FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 06 - Redes *RBF*

(*Radial Basis Function Networks*)

Prof. Rafael G. Mantovani

Londrina - PR, Brasil
Outubro, 2018

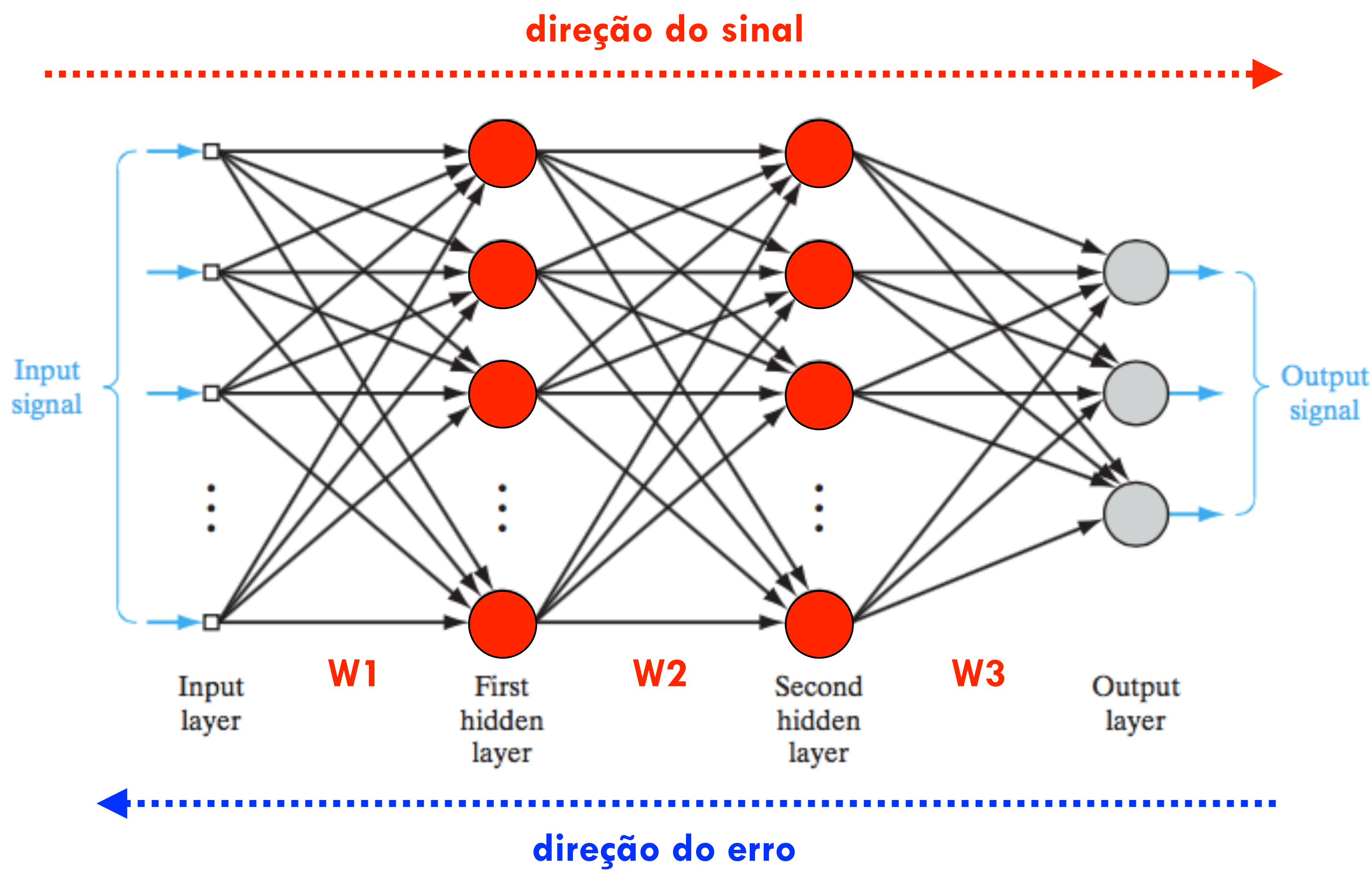
Universidade Estadual de Londrina - UEL
Departamento de Computação - DC

Roteiro

1. Introdução
2. Redes com Funções de Base Radial (RBF)
3. Algoritmo de Treinamento
4. Exemplo
5. Síntese / Próximas Aulas
6. Referências

Roteiro

Introdução

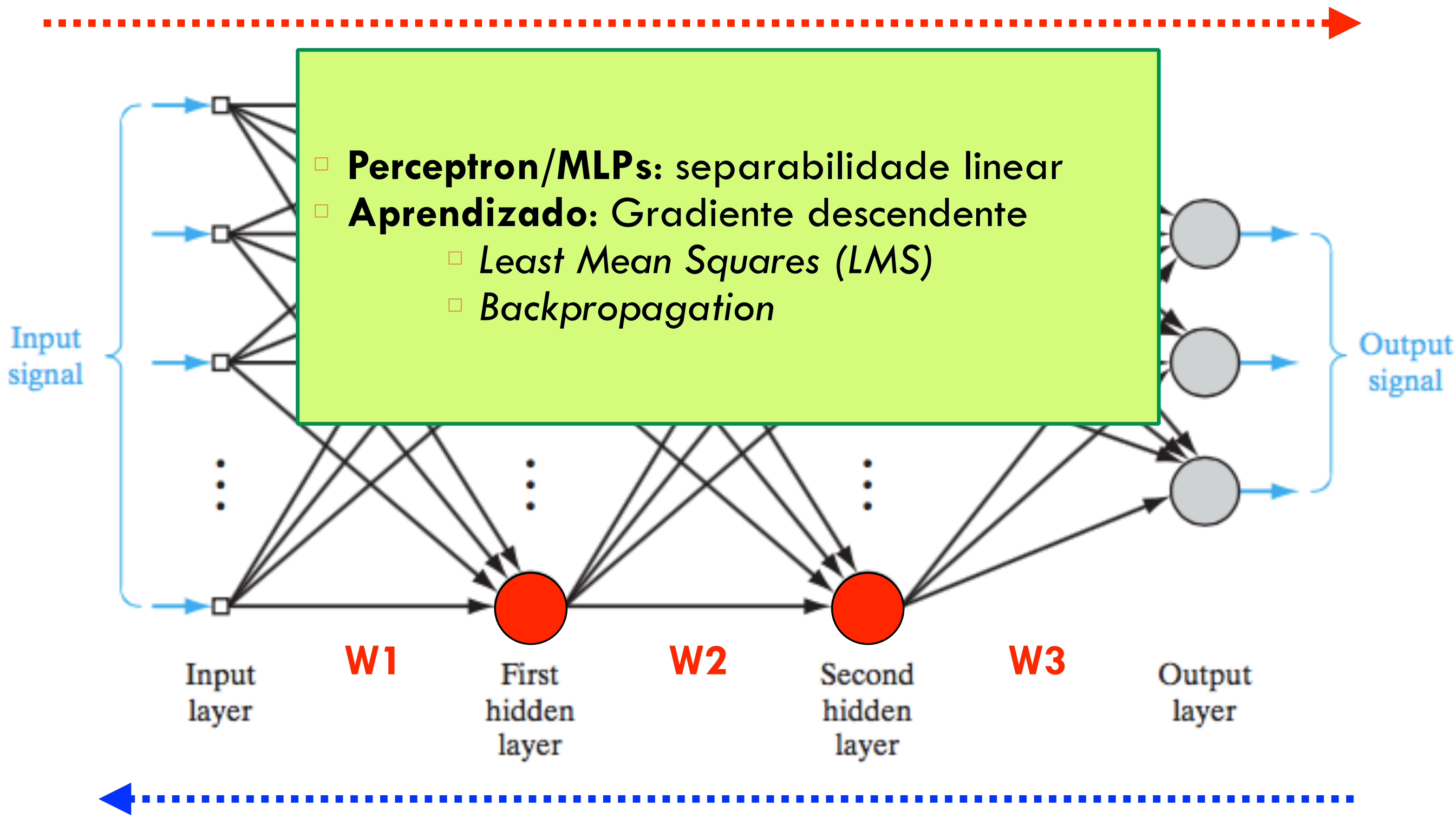


direção do sinal
Input signal
Output signal
Input layer
W1
First hidden layer
W2
Second hidden layer
W3
Output layer
direção do erro

