

Redes Neurais Artificiais – Parte 2

Índice

1.	Contribuição de cada neurônio em uma rede MLP	2
2.	O papel dos pesos sinápticos	10
3.	Aprendizado a partir de dados amostrados (revisão)	13
4.	Aprendizado supervisionado em RBF (revisão)	19
5.	O problema do OU-exclusivo em MLP	20
6.	Processo Iterativo para MLP – Método Padrão-a-Padrão	27
7.	Processo Iterativo para MLP – Método em Lote ou Batelada.....	28
8.	Exemplos de auto-organização na natureza	29
9.	O Jogo da Vida	34
10.	Arranjo bidimensional de um mapa de Kohonen (revisão).....	35
11.	Um passo de ajuste num arranjo unidimensional (revisão).....	36
12.	O caso Phineas Gage	38
13.	Monitorando a atividade cerebral	39

1. Contribuição de cada neurônio em uma rede MLP

- O mapeamento não-linear realizado por uma rede neural do tipo perceptron de uma camada intermediária é uma **combinação linear de funções de expansão ortogonal**, ou seja, funções que têm a forma de tangente hiperbólica em uma direção e são constantes nas demais direções ortogonais a esta única direção em que a forma da função se manifesta.
- Como um exemplo, vamos tomar amostras de um mapeamento do \mathcal{R}^2 para o \mathcal{R}^1 , e utilizar uma rede neural com 5 neurônios na camada intermediária para buscar aproximar este mapeamento, o qual pode ser visualizado no \mathcal{R}^3 .
- Os pesos sinápticos resultantes do processo de treinamento estão apresentados na sequência, sendo que a rede neural tem ao todo $3 \times 5 + 6 \times 1 = 21$ pesos ajustáveis. São 2 entradas, 5 neurônios na camada intermediária e 1 saída, mais as entradas constantes (entradas de polarização) de todos os 6 neurônios da rede neural.

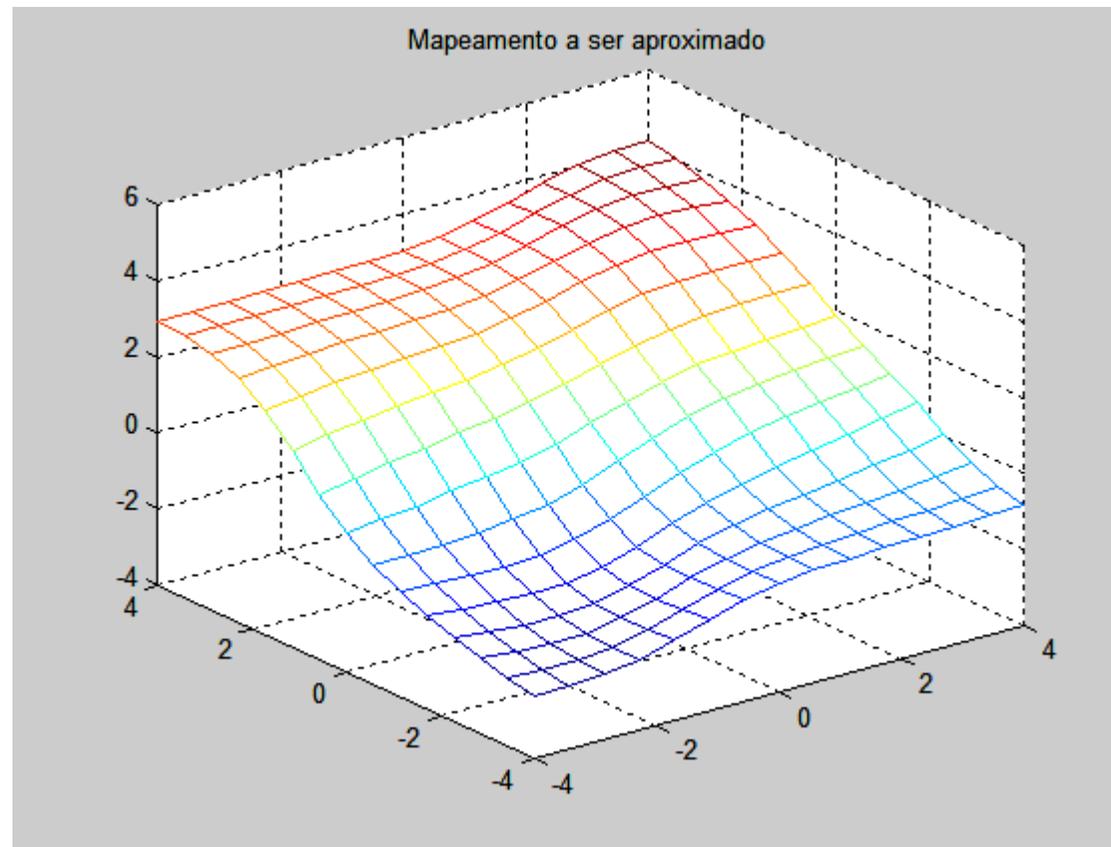
- Pesos sinápticos da camada intermediária (cada coluna representa os pesos de um neurônio):

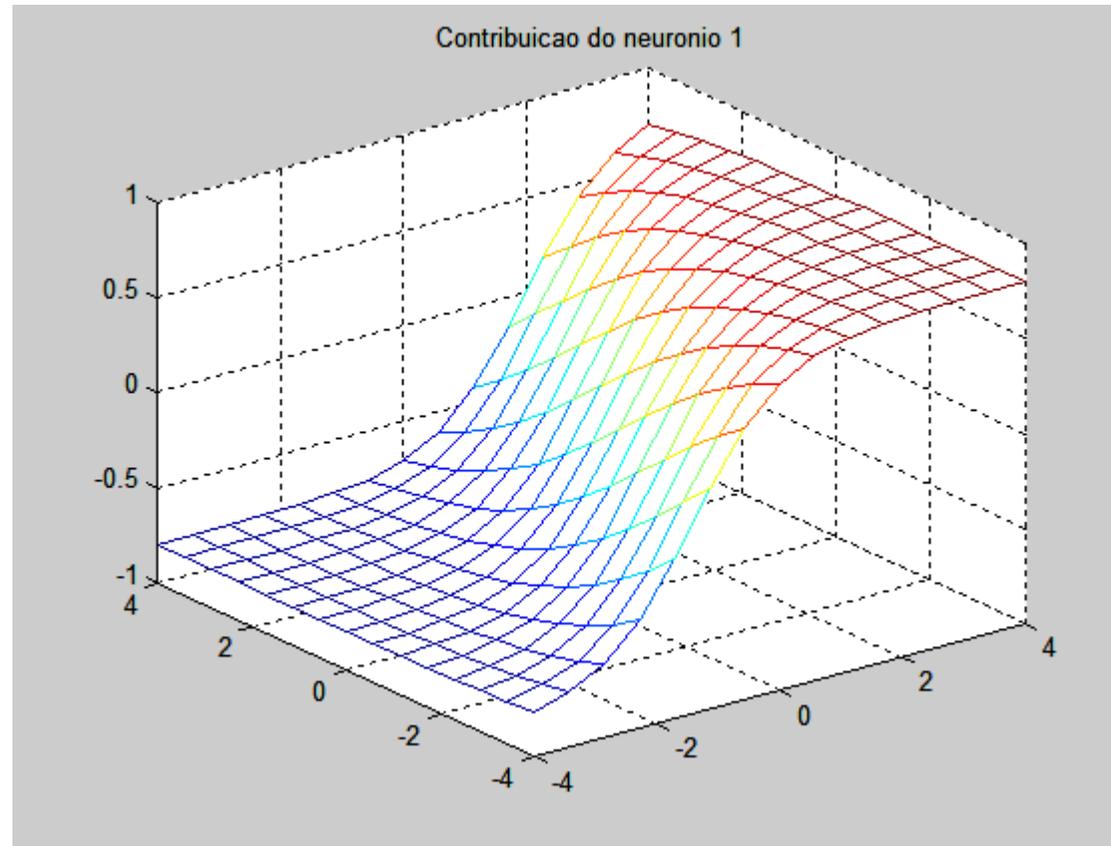
-0.20008939714462 -0.70051908010040 0.39699221844113 -0.10003863267278 0.69606262467282
0.70018168528932 0.10015860417667 0.19860028823484 -0.29996195303800 0.29869112235480
-0.30006398146599 0.80022209855791 0.49372400421686 0.50005427222963 0.89515012131364

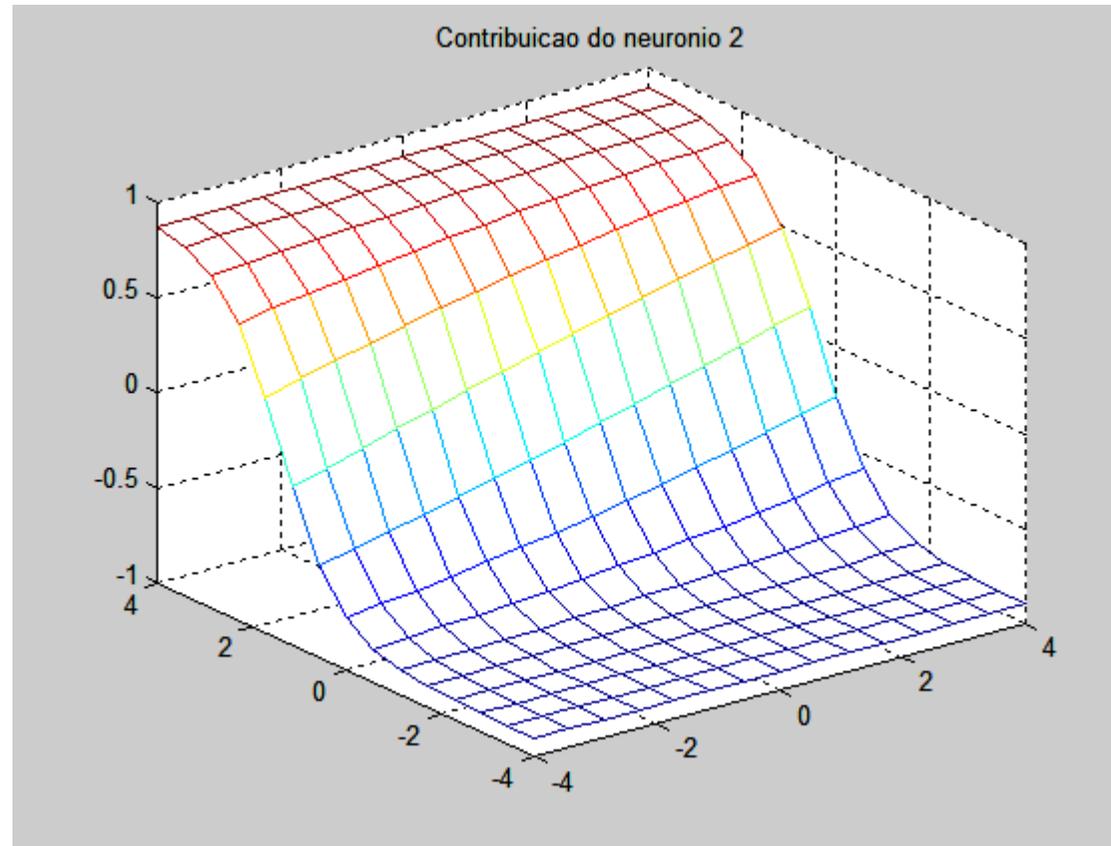
- Pesos sinápticos da camada de saída:

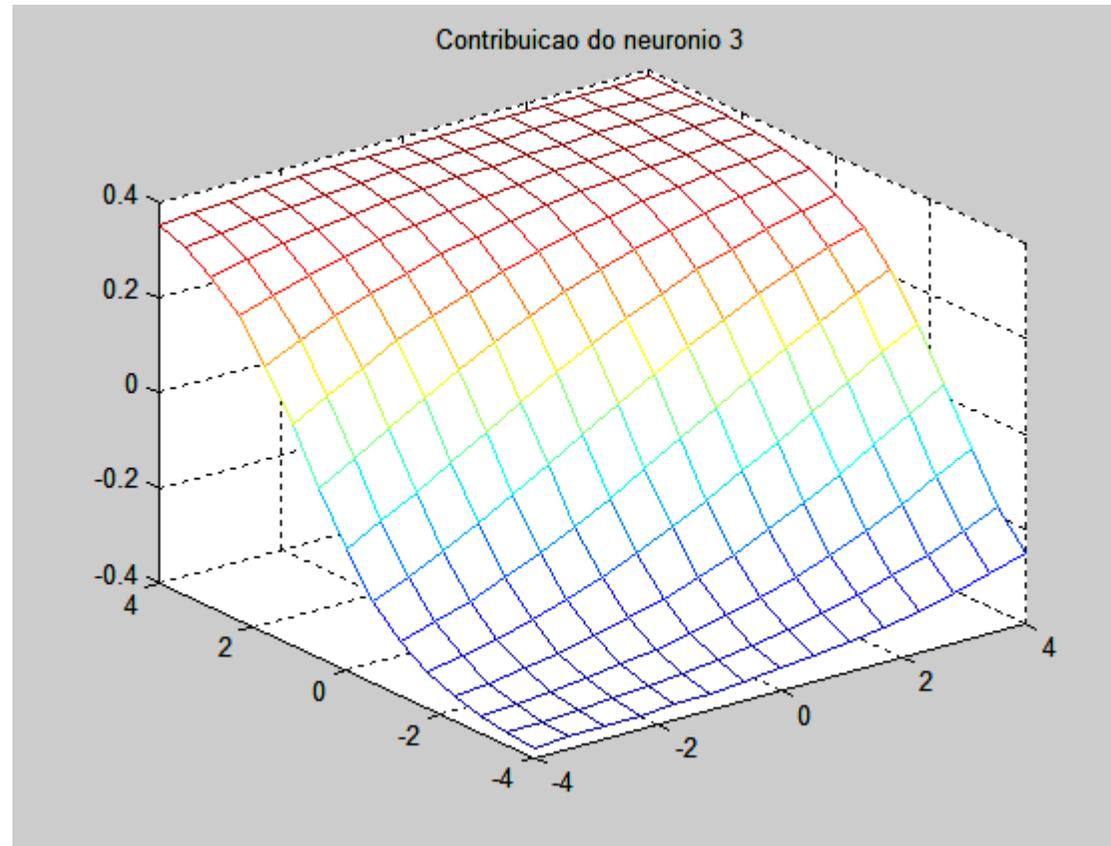
0.99989340388393
0.79971888341317
0.90007841696146
0.38564988369799
0.79996881679466
0.71442550587375

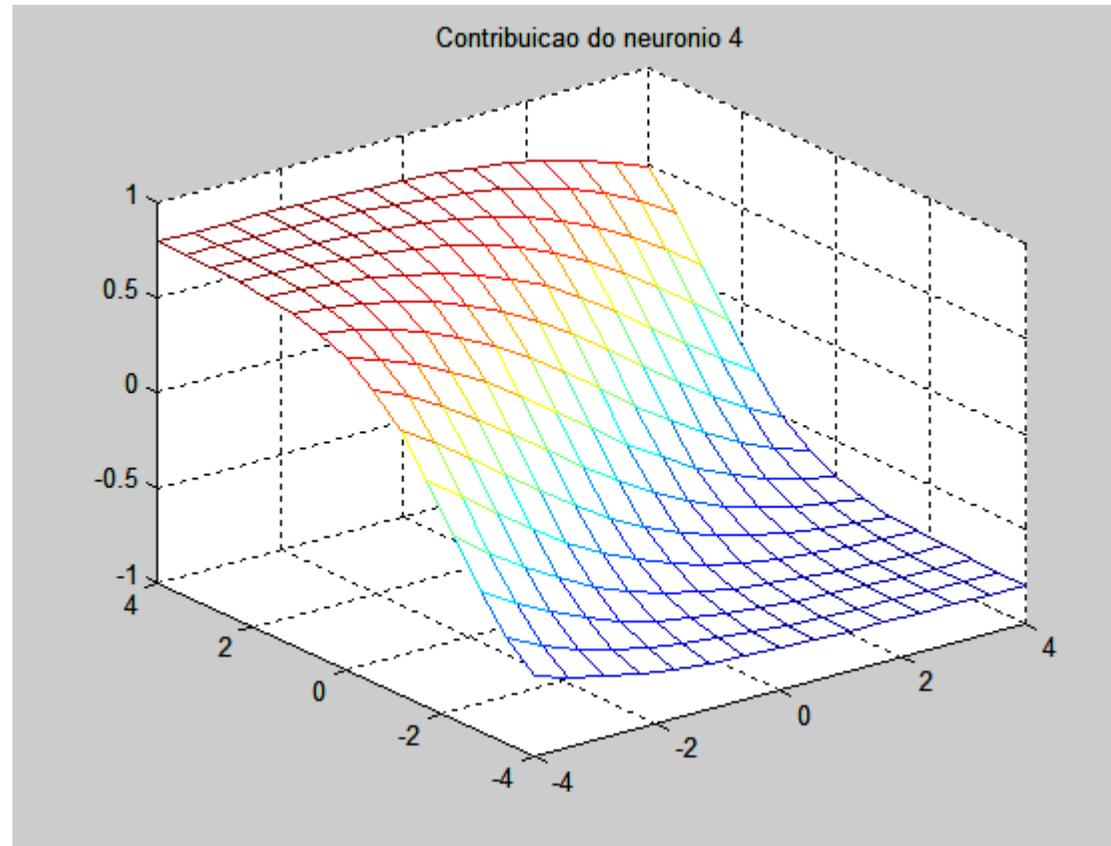
- Obs: O peso de polarização é o primeiro peso de cada neurônio.

