproximação e não de interpolação é de duas naturezas:
  - » Os dados disponíveis geralmente estão sujeitos a ruído;
  - » Como a forma final do mapeamento não é conhecida, os dados disponíveis não contêm toda a informação necessária sobre o mapeamento.
- 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.

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

- 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 $\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_j & \cdots & w_m \end{bmatrix}^T$.

- O sistema de equações resultante é dado na forma:

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

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

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

- Portanto, existem m equações para obter as m incógnitas $\{w_r, r=1,...,m\}$. Exceto sob condições "patológicas", este sistema de equações vai apresentar uma solução única.
- Para encontrar esta solução única do sistema de equações lineares, é interessante recorrer à notação vetorial, fornecida pela álgebra linear, para obter:

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

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 **H**, com sua *j*-ésima coluna dada por $\mathbf{h}_j$, temos:

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

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 **f** pode ser apresentado na forma:

$$f_i = f(\mathbf{x}_i) = \sum_{r=1}^{m} w_r h_r(\mathbf{x}_i) = \begin{bmatrix} h_1(\mathbf{x}_i) & h_2(\mathbf{x}_i) & \cdots & h_m(\mathbf{x}_i) \end{bmatrix}\mathbf{w}$$

permitindo expressar **f** em função da matriz **H**, de modo que:

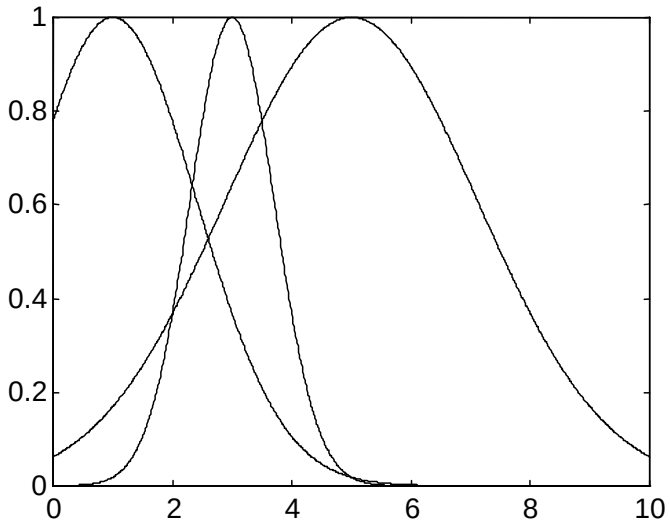
$$\mathbf{f} = \mathbf{H}\mathbf{w}$$

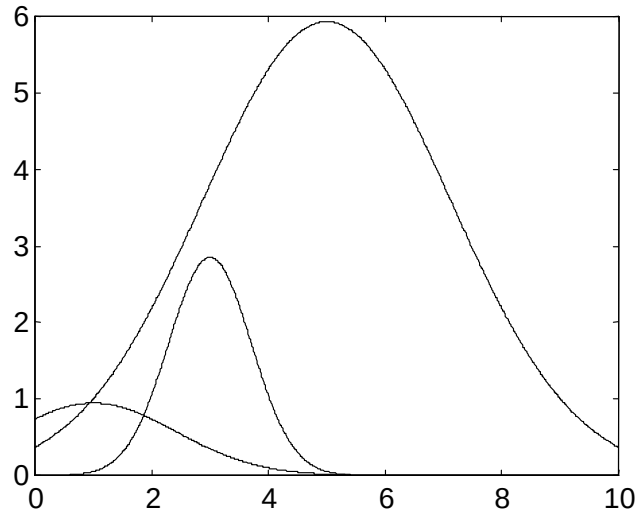
- Substituindo no sistema de equações lineares, resulta a solução ótima para o vetor de coeficientes da combinação linear (que correspondem aos pesos da camada de saída da rede neural de base radial):

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

- 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 $m \leq N$.
- A denominação de equação normal se deve ao fato de que $\mathbf{e} = \mathbf{H}\mathbf{w} - \mathbf{s}$, que é o erro cuja norma deve ser minimizada, é um vetor ortogonal (ou normal) ao espaço gerado pelas colunas de $\mathbf{H}$, sendo que $\mathbf{H}\mathbf{w}$ representa qualquer vetor neste espaço.

11.5 Exemplo didático 1



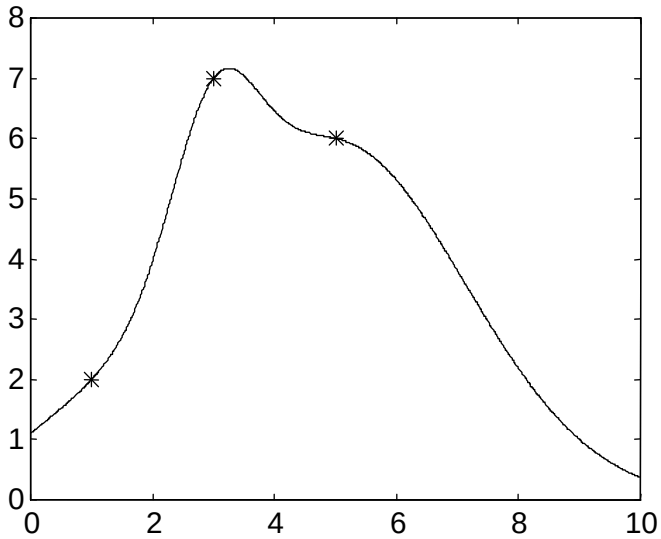


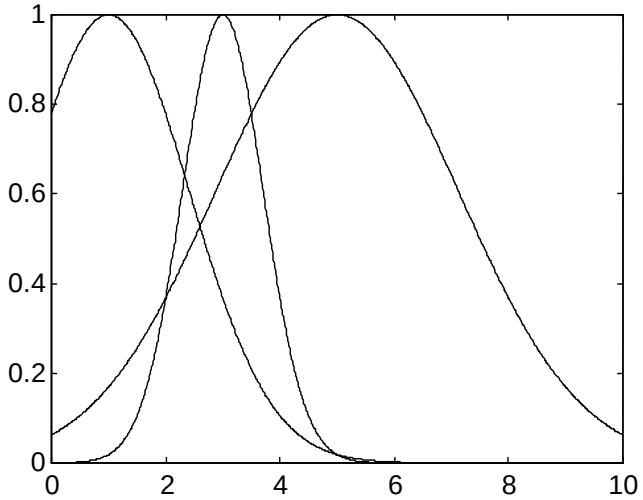
Caso 1: $m = N$

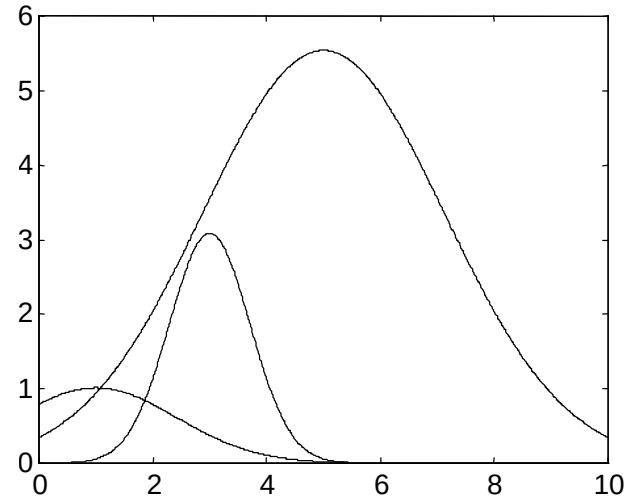
Pontos amostrados: (1,2); (3,7); (5,6)

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

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





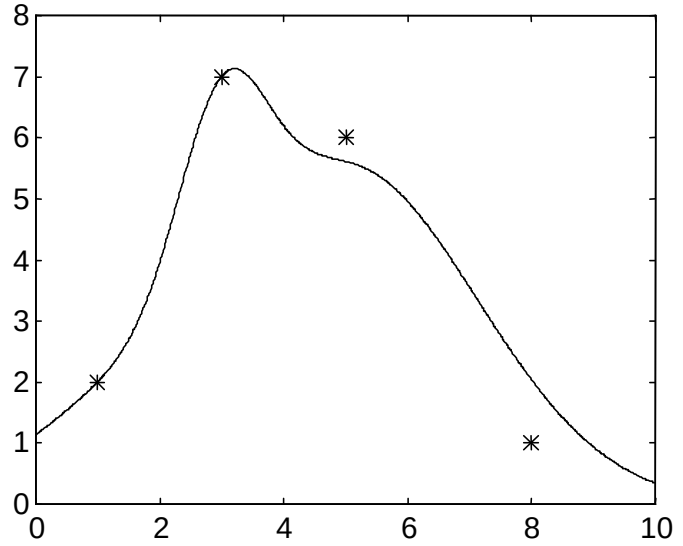


Caso 2: $m < N$

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.



11.6 Definição da dispersão

- Quanto às dispersões das funções de base radial, embora elas possam ser distintas (e até ajustáveis) para cada centro, usualmente se adota uma única dispersão para todos os centros, na forma (HAYKIN, 1999):

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

  onde m é o número de centros, e $d_{\max}$ é a distância máxima entre os centros.
- Com este critério de dispersão, evita-se que as funções de base radial sejam excessivamente pontiagudas, ou então com uma base demasiadamente extensa.

11.7 Seleção dos centros por auto-organização

- Para auto-organizar os centros, é suficiente aplicar algum algoritmo capaz de refletir a distribuição dos dados de entrada.
- O algoritmo a ser apresentado a seguir, é um algoritmo de clusterização denominado *k*-means (COVER & HART, 1967; MACQUEEN, 1967). Este algoritmo se assemelha ao

adotado para as redes de Kohonen, mas não leva em conta noções de vizinhança entre os centros.

- Seja $\{\mathbf{c}_j(t)\}_{j=1}^{m}$ os centros das funções de base radial na iteração *t*. O algoritmo *k*-means pode ser descrito da seguinte forma:

1. *Inicialização*: Escolha valores aleatórios distintos para os centros $\mathbf{c}_j(t)$.
2. *Amostragem*: Tome aleatoriamente um vetor $\mathbf{x}_i$ do conjunto de padrões de entrada;
3. *Matching*: Determine o índice *k* do centro que mais se aproxima deste padrão, na forma: $k(\mathbf{x}_i) = \arg\min_j \|\mathbf{x}_i(t) - \mathbf{c}_j(t)\|, \quad j = 1,...,m,$
4. *Ajuste*: Ajuste os centros usando a seguinte regra:

$$\mathbf{c}_j(t+1) = \begin{cases} \mathbf{c}_j(t) + \gamma[\mathbf{x}_i(t) - \mathbf{c}_j(t)], & k = k(\mathbf{x}_i) \\ \mathbf{c}_j(t), & \text{alhures} \end{cases}$$

   onde $\gamma \in (0,1)$ é a taxa de ajuste.

5. *Ciclo*: Repita os Passos 2 a 5 para todos os *N* padrões de entrada e até que os centros não apresentem deslocamento significativo após cada apresentação completa dos *N* padrões.

11.8 Exemplo didático 2

- Dados *N* pontos no plano, na forma $\{x_i, y_i\}_{i=1}^{N}$, obtenha uma fórmula fechada para os coeficientes do polinômio $P_r(x) = a_r x^r + a_{r-1}x^{r-1} + \cdots + a_1 x + a_0$ que melhor se ajusta aos pontos, segundo o método dos quadrados mínimos e com $r \leq N-1$.

Solução:

Deve-se minimizar a seguinte função: $J(a_r, a_{r-1}, ..., a_1, a_0) = \sum_{i=1}^{N} (y_i - P_r(x_i))^2$

Esta função pode ser expandida na forma:

$$J(a_r, a_{r-1}, ..., a_1, a_0) = \sum_{i=1}^{N} {y_i}^2 - 2\sum_{i=1}^{N} P_r(x_i) y_i + \sum_{i=1}^{N} (P_r(x_i))^2 =$$

$$= \sum_{i=1}^{N} {y_i}^2 - 2\sum_{i=1}^{N} \left(\sum_{j=0}^{r} a_j {x_i}^j \right) y_i + \sum_{i=1}^{N} \left(\sum_{j=0}^{r} a_j {x_i}^j \right)^2 =$$

$$= \sum_{i=1}^{N} {y_i}^2 - 2\sum_{j=0}^{r} a_j \left(\sum_{i=1}^{N} y_i {x_i}^j \right) + \sum_{j=0}^{r} a_j \sum_{k=0}^{r} a_k \left(\sum_{i=1}^{N} {x_i}^{j+k} \right)$$

Condição necessária de otimalidade: $\frac{\partial J}{\partial a_j} = 0$ para j=0,1,...,r. Assim, para cada j resulta:

$$\frac{\partial J}{\partial a_j} = -2\sum_{i=1}^{N} y_i x_i^{\,j} + 2\sum_{k=0}^{r} a_k \sum_{i=1}^{N} x_i^{\,j+k} = 0 \text{ para } j=0,1,...,r.$$

$$\sum_{k=0}^{r} a_k \sum_{i=1}^{N} x_i^{\,j+k} = \sum_{i=1}^{N} y_i x_i^{\,j} \text{ para } j=0,1,...,r.$$

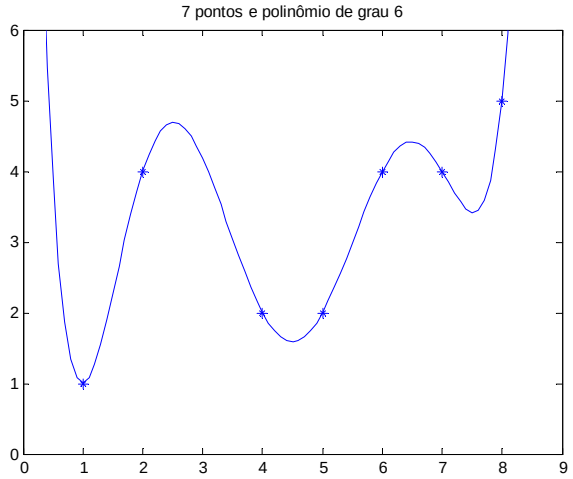
Expandindo para k e para j resulta:

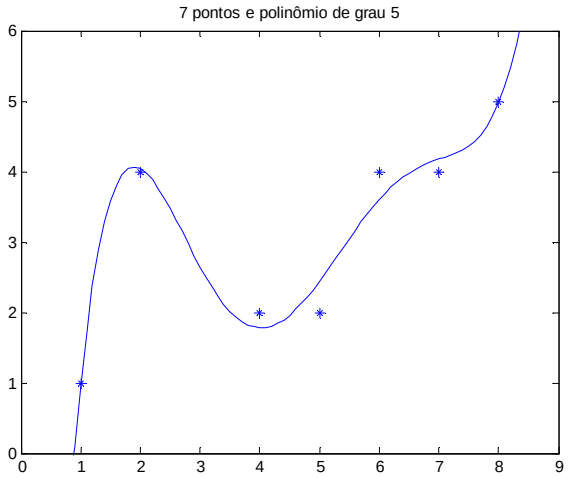
$$\begin{cases} a_0 \sum_{i=1}^{N} x_i^{\,0} + a_1 \sum_{i=1}^{N} x_i^{\,1} + a_2 \sum_{i=1}^{N} x_i^{\,2} + \cdots + a_r \sum_{i=1}^{N} x_i^{\,r} = \sum_{i=1}^{N} y_i x_i^{\,0} \\ a_0 \sum_{i=1}^{N} x_i^{\,1} + a_1 \sum_{i=1}^{N} x_i^{\,2} + a_2 \sum_{i=1}^{N} x_i^{\,3} + \cdots + a_r \sum_{i=1}^{N} x_i^{\,r+1} = \sum_{i=1}^{N} y_i x_i^{\,1} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_0 \sum_{i=1}^{N} x_i^{\,r} + a_1 \sum_{i=1}^{N} x_i^{\,r+1} + a_2 \sum_{i=1}^{N} x_i^{\,r+2} + \cdots + a_r \sum_{i=1}^{N} x_i^{\,2r} = \sum_{i=1}^{N} y_i x_i^{\,r} \end{cases}$$

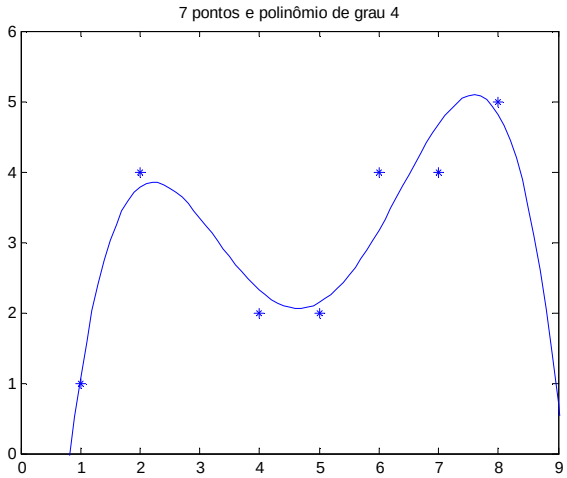
Em forma matricial, fica:

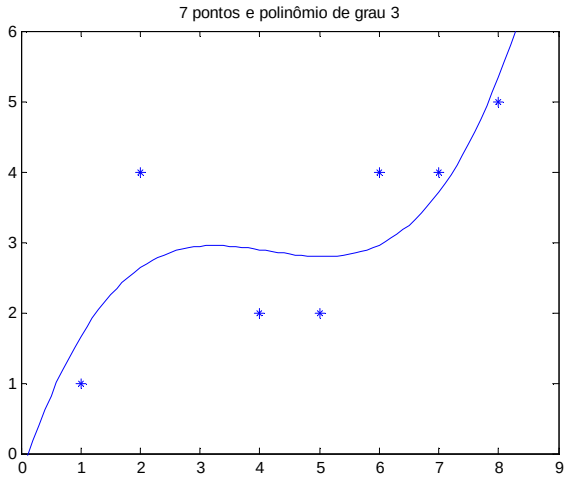
$$\begin{bmatrix} \sum_{i=1}^{N} x_i^0 & \sum_{i=1}^{N} x_i^1 & \sum_{i=1}^{N} x_i^2 & \cdots & \sum_{i=1}^{N} x_i^r \\ \sum_{i=1}^{N} x_i^1 & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i^3 & \cdots & \sum_{i=1}^{N} x_i^{r+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_i^r & \sum_{i=1}^{N} x_i^{r+1} & \sum_{i=1}^{N} x_i^{r+2} & \cdots & \sum_{i=1}^{N} x_i^{2r} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_r \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} y_i x_i^0 \\ \sum_{i=1}^{N} y_i x_i^1 \\ \vdots \\ \sum_{i=1}^{N} y_i x_i^r \end{bmatrix} \Rightarrow Za = q$$

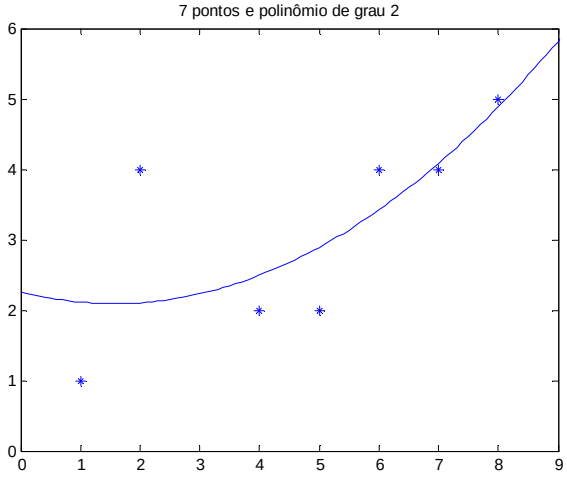
Se os x_i's forem todos distintos, então a matriz $Z \in \Re^{(r+1)\times(r+1)}$ é inversível e a solução assume a forma: $a = Z^{-1}q$. Exemplos:

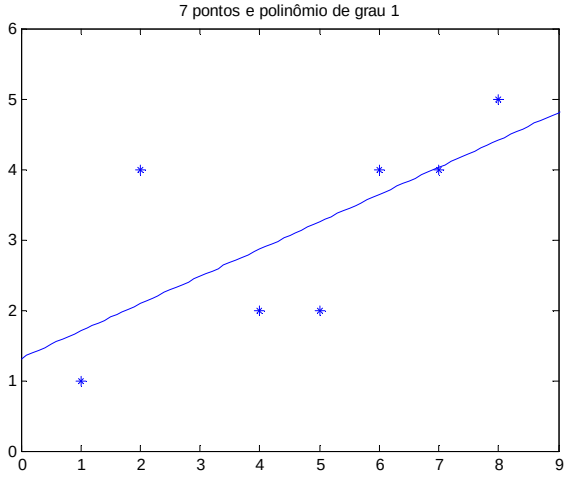





7 pontos e polinômio de grau 4


7 pontos e polinômio de grau 3


7 pontos e polinômio de grau 2


7 pontos e polinômio de grau 1

Programa em Matlab para se chegar aos resultados acima:

```
% Programa para aproximação polinomial de pontos no plano
clear all;
ptos = [1 1;2 4;4 2;5 2;6 4;7 4;8 5];
N = length(ptos(:,1));
x = ptos(:,1);
y = ptos(:,2);
r = 6; % Grau do polinômio
% Cálculo da matriz Z e do vetor q
Z = zeros(r+1,r+1);
q = zeros(r+1,1);
for j=0:r,
    for k=0:r,
        for i=1:N,
            Z(j+1,k+1) = Z(j+1,k+1)+power(x(i),j+k);
        end
    end
    for i=1:N,
        q(j+1) = q(j+1)+y(i)*power(x(i),j);
    end
end
% Obtenção dos coeficientes do polinômio
a = inv(Z)*q;
% Produção dos resultados gráficos
t = [0:0.1:10];
ind = 1;
f = zeros(length(t),1);
for i=1:length(t),
    for j=0:r,
        f(i,1) = f(i,1)+a(j+1)*power(t(i),j);
    end
end
figure(1);
% Plotagem dos pontos fornecidos
for i=1:N,
    plot(x(i),y(i),'*');
    hold on;
end
% Plotagem da função polinomial
plot(t,f);
axis([0 9 0 6]);
title(sprintf('%d pontos e polinômio de grau %d',N,r));
hold off;
```

12. Organização e ordem

- Um cristal tende a apresentar mais ordem que uma célula. No entanto, a célula tende a apresentar mais organização.
- Um papel de parede que apresenta um padrão de repetição tende a apresentar mais ordem espacial que uma pintura. No entanto, a pintura tende a apresentar mais organização espacial.
- Um som de alarme tende a apresentar mais ordem temporal que uma música. No entanto, a música tende a apresentar mais organização temporal.
- Objetos estão dispostos no tempo e/ou no espaço de forma ordenada se eles seguem um regra específica de disposição temporal e/ou espacial.
- Objetos estão dispostos no tempo e/ou no espaço de forma organizada se eles contribuem em conjunto para produzir alguma funcionalidade.
- Ideias podem estar logicamente ordenadas (para efeito de apresentação) ou logicamente organizadas (para efeito de argumentação).

13. Auto-Organização

- O estudo de sistemas auto-organizados é recente, embora a humanidade tenha sempre se ocupado com questões vinculadas à origem de sistemas organizados.
- As formas que podem ser observadas no mundo à nossa volta representam apenas uma pequena parcela de todas as formas possíveis. Logo, por que não existe mais variedade?
- Para procurar respostas a questões como esta é que se estuda sistemas auto-organizados e teoria da complexidade.
- Exemplos de sistemas naturais que apresentam organização: galáxias, planetas, componentes químicos, células, organismos, sociedades.
- A auto-organização pode ser abordada recorrendo-se às propriedades e leis comuns a todos os sistemas organizados, independente de suas particularidades.

- Neste caso, a atual disponibilidade de recursos computacionais é fundamental para viabilizar a investigação dos processos envolvidos, através de simulações que envolvem um grande número de etapas e uma grande variedade de parâmetros e condições iniciais.
- Mesmo assim, o estudo está restrito a fenômenos (concretos ou abstratos) que são facilmente reprodutíveis, os quais certamente representam um subconjunto de todos os fenômenos possíveis.
- A reprodução em computador de fenômenos auto-organizados tem levado à geração de teorias que procuram descrever sistemas complexos e sua organização espontânea.
- Um sistema complexo pode ser caracterizado como o resultado da auto-organização de componentes sob forte interação, produzindo estruturas sistêmicas cujas propriedades geralmente não estão presentes em nenhum de seus componentes, já que estas dependem de níveis mais elevados de organização.

- Um sistema, por sua vez, pode ser definido como um agrupamento coerente de componentes que operam como um todo e que apresentam uma individualidade, ou seja, se distinguem de outras entidades por fronteiras reconhecíveis. Há muitas variedades de sistemas, as quais podem ser classificadas em 3 grandes grupos:

    1. Quando as interações de seus componentes são fixas. Ex: máquina.
    2. Quando as interações de seus componentes são irrestritas. Ex: gás.
    3. Quando existem interações fixas e variáveis de seus componentes. Ex: célula.

- Os sistemas de maior interesse são aqueles pertencentes à classe 3, já que dependem da natureza e forma das interações de seus componentes ao longo de sua existência. Assim, o sistema vai apresentar um novo comportamento sempre que componentes forem adicionados, removidos ou rearranjados, ou então sempre que houver modificação nas interações.
- A essência da auto-organização está no surgimento de estrutura (formas restritas) e organização sem que estas sejam impostas de fora do sistema. Isto implica que este

fenômeno é interno ao sistema, resulta da interação de seus componentes e, em essência, não depende da natureza física destes componentes.

- A organização pode se dar no espaço, no tempo, ou em ambos.
- O que se busca são regras gerais para o crescimento e evolução de estruturas sistêmicas. Com isso, espera-se poder prever a organização futura que irá resultar de alterações promovidas junto aos componentes de um dado sistema, além de poder estender estes resultados a outros sistemas semelhantes.
- Em geral, os mecanismos estabelecidos pelos componentes de um dado sistema capaz de expressar auto-organização são: **realimentação positiva, realimentação negativa e interação local**.
- Esses mecanismos devem ocorrer simultaneamente no espaço-tempo.
- São distintos, portanto, de mecanismos de regulação de pressão arterial ou regulação de temperatura corpórea, que envolvem apenas realimentação negativa, e do mecanismo de contração no parto, que envolve apenas realimentação positiva.

14. Motivação para treinamento não-supervisionado: clusterização

- Dados rotulados são aqueles que assumem valores em um mesmo espaço vetorial multidimensional e que vêm acompanhados da classe a que cada um pertence (rótulo), podendo haver múltiplas classes, com variâncias e número de dados distintos ou não para cada classe.
- Dados não-rotulados são aqueles que assumem valores em um mesmo espaço vetorial multidimensional, e que não se conhece a priori a classe a que cada um pertence, embora cada um pertença a uma classe específica. O número de classes pode ser conhecido a priori ou não. A variância e o número de dados de cada classe pode diferir ou não.

14.1 Modelo simples de classificação para dados rotulados

- Hipótese: as classes apresentam propriedades distintas (seus elementos pertencem a regiões distintas do espaço vetorial multidimensional);
- Modelagem: um representante para cada classe;

- Objetivo: minimizar o somatório das distâncias entre os dados e o respectivo representante da classe a que pertencem;
- Aplicação: após finalizar o posicionamento de todos os representantes, definir o rótulo de cada novo dado não-rotulado como aquele associado ao representante que possuir a menor distância ao dado.

- Trata-se, portanto, de um problema de otimização, que pode ser resolvido por intermédio de técnicas de treinamento supervisionado.
- No caso, basta dividir o presente problema em *C* problemas distintos, sendo *C* o número de classes.
- Exemplos gráficos.
- Limitações.

14.2 Modelo composto de classificação para dados rotulados

- Hipótese: as classes apresentam propriedades distintas (seus elementos pertencem a regiões distintas do espaço vetorial multidimensional);

- Modelagem: múltiplos representantes para cada classe;
- Objetivo: minimizar o somatório da distância entre cada dado e o representante mais próximo da classe a que pertence;
- Aplicação: após finalizar o posicionamento de todos os representantes, definir o rótulo de cada novo dado não-rotulado como aquele associado ao representante que possuir a menor distância ao dado.

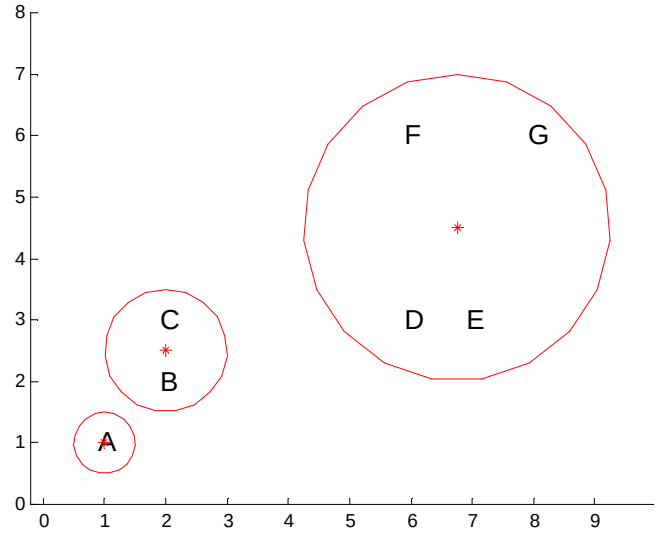
- Trata-se também de um problema de otimização, mas neste caso técnicas de treinamento não-supervisionado devem ser empregadas, pois os representantes de cada classe devem se auto-organizar no espaço de acordo com a distribuição apresentada pelos dados da respectiva classe.
- Como no caso anterior, pode-se dividir o presente problema em C problemas distintos.
- Exemplos gráficos.

14.3 Modelo composto de classificação para dados não-rotulados

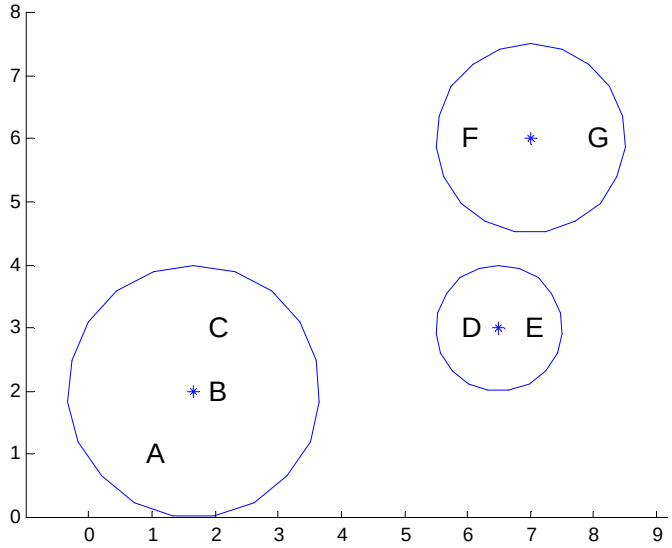
- Hipótese: não se conhece o número de classes, mas sabe-se que elas apresentam propriedades distintas (seus elementos pertencem a regiões distintas do espaço vetorial multidimensional);
- Modelagem: múltiplos representantes não-rotulados;
- Objetivo: minimizar o somatório da distância entre cada dado e o representante não-rotulado mais próximo.
- Rotulagem: após finalizar o posicionamento de todos os representantes (também por auto-organização), aplicar alguma técnica de discriminação que agrupe representantes de acordo com as posições relativas entre eles. A seguir, atribuir um rótulo diferente a cada grupo de representantes.
- Aplicação: definir o rótulo de cada novo dado não-rotulado como aquele associado ao representante que possuir a menor distância ao dado.

- Exemplos gráficos.

14.4 O algoritmo *k*-médias (do inglês *k*-means)



Erro total = 12,250



Erro total = 5,167

- Figuras extraídas de ZUCHINI (2003) e baseadas em JAIN *et al*. (1999)
- Sensibilidade à condição inicial
- Ausência de vizinhança topológica e *k* fornecido a priori

15. 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** e ao **treinamento por reforço**, 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.

- Duas abordagens têm sido propostas para aprendizado não-supervisionado:
  1. Técnicas para estimação de densidades de probabilidade, que produzem modelos estatísticos explícitos para descrever os fenômenos responsáveis pela produção dos dados de entrada. Ex: redes bayesianas e redes gaussianas.
  2. Técnicas de extração de regularidades estatísticas diretamente dos dados de entrada. Ex: redes de Kohonen.
- A história do aprendizado não-supervisionado é longa e diversificada:
  - (BARLOW, 1989), HEBB (1949), HINTON & SEJNOWSKI (1986), MACKAY (1956), MARR (1970);
  - BECKER & PLUMBLEY (1996), HINTON (1989), LINSKER (1988), KOHONEN (1989), KOHONEN (1997);

16. 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 (interpretação possível para problemas de otimização combinatória).
- 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 dimensionalidade** sempre que o espaço original (em que os dados se encontram) apresentar uma dimensão mais elevada.

- Toda redução de dimensionalidade (relativa à dimensionalidade intrínseca dos dados) pode implicar na perda de informação (por exemplo, violação topológica). Sendo assim, este mapeamento deve ser tal que minimiza a perda de informação.
- A informação é uma medida da redução da incerteza, sobre um determinado estado de coisas. Neste sentido, a informação não deve ser confundida com o dado ou seu significado, e apresenta-se como função direta do grau de originalidade, imprevisibilidade ou valor-surpresa do dado (ou conjunto de dados).
- Espaços normados são aqueles que permitem o estabelecimento de propriedades topológicas entre seus elementos. Ex: medida de distância (fundamental para a definição do conceito de vizinhança), ordenamento.

16.1 Arranjo unidimensional

- Um mapa de Kohonen unidimensional é dado por uma sequência ordenada de neurônios lineares, sendo que o número de pesos 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.**

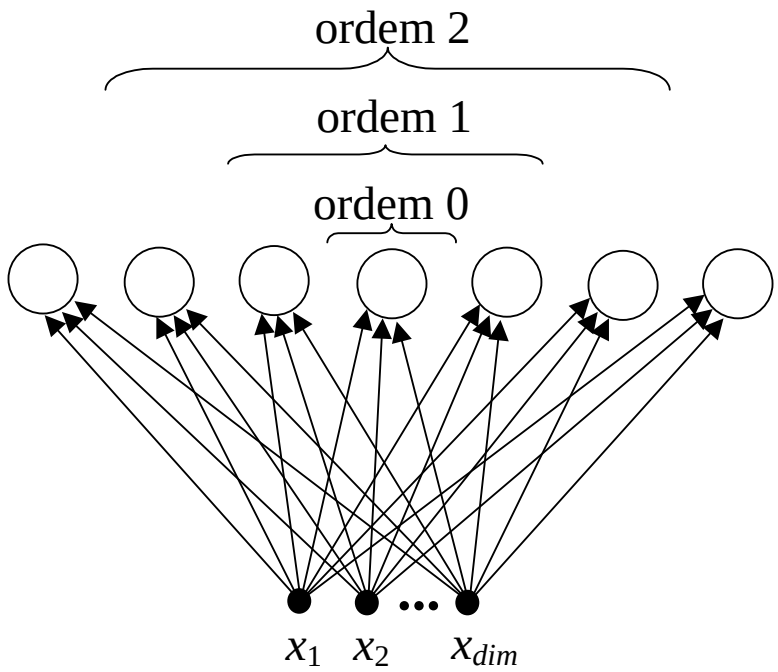


Figura 2 – Rede de Kohonen em arranjo unidimensional: ênfase na vizinhança

Exemplo: Para *dim* = 2, considere *n* = 4, sendo que os vetores de pesos são dados na forma:

$$\mathbf{w}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} -3 \\ 0 \end{bmatrix} \quad \text{e} \quad \mathbf{w}_4 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

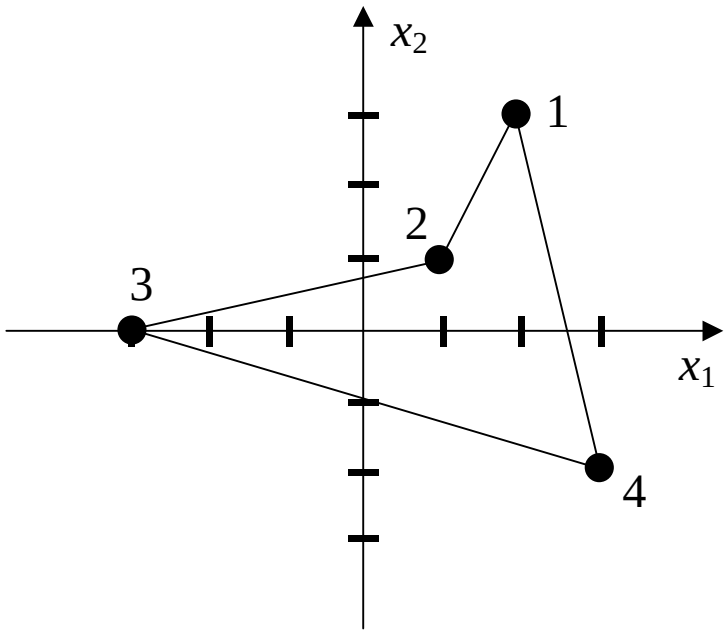


Figura 3 – Mapa de Kohonen em arranjo unidimensional (nesta caso, existe vizinhança entre primeiro e último neurônios)