Introdução

direção do sinal

- **Perceptron/MLPs:** separabilidade linear
- **Aprendizado:** Gradiente descendente
  - *Least Mean Squares (LMS)*
  - *Backpropagation*



direção do erro

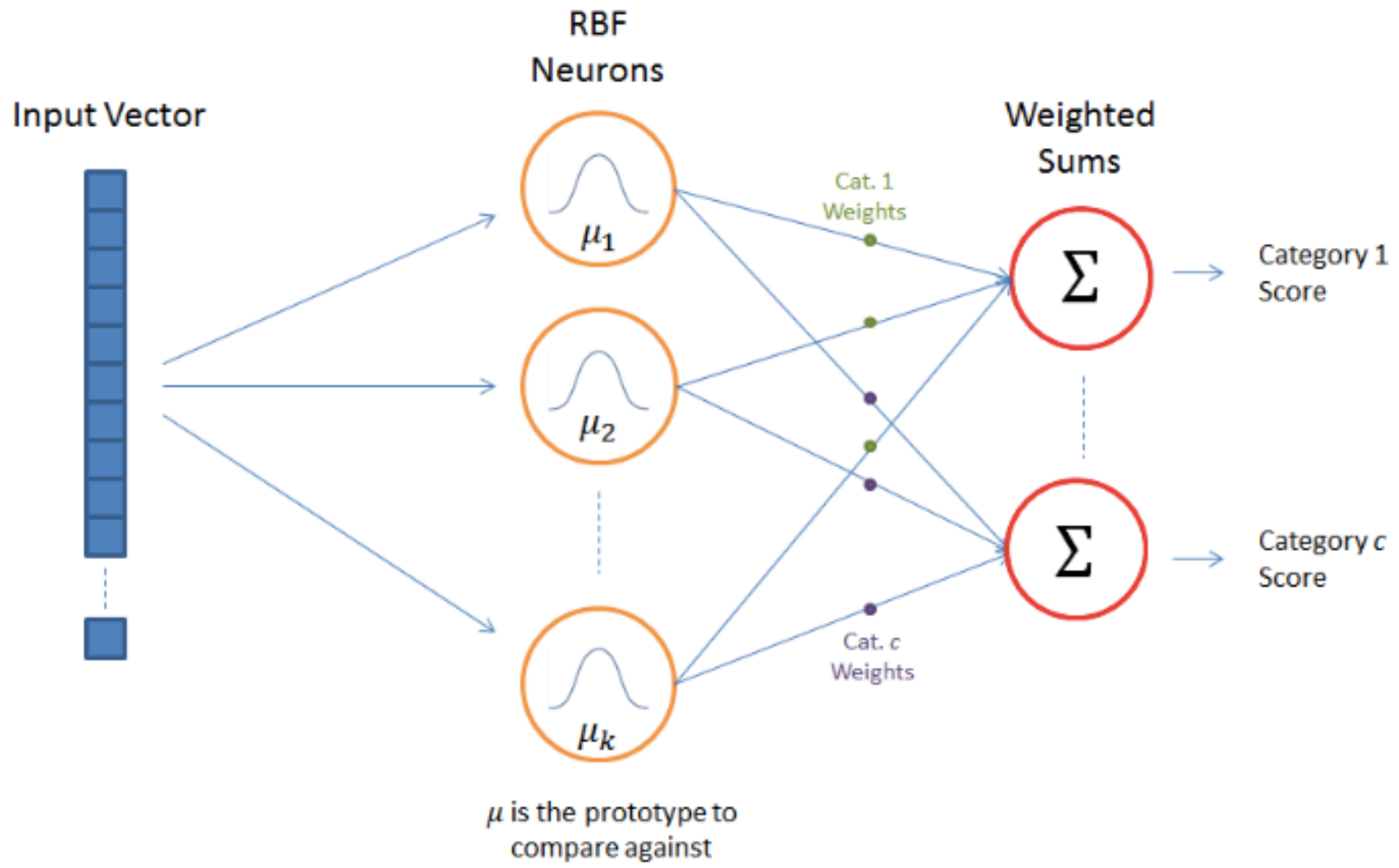
Introdução

- Treinamento pode ser feito de várias maneiras
  - ***Backpropagation*** é apenas uma delas

- Abordagem diferente (híbrida)
  1. Transformar um conjunto de exemplos não linearmente separáveis em outro, com grande chances de ser linearmente separável (**Teorema de Cover, 1965**)
  2. Segunda fase que completa a classificação.

- Redes com Funções de Base Radial (RBFs)

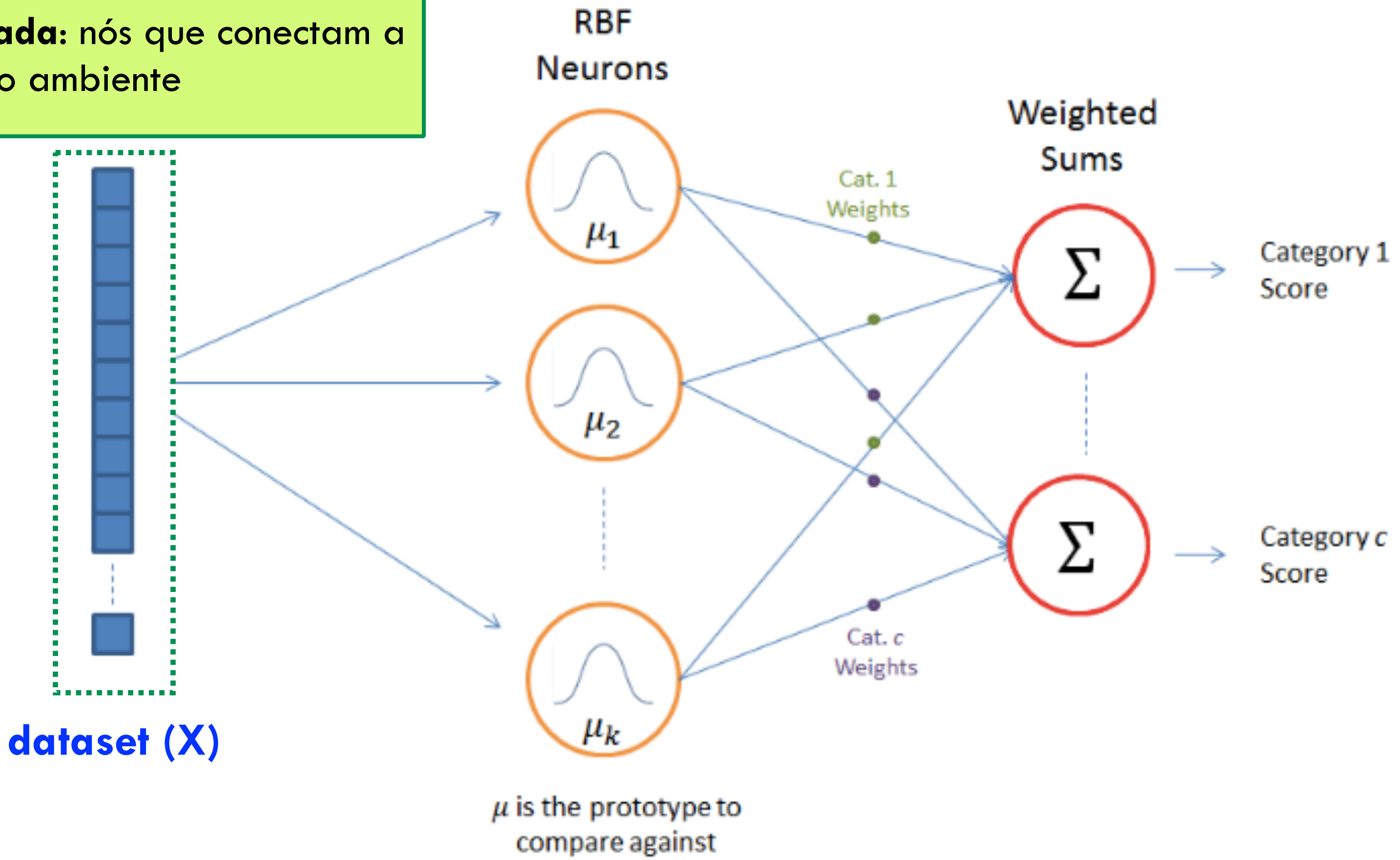
Introdução



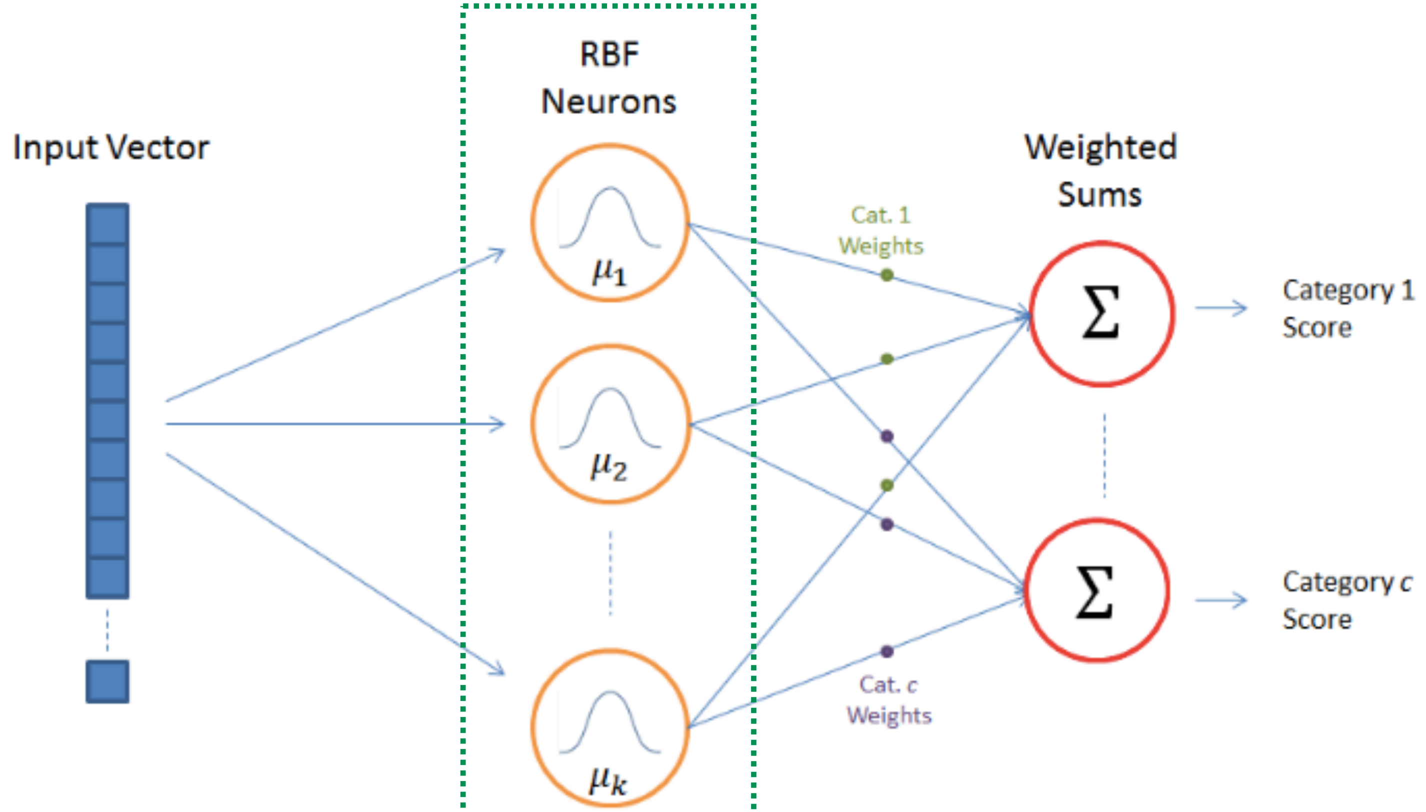
RBF Neurons
Input Vector
Weighted Sums
Cat. 1 Weights
Cat. c Weights
μ_1
μ_2
μ_k
Category 1 Score
Category c Score
μ is the prototype to compare against