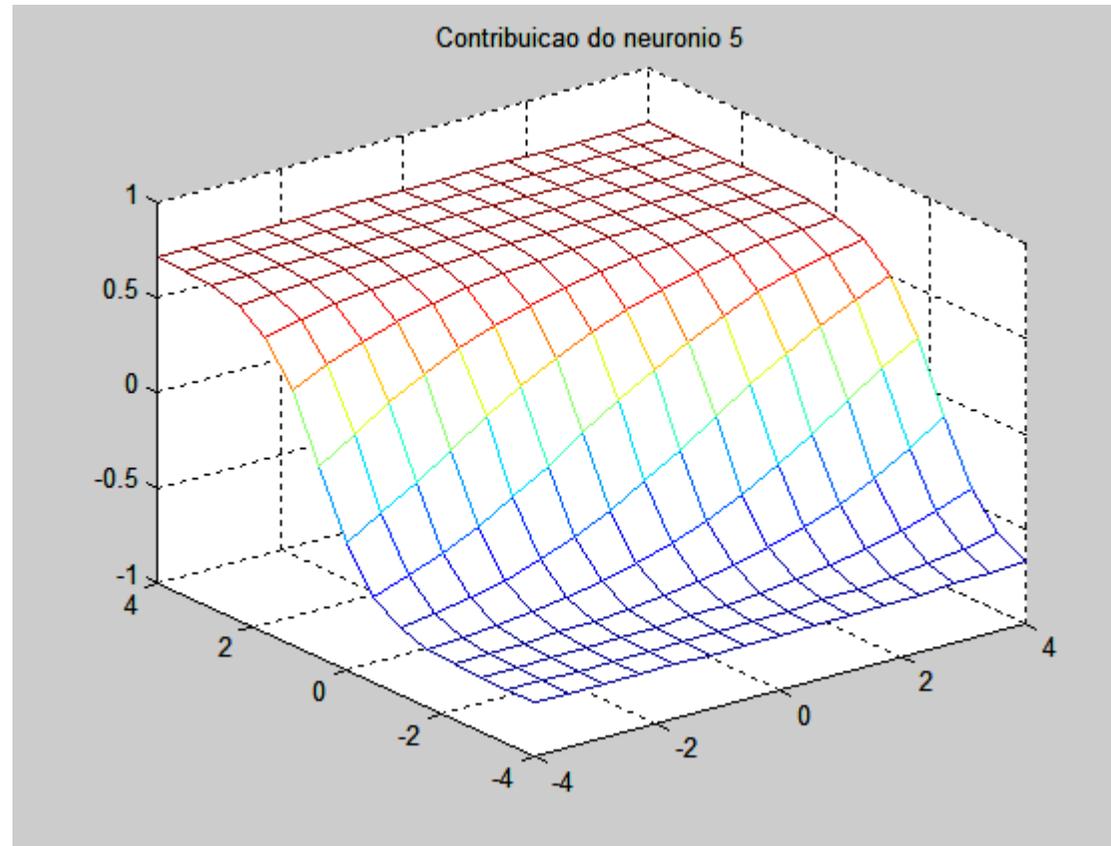






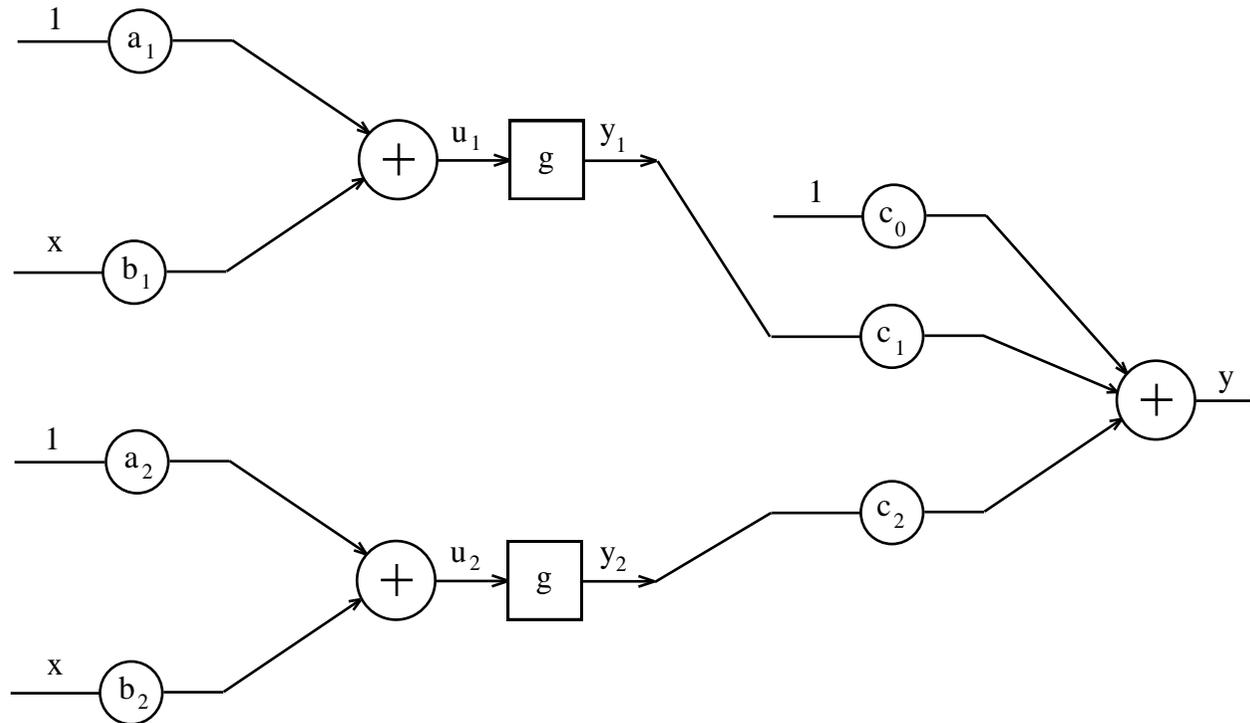






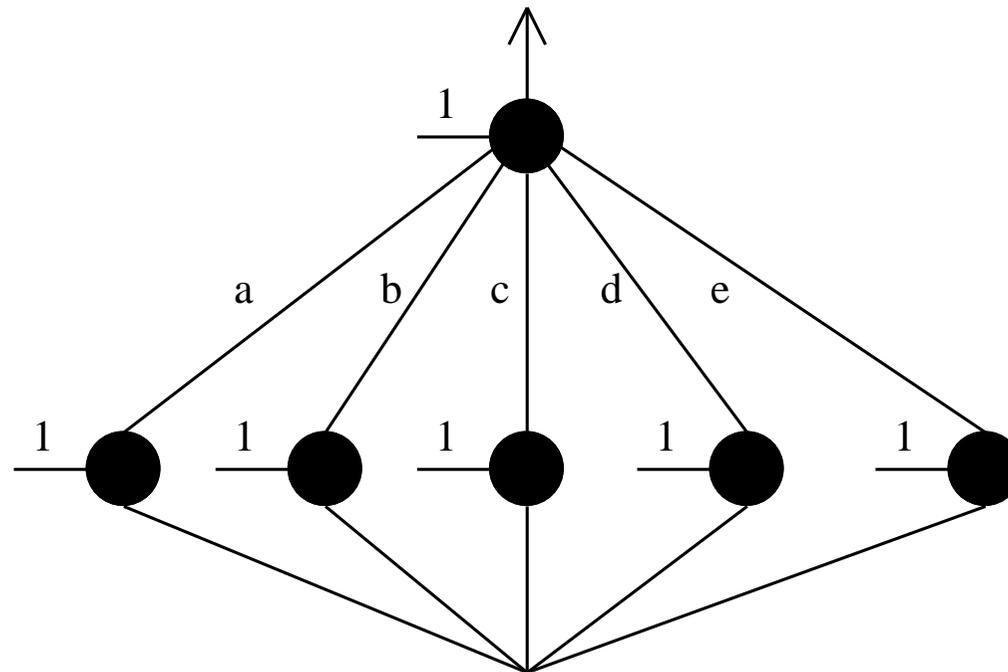
2. O papel dos pesos sinápticos

$$y = c_0 + \sum_{n=1}^p c_n g(b_n x + a_n)$$



$$y = c_0 + c_1 g(b_1 x + a_1) + c_2 g(b_2 x + a_2) \Rightarrow \begin{cases} a : \text{deslocamento no eixo } x \\ b : \text{inclinação da sigmóide} \\ c : \text{amplitude da sigmóide} \end{cases}$$

Exemplo: Forma “construtiva” de aproximação de um mapeamento não-linear empregando neurônios com função de ativação do tipo tangente hiperbólica. Exemplo considerando um único estímulo de entrada.



$$f(\mathbf{w}) = \underbrace{c_1 g(b_1 x + a_1)}_a + \underbrace{c_2 g(b_2 x + a_2)}_b + \underbrace{c_3 g(b_3 x + a_3)}_c + \underbrace{c_4 g(b_4 x + a_4)}_d + \underbrace{c_5 g(b_5 x + a_5)}_e + \underbrace{c_0}_{\text{bias}}$$

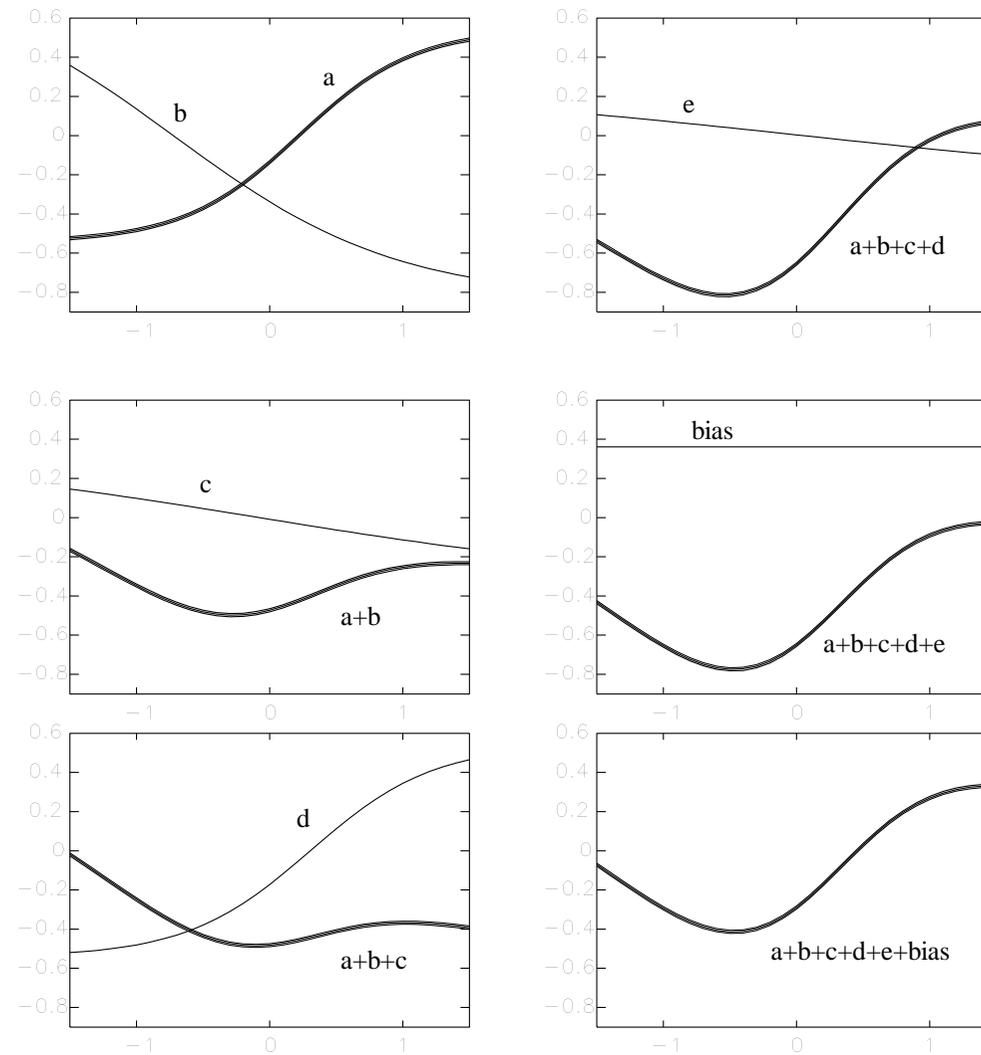


Figura 1 – Composição aditiva de ativações na reprodução de um mapeamento não-linear

O mapeamento a ser aproximado encontra-se na última figura à direita.

3. Aprendizado a partir de dados amostrados (revisão)

- O aprendizado supervisionado visto como um problema de otimização não-linear.
- A função objetivo (critério de desempenho a ser otimizado) e os parâmetros ajustáveis:

$$\min_{\vec{w}} J(\vec{w})$$

- Formalização matemática do que se quer otimizar + método de solução
- Solução na forma fechada × Busca iterativa
- Os dados de entrada/saída e a questão dos 3 mapeamentos envolvidos no processo:
 1. O mapeamento a ser aproximado (do qual se conhece apenas dados amostrados);
 2. O mapeamento resultante do processo de aproximação;
 3. O mapeamento entre cada vetor de pesos e o erro: superfície de erro.