16.2 Arranjo bidimensional

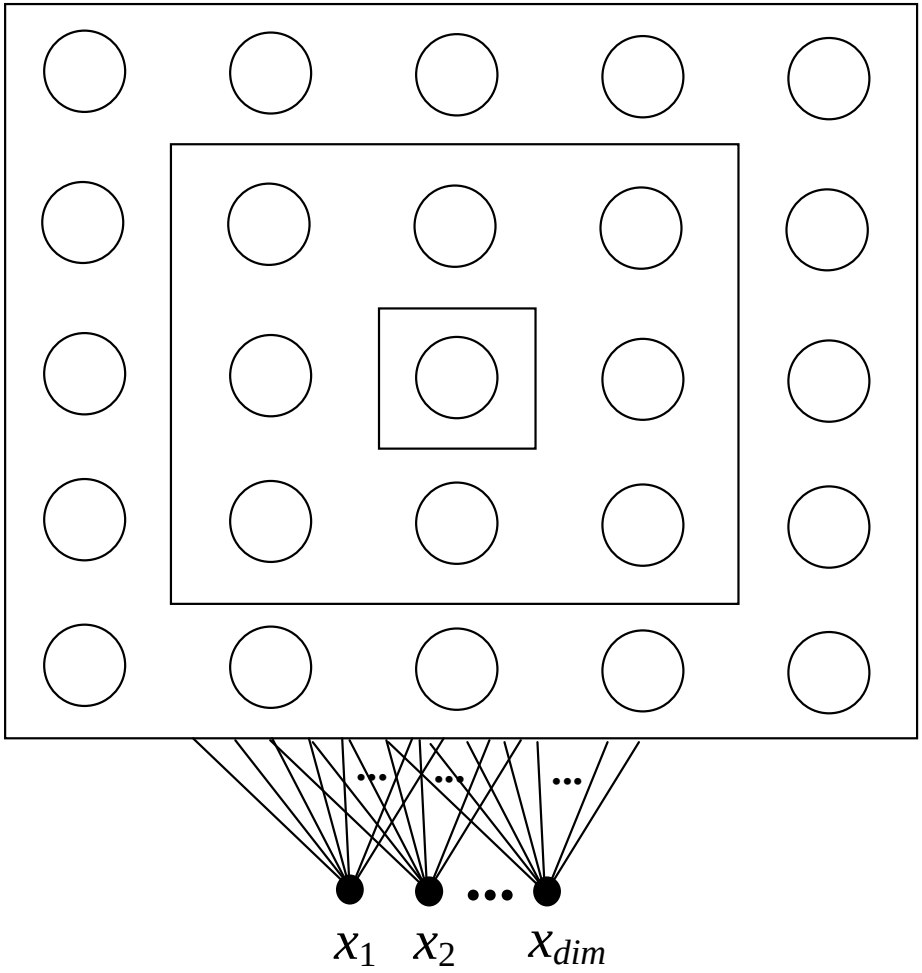
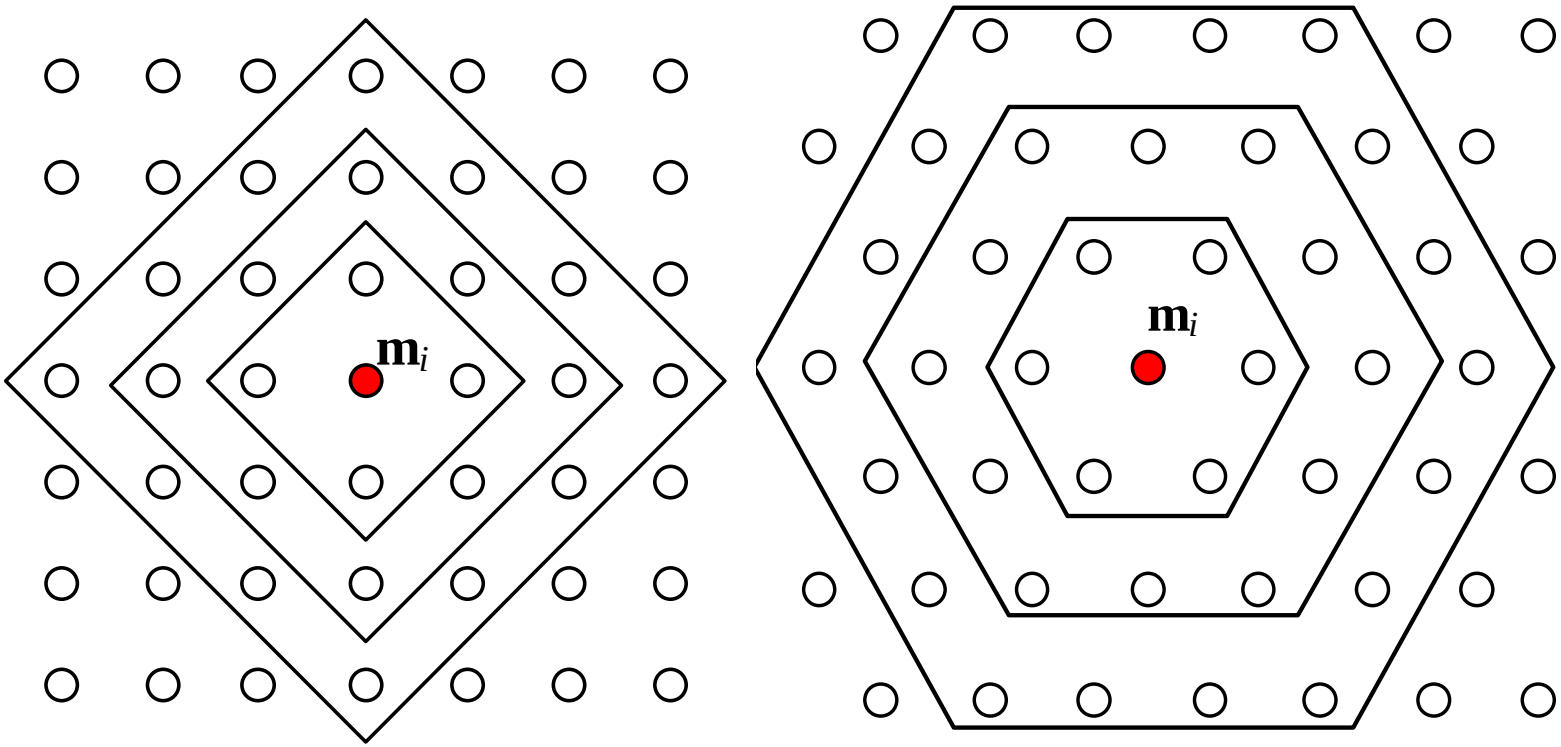
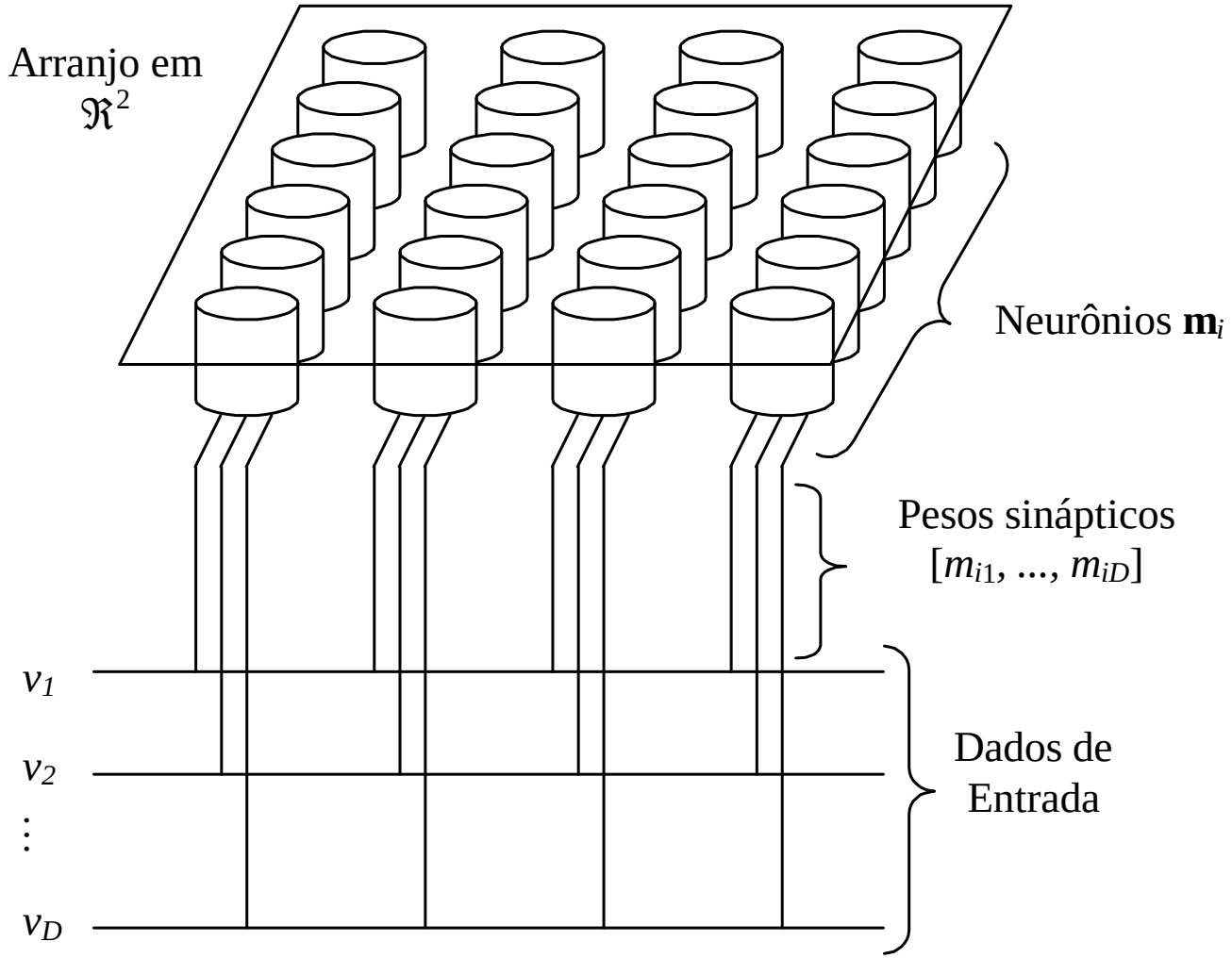


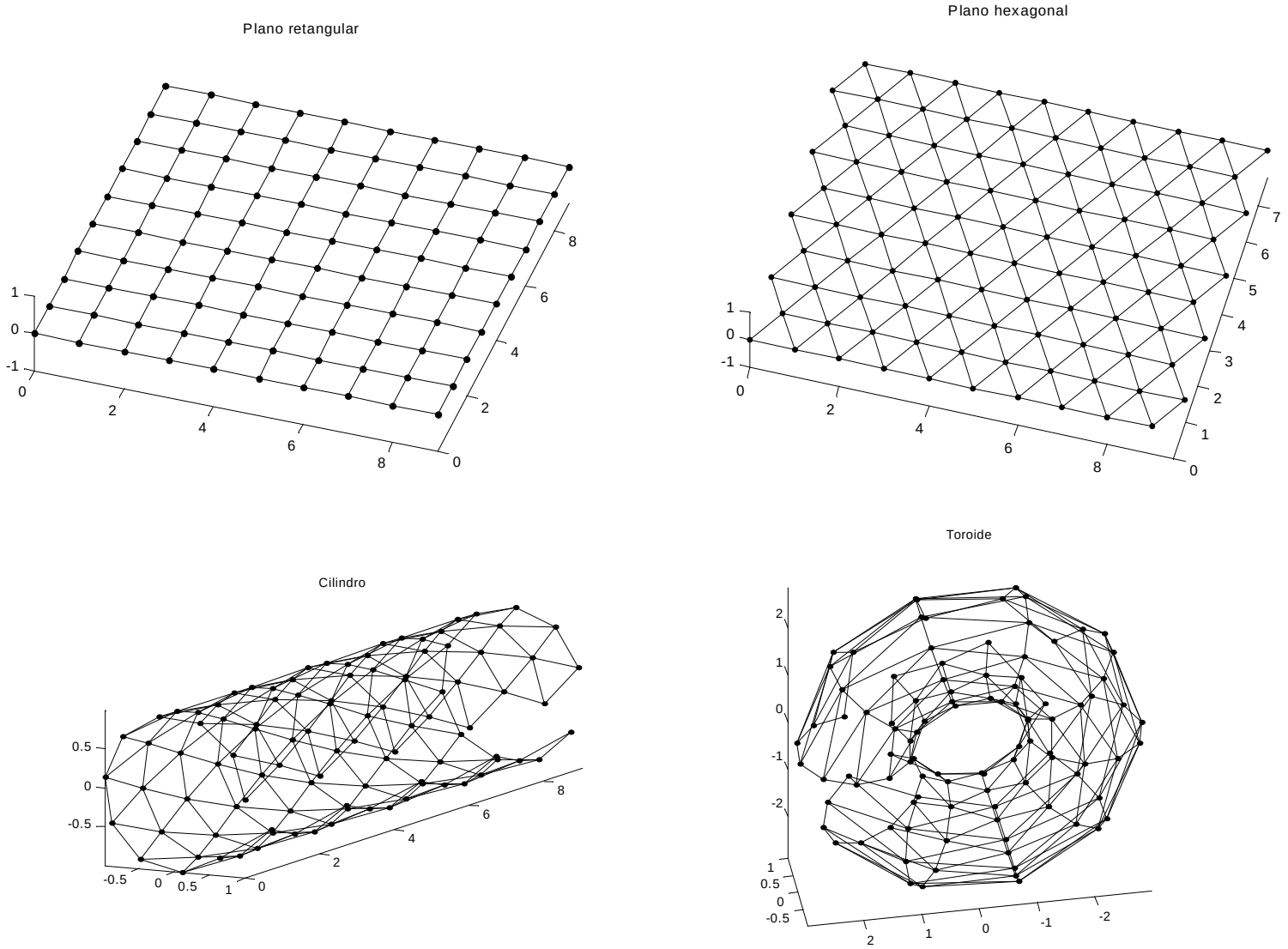
Figura 4 – Rede de Kohonen em arranjo bidimensional: ênfase na vizinhança



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



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



Arranjos com vizinhança nos extremos (figuras extraídas de ZUCHINI, 2003)

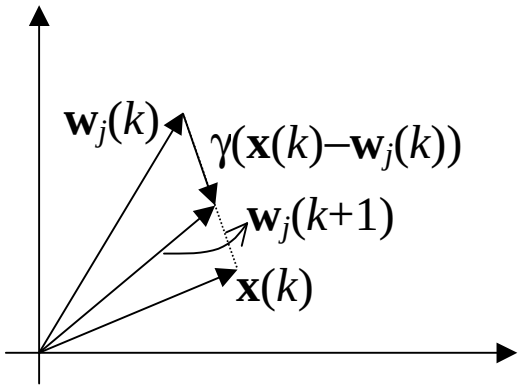
16.3 Fase de aprendizado não-supervisionado

- Se os pesos se mantiverem com norma unitária, então a definição do neurônio vencedor pode se dar de duas formas:
  1. produto escalar;
  2. norma euclidiana (é a única válida também no caso de pesos não-normalizados).
- Seja *j* o neurônio vencedor:

Alternativa 1

- Somente o neurônio *j* é ajustado na forma:

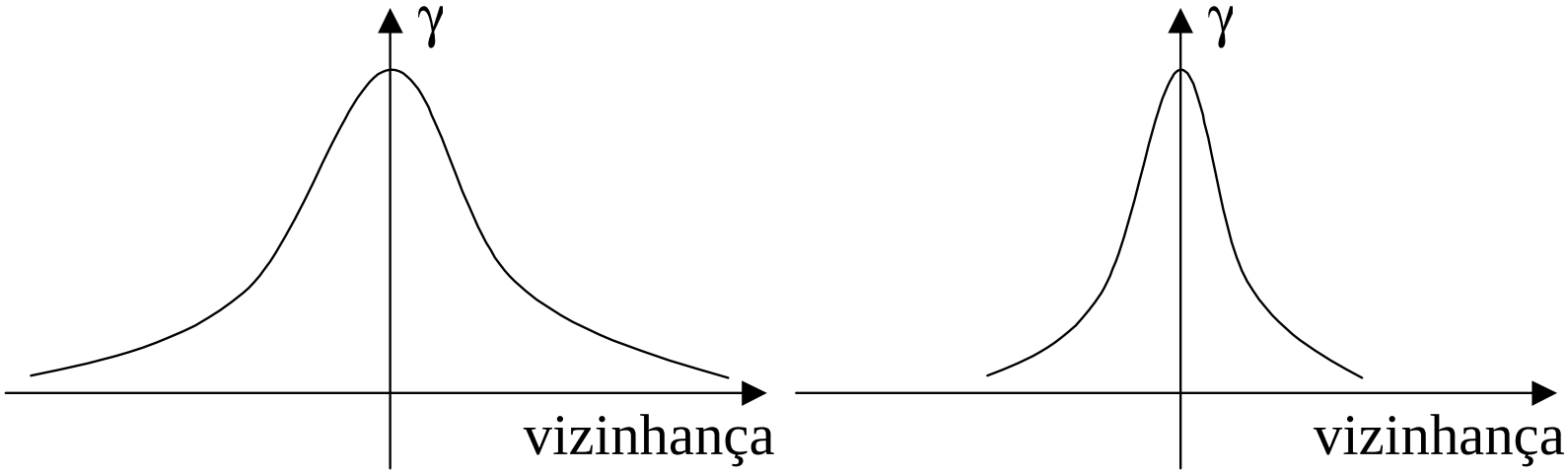
$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \gamma\left(\mathbf{x}(k) - \mathbf{w}_j(k)\right)$$



Alternativa 2

- Caso existam múltiplos representantes para cada agrupamento de dados, então é interessante ajustar o neurônio vencedor e seus vizinhos mais próximos.

Implementação



- É importante que a influência de cada neurônio vencedor seja ampla no início do processo e sofra uma redução continuada com o decorrer das iterações.

16.4 Algoritmo de ajuste dos pesos

while <condição de parada> é falso,

ordene aleatoriamente os N padrões de entrada;

for i=1 até N,

$$j = \arg\min_j \left\| \mathbf{x}_i - \mathbf{w}_j \right\|$$

$\forall\, J \in \text{Viz}(j)$ **do**:

$$\mathbf{w}_J = \mathbf{w}_J + \gamma\big(dist(j, J)\big)\big(\mathbf{x}_i - \mathbf{w}_J\big);$$

end do

end for

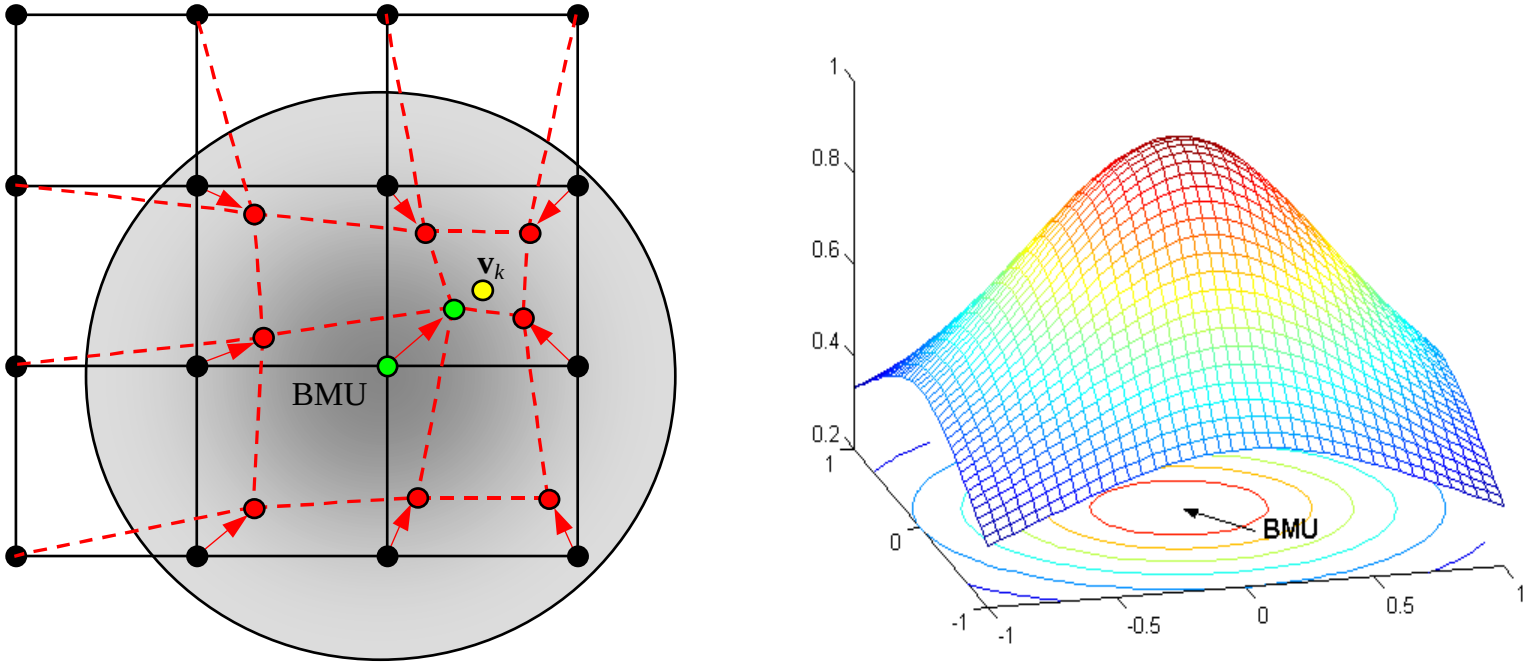
atualize a taxa de aprendizado γ;

atualize a vizinhança;

avalie a condição de parada;

end while

16.5 Ajuste de pesos com restrição de vizinhança



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.

