UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Redes RBF
Radial Basis Function

Gisele L. Pappa

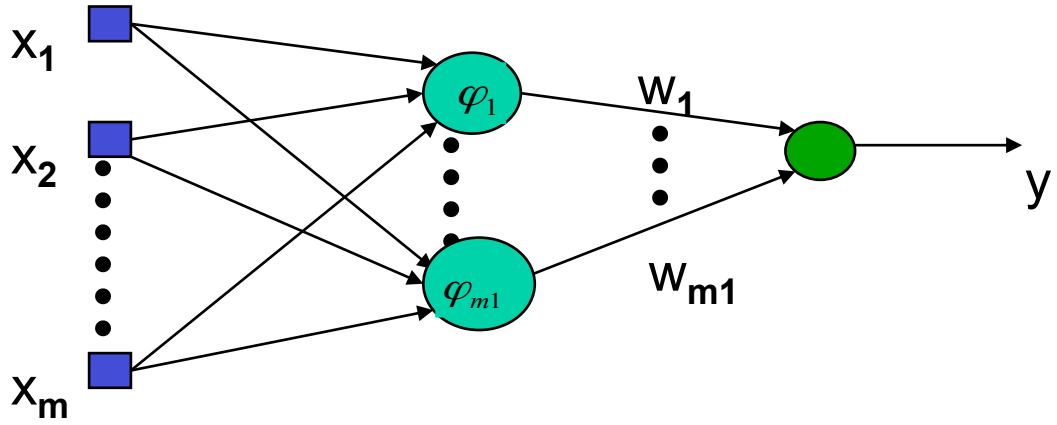
DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Introdução

- São redes de 3 camadas
- Os neurônios da camada escondida implementam funções de base radial
- Os nós de saída implementam funções lineares
- O treinamento é dividido em 2 estágios:
  - Determina os pesos da camada de entrada para a escondida
  - Determina os pesos da camada escondida para a de saída
- O aprendizado é rápido

Arquitetura

- Camada de entrada

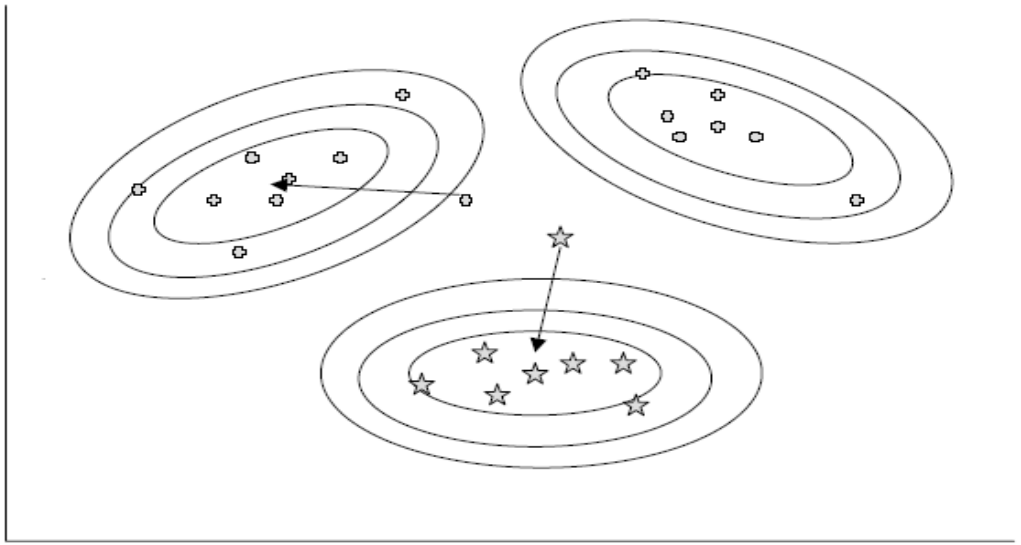


- Camada escondida
  - Aplica uma transformação não linear do espaço de entrada para o espaço escondido
- Camada de saída
  - Aplica uma transformção linear do espaço "escondido" para o espaço de saída

RBF

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

- A saída de rede é aproximadamente a combinação linear de funções de base radial
- RBFs capturam o comportamento local das funções



O que é uma RBF ?