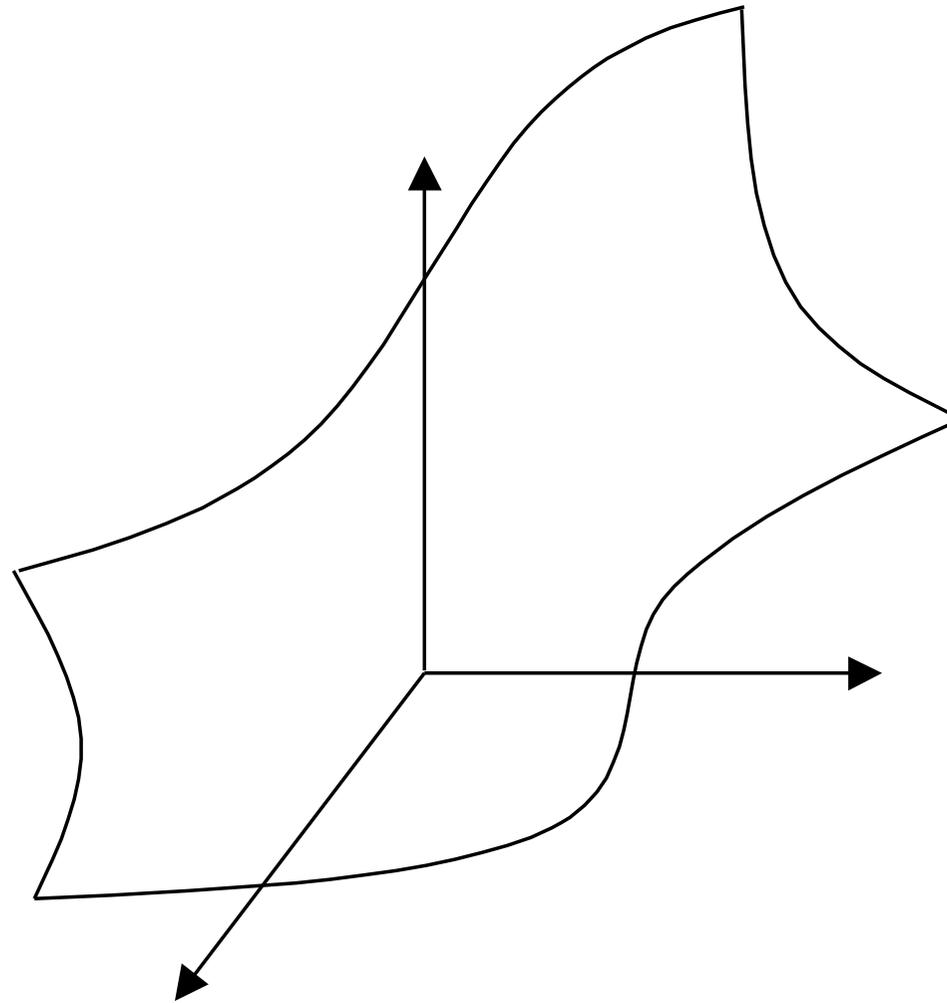


Figura 2– Mapeamento desconhecido a ser aproximado

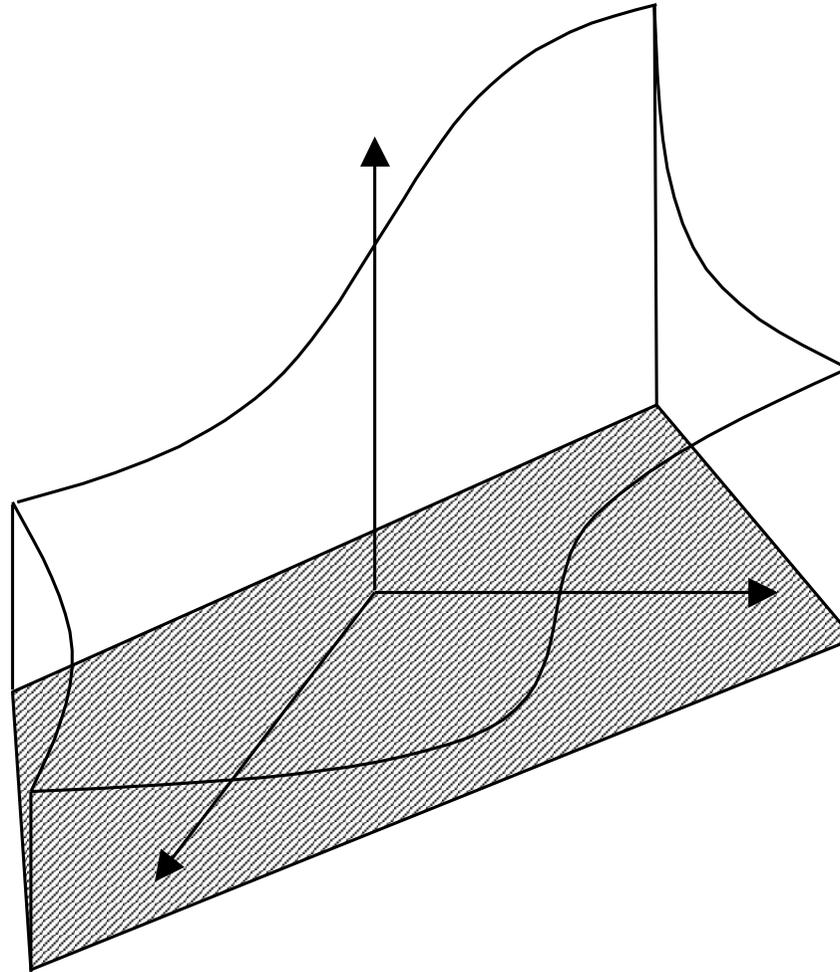


Figura 3 – Exemplo de região de operação. É uma região compacta (fechada e limitada).

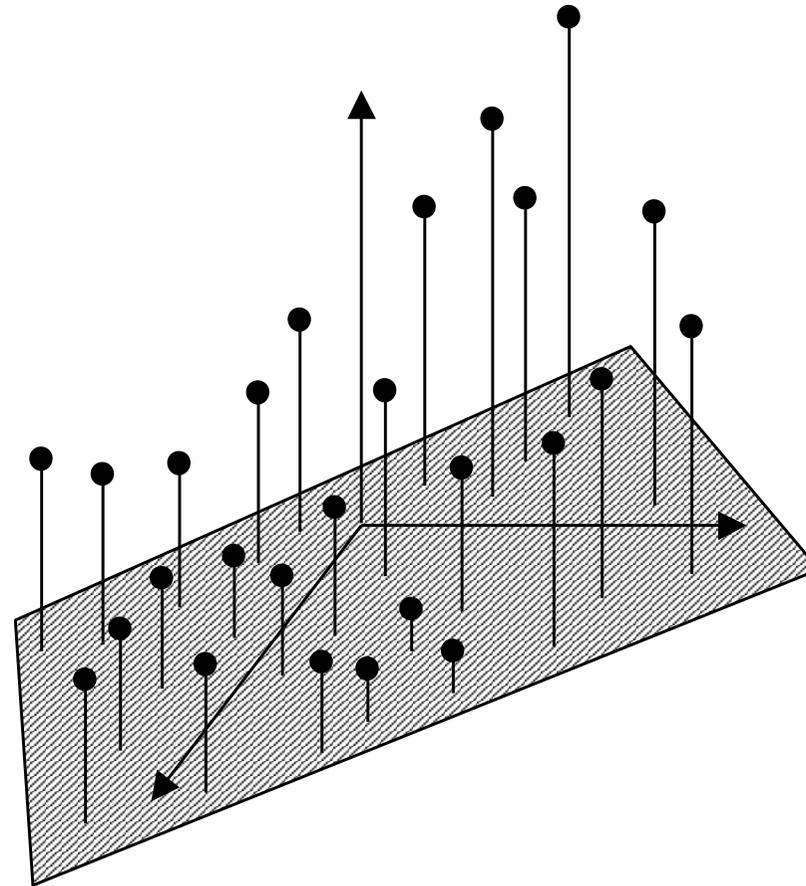


Figura 4 – Amostras expressando o comportamento da função para pontos específicos da região de operação. Essas amostras comporão os conjuntos de treinamento e validação (sendo que os dois conjuntos são independentes entre si)

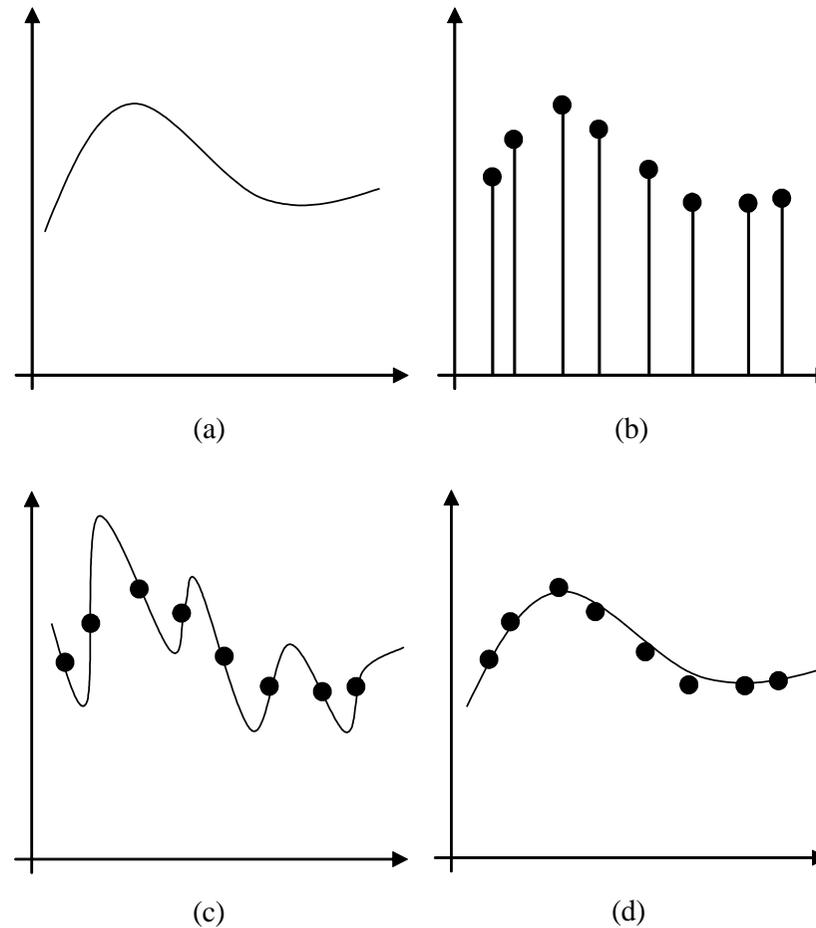


Figura 5 – (a) Função a ser aproximada (agora considerando apenas uma entrada); (b) Amostras disponíveis; (c) Resultado de um processo de aproximação com sobre-treinamento; (d) Resultado de um processo de aproximação sem sobre-treinamento.