Introdução

- 1. **Entrada:** nós que conectam a rede ao ambiente

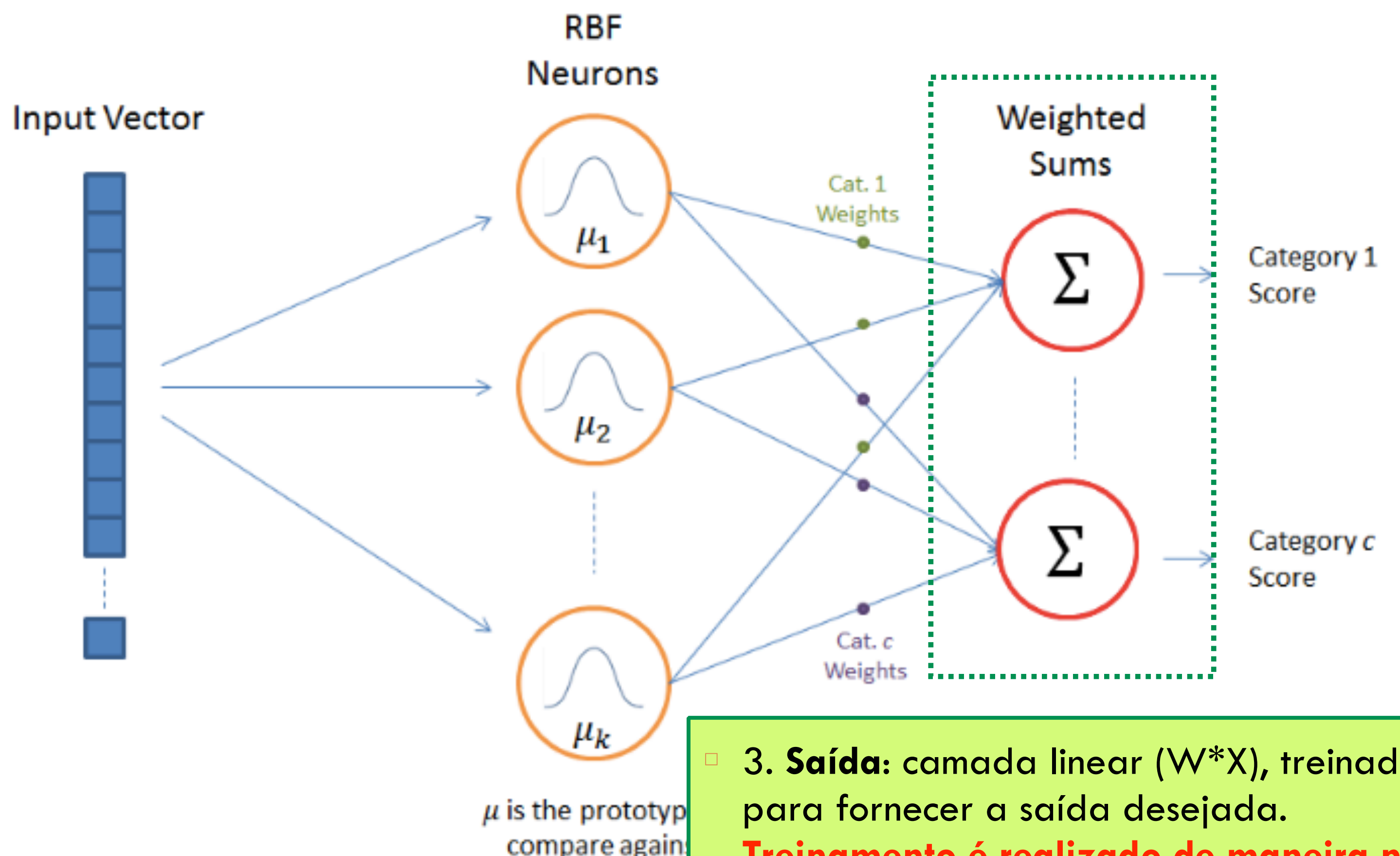


Introdução



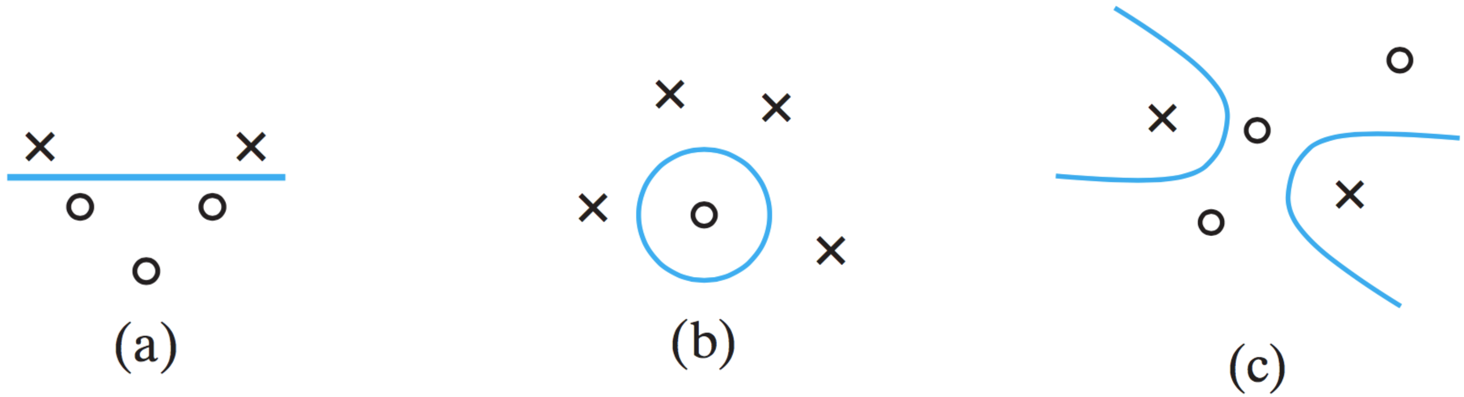
- 2. **Escondida:** transformação não-linear das entradas. **Treinamento é realizado de maneira não supervisionada.**

Introdução

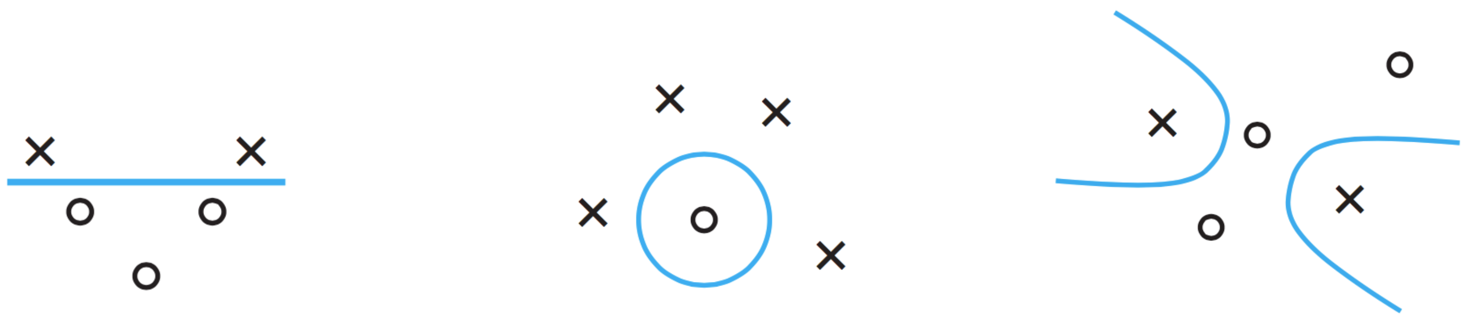


- 3. **Saída**: camada linear (W*X), treinada para fornecer a saída desejada. **Treinamento é realizado de maneira não não-supervisionada.**

Introdução



Introdução



(a) **linearmente separável** (b) **esfericamente separável** (c) **quadraticamente separável**

- Exemplos de dicotomias separáveis por diferentes superfícies de decisão, considerando um conjunto de cincos pontos em duas dimensões

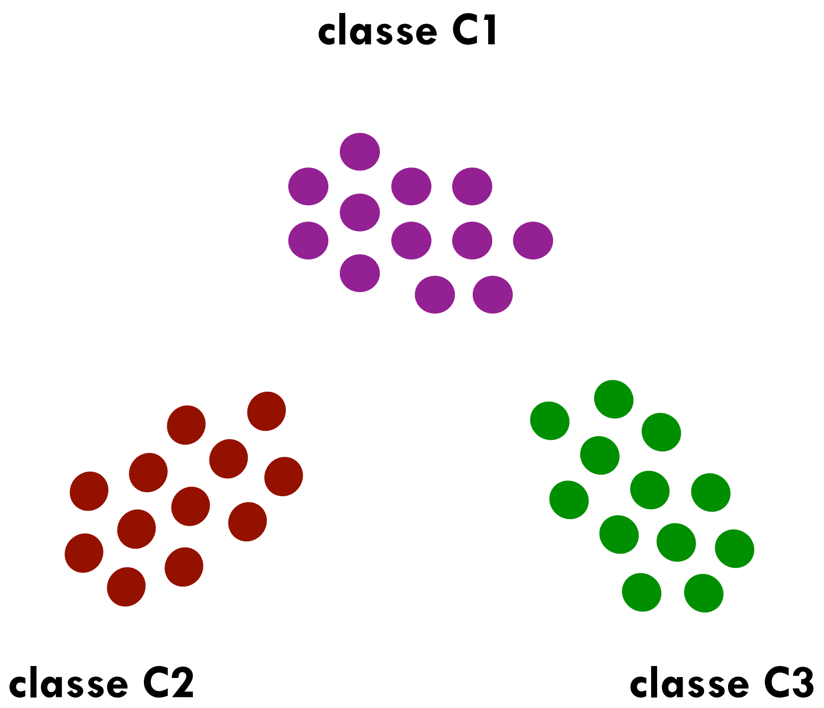
Pergunta?