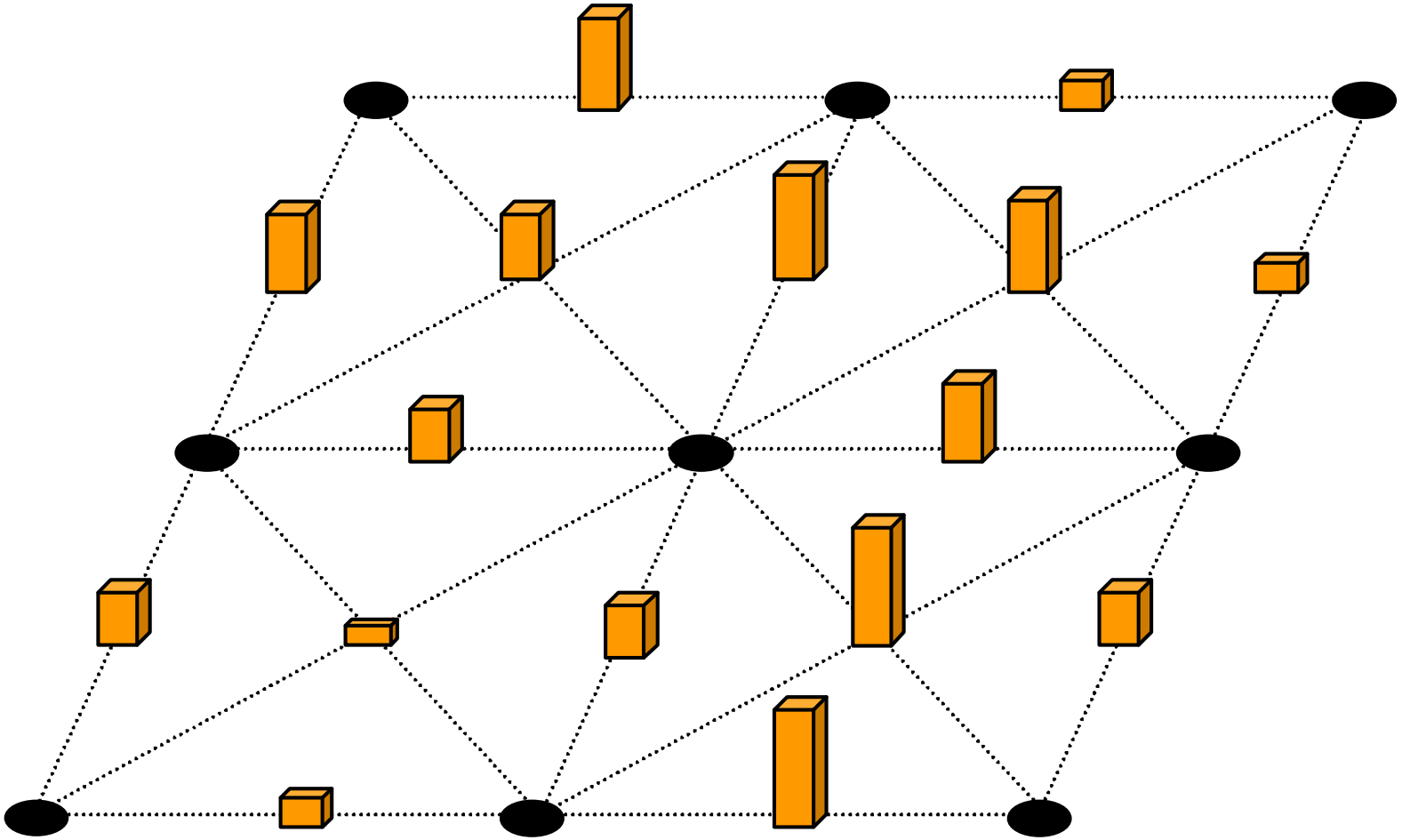
16.6 Discriminação dos agrupamentos

- Dada a conformação final de neurônios (não rotulados), como realizar a clusterização, ou seja, definição de agrupamentos e atribuição do mesmo rótulo a todos os neurônios pertencentes a um dado agrupamento?
- Solução: matriz(vetor)-U (ULTSCH, 1993; COSTA, 1999)
- Aspecto a ser explorado: após o processo de auto-organização, dados de entrada com características semelhantes passam a promover reações semelhantes da rede neural.
- Assim, comparando-se as reações da rede neural treinada, é possível agrupar os dados pela análise do efeito produzido pela apresentação de cada um à rede.
- A matriz-U é uma ferramenta que permite realizar a discriminação dos agrupamentos, a partir de uma medida do grau de similaridade entre os pesos de neurônios adjacentes na rede. O perfil apresentado pelas distâncias relativas entre neurônios vizinhos representa uma forma de visualização de agrupamentos.

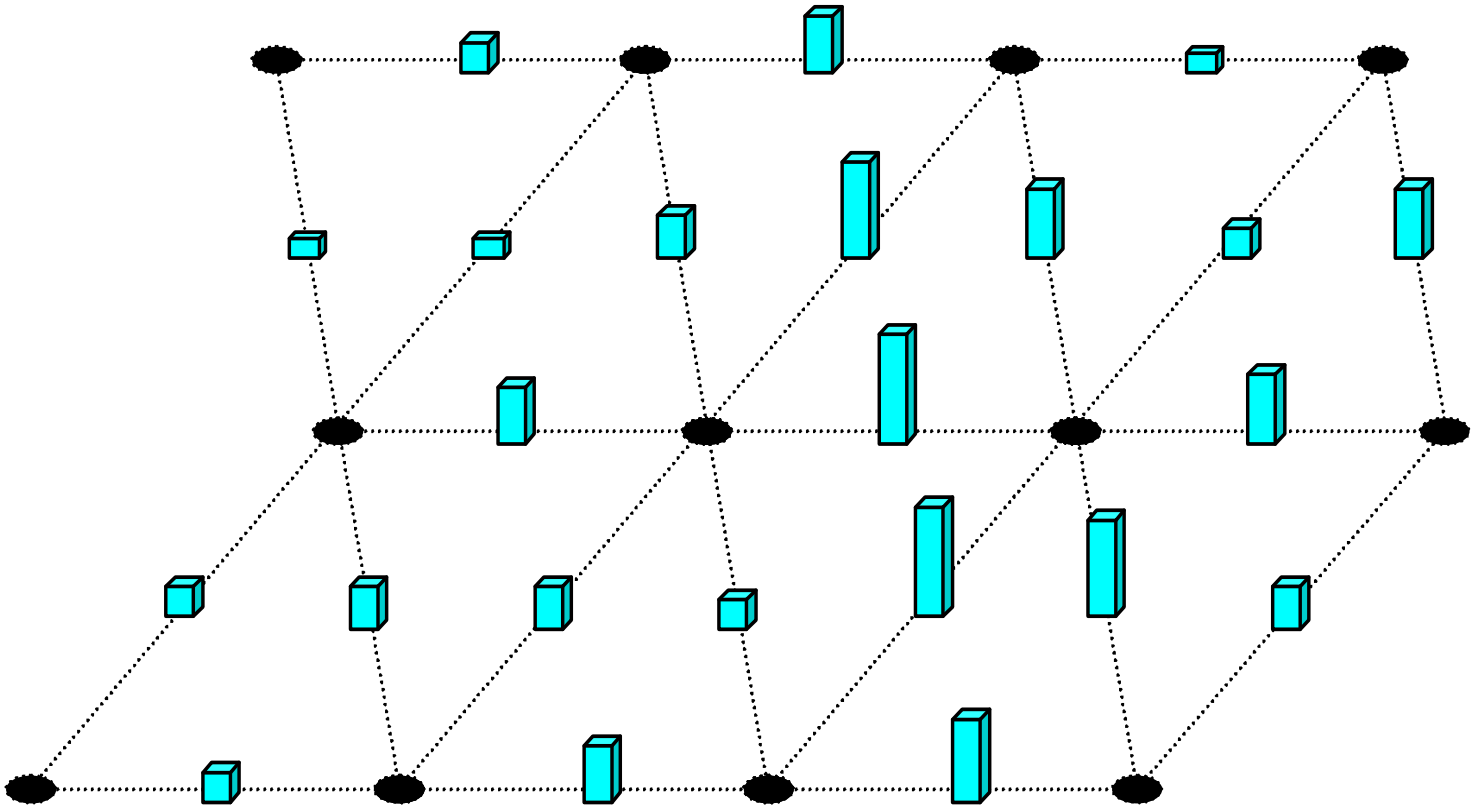
- Recebeu a denominação de matriz por ter sido proposta no caso de mapas bidimensionais, sendo que o grau de similaridade é plotado na terceira dimensão gerando uma superfície em relevo em 3D.
- Para o caso de mapas unidimensionais, tem-se o vetor-U.
- Topologicamente, as distâncias entre neurônios vizinhos refletem os agrupamentos, pois uma "depressão" ou um "vale" da superfície de relevo representa neurônios pertencentes a um mesmo agrupamento. Neurônios que têm uma distância grande em relação ao neurônio adjacente, a qual é representada por um pico da superfície de relevo, são neurônios discriminantes de agrupamentos.

16.7 Aplicação

- Como utilizar o mapa de Kohonen, após a fase de treinamento não-supervisionado e depois de ter as classes devidamente discriminadas, para classificação de padrões?
- Exemplos de aplicação

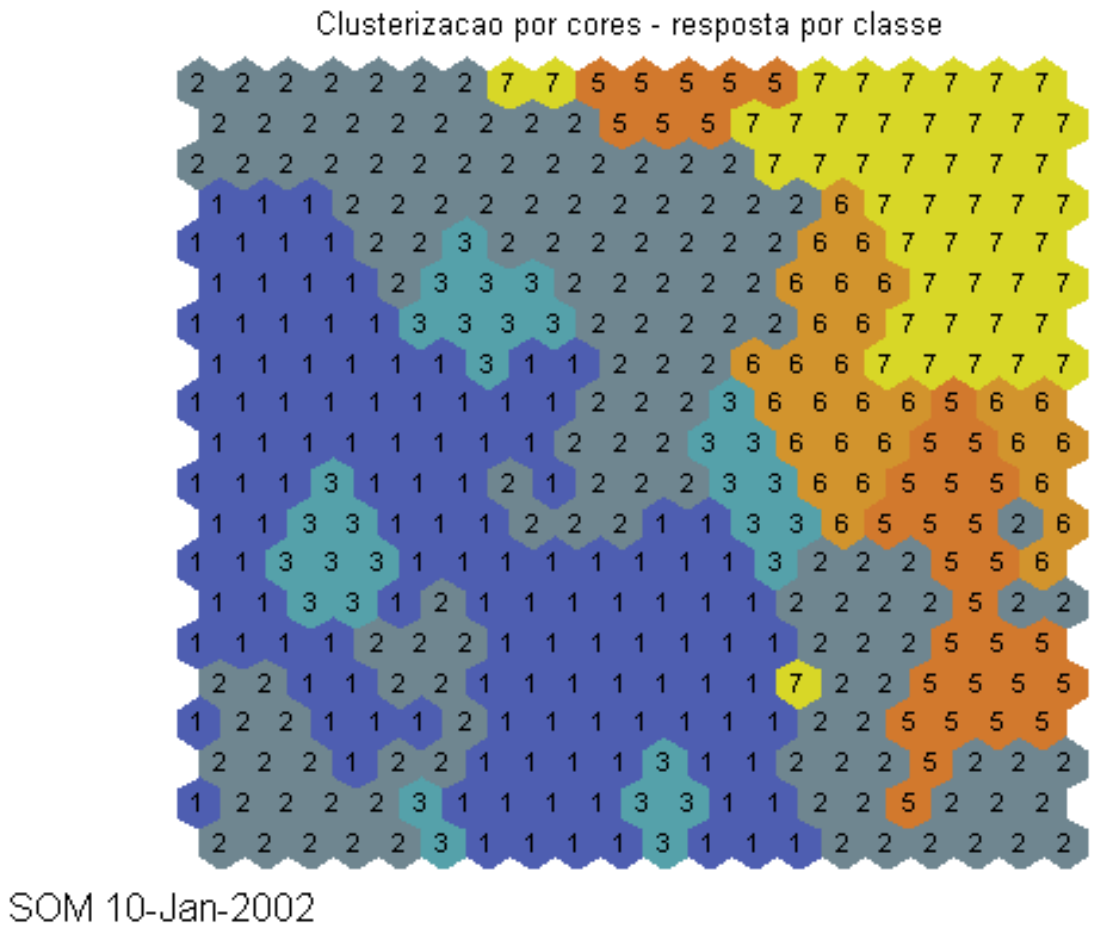


Exemplo de matriz-U para arranjo retangular (figura extraída de ZUCHINI, 2003)

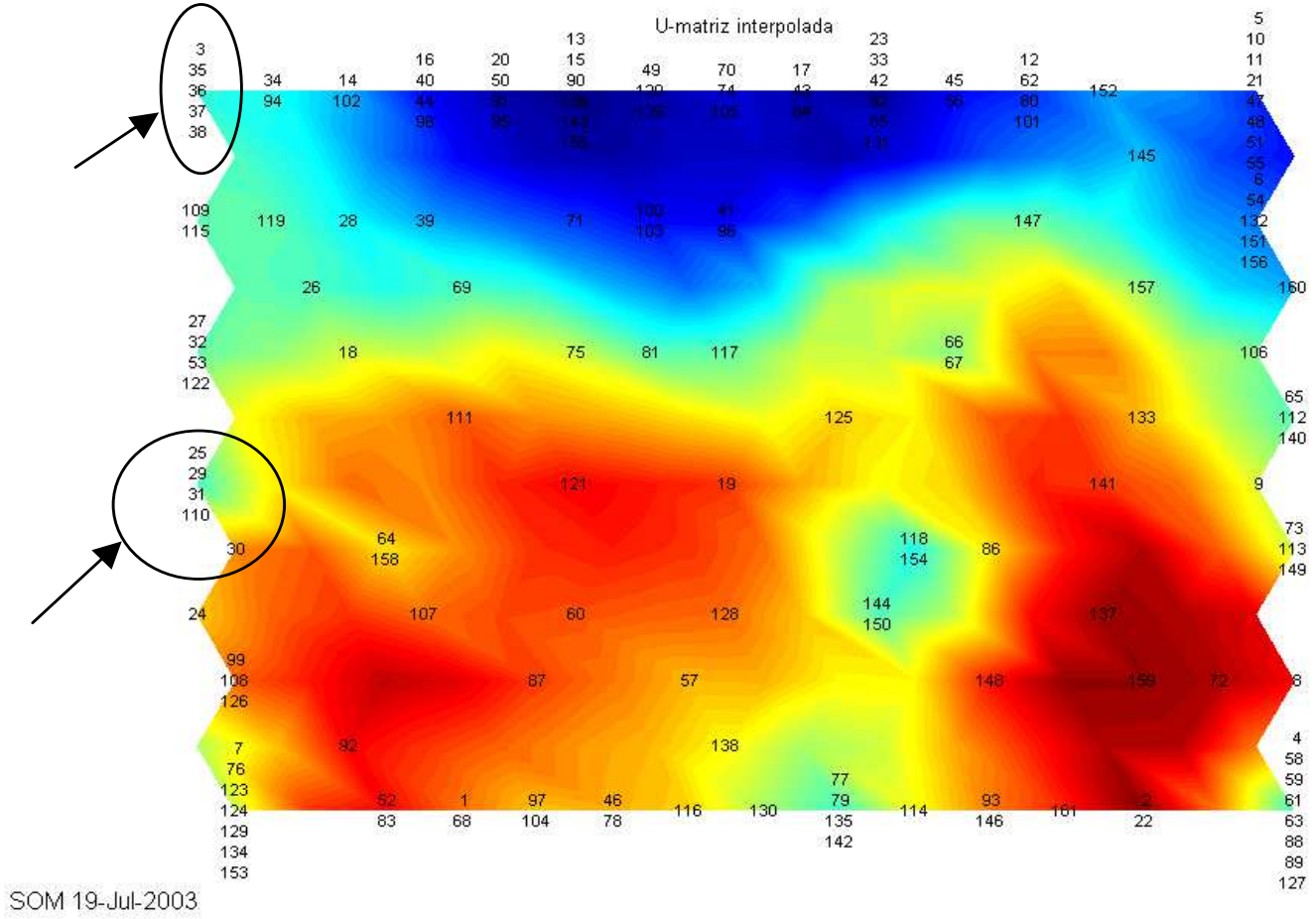


Exemplo de matriz-U para arranjo hexagonal (figura extraída de ZUCHINI, 2003)

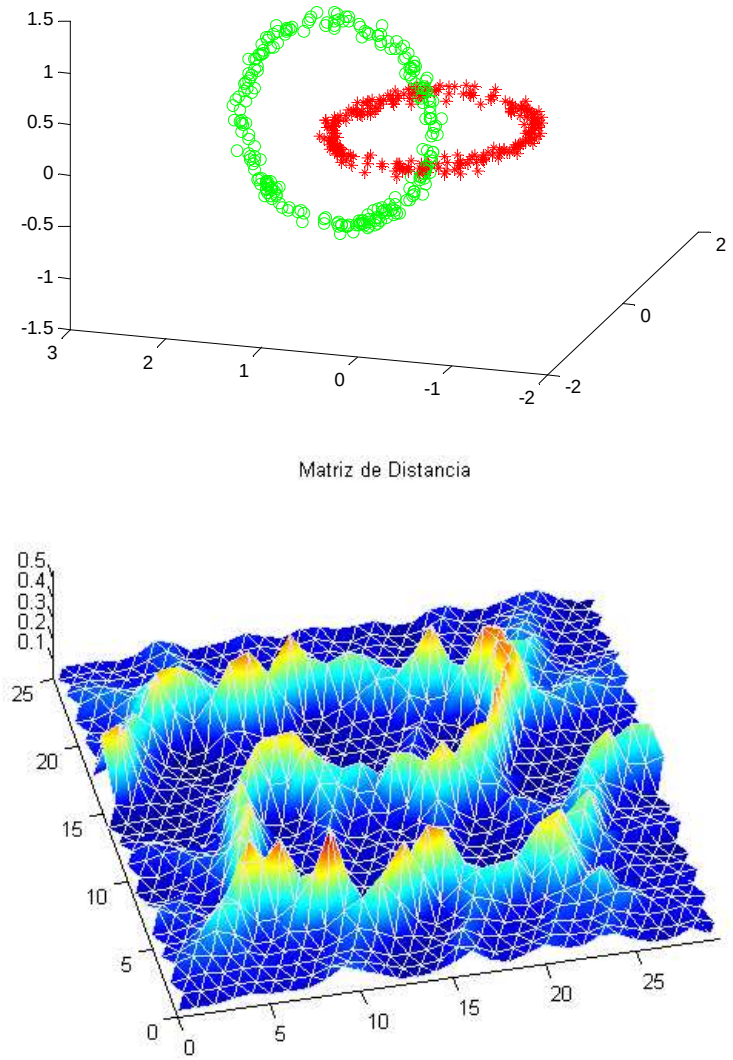
16.8 Ferramentas de visualização e discriminação