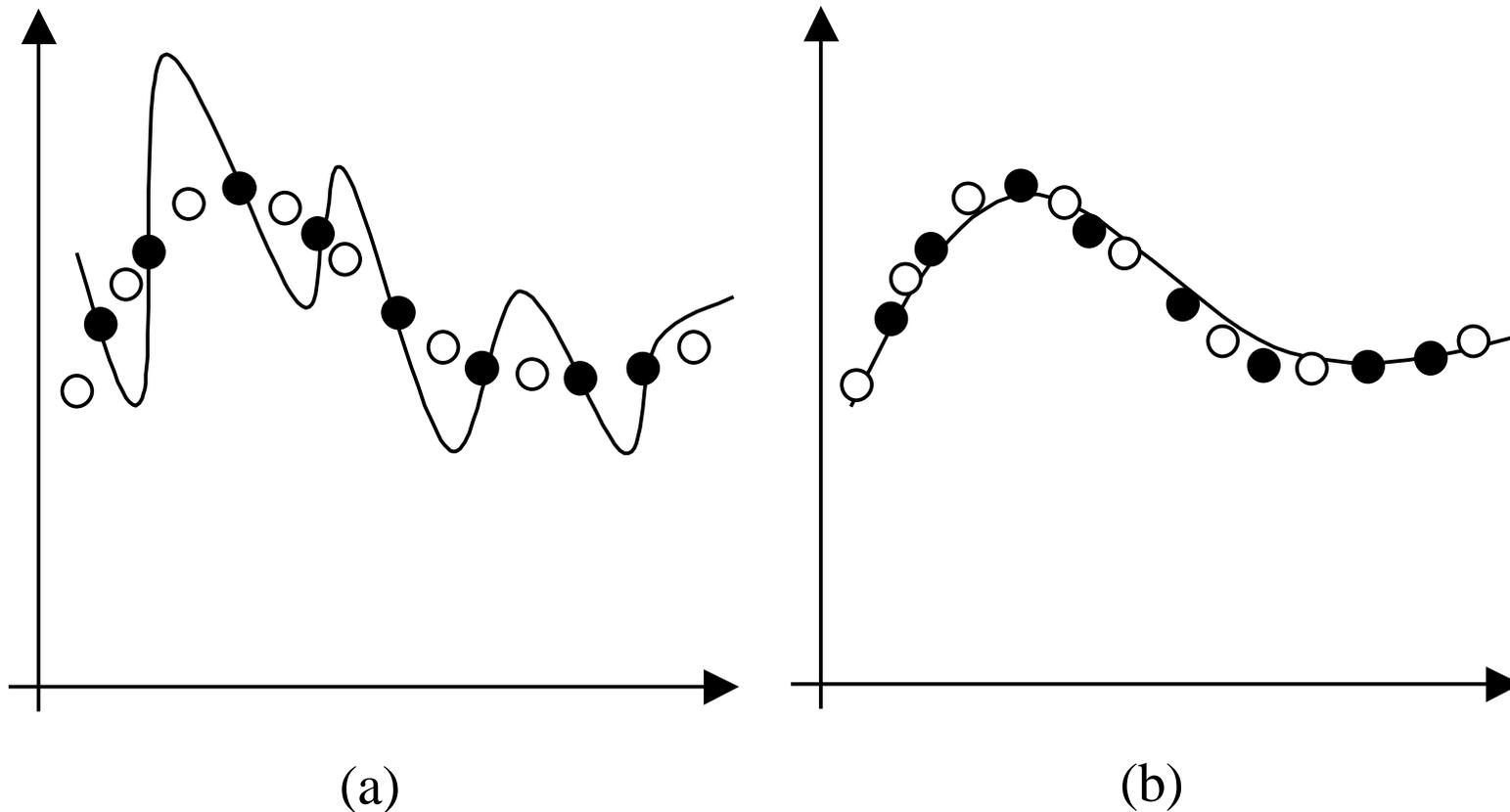


Figura 6 – Comparação de desempenho para dados de treinamento e validação, de modo a medir a capacidade de generalização dos mapeamentos produzidos.

- O mapeamento da esquerda apresenta um erro de treinamento muito baixo, mas um erro de validação bastante elevado, quando comparado ao mapeamento da direita.

4. Aprendizado supervisionado em RBF (revisão)

$$\{(\mathbf{x}_i, s_i)\}_{i=1}^N$$

$$f(\mathbf{w}, \mathbf{x}) = w_0 + \sum_{j=1}^n w_j h_j(\mathbf{x}) = \sum_{j=0}^n w_j h_j(\mathbf{x}), \text{ com } h_0(\mathbf{x}) = 1.$$

$$\mathbf{H} = [\mathbf{h}_0 \quad \mathbf{h}_1 \quad \cdots \quad \mathbf{h}_n] = \begin{bmatrix} h_0(\mathbf{x}_1) & h_1(\mathbf{x}_1) & \cdots & h_n(\mathbf{x}_1) \\ h_0(\mathbf{x}_2) & h_1(\mathbf{x}_2) & \cdots & h_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_0(\mathbf{x}_N) & h_1(\mathbf{x}_N) & \cdots & h_n(\mathbf{x}_N) \end{bmatrix}$$

$$\text{Resolver } \mathbf{H}\mathbf{w} = \mathbf{s} \text{ com } \mathbf{w} = [w_0 \quad w_1 \quad \cdots \quad w_n]^T \equiv \min_{\mathbf{w}} \|\mathbf{H}\mathbf{w} - \mathbf{s}\|_2^2$$

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

5. O problema do OU-exclusivo em MLP

- Considere os pontos $(0,0)$; $(0,1)$; $(1,0)$ e $(1,1)$ no plano \mathbb{R}^2 , conforme apresentado na Figura 7. O objetivo é determinar uma rede com duas entradas $\mathbf{x}_i \in \{0,1\}$ ($i=1,2$), e

uma saída $\mathbf{y} \in \{0,1\}$ de maneira que:
$$\begin{cases} (\mathbf{x}_1, \mathbf{x}_2) = (0,0) \text{ ou } (1,1) \Rightarrow \mathbf{y} = 0 \\ (\mathbf{x}_1, \mathbf{x}_2) = (1,0) \text{ ou } (0,1) \Rightarrow \mathbf{y} = 1 \end{cases}$$

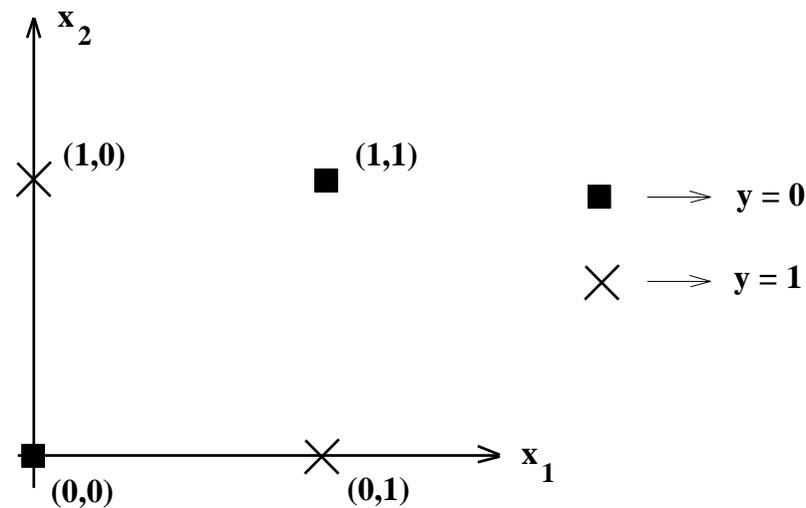


Figura 7 – O problema do OU-exclusivo

- Inicialmente será analisado o comportamento de um neurônio tipo perceptron (veja Figura 8) no processo de solução do problema exposto acima. A saída y pode ser representada na forma:

$$y = g(\mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \mathbf{w}_0) \quad \text{onde} \quad \begin{cases} g(\mathbf{u}) = 1 & \text{se } \mathbf{u} \geq 0 \\ g(\mathbf{u}) = 0 & \text{se } \mathbf{u} < 0 \end{cases}$$

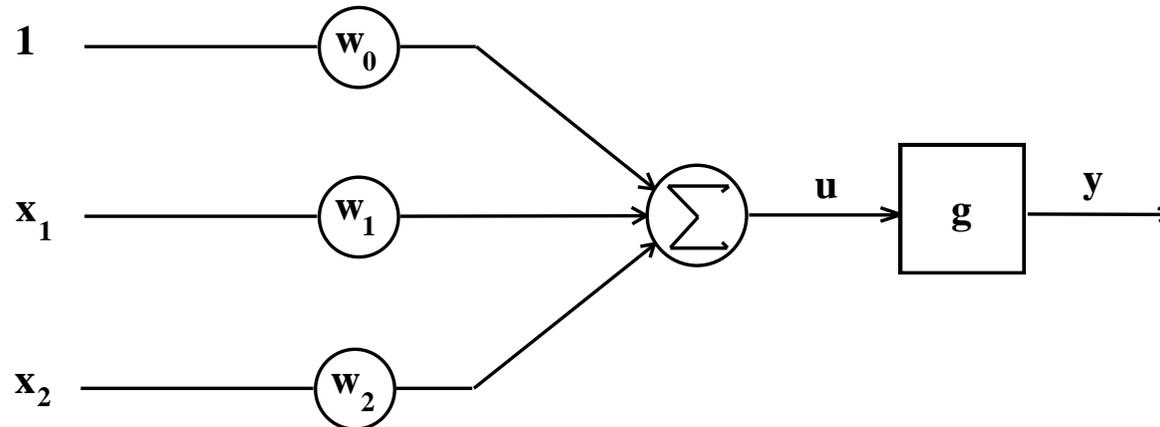


Figura 8 – Neurônio tipo perceptron, com duas entradas (mais a polarização)

- Para quaisquer valores dos parâmetros \mathbf{w}_0 , \mathbf{w}_1 e \mathbf{w}_2 , a função $g(\mathbf{u})$ separa o espaço de entradas em duas regiões, sendo que a curva de separação é uma linha reta.