- Funções de base radial
  - Radial: Simétrica em torno do centro
  - Funções base
    - Também conhecidas como kernels
    - Conjunto de funções cuja combinação linear pode gerar uma função arbitrária em um dado espaço de funções

Qual a função de um kernel?

- Digamos que você tenha um problema em que deseja separar 2 classes, mas a borda que você tem que utilizar se confunde
- Você pode encontrar um algoritmo que separe os pontos, mas ele irá demorar para convergir
- Funções de kernel mapeam esses dados em espaços de maior dimensão, na esperança de que os dados possam ser mais facilmente separados

RBFs

- Qual a motivação para utilizar uma função não-linear seguida de uma linear?
  - Teorema de Cover sobre a separabilidade de padrões:
    - "Um problema de classificação de padrões complexos moldado não-linearmente em um espaço com muitas dimensões tem maior probabilidade de ser linearmente separável que quando moldado em um espaço com poucas dimensões"
  - Quando a camada escondida aplica uma transformação não-linear no espaço de entrada, ela cria um novo espaço tipicamente com mais dimensões que o espaço de entrada

* Mesmo argumento para trabalhar com SVM

Tipos de Função RBF

- Multiquadráticas:

$$\phi(r) = (r^2 + \sigma^2)^{1/2} \qquad \sigma > 0$$

- Multiquadráticas inversas:

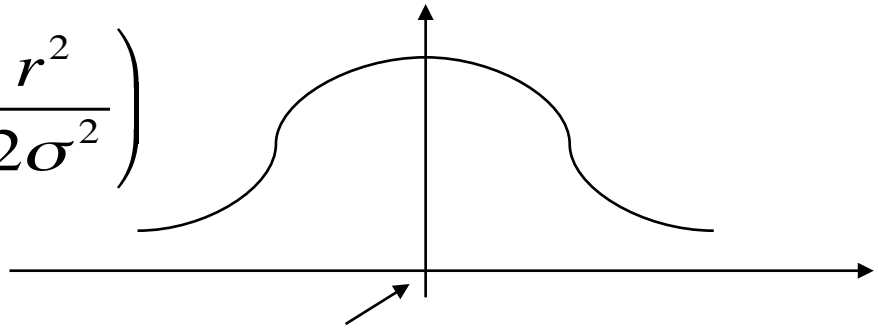
$$\phi(r) = \frac{1}{(r^2 + \sigma^2)^{1/2}} \qquad \sigma > 0$$

- Gaussianas:

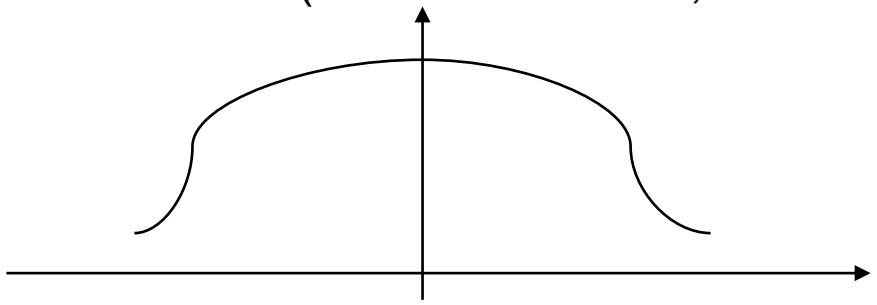
$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad \sigma > 0$$