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



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



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

16.9 Ordenamento de pontos em espaços multidimensionais

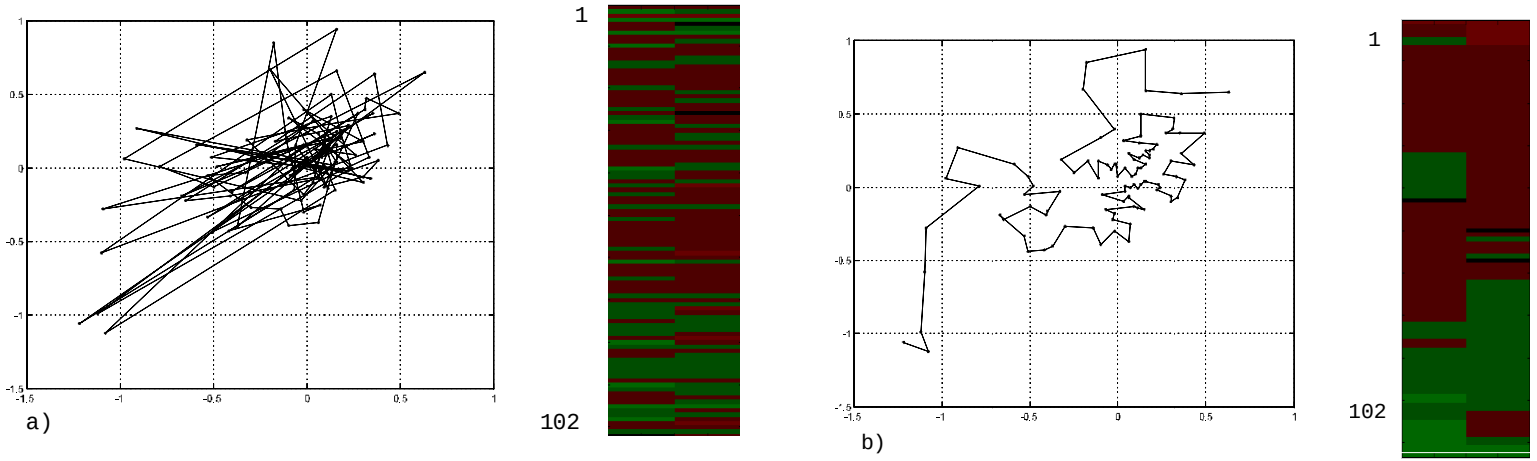
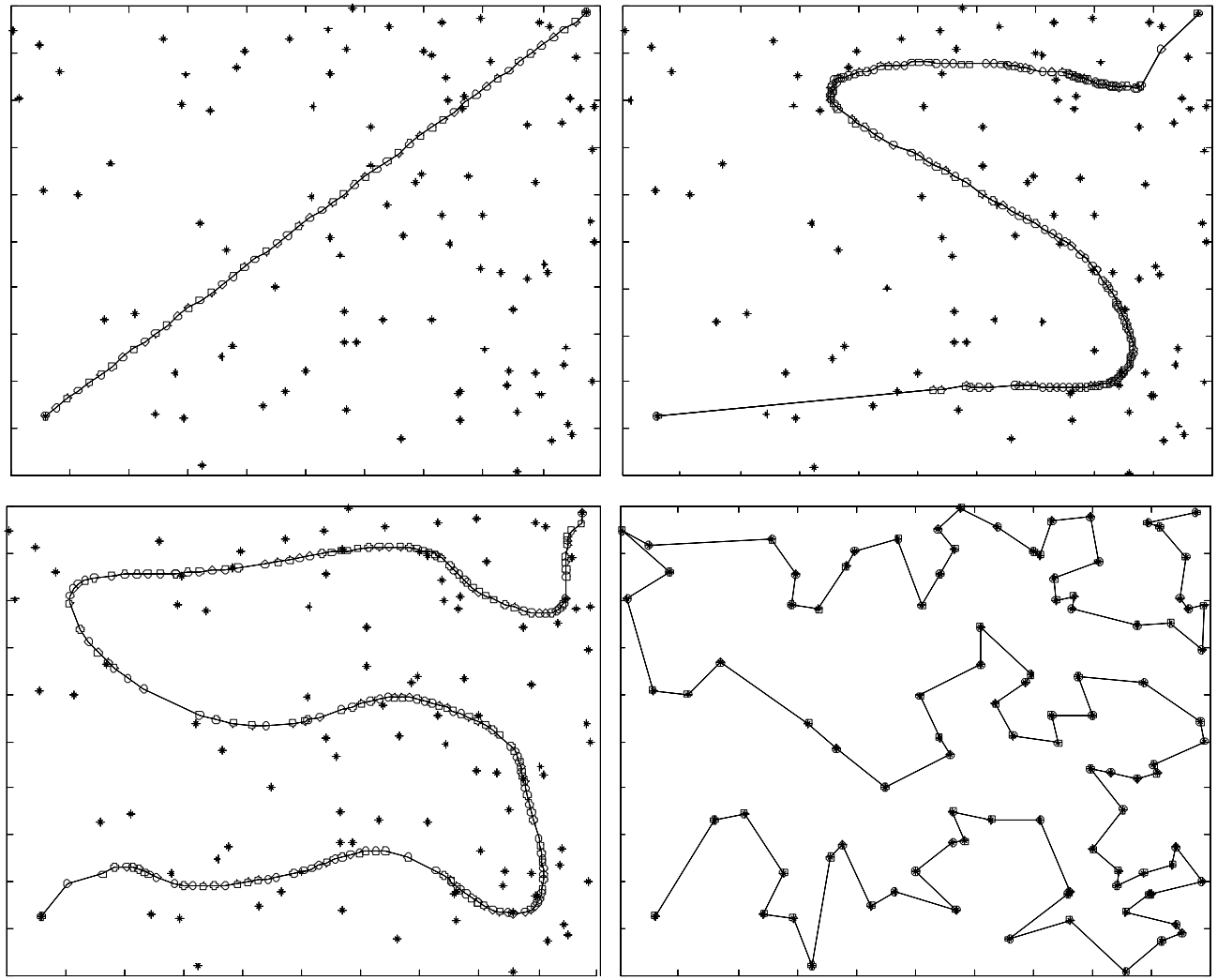


Figura 5 – Situação anterior (esquerda) e situação posterior (direita) ao ordenamento de pontos no $\Re^2$ (GOMES *et al.*, 2004).

- A extensão para pontos em espaços de maior dimensão é imediata (generalização do problema do caixeiro viajante)



Modo de operação (GOMES *et al.*, 2004)

16.10 Roteamento de veículos (múltiplos mapas auto-organizáveis)

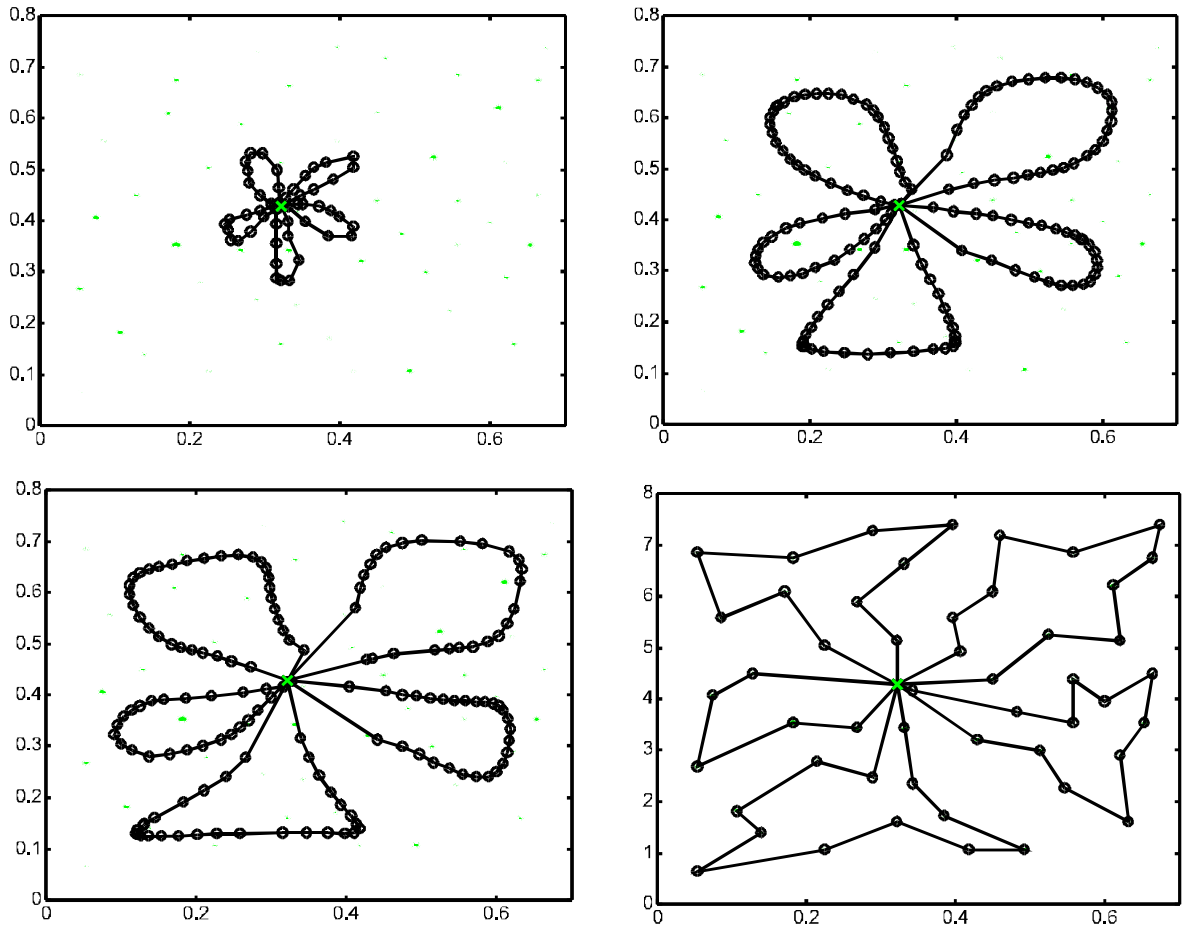
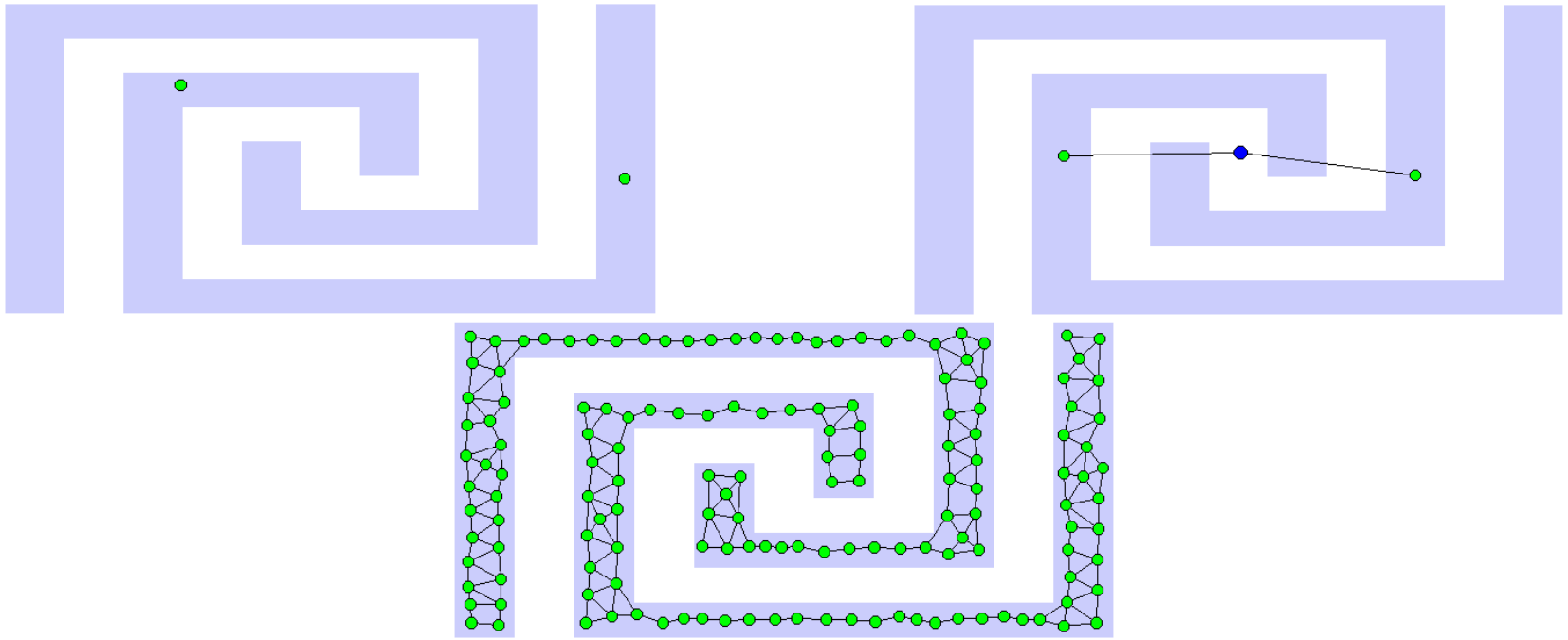


Figura 6 – Várias etapas do processo de auto-organização (GOMES & VON ZUBEN, 2002)

16.11 Mapas auto-organizáveis construtivos



Growing Neural Gas (FRITZKE, 1995)

16.12 Learning Vector Quantization

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

17. 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).

17.1 Rede neural de Hopfield

- 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)

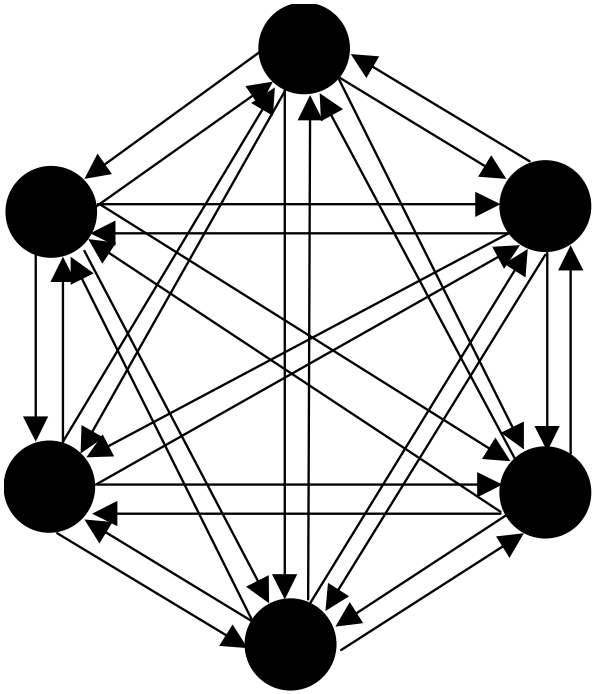


Figura 7 – Rede Neural de Hopfield: ênfase nas conexões

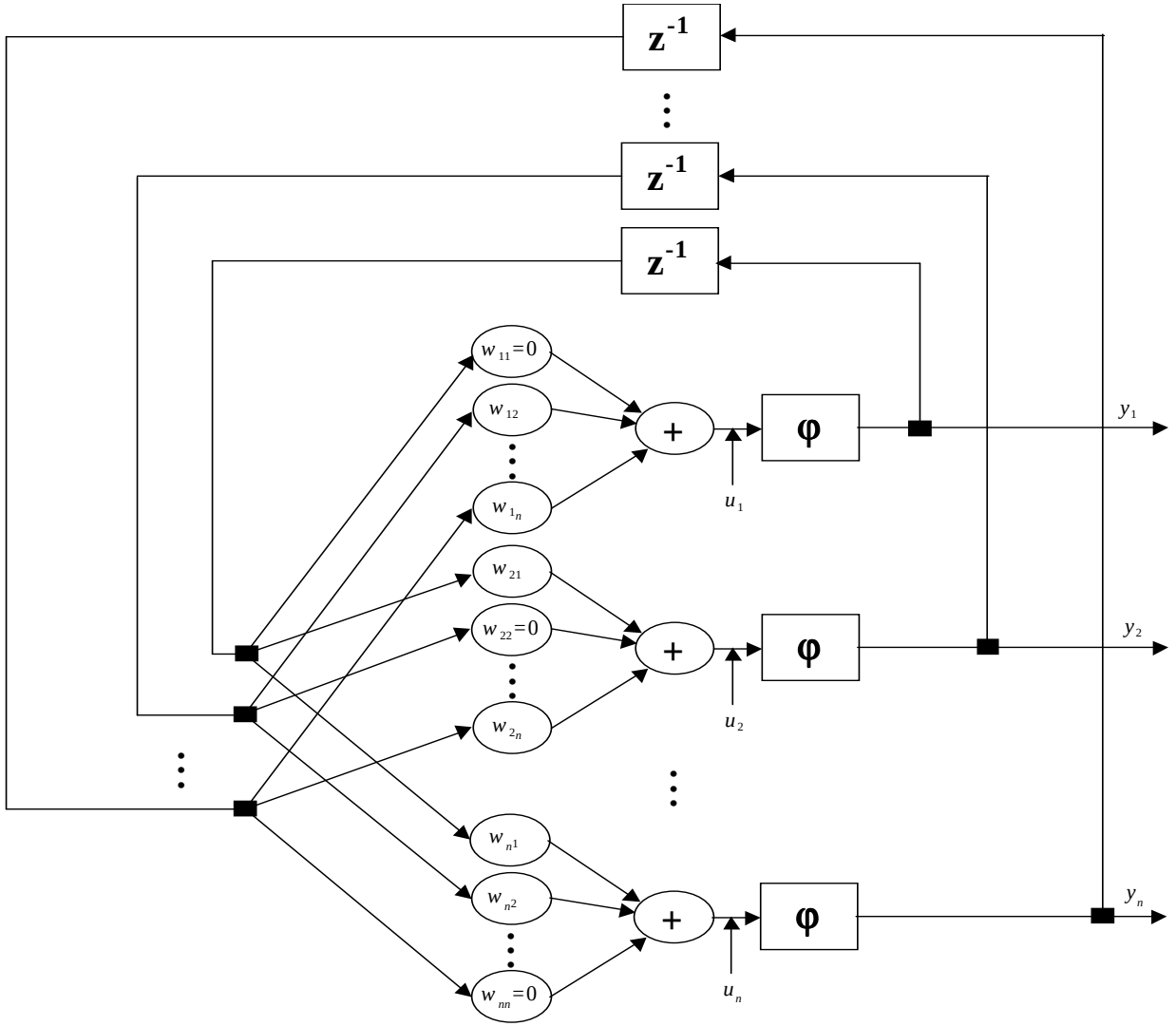


Figura 8 – Rede Neural de Hopfield: ênfase no processamento dinâmico (caso discreto)

- Percebe-se que a rede neural de Hopfield corresponde a um sistema dinâmico não-linear autônomo, pois não tem entrada externa.
- Em regime (após vencido o transitório), sistemas dinâmicos não-lineares autônomos, sejam de tempo discreto ou contínuo, podem apresentar quatro comportamentos possíveis: pontos de equilíbrio, ciclos limites (soluções periódicas), soluções quase-periódicas e caos. Um mesmo sistema dinâmico pode apresentar múltiplos casos desses quatro comportamentos, dependendo da condição inicial (estado inicial do sistema dinâmico).
- 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.
- Dentre todas as possibilidades de dinâmicas, buscam-se dinâmicas que expressem múltiplos pontos de equilíbrio em pontos específicos do espaço de estados.
- 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 → dinâmica de relaxação → comportamento de estado estacionário).