Treinamento/ Aprendizado Supervisionado	VS	Treinamento/ Aprendizado Não-Supervisionado

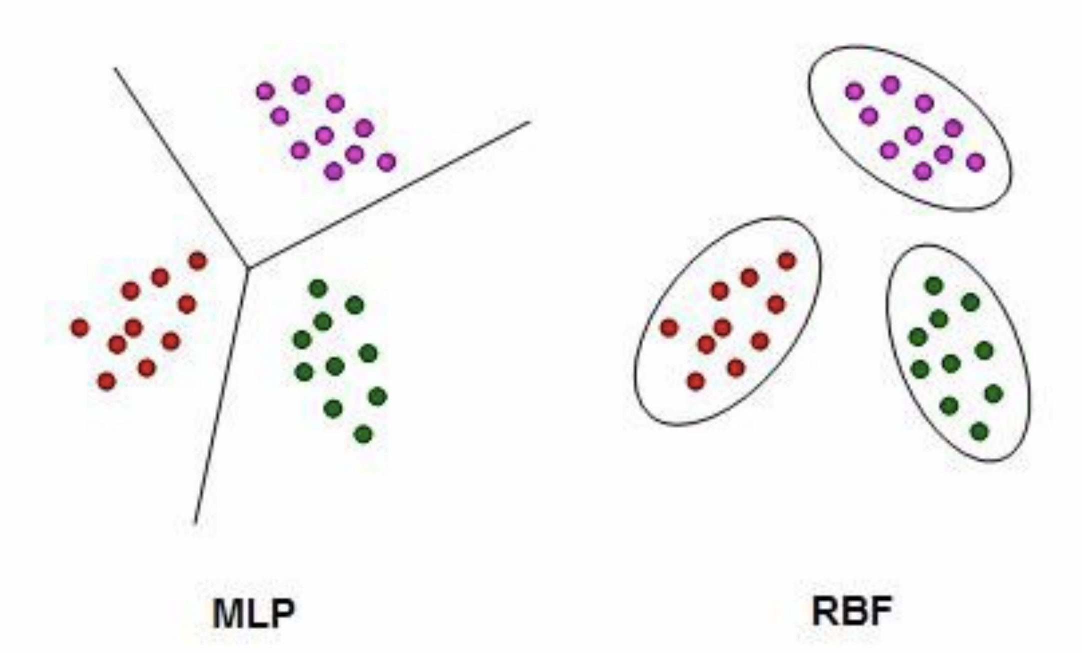
Roteiro

1. Introdução
2. Redes com Funções de Base Radial (RBF)
3. Algoritmo de Treinamento
4. Exemplo
5. Síntese / Próximas Aulas
6. Referências