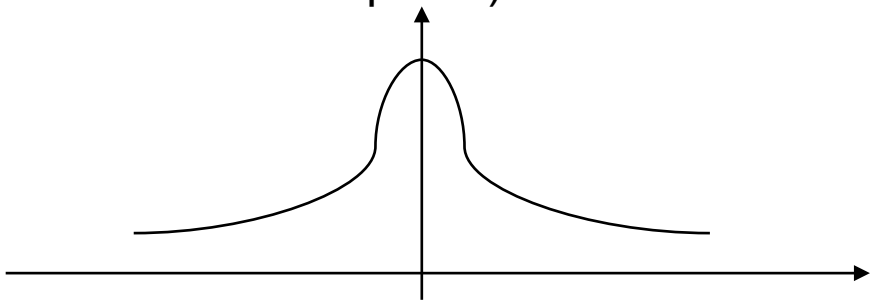
Função RBF Gaussiana

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$$



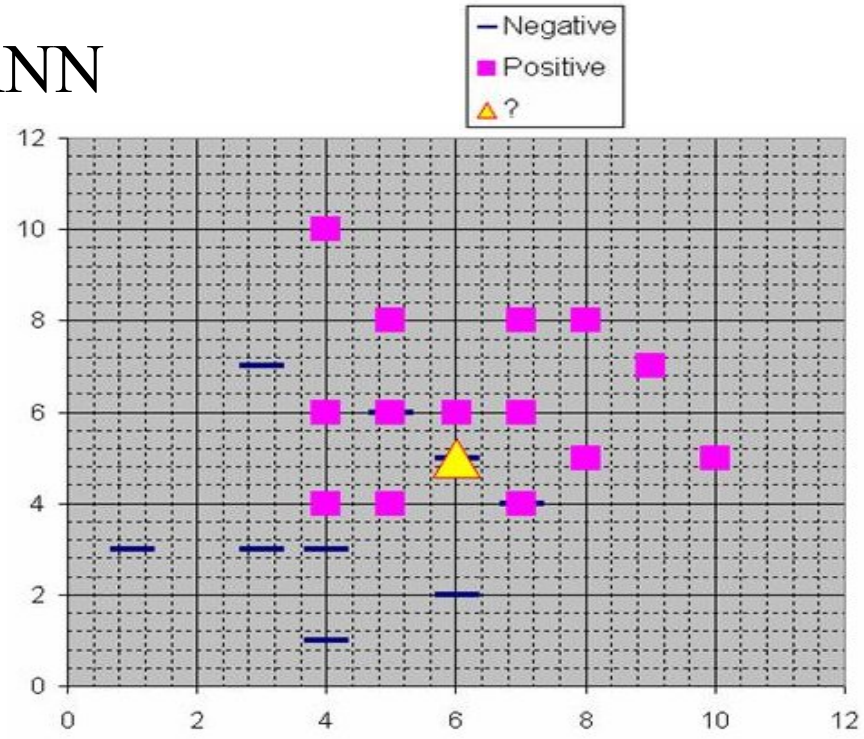
σ é a medida do quão "achatada" a curva é
(dentro de um raio, define a influência de um ponto):


Valor de σ é grande


Valor de σ é pequeno

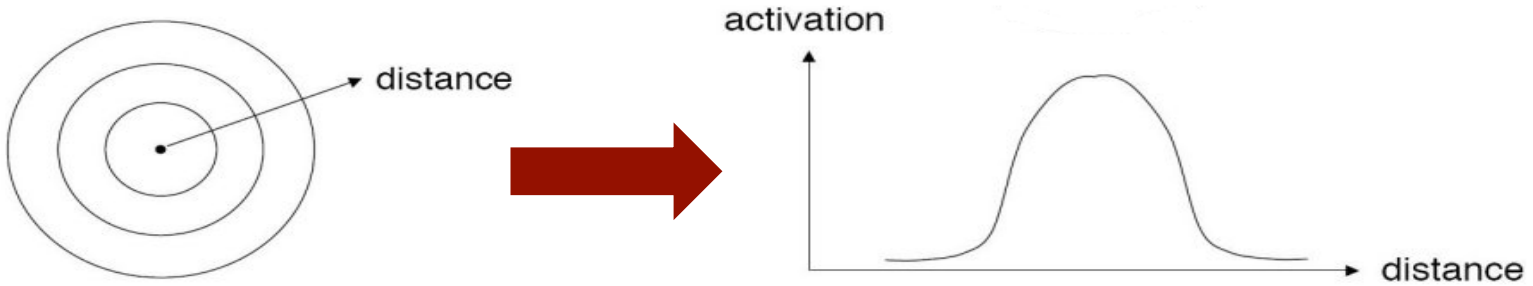
Como uma Rede RBF funciona?