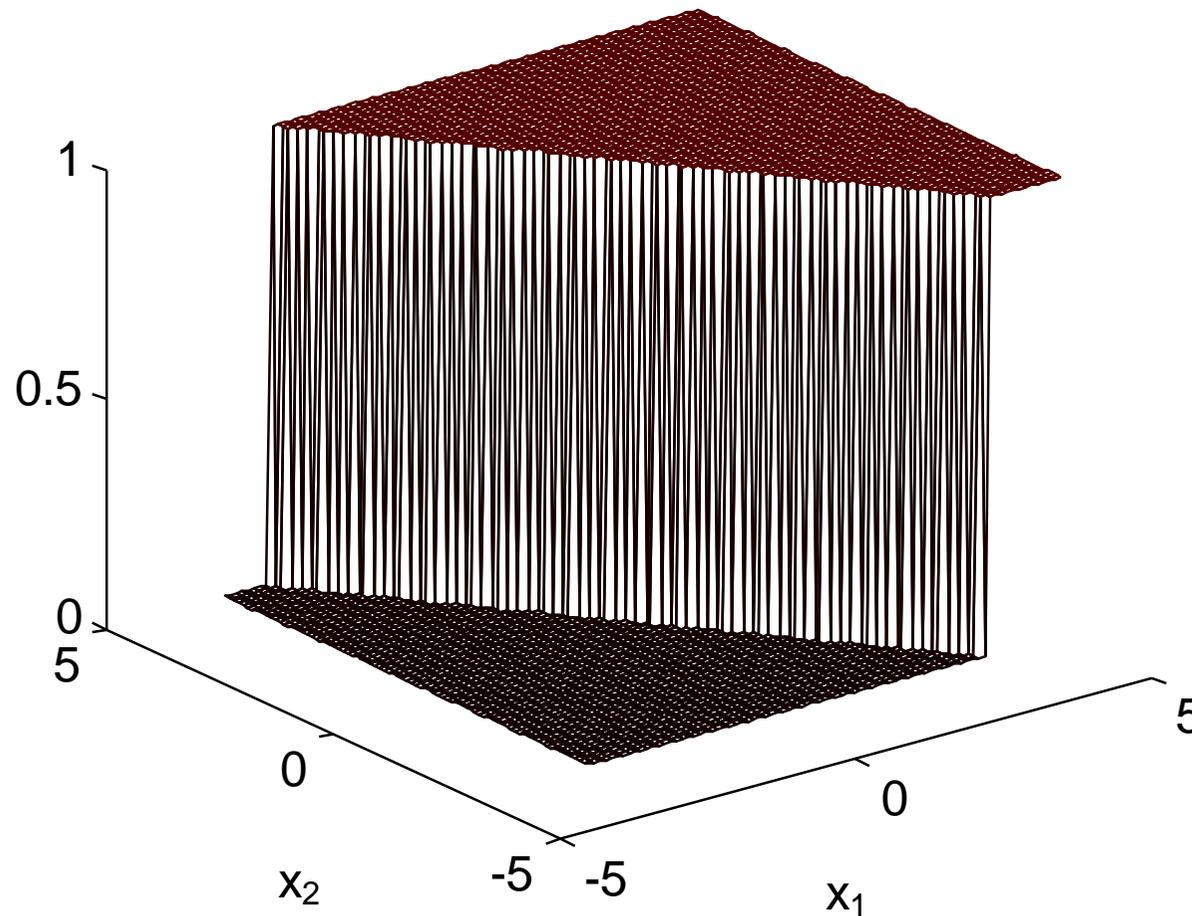


Figura 9 – Mapeamento de entrada-saída para o perceptron da Figura 8,
com $w_0 = -6$, $w_1 = 4$ e $w_2 = 3$

- Aqui tomou-se a função $g(\cdot)$ como sendo a função sinal, pois as saídas são binárias.

- No problema do OU-exclusivo (Figura 7), pode-se constatar que não existe uma única linha reta divisória de forma que os pontos (0,0) e (1,1) se posicionem de um lado enquanto que (0,1) e (1,0) permaneçam do outro lado da linha.
- Logo, pode-se imediatamente concluir que um neurônio tipo perceptron não apresenta grau de liberdade suficiente para resolver o problema proposto, o que foi corretamente constatado por Minsky & Papert, em 1969.
- No entanto, esses autores também acreditavam que não havia razão para supor que redes multicamadas pudessem conduzir a uma solução para o problema proposto. Esta hipótese só foi definitivamente rejeitada com o desenvolvimento do algoritmo de retro-propagação (*back-propagation*), já nos anos 80, o qual permite o ajuste automático de pesos para redes neurais multicamadas, arquitetura necessária para a realização de mapeamentos não-lineares.
- Considere o problema de mapeamento de uma rede neural tipo perceptron, com uma camada intermediária (Figura 10), aplicada ao problema do OU-exclusivo.

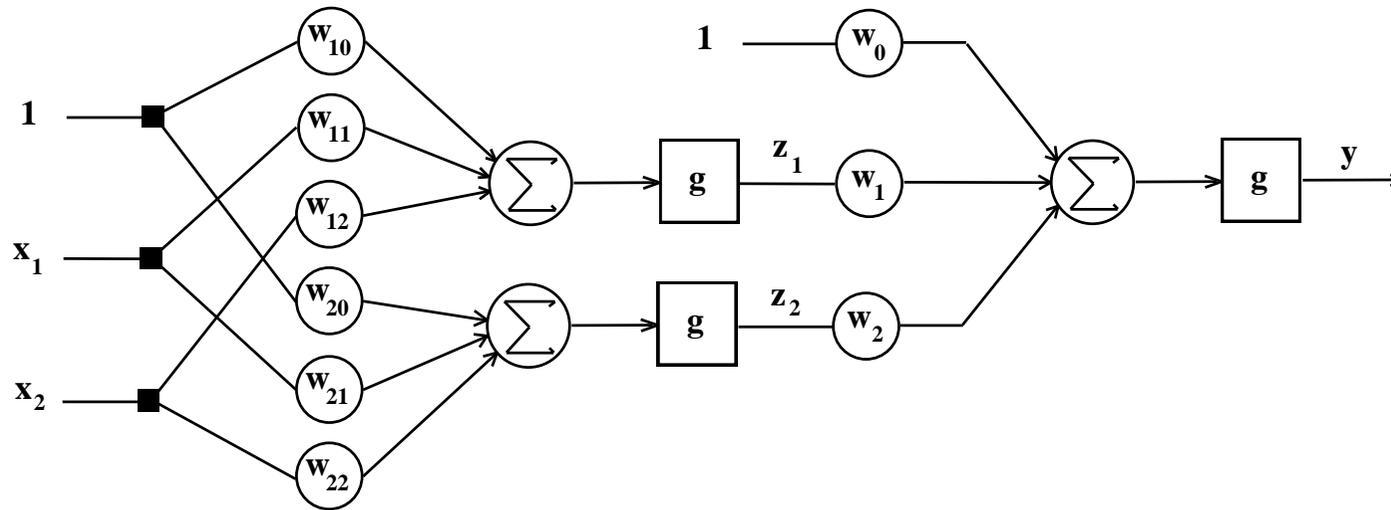


Figura 10 – Perceptron de três camadas (uma camada intermediária)

- A camada de entrada fornece um vetor de entrada (x_1, x_2) para a camada intermediária, enquanto que a camada intermediária produz duas saídas $z_1 = \text{sgn}(w_{10} + w_{11}x_1 + w_{12}x_2)$ e $z_2 = \text{sgn}(w_{20} + w_{21}x_1 + w_{22}x_2)$. Na camada de saída, o sinal de saída da rede neural é dado por $y = \text{sgn}(w_0 + w_1z_1 + w_2z_2)$.
- Surge uma questão: Existem parâmetros w_{ij} ($i=1,2; j=0,1,2$) e w_k ($k=0,1,2$) tais que $y = 0$ para as entradas $(0,0)$ e $(1,1)$ e $y = 1$ para as entradas $(1,0)$ e $(0,1)$?

- As saídas da primeira camada (\mathbf{z}_1 e \mathbf{z}_2) podem ser consideradas como variáveis intermediárias utilizadas na geração da saída \mathbf{y} .
- Do que já foi visto a respeito de um neurônio tipo perceptron, sabe-se que existem pesos \mathbf{w}_{1j} ($j=0,1,2$) tais que (veja curva de separação \mathbf{L}_1 na Figura 11(a)):

(0,1) produza $\mathbf{z}_1 = 1$

(0,0), (1,0) e (1,1) produzam $\mathbf{z}_1 = 0$.

- De forma similar, existem pesos \mathbf{w}_{2j} ($j=0,1,2$) tais que (veja curva de separação \mathbf{L}_2 na Figura 11(a)):

(0,1), (0,0) e (1,1) produzam $\mathbf{z}_2 = 1$

(1,0) produza $\mathbf{z}_2 = 0$

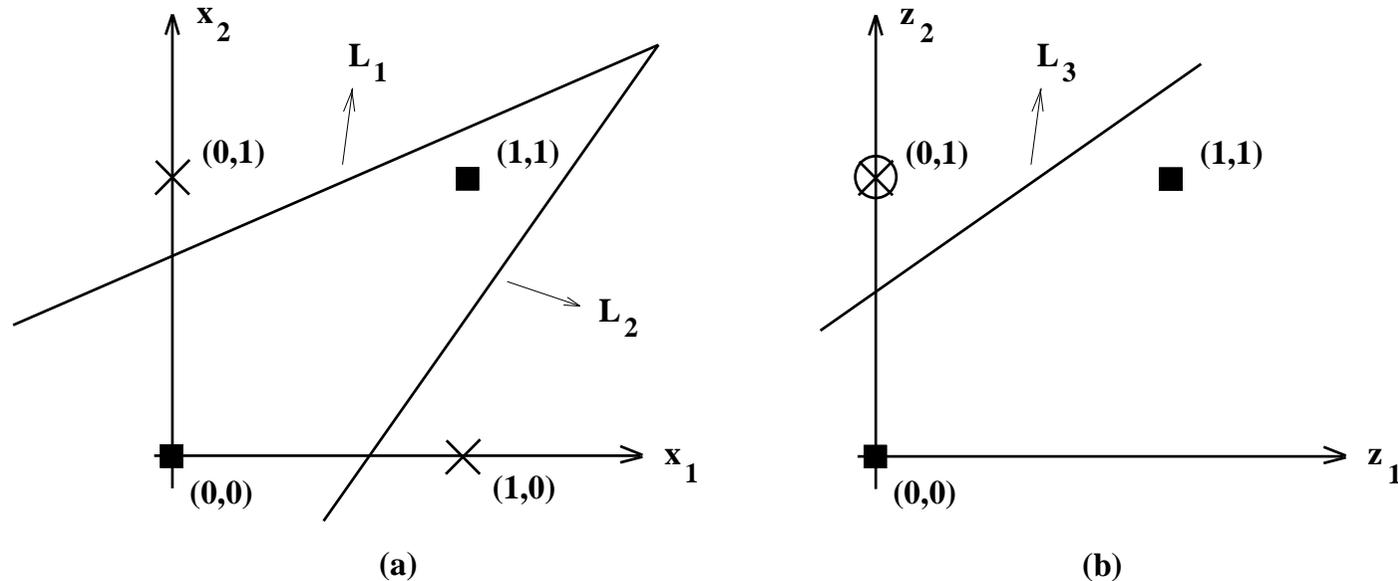


Figura 11 – Realização da função OU-exclusivo

- A discussão acima mostra que existem pesos w_{ij} ($i=1,2$; $j=0,1,2$) de maneira que a entrada $(0,1)$ resulte em $z_1 = 1$, $z_2 = 1$, e a entrada $(1,0)$ resulte em $z_1 = 0$, $z_2 = 0$, enquanto que $(0,0)$ e $(1,1)$ produzam $z_1 = 0$, $z_2 = 1$. Já que $(0,0)$ e $(1,1)$ podem ser separados linearmente de $(0,1)$, como mostrado na Figura 11(b) pela curva de separação L_3 , pode-se concluir que a função booleana desejada pode ser obtida utilizando-se perceptrons em cascata, ou seja, 3 neurônios do tipo perceptron.