Redes RBF

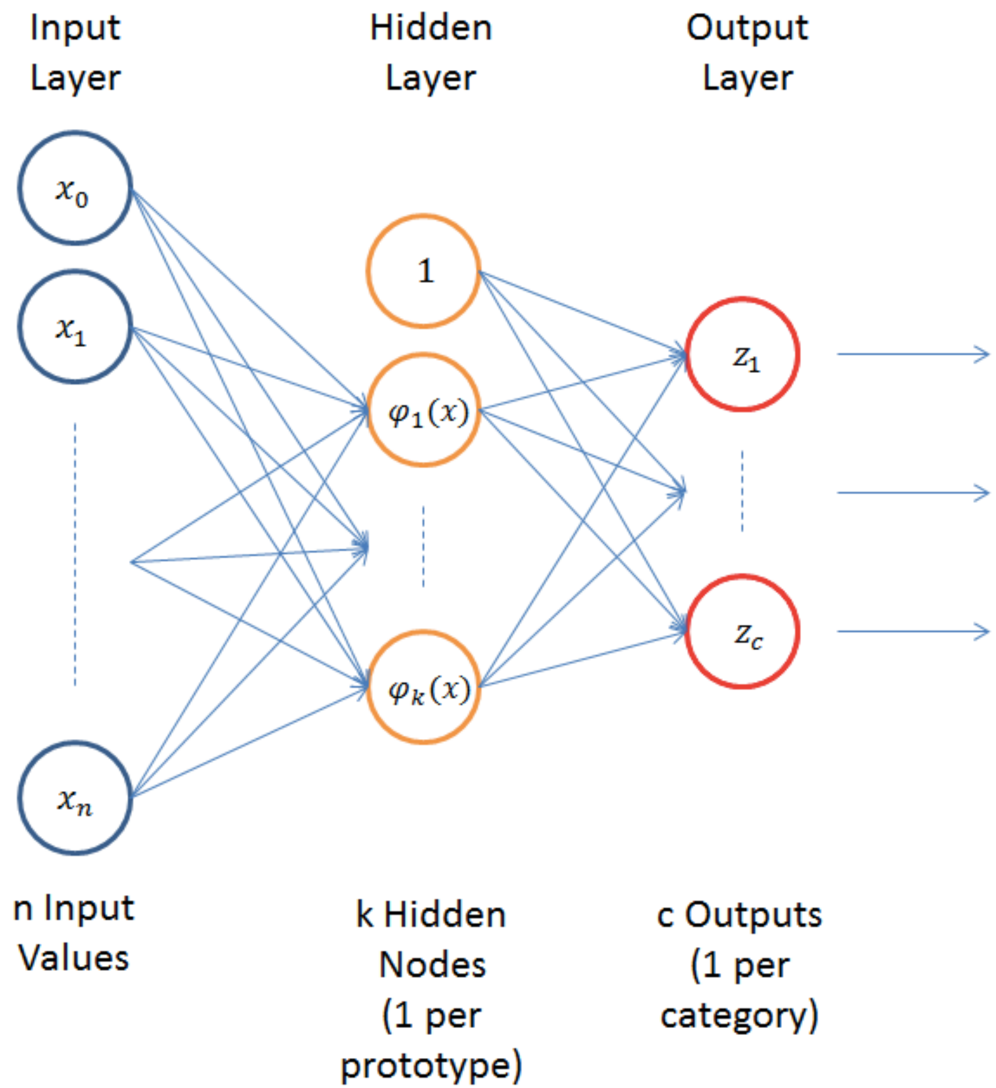


Redes RBF



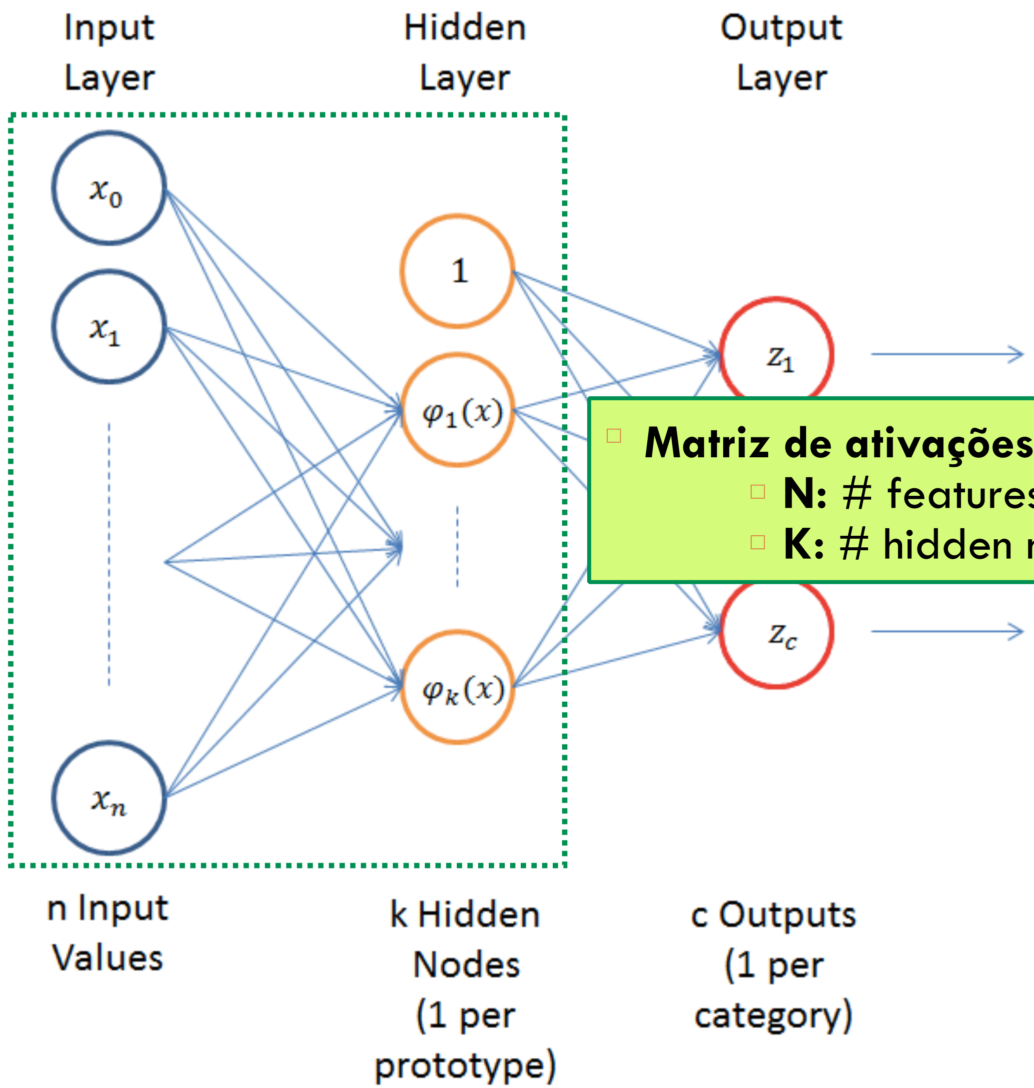
Distinction between MLP and RBF

Redes RBF



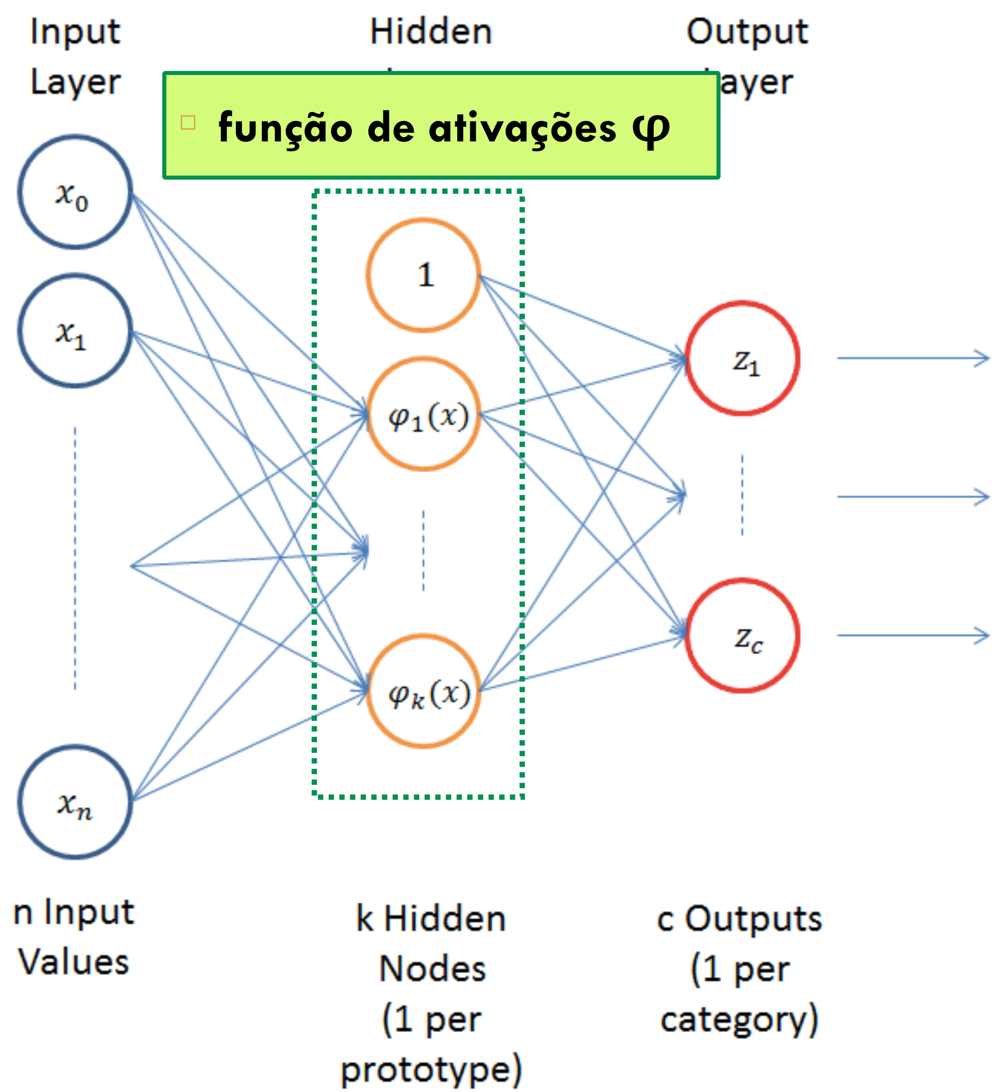
Input Layer
Hidden Layer
Output Layer
x_0
x_1
x_n
1
$\varphi_1(x)$
$\varphi_k(x)$
z_1
z_c
n Input Values
k Hidden Nodes (1 per prototype)
c Outputs (1 per category)

Redes RBF



- **Matriz de ativações Φ_{NxK}**
  - **N:** # features
  - **K:** # hidden neurons

Redes RBF

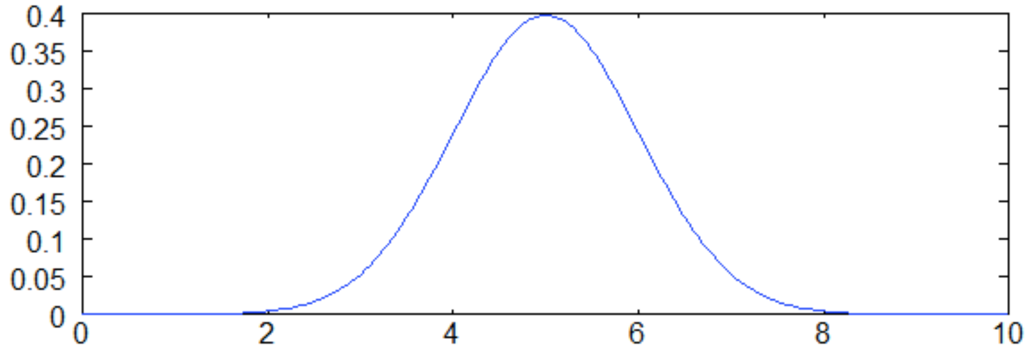


Input
Layer
Hidden
Output
ayer
função de ativações φ
x_0
x_1
x_n
1
$\varphi_1(x)$
$\varphi_k(x)$
z_1
z_c
n Input
Values
k Hidden
Nodes
(1 per
prototype)
c Outputs
(1 per
category)

Redes RBF

- função de ativação dos neurônios RBF:

$$\varphi(x) = exp^{-\gamma||x-\mu||^2}$$

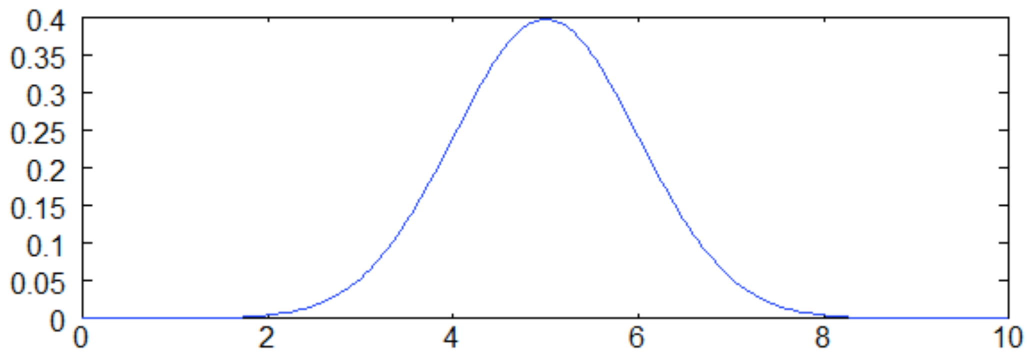


Redes RBF

- função de ativação dos neurônios RBF:

$$\varphi(x) = exp^{-\gamma||x-\mu||^2}$$

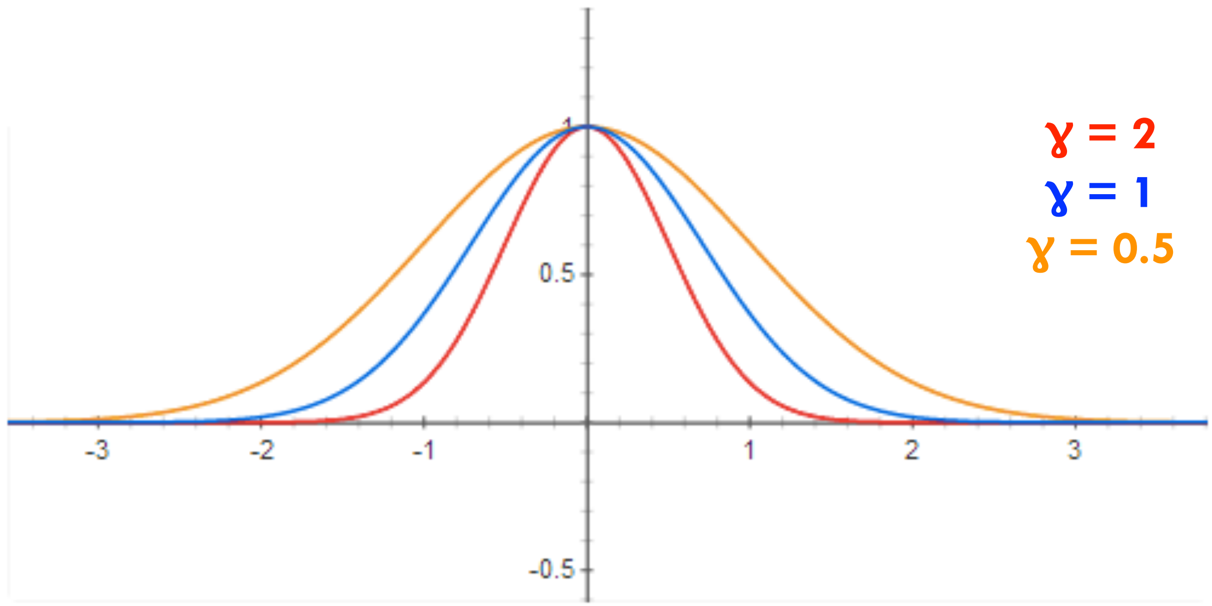
- **μ é um padrão/ exemplo de X**



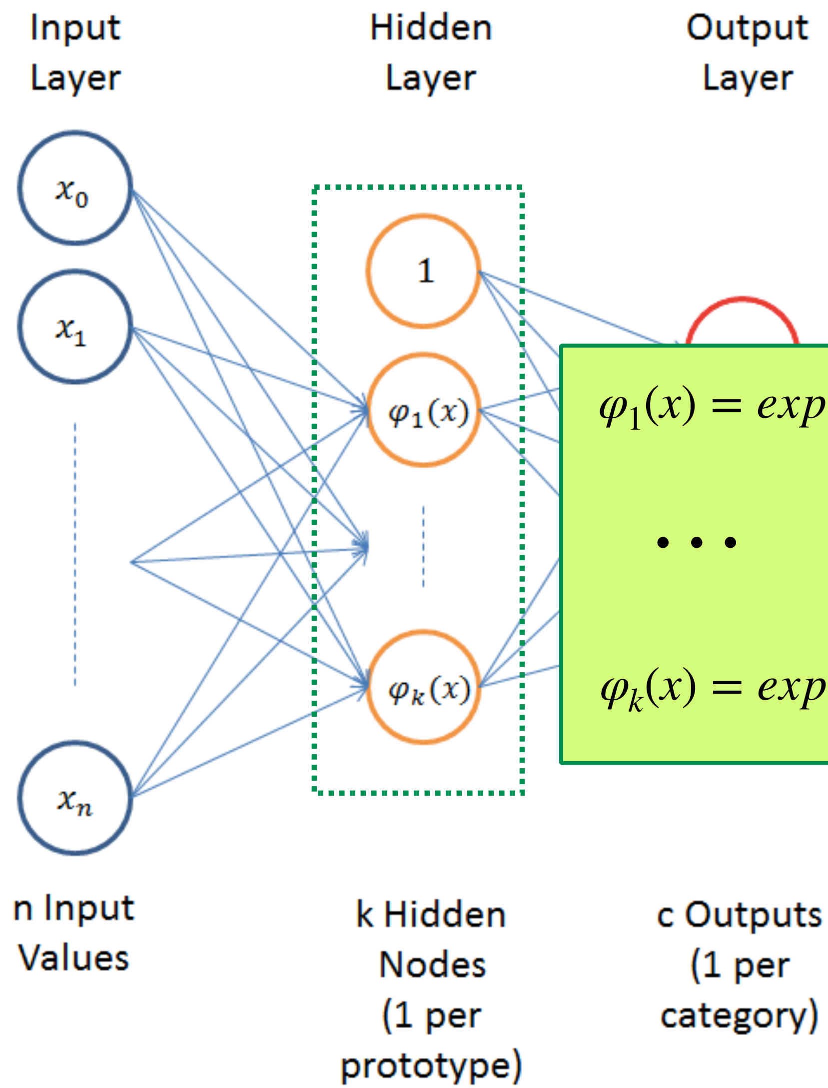
Redes RBF

- Variando γ:

$$\varphi(x) = exp^{-\gamma||x-\mu||^2}$$



Redes RBF

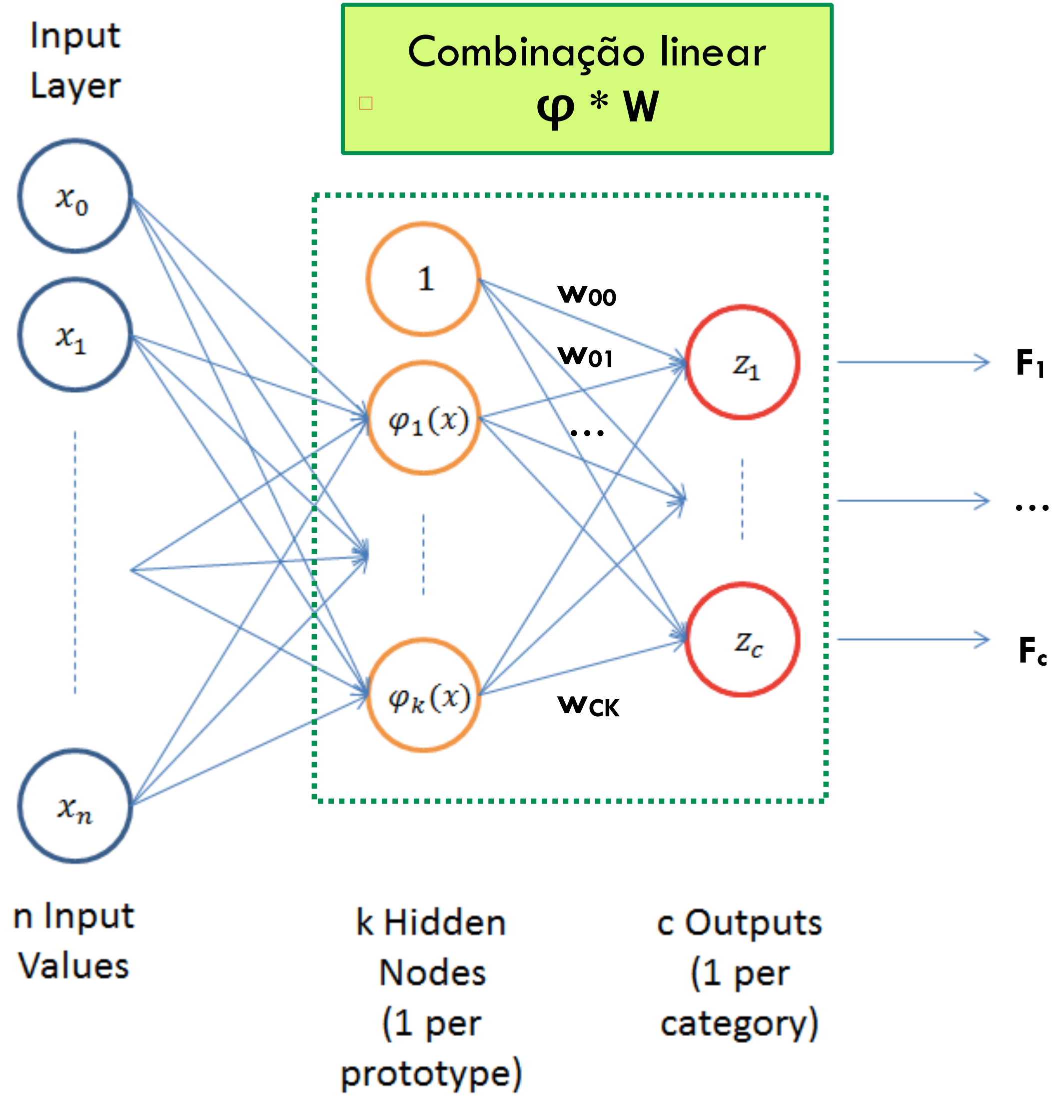


$$\varphi_1(x) = exp(-\gamma||x - \mu_1||^2)$$

$$\cdots$$

$$\varphi_k(x) = exp(-\gamma||x - \mu_k||^2)$$

Redes RBF


Combinação linear
φ * W
Input
Layer
x_0
x_1
x_n
n Input
Values
1
$\varphi_1(x)$
$\varphi_k(x)$
k Hidden
Nodes
(1 per
prototype)
w_{00}
w_{01}
...
w_{CK}
z_1
z_c
c Outputs
(1 per
category)
F_1
...
F_c