- Memória ↔ 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.

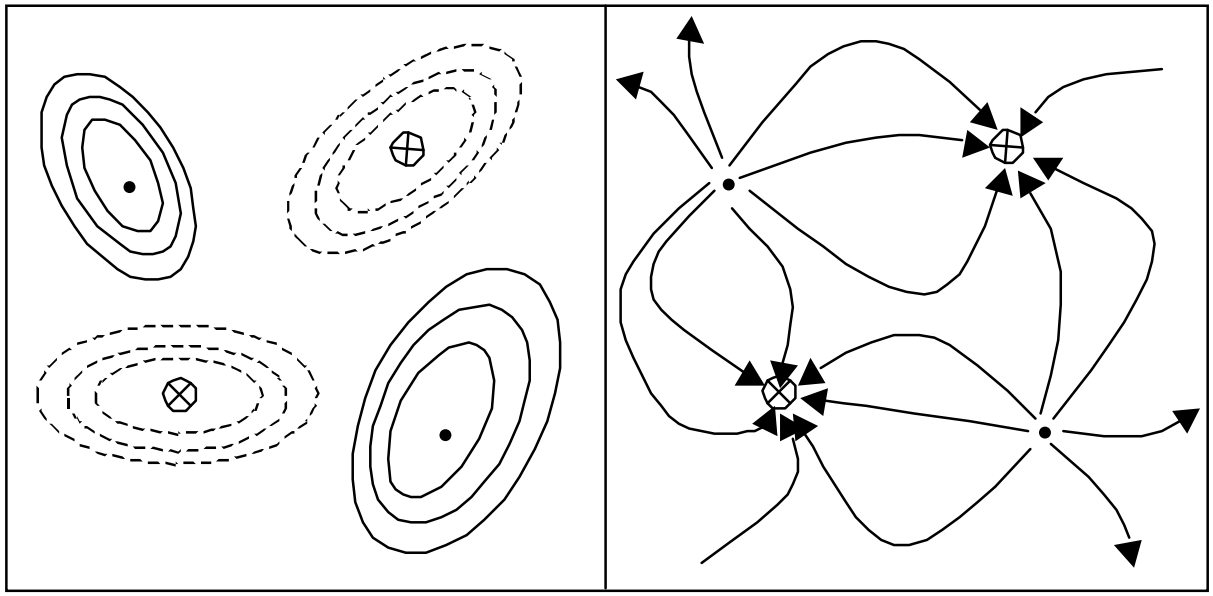


Figura 9 – Superfície de energia: pontos de equilíbrio e bases de atração

- 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.

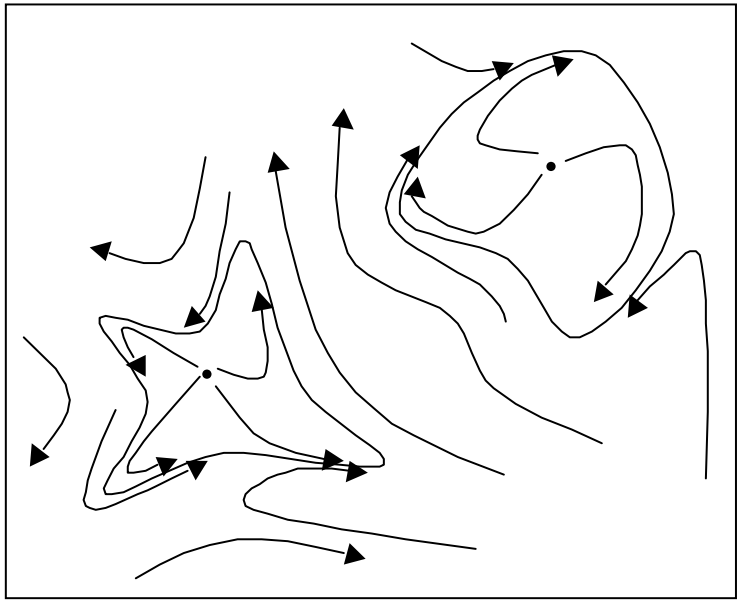


Figura 10 – Dinâmica de uma rede recorrente generalizada: presença de ciclos limites (não desejados no caso da rede de Hopfield)

17.2 Modelagem de sistemas dinâmicos lineares

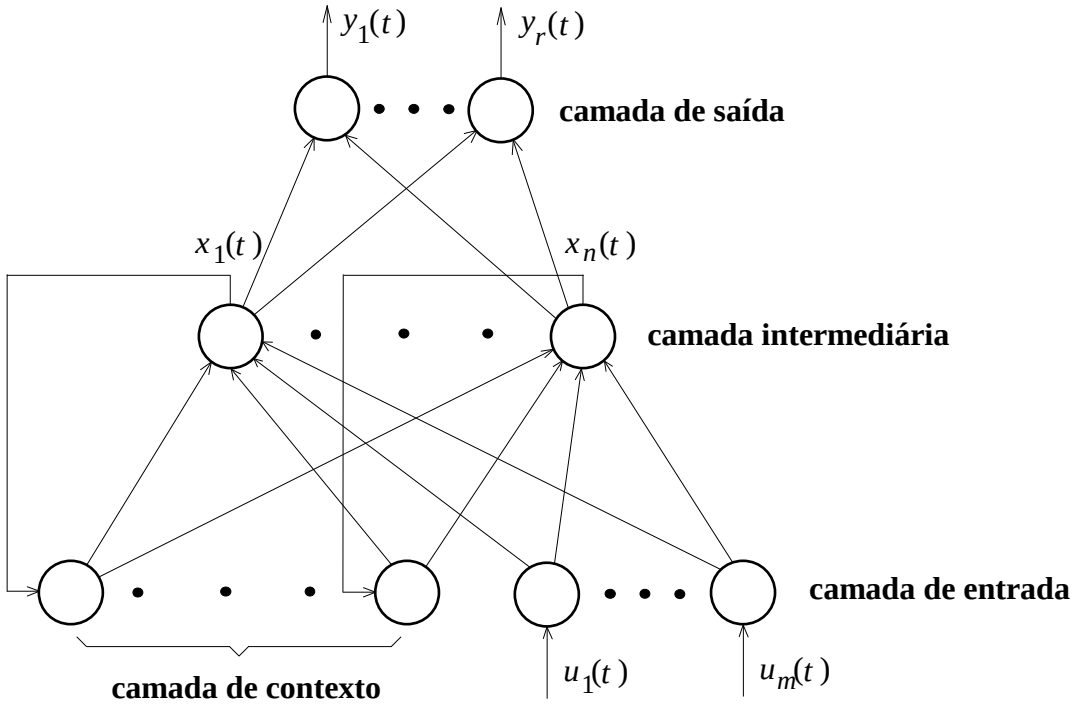


Figura 11 – Estrutura detalhada da Rede de Elman (ELMAN, 1990)

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

17.3 Modelagem de sistemas dinâmicos não-lineares

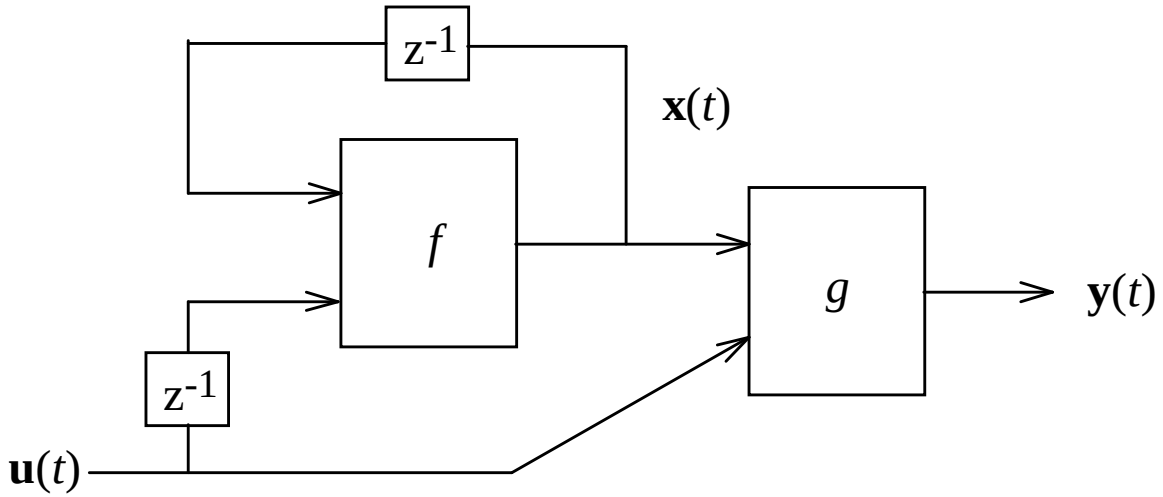


Figura 12 – Representação por espaço de estados de um sistema dinâmico não-linear

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

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

17.4 Excursão ilustrativa por algumas arquiteturas recorrentes

- Repare que o processo de treinamento vai envolver duas dinâmicas acopladas: a dinâmica da rede neural e a dinâmica do ajuste de pesos.

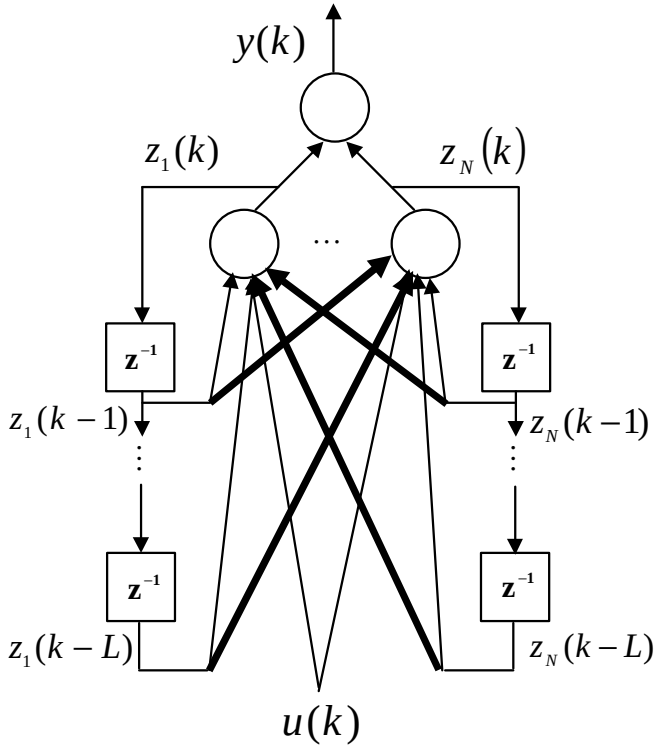


Figura 13 – *Globally recurrent neural network* (GRNN)

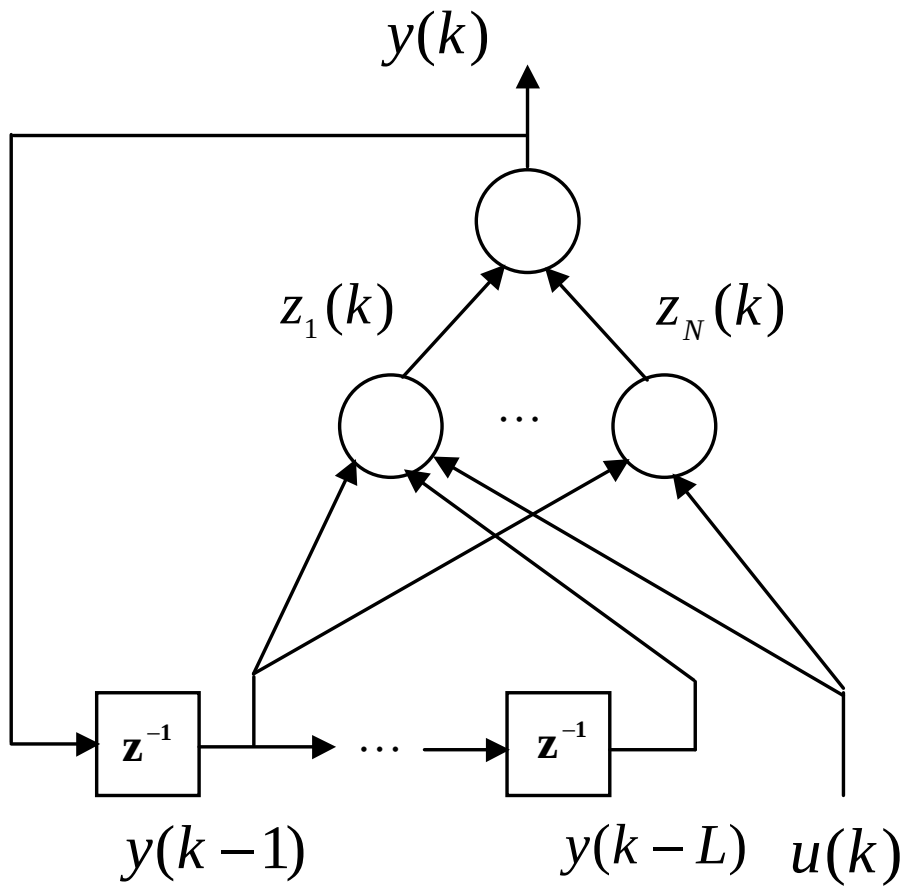


Figura 14 – *Output-feedback recurrent neural network* (OFRNN)

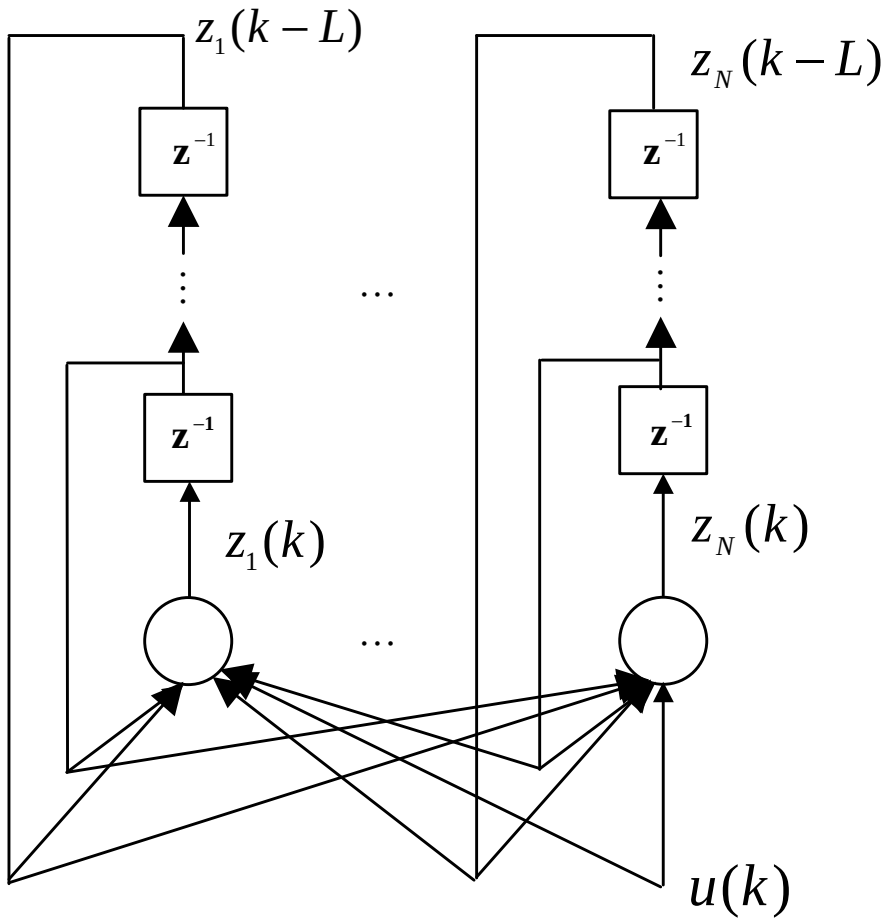


Figura 15 – *Fully recurrent neural network* (FRNN)

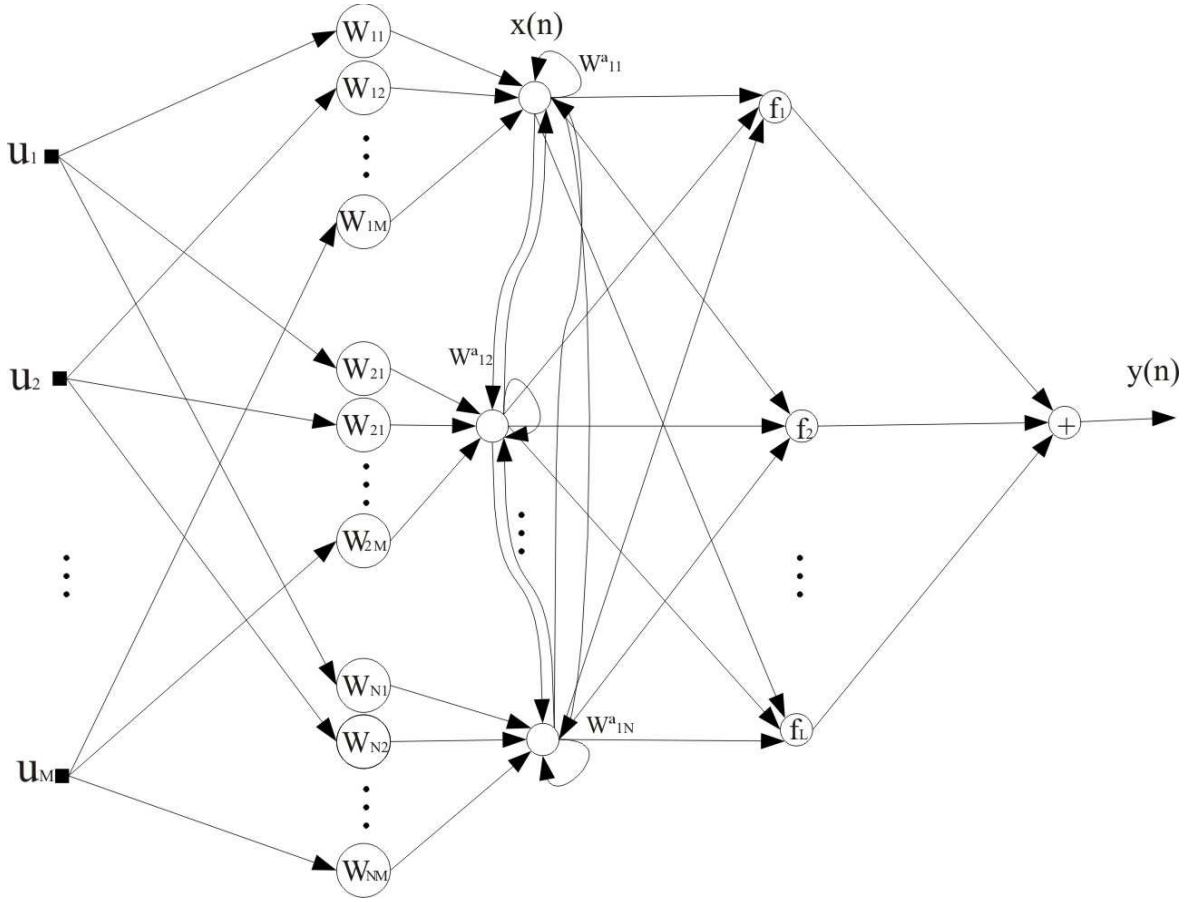


Figura 16 – *Echo state neural network*

(pesos da parte dinâmica da rede neural não são ajustáveis)