6. Processo Iterativo para MLP – Método Padrão-a-Padrão

- Defina uma condição inicial para o vetor de pesos \mathbf{w} e escolha um passo α pequeno;
- Faça $k = 0$ e calcule $J(\mathbf{w}(k))$;
- Enquanto o critério de parada não for atendido, faça:

- ◆ Ordene aleatoriamente os padrões de entrada-saída;

- ◆ Para l variando de 1 até N , faça:

Apresente o padrão l de entrada à rede;

Calcule $J_l(\mathbf{w}(k))$ e $\nabla J_l(\mathbf{w}(k))$;

$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha \nabla J_l(\mathbf{w}(k))$;

- ◆ $k = k + 1$;

- ◆ Calcule $J(\mathbf{w}(k))$;

7. Processo Iterativo para MLP – Método em Lote ou Batelada

- Defina uma condição inicial para o vetor de pesos \mathbf{w} e escolha um passo α pequeno;
- Faça $k = 0$ e calcule $J(\mathbf{w}(k))$;
- Enquanto o critério de parada não for atendido, faça:

- ◆ Para l variando de 1 até N , faça:

Apresente o padrão l de entrada à rede;

Calcule $J_l(\mathbf{w}(k))$ e $\nabla J_l(\mathbf{w}(k))$;

- ◆ $\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\alpha}{N} \sum_{l=1}^N \nabla J_l(\mathbf{w}(k))$;

- ◆ $k = k + 1$;

- ◆ Calcule $J(\mathbf{w}(k))$;

8. Exemplos de auto-organização na natureza

Conchas



Processos Físicos

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



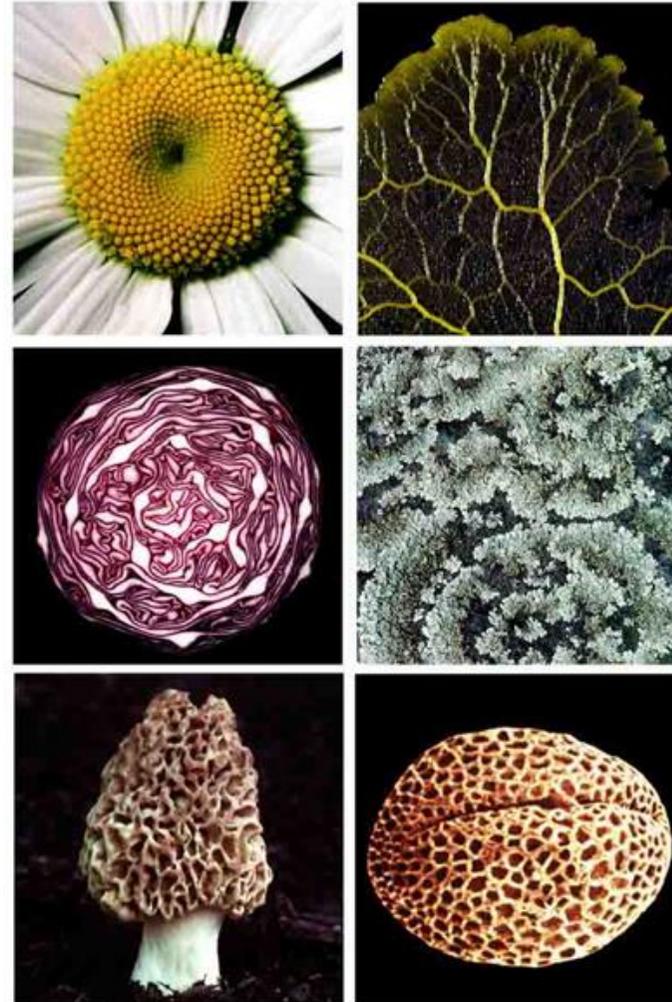
Em animais

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



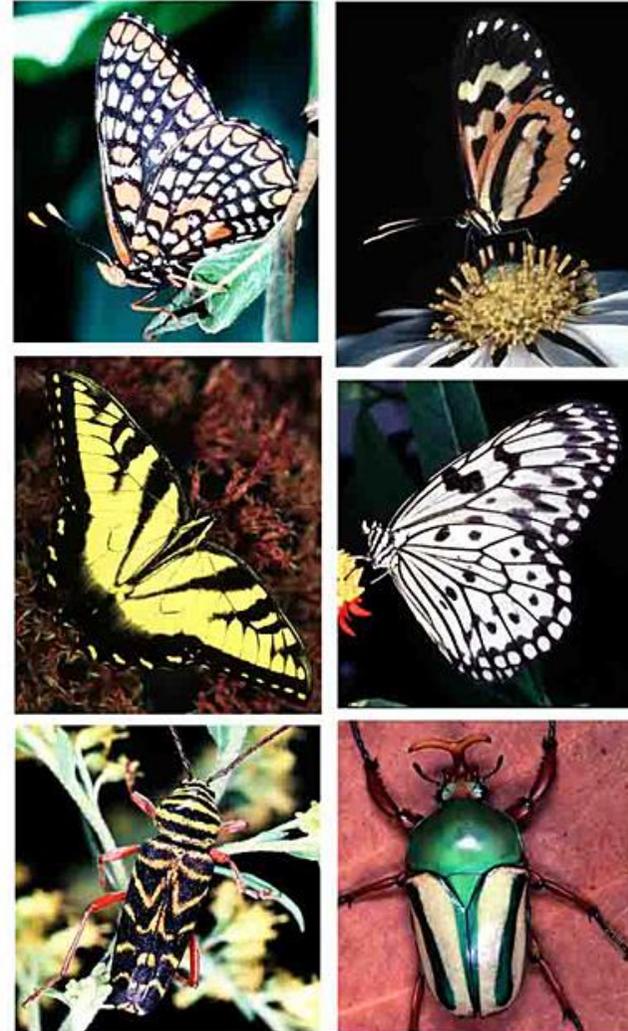
Em plantas

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



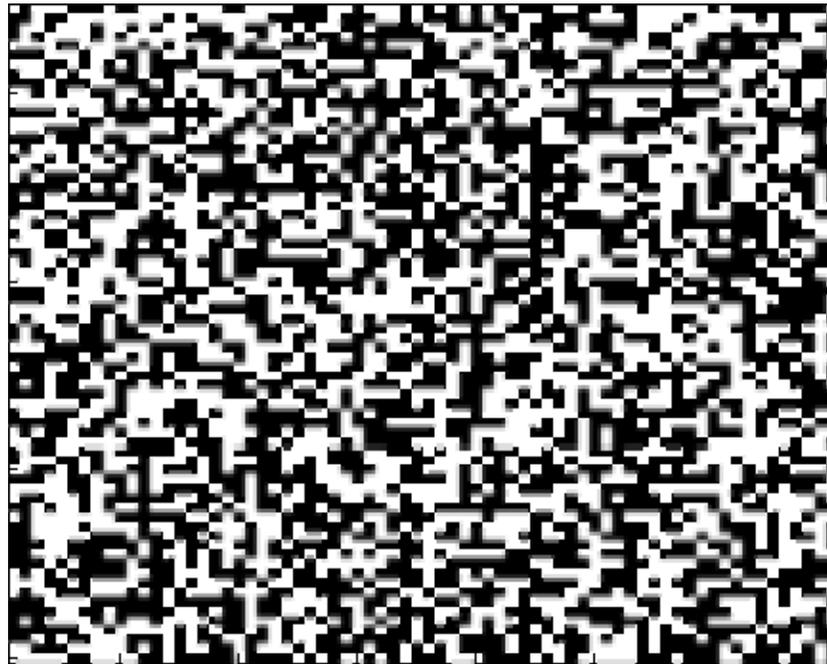
Em insetos

- ✓ Asas de borboletas
- ✓ Exoesqueletos de besouros



9. O Jogo da Vida

- Concebido por John Horton Conway, matemático britânico, em 1970. Procure assistir ao vídeo em <http://www.youtube.com/watch?v=XcuBvj0pw-E&feature=related> ou em <http://www.youtube.com/watch?v=C2vgICfQawE>.