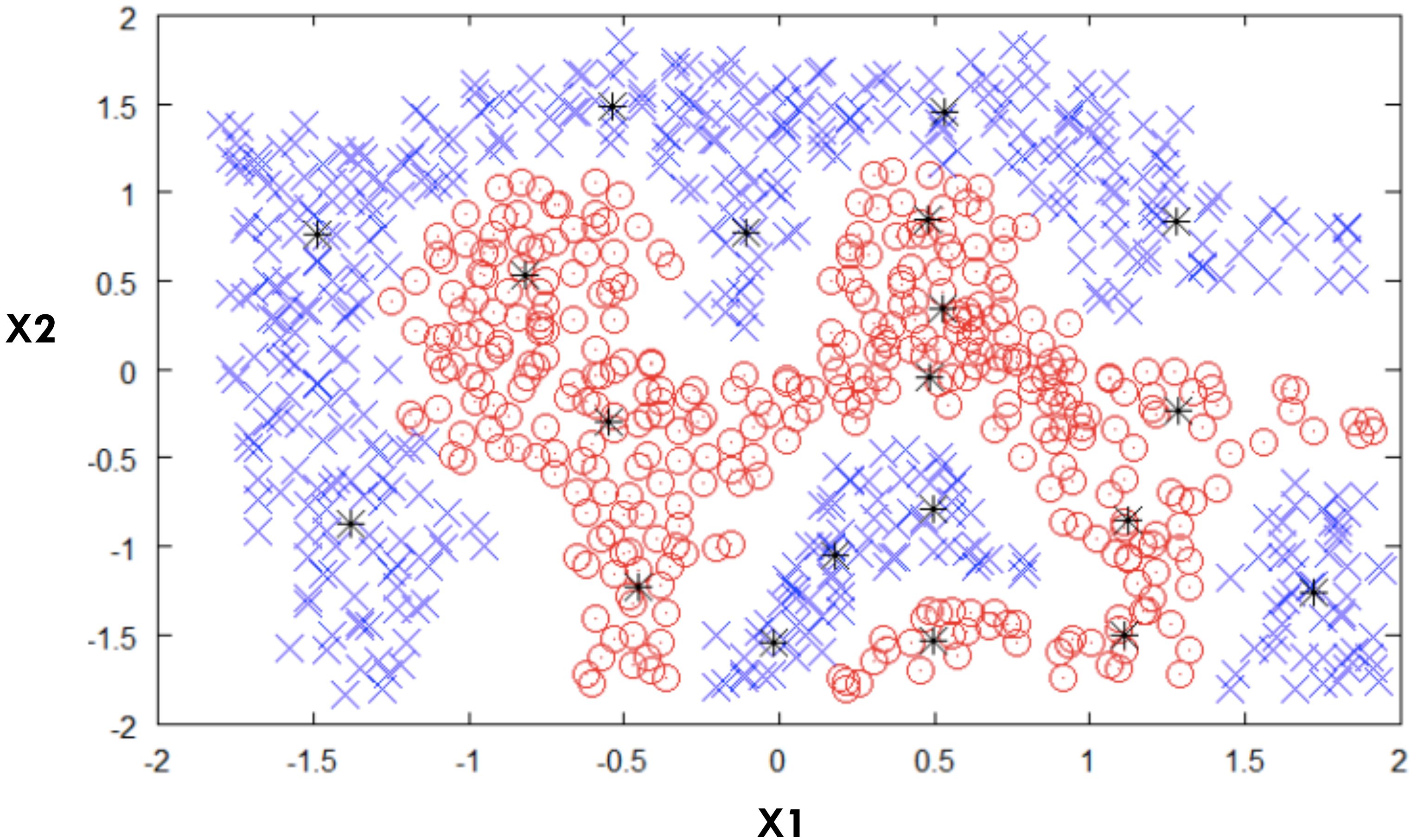
Redes RBF

- Cada neurônio de saída tem a forma:

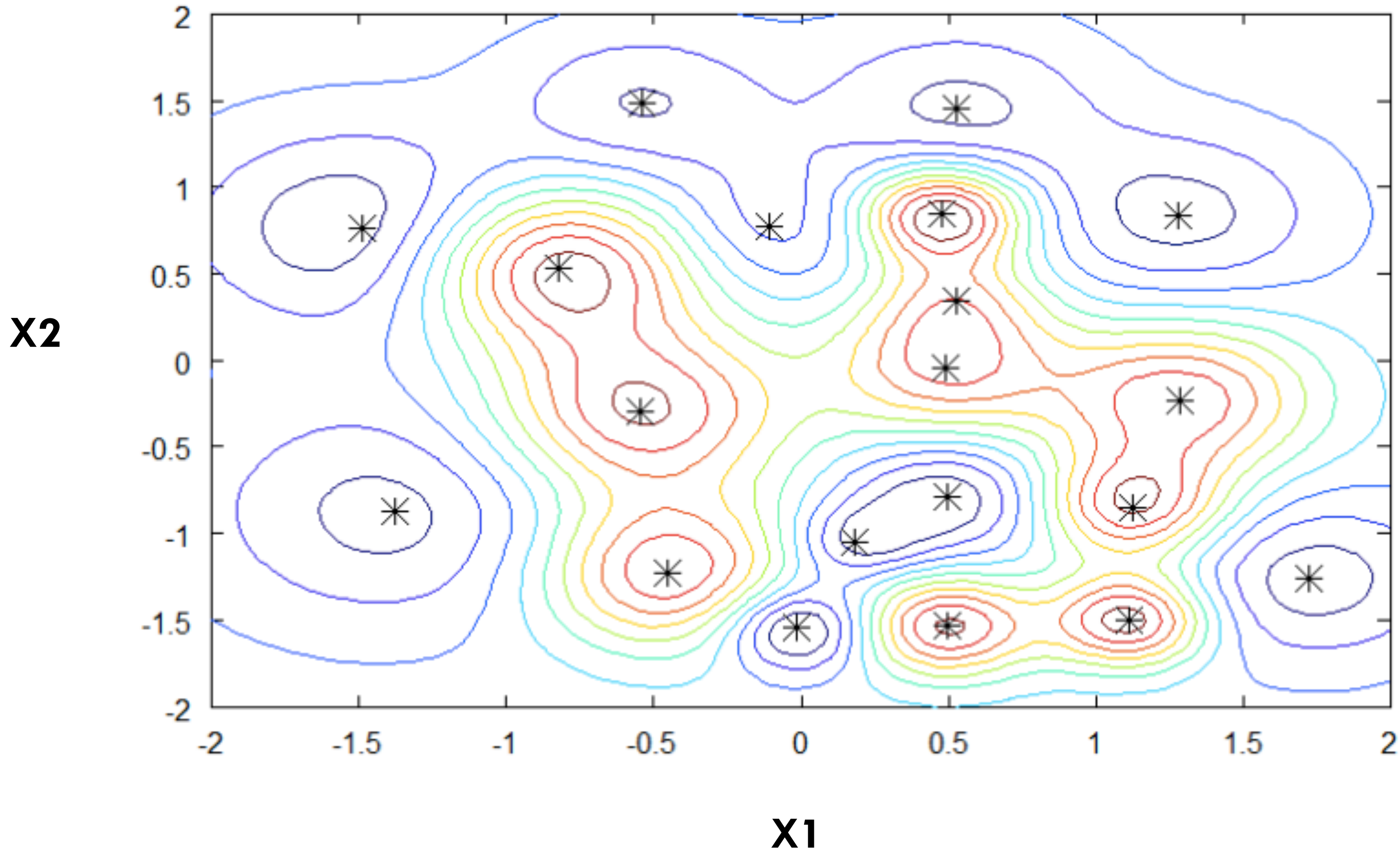
$$F(\mathbf{x}) = \sum_{j=1}^{K} w_j \varphi(\mathbf{x}, \mathbf{x}_j)$$

  - cada unidade ocupa é uma RBF φ (x,μ)
  - J = 1, ..., K
  - K é menor que N

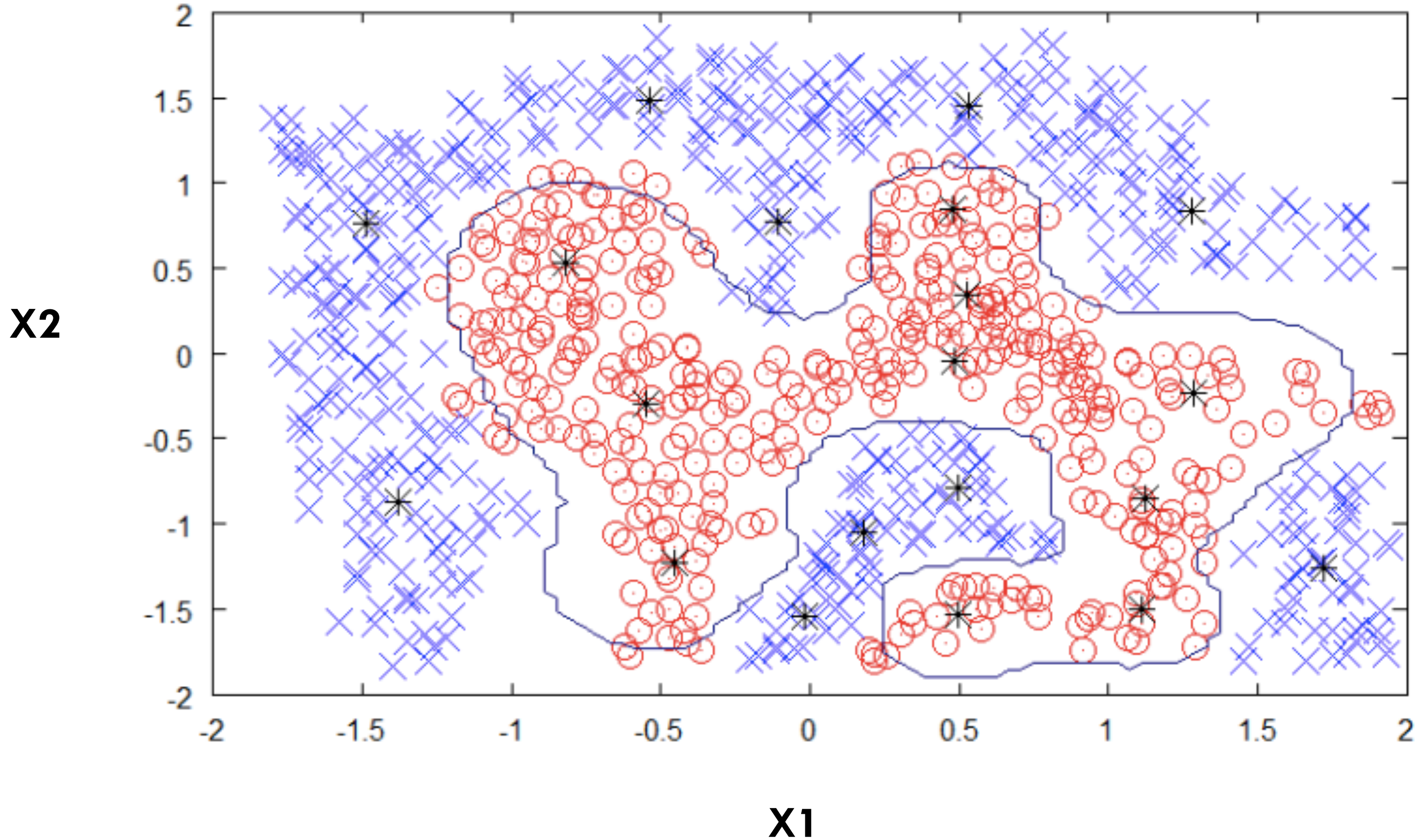
Exemplo



Exemplo



Exemplo



Roteiro

Algoritmo de treinamento

- $F(\mathbf{x}) = \sum_{i=i}^{N} w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|)$ pode ser vista como um conjunto de equações lineares para os coeficientes desconhecidos (pesos w_i):