17.5 Predição de séries temporais: um ou múltiplos passos à frente

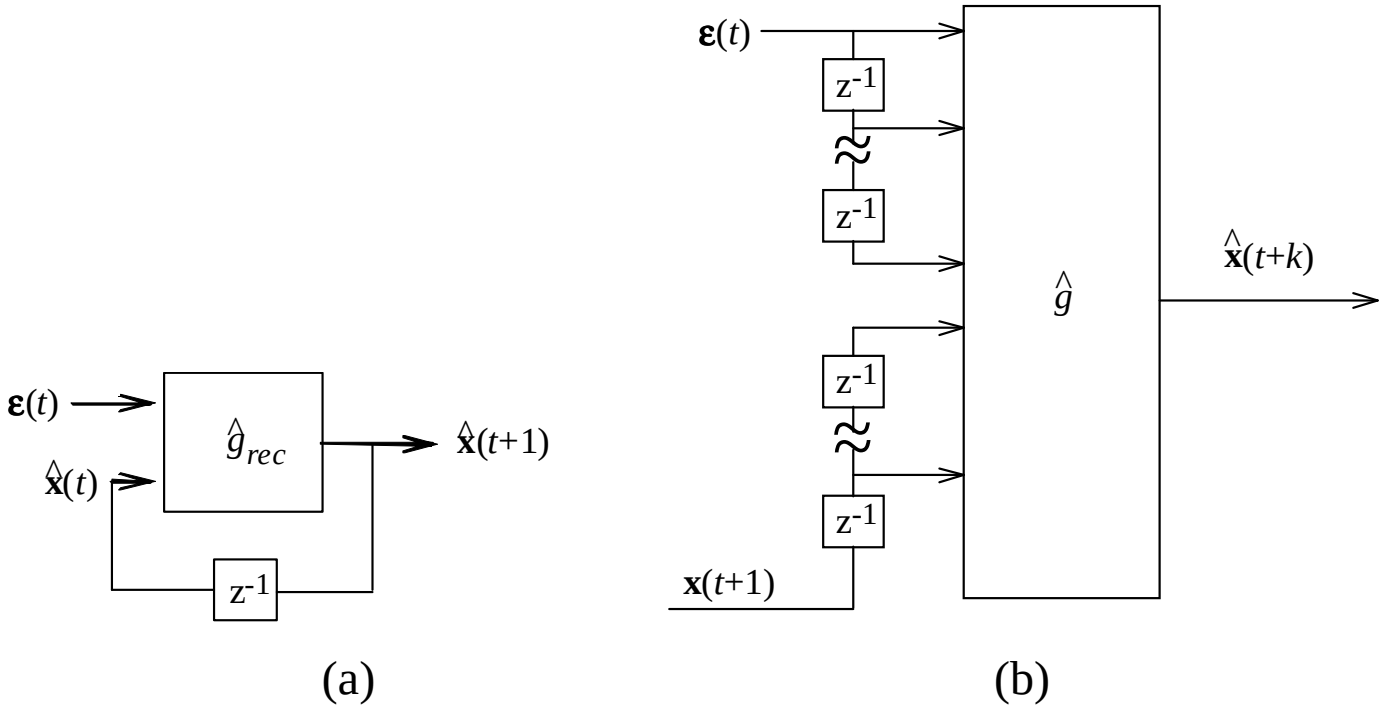


Figura 17 – Modelos para predição de séries temporais

(a) Predição de múltiplos passos à frente

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

18. Extreme Learning Machines (ELM)

- Correspondem a abordagens que definem arbitrariamente os pesos dos neurônios da camada intermediária de redes MLP, os quais não são passíveis de ajuste.
- Os pesos ajustáveis correspondem àqueles da camada de saída. Aplica-se o método dos quadrados mínimos regularizados, que permitem obter soluções que generalizam bem. Já há extensões para outros tipos de funções de ativação.

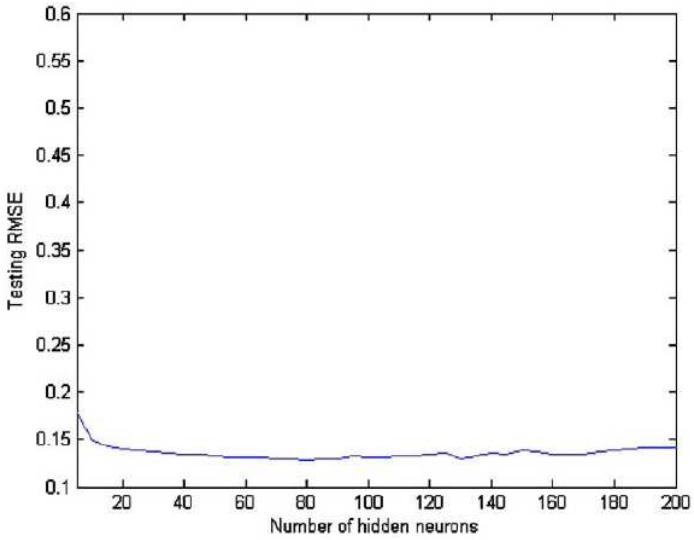


Fig. 3. The generalization performance of ELM is stable on a wide range of number of hidden nodes.

HUANG, G.B., ZHOU, H., DING, X. & ZHANG, R. (2012) "Extreme Learning Machine for Regression and Multiclass Classification", IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 42, no. 2, pp. 513-529.

19. LASSO & Elastic Nets

Suppose that we have data $(\mathbf{x}^i, y_i)$, $i = 1, 2, \ldots, N$, where $\mathbf{x}^i = (x_{i1}, \ldots, x_{ip})^T$ are the predictor variables and y_i are the responses. As in the usual regression set-up, we assume either that the observations are independent or that the y_is are conditionally independent given the x_{ij}s. We assume that the x_{ij} are standardized so that $\Sigma_i x_{ij}/N = 0$, $\Sigma_i x_{ij}^2/N = 1$.

Letting $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T$, the lasso estimate $(\hat{\alpha}, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min\left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \qquad \text{subject to} \sum_j |\beta_j| \leqslant t. \quad (1)$$

Here $t \geqslant 0$ is a tuning parameter. Now, for all t, the solution for α is $\hat{\alpha} = \bar{y}$. We can assume without loss of generality that $\bar{y} = 0$ and hence omit α.

We estimate the prediction error for the lasso procedure by fivefold cross-validation as described (for example) in chapter 17 of Efron and Tibshirani (1993). The lasso is indexed in terms of the normalized parameter $s = t/\Sigma\, \hat{\beta}_j^{\circ}$, and the prediction error is estimated over a grid of values of s from 0 to 1 inclusive. The value $\hat{s}$ yielding the lowest estimated PE is selected.

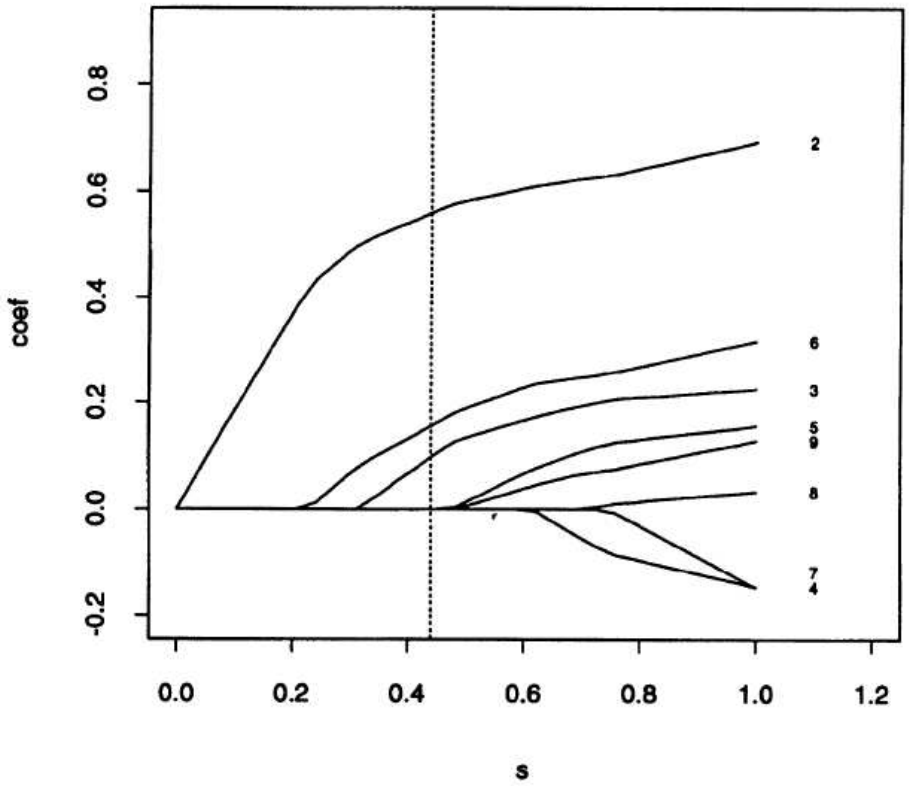


Fig. 5 shows the lasso estimates as a function of standardized bound $s = t/\Sigma|\hat{\beta}_j^{\circ}|$. Notice that the absolute value of each coefficient tends to 0 as s goes to 0. In this example, the curves decrease in a monotone fashion to 0, but this does not always happen in general. This lack of monotonicity is shared by ridge regression and subset regression, where for example the best subset of size 5 may not contain the best subset of size 4. The vertical broken line represents the model for $\hat{s} = 0.44$, the optimal value selected by generalized cross-validation. Roughly, this corresponds to keeping just under half of the predictors.

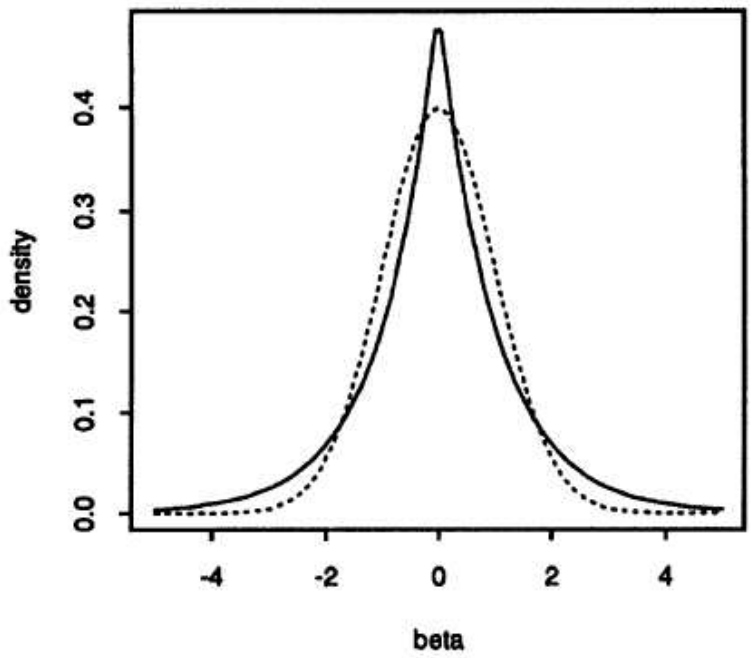


Fig. 7. Double-exponential density (———) and normal density (- - - - -): the former is the implicit prior used by the lasso; the latter by ridge regression

Elastic Net - Definição

We consider the usual setup for linear regression. We have a response variable $Y \in \mathbb{R}$ and a predictor vector $X \in \mathbb{R}^p$, and we approximate the regression function by a linear model $E(Y|X = x) = \beta_0 + x^T\beta$. We have N observation pairs (x_i, y_i). For simplicity we assume the x_{ij} are standardized: $\sum_{i=1}^N x_{ij} = 0$, $\frac{1}{N}\sum_{i=1}^N x_{ij}^2 = 1$, for $j = 1, \ldots, p$. Our algorithms generalize naturally to the unstandardized case. The elastic net solves the following problem

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \left[\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda P_\alpha(\beta)\right], \qquad (1)$$

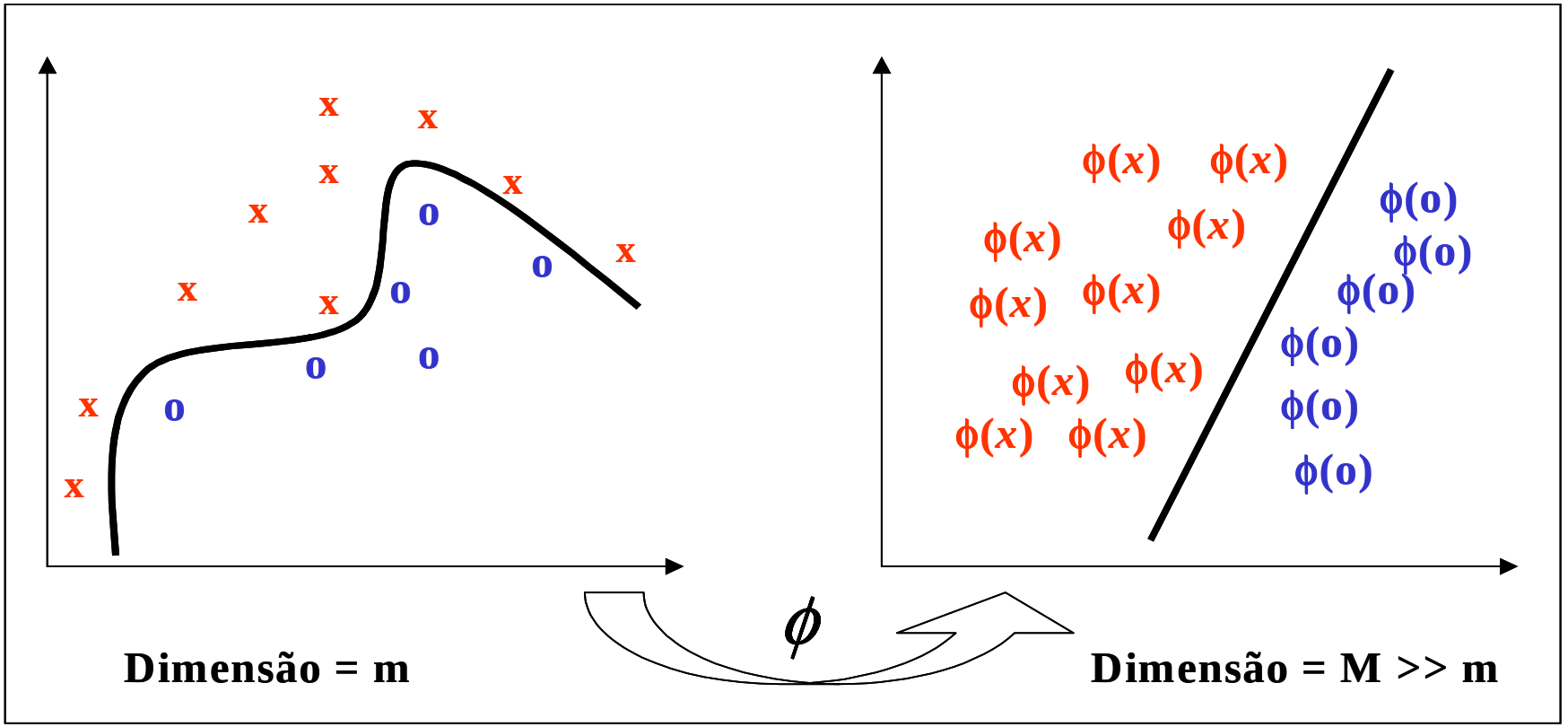
where

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}||\beta||_{\ell_2}^2 + \alpha||\beta||_{\ell_1} \qquad (2)$$

$$= \sum_{j=1}^{p}\left[\tfrac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right] \qquad (3)$$

is the *elastic-net penalty* [Zou and Hastie, 2005]. P_α is a compromise between the ridge-regression penalty ($\alpha = 0$) and the lasso penalty ($\alpha = 1$). This penalty is particularly useful in the $p \gg N$ situation, or any situation where there are many correlated predictor variables.

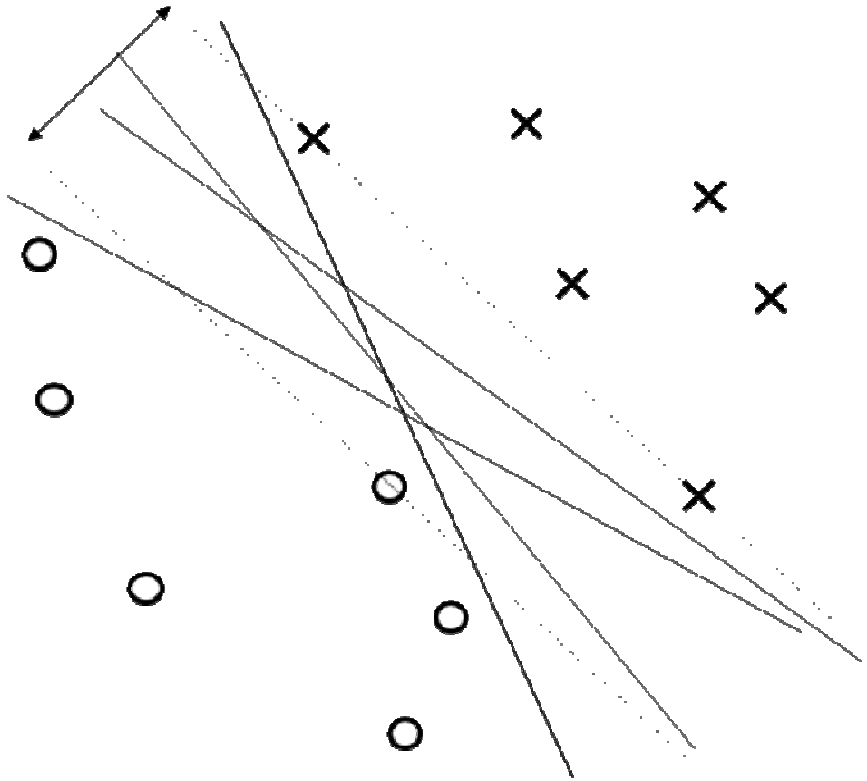
20. Máquinas de Vetores-Suporte (Support Vector Machines – SVM)

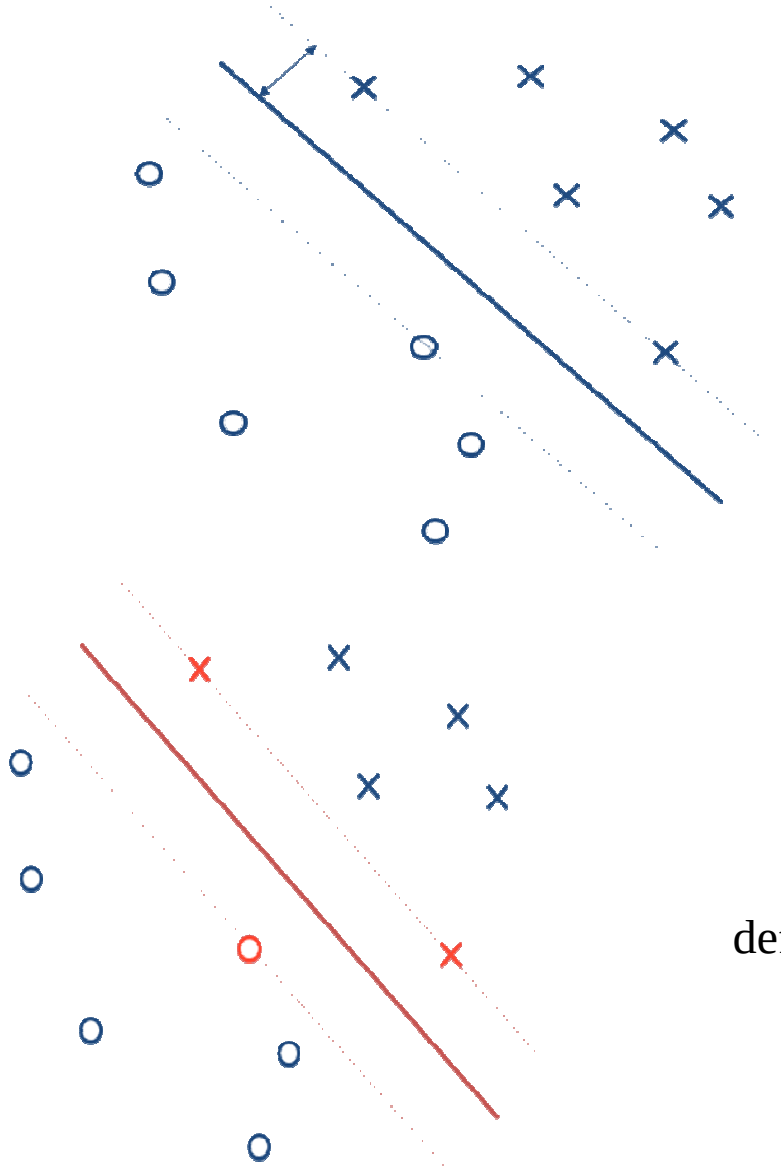


- Supera dois problemas:
  1. Como controlar a complexidade do espaço de hipóteses?
  2. Como evitar o problema de mínimos locais no ajuste de parâmetros?