- Conceitualmente, similar ao KNN
- A rede posiciona 1 ou mais neurônios no espaço n-D descrito pelos n atributos que descrevem o exemplo
- Calcula a distância Euclidiana do ponto sendo avaliado para o centro de cada neurônio

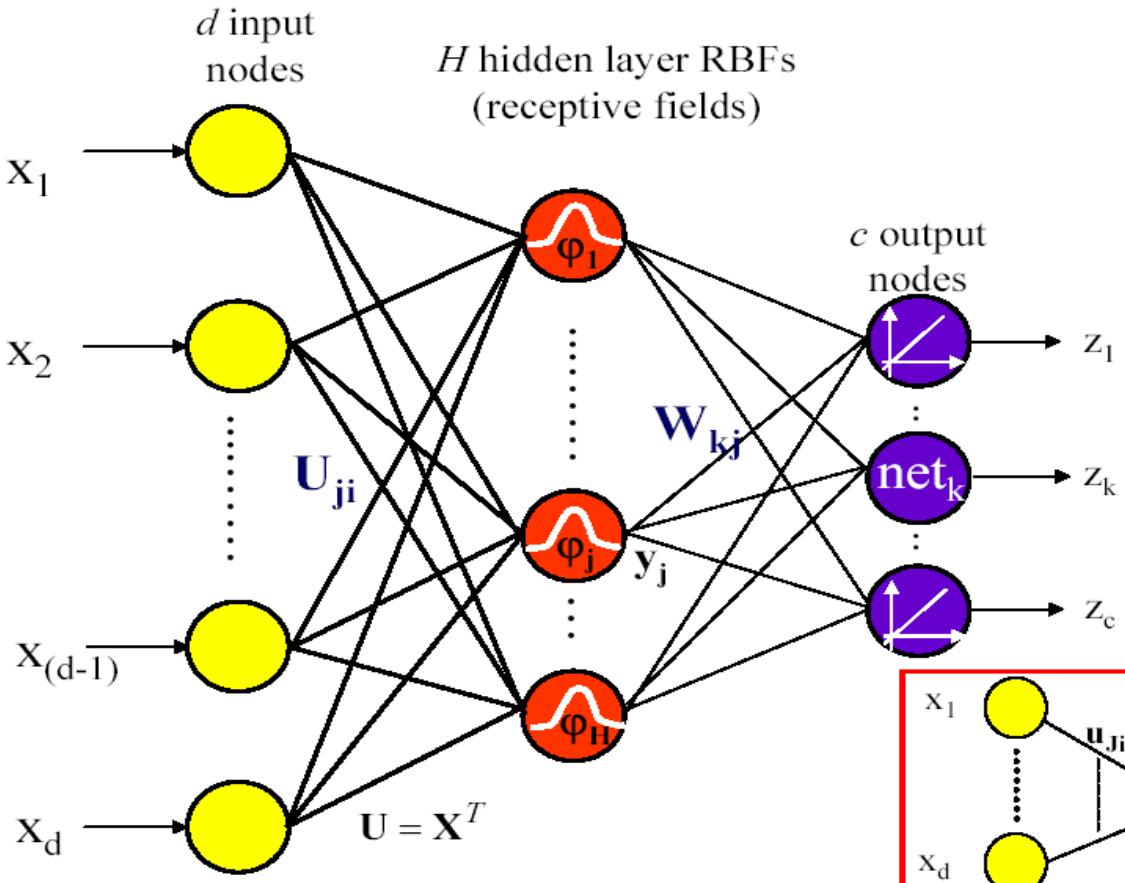


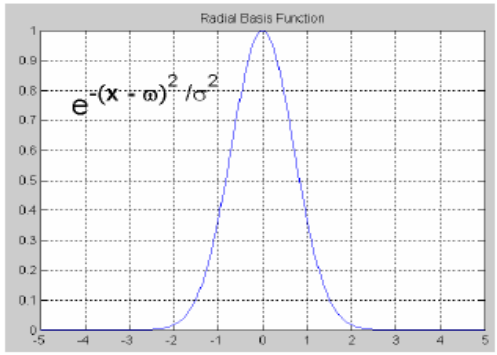
Como uma Rede RBF funciona?

- Aplica uma função RBF a cada distância ponto-neurônio, para computar o peso (influência) de cada neurônio
  - Peso(neurônio) = RBF (distância)
- Quanto mais longe o neurônio está do ponto sendo avaliado, menor seu peso.