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}$$

$$\varphi_{ij} = \varphi(\|x_i - x_j\|), \quad i, j = 1, 2, \ldots, N$$

Algoritmo de treinamento

$$\mathbf{d} = [d_1,\ d_2,\ \ldots,\ d_N]^T$$

$$\mathbf{w} = [w_1,\ w_2,\ \ldots,\ w_N]^T$$

- Os vetores **d** e **w** representam o vetor de saídas desejadas, e o vetor de pesos, respectivamente. N é o número de exemplos de treinamento

- ϕ é a matriz N **x** N de elementos φ_{ij} : $\phi = \{\varphi_{ij}\}_{i,j=1}^{N}$

Algoritmo de treinamento

- A matriz $\phi = \left\{ \varphi_{ij} \right\}_{i,j=1}^{N}$ é chamada **matriz de interpolação.** Ela pode ser reescrita na forma:

$$\boldsymbol{\phi}\mathbf{w} = \mathbf{x}$$

- Assumindo que ϕ é invertível, pode-se encontrar os pesos $\mathbf{w}$ por meio de:

$$\mathbf{w} = \boldsymbol{\phi}^{-1}\mathbf{x}$$

- Para a não singularidade de ϕ, todos os pontos $\left\{ \mathbf{x}_i \right\}_{i=1}^{N}$ devem ser diferentes (distintos)

Algoritmo de treinamento

- Pode-se ter um bias alimentando a rede:

$$\underbrace{\begin{bmatrix} 1 & \exp(-\gamma\|x_1 - x_1\|^2) & \cdots & \exp(-\gamma\|x_1 - x_N\|^2) \\ 1 & \exp(-\gamma\|x_2 - x_1\|^2) & \cdots & \exp(-\gamma\|x_2 - x_N\|^2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \exp(-\gamma\|x_N - x_1\|^2) & \cdots & \exp(-\gamma\|x_N - x_N\|^2) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{d}}$$

  - Φ deixa de ser quadrada (NxN), a formula de treinamento de W precisa ser adaptada para:

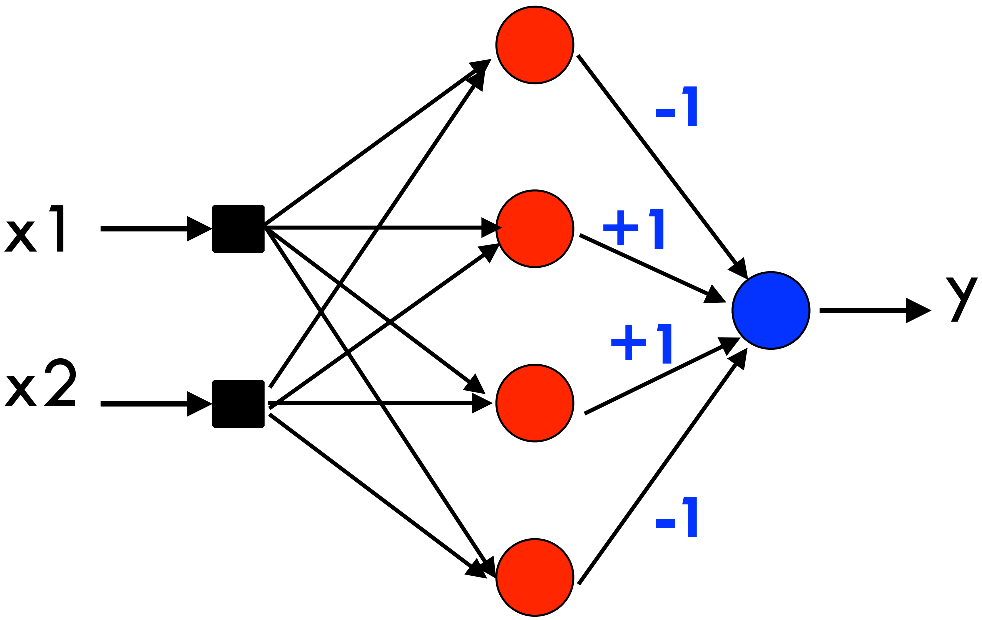
$$\mathbf{w} = \left(\phi^T \phi\right)^{-1} \phi^T \mathbf{d}$$

Algoritmo de treinamento

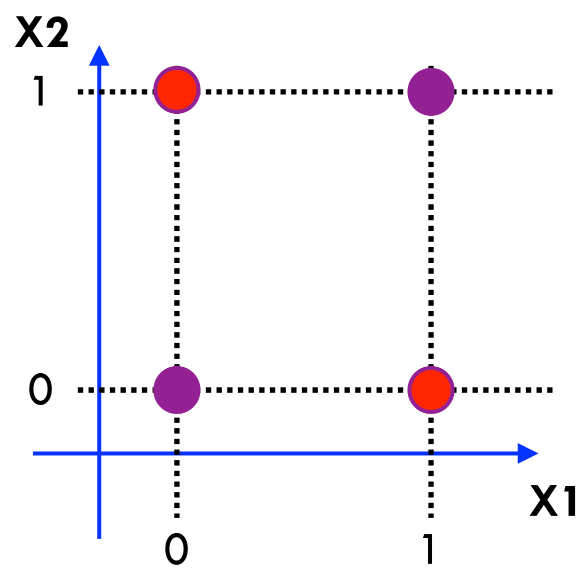
- **Passos:**
  1. definir a função **φ (gaussiana)**
  2. definir o tamanho da camada oculta (**K**)
  3. selecionar os **K protótipos** usados em cada uma das funções RBFs (unidades ocultas)
     1. agrupamento → k-MEANS (centróides do algoritmo)
  4. Computar os valores de Φ (matriz)
  5. Estimar os pesos sinápticos W, baseados em Y

Roteiro

Exemplo 2



$$\varphi_n(x) = e^{\frac{-\gamma||x-\mu_n||^2}{2}}$$



- φ1 → μ1 = (0,0)
- φ2 → μ2 = (0,1)
- φ3 → μ3 = (1,0)
- φ4 → μ4 = (1,1)

Roteiro

Síntese

- RBFs
  - três camadas (entrada, oculta, saída)
  - mapeamento não linear (entrada, oculta)
    - unidades gaussianas (RBFs)
    - k centróides/protótipos
  - mapeamento linear (oculta, saída)
  - treinamento é uma regressão de várias equações lineares

Próxima Aula

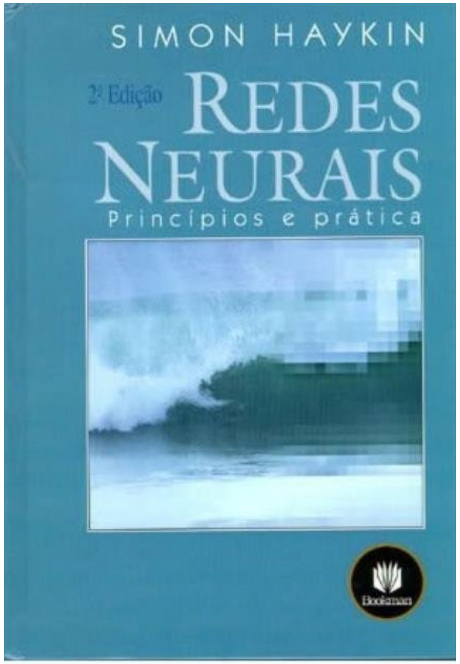
- Implementação de Redes RBF
- Deep Learning
- SVMs

- Revisão/atendimento (quinta)
- Prova (sexta)

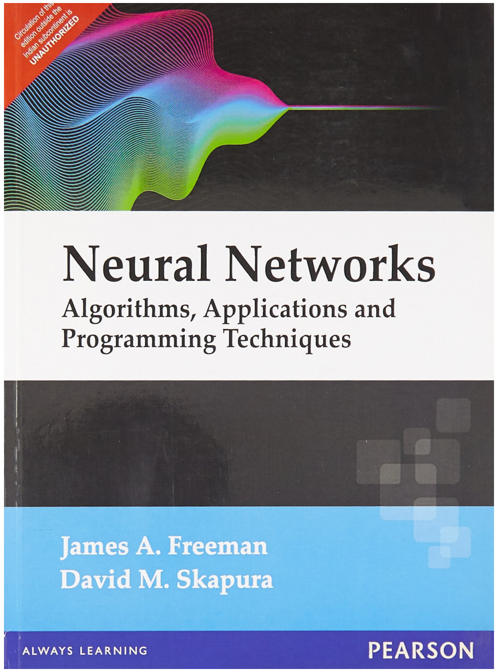
Roteiro

1. Introdução
2. Redes com Funções de Base Radial (RBF)
3. Algoritmo de Treinamento
4. Exemplo
5. Síntese / Próximas Aulas
6. **Referências**

Literatura Sugerida



(Haykin, 1999)



(Freeman & Skapura, 1991)

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@uel.br

Exercício

epoch	$\theta^h 0$	$\theta^h 1$	$\theta^o 0$	$W^h 00$	$W^h 10$	$W^h 01$	$W^h 11$	$W^o 00$	$W^o 01$
0	0.05	0.06	0.07	0.2	0.15	0.35	0.18	0.10	0.12
1									
2									

- X = XOR dataset
- $\eta = 0.2$
- **$f(net) = net^3 + 0.5$**
- **$f'(net) = 3*net^2$**