20.1 Hiperplano de máxima margem de separação

- Há infinitos hiperplanos que separam os dados no espaço de características (espaço expandido). Qual deles escolher?

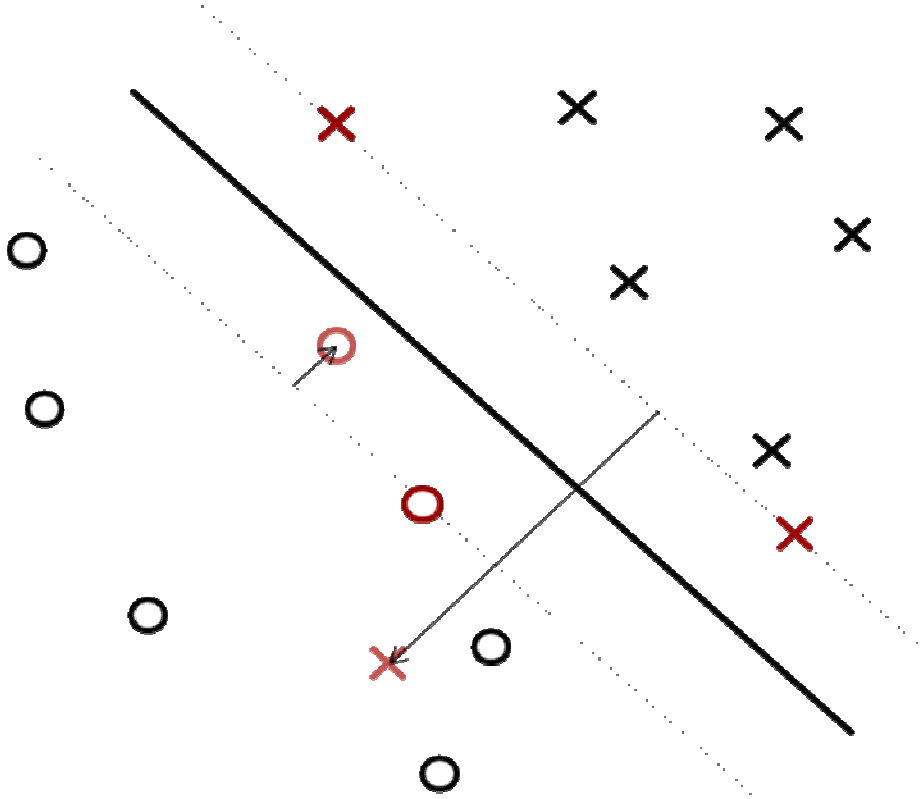




$$y = \text{sign}(\mathbf{w} \cdot \mathbf{X} + b)$$

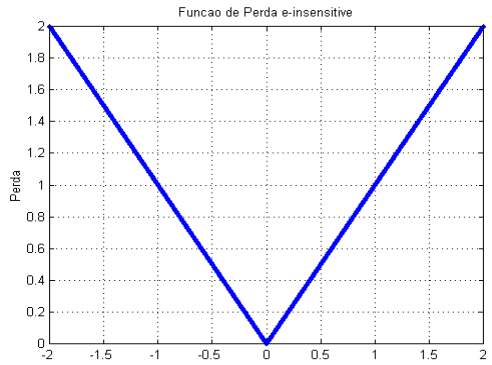
$$\min_{\mathbf{w}} : \|\mathbf{w}\|$$

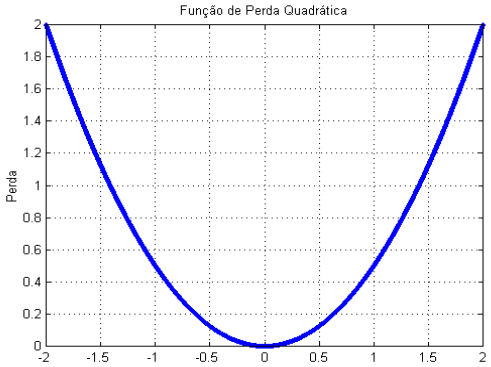
Os vetores-suporte são os pontos que definem o hiperplano de máxima margem.

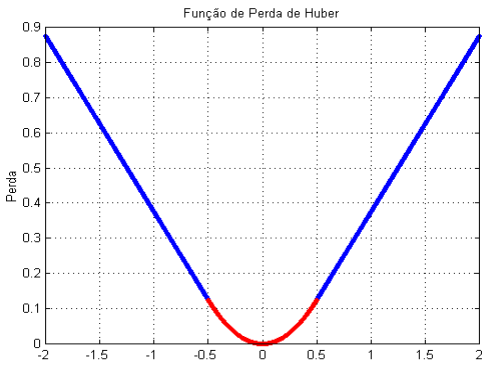


Problema primal:

$$\Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} F(\xi_i) \qquad \text{s.a.} \quad y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \geq 1-\xi_i, \qquad i = 1,\ldots,N$$







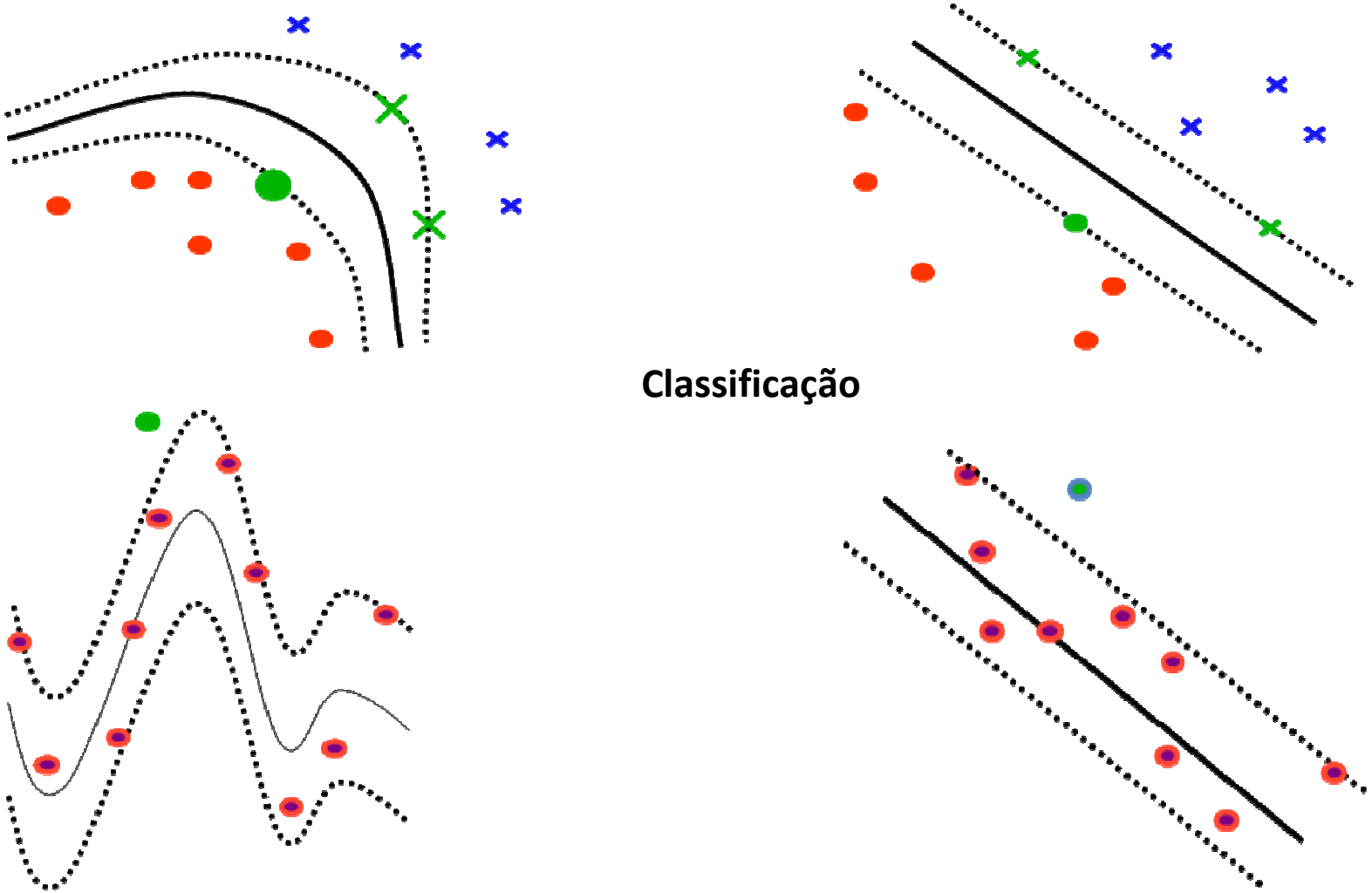
Problema dual:

$$\min_{\alpha} W(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i$$

s.a. $\quad 0 \le \alpha_i \le C, \qquad i = 1,\ldots,N$

$$\sum_{i=1}^{N}\alpha_i y_i = 0$$

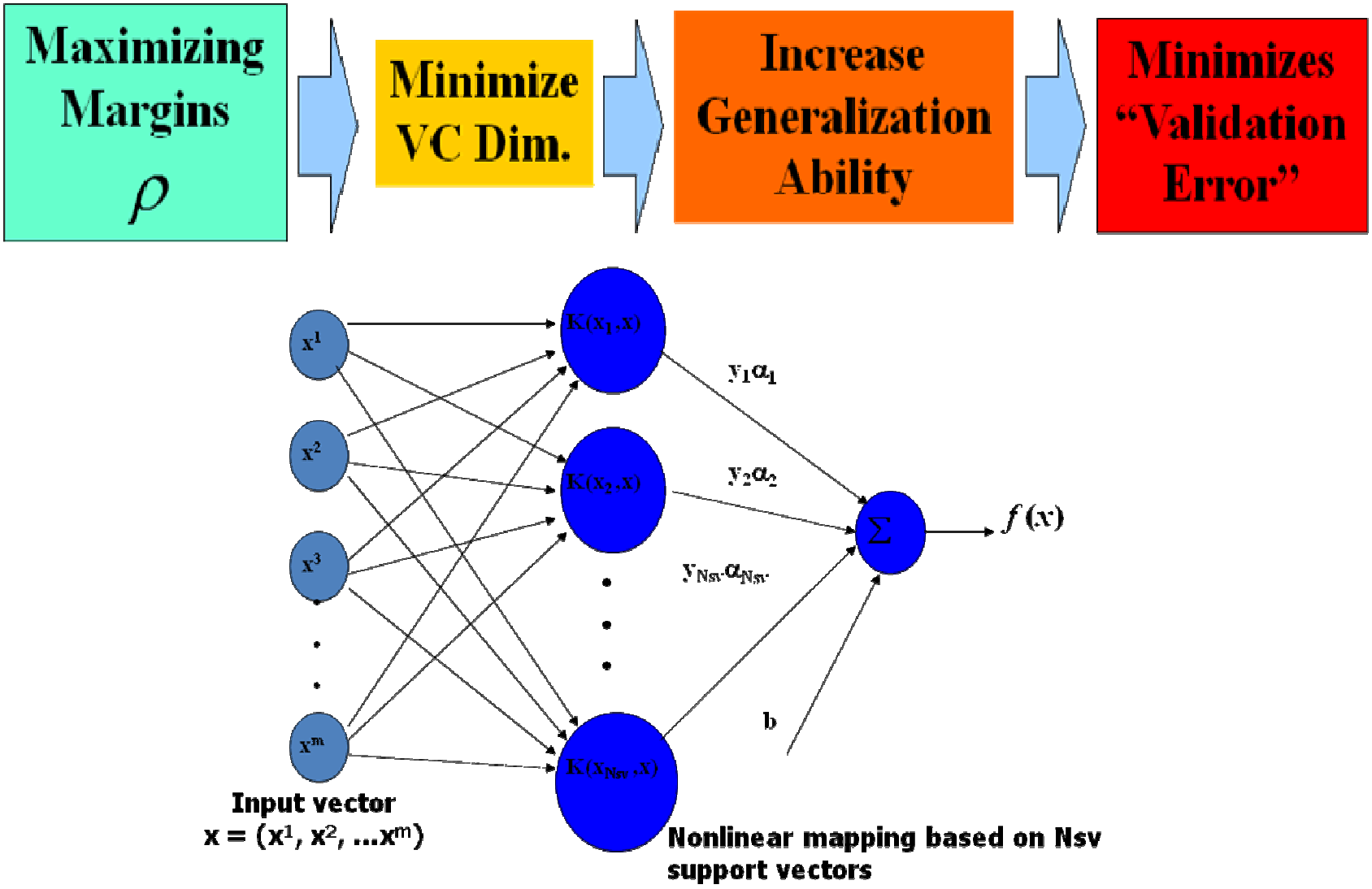
20.2 Classificação × Regressão



Classificação

Regressão

20.3 Embasamento teórico



20.4 Funções de kernel mais empregadas

i. Linear

$$K(x, y) = x \cdot y$$

ii. Polynomial

$$K(x, y) = (x \cdot y + 1)^d$$

iii. Gaussian Radial Basis Function

$$K(x, y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

iv. Exponential Radial Basis Function

$$K(x, y) = \exp\left(-\frac{|x-y|}{2\sigma^2}\right)$$

v. Sigmoid

$$K(x, y) = \tanh(b(x \cdot y) + c)$$

vi. Fourier Series

$$K(x, y) = \frac{sin(N + \frac{1}{2})(x-y)}{sin\left(\frac{1}{2}(x-y)\right)}$$

vii. Linear Splines

$$K(x, y) = 1 + xy + xy\min(x, y) - \frac{(x+y)}{2}(\min(x, y))^2 + \frac{1}{3}(\min(x, y))^3$$

viii. Bn-splines

$$K(x, y) = B_{2n+1}(x-y)$$

21. Referências

BARLOW, H. B. "Unsupervised learning", *Neural Computation*, 1: 295-311, 1989.

BATTITI, R. First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method. *Neural Computation*, vol. 4, no. 2, pp. 141-166, 1992.

BECKER, S. & PLUMBLEY, M. "Unsupervised neural network learning procedures for feature extraction and classification", *International Journal of Applied Intelligence*, 6: 185-203, 1996.

BISHOP, C. Exact Calculation of the Hessian Matrix for the Multilayer Perceptron. *Neural Computation*, vol. 4, no. 4, pp. 494-501, 1992.

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.

COVER, T.M. & HART, P.E. "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, 13:21-27, 1967.

CYBENKO, G. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303-314, 1989.

ELMAN, J. L. Finding structure in time. *Cognitive Science*, vol. 14, p. 179-211, 1990.

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.

HARTMAN, E.J., KEELER, J.D., KOWALSKI, J.M. Layered neural networks with gaussian hidden units as universal approximators. *Neural Computation*, vol. 2, no. 2, pp. 210-215, 1990.

HAYKIN, S. "Neural Networks: A Comprehensive Foundation", 2nd edition, Prentice-Hall, 1999.

HEBB, D. O. "The Organization of Behavior", Wiley, 1949.

HINTON, G.E. "Connectionist learning procedures", *Artificial Intelligence*, 40: 185-234, 1989.

HINTON, G. E. & SEJNOWSKI, T.J. "Learning and relearning in Boltzmann machines", *in* D. E. Rumelhart, J. L. McClelland & The PDP Research Group (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, vol. 1, pp. 282-317, 1986.

HORNIK, K., STINCHCOMBE, M., WHITE, H. Multi-layer feedforward networks are universal approximators. *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

HORNIK, K., STINCHCOMBE, M., WHITE, H. Universal approximation of an unknown function and its derivatives using multilayer feedforward networks. *Neural Networks*, vol. 3, no. 5, pp. 551-560, 1990.

HORNIK, K., STINCHCOMBE, M., WHITE, H., AUER, P. Degree of Approximation Results for Feedforward Networks Approximating Unknown Mappings and Their Derivatives. *Neural Computation*, vol. 6, no. 6, pp. 1262-1275, 1994.

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.

KOHONEN, T. "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, 43:59-69, 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.

LINSKER, R. "Self-organization in a perceptual network", *Computer*, 21: 105-128, 1988.

MACKAY, D. M. "The epistemological problem for automata", *in* C. E. Shannon & J. McCarthy (eds.) *Automata Studies*, Princeton University Press, pp. 235-251, 1956.

MACQUEEN, J. "Some methods for classification and analysis of multivariate observation", *in* L.M. LeCun and J Neyman (eds.) *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1:281-297, 1967.

MARR, D. "A theory for cerebral neocortex", *Proceedings of the Royal Society of London, Series B*, 176: 161-234, 1970.

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.

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.

PARK, J., SANDBERG, I.W. Universal approximation using radial basis function networks. *Neural Computation*, vol. 3, no. 2, pp. 246-257, 1991.

POTVIN, J.-I. & ROBILLARD, C. "Clustering for Vehicle Routing wih 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.

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.

Muitos não fizeram quando deviam porque não quiseram quando podiam.

François Rabelais (1483/94-1553)