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

10. Arranjo bidimensional de um mapa de Kohonen (revisão)

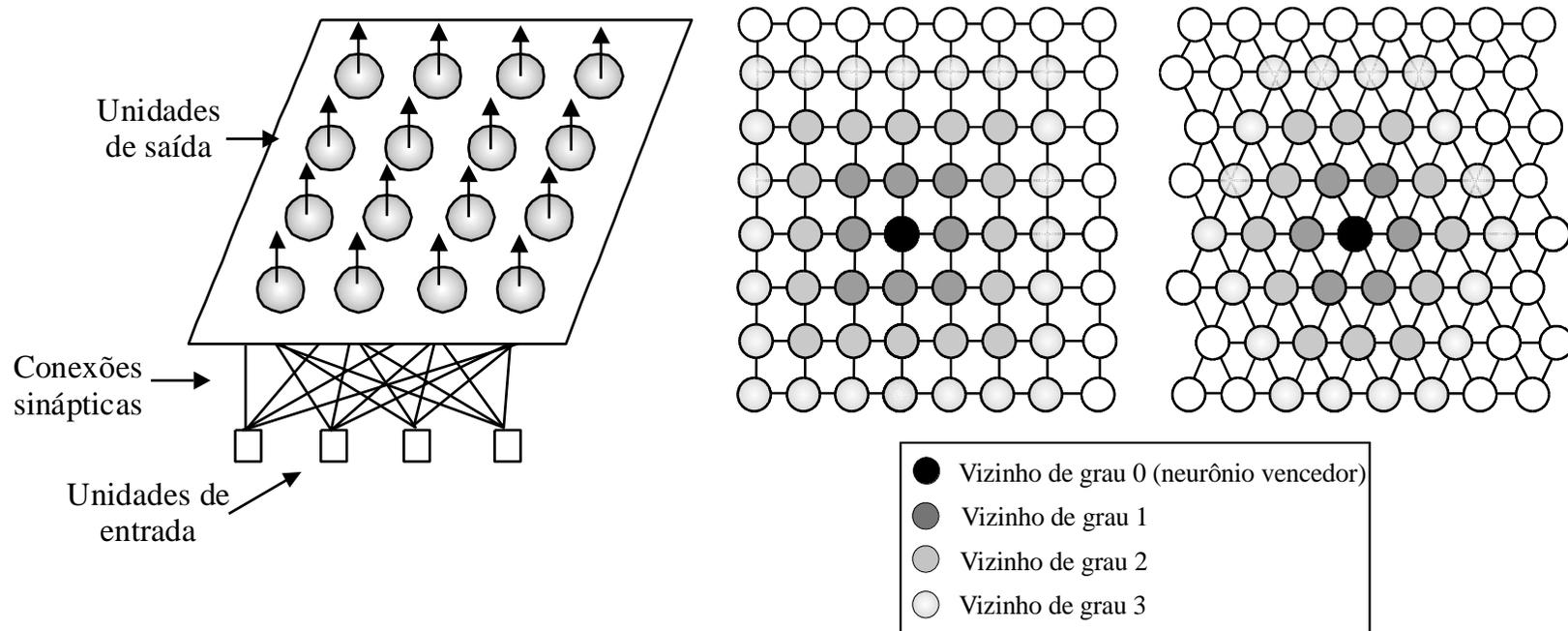


Figura 12 - Rede de Kohonen em arranjo bidimensional

11. Um passo de ajuste num arranjo unidimensional (revisão)

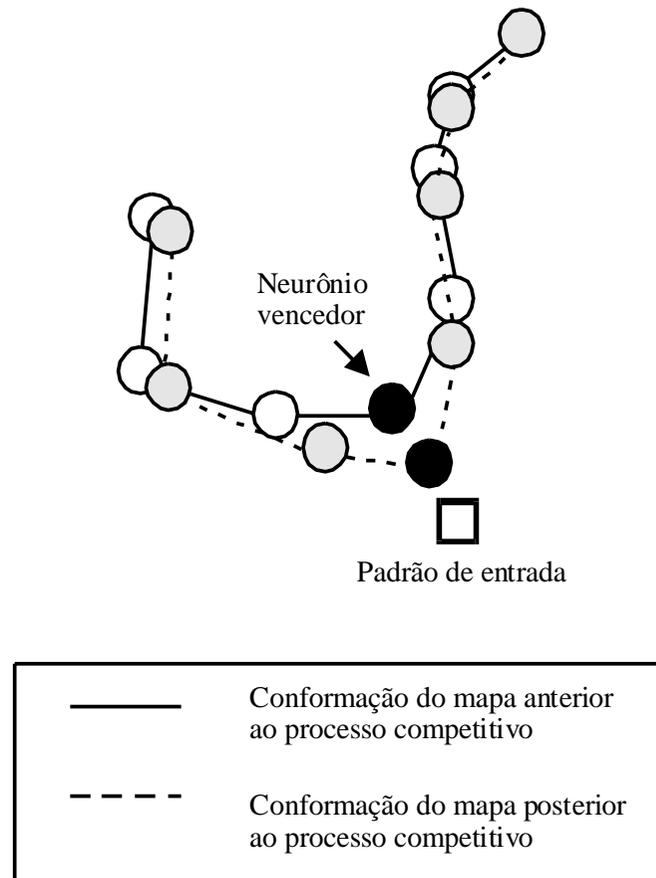


Figura 13 – Ajuste do neurônio vencedor e de seus vizinhos mais próximos

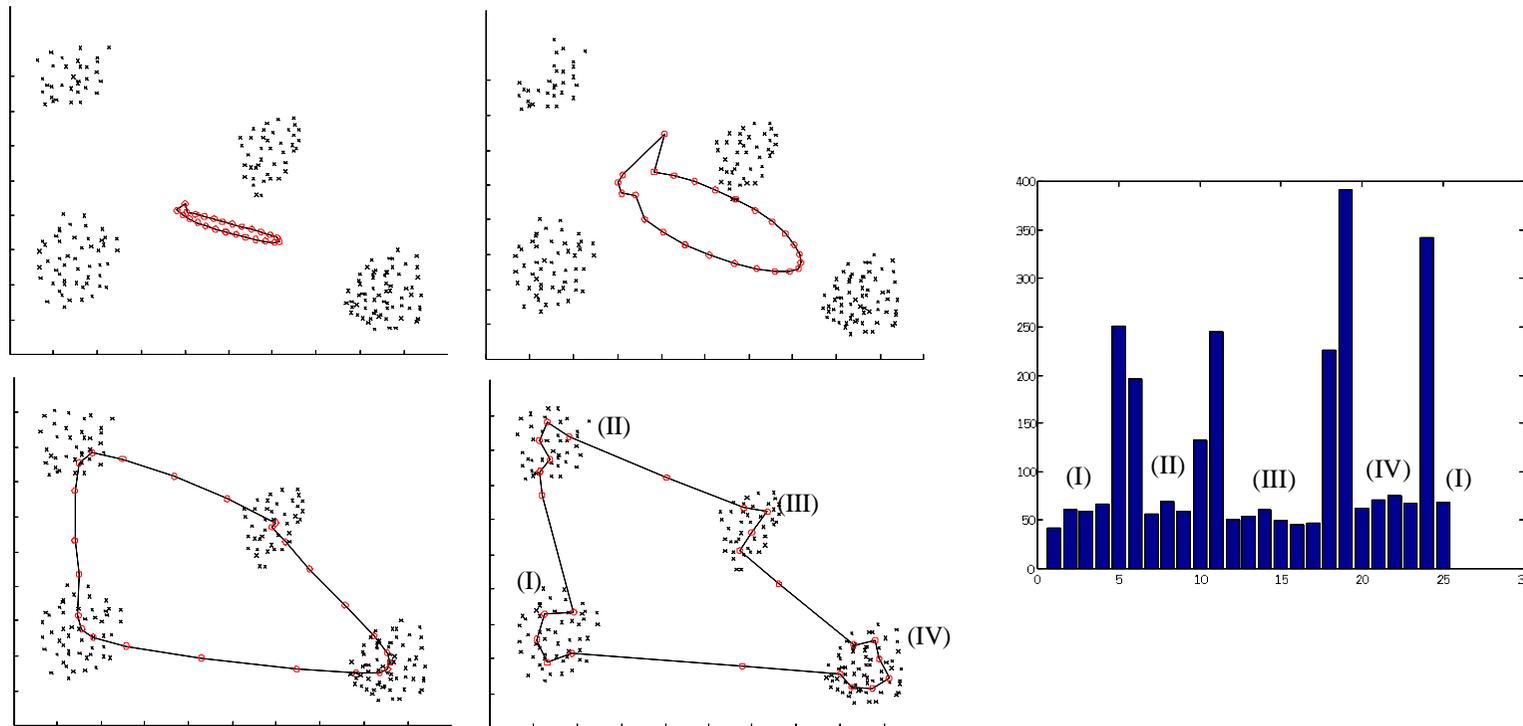


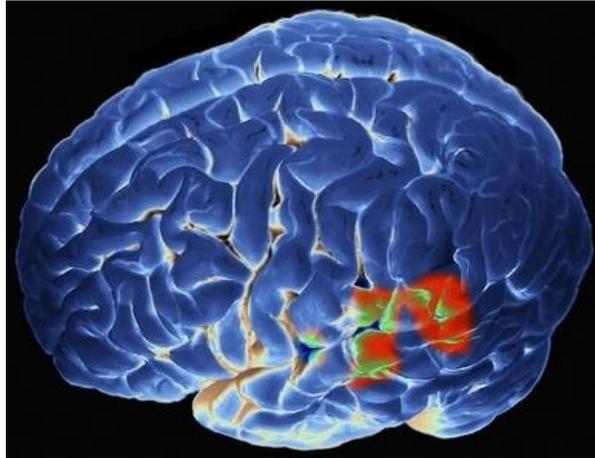
Figura 14 – Determinação automática do número de agrupamentos com base no vetor-U para arranjo unidimensional (Exemplo didático em 2 dimensões)

12. O caso Phineas Gage

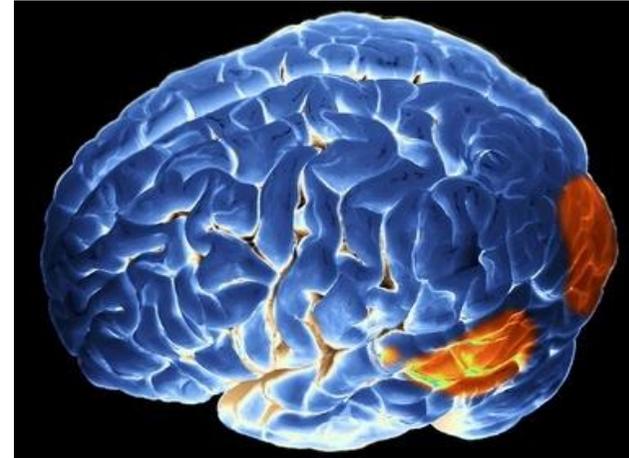


Figura 15 – Reconstrução computadorizada da destruição do cérebro de Phineas Gage por uma barra de ferro no ano de 1848.

13. Monitorando a atividade cerebral



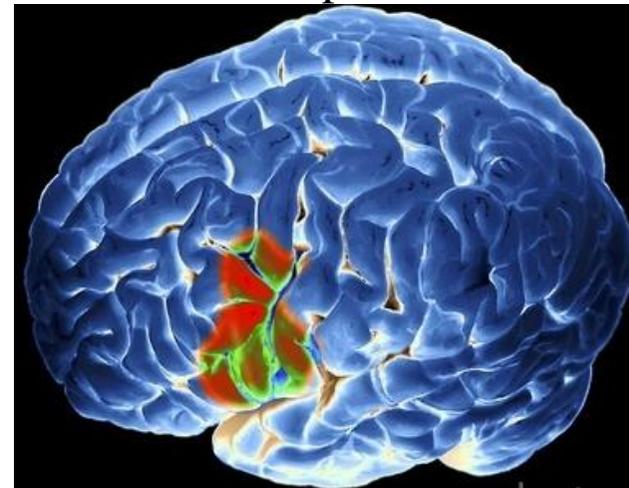
Ouvindo palavras



Lendo palavras



Falando palavras



Pensando em palavras

Functional magnetic resonance imaging or functional MRI (fMRI)