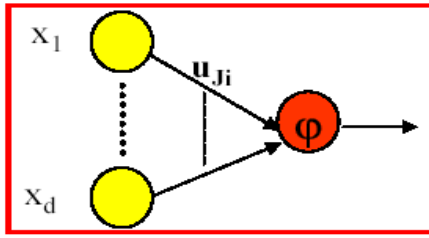
RBF





$$z_k = f(net_k) = f\left(\sum_{j=1}^{H} w_{kj} y_j\right) = \sum_{j=1}^{H} w_{kj} y_j$$

Linear act. function



$$\varphi(net_J = \|\mathbf{x} - \mathbf{u_J}\|)$$

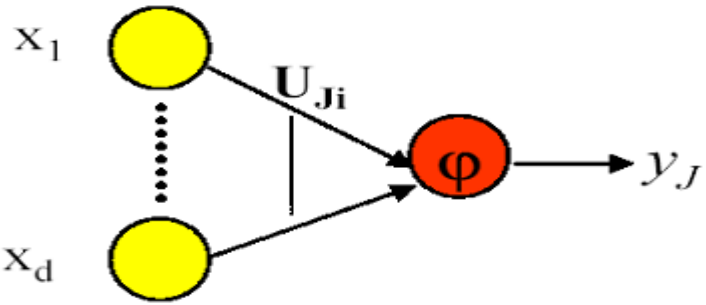
$$= e^{-\left(\frac{\|\mathbf{x}-\mathbf{u_J}\|}{\sigma}\right)^2}$$

σ: spread constant

Neurônios da Camada Escondida

- Utilizam uma função de base radial

  $\phi(|| x - u||^2)$ A saída depende da distância da entrada x e do centro u



Norma euclidiana

$$= e^{-\left(\frac{\|\mathbf{x}-\mathbf{u}\|}{\sigma}\right)^2}$$

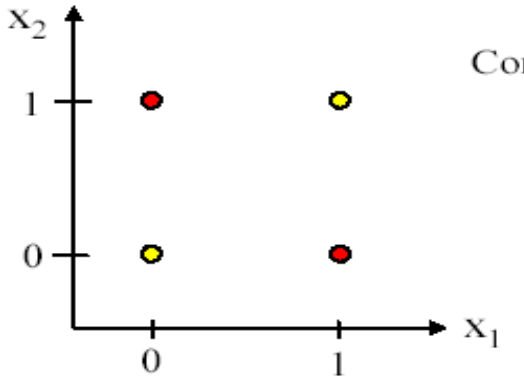
$\phi_j(|| x - u||^2)$

u é o centro da função
σ é o *spread* da função
* Ambos são parâmetros

Neurônios da Camada Escondida

- Os neurônios da camada escondida são mais sensíveis as entradas de dados próximas ao seu centro. Essa sensibilidade pode ser controlada pelo parâmetro σ.
  - Quanto maior o valor de σ, menor a sensibilidade

Exemplo -XOR



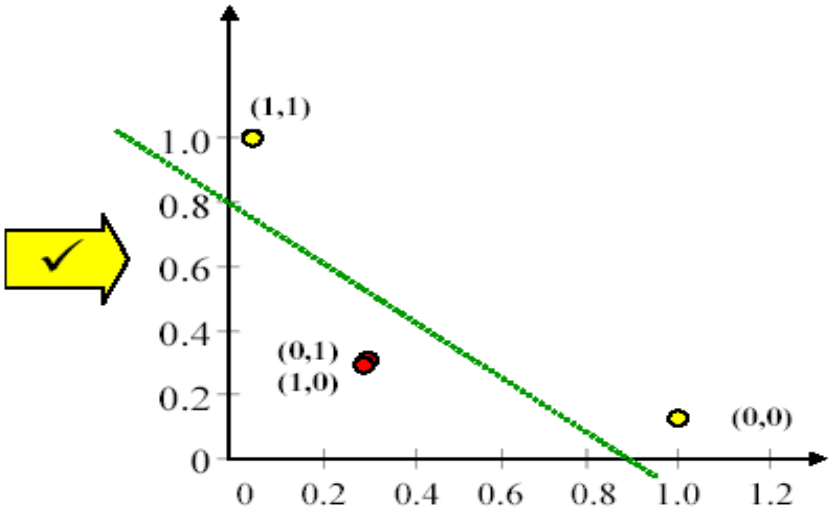
Consider the nonlinear functions to map the input vector **x** to the φ_1- φ_2 space

$\mathbf{x}=[x_1\ x_2]$

$$\varphi_1(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{u}_1\|^2} \qquad U_1 = [1\ \ 1]^T$$

$$\varphi_2(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{u}_2\|^2} \qquad U_2 = [0\ \ 0]^T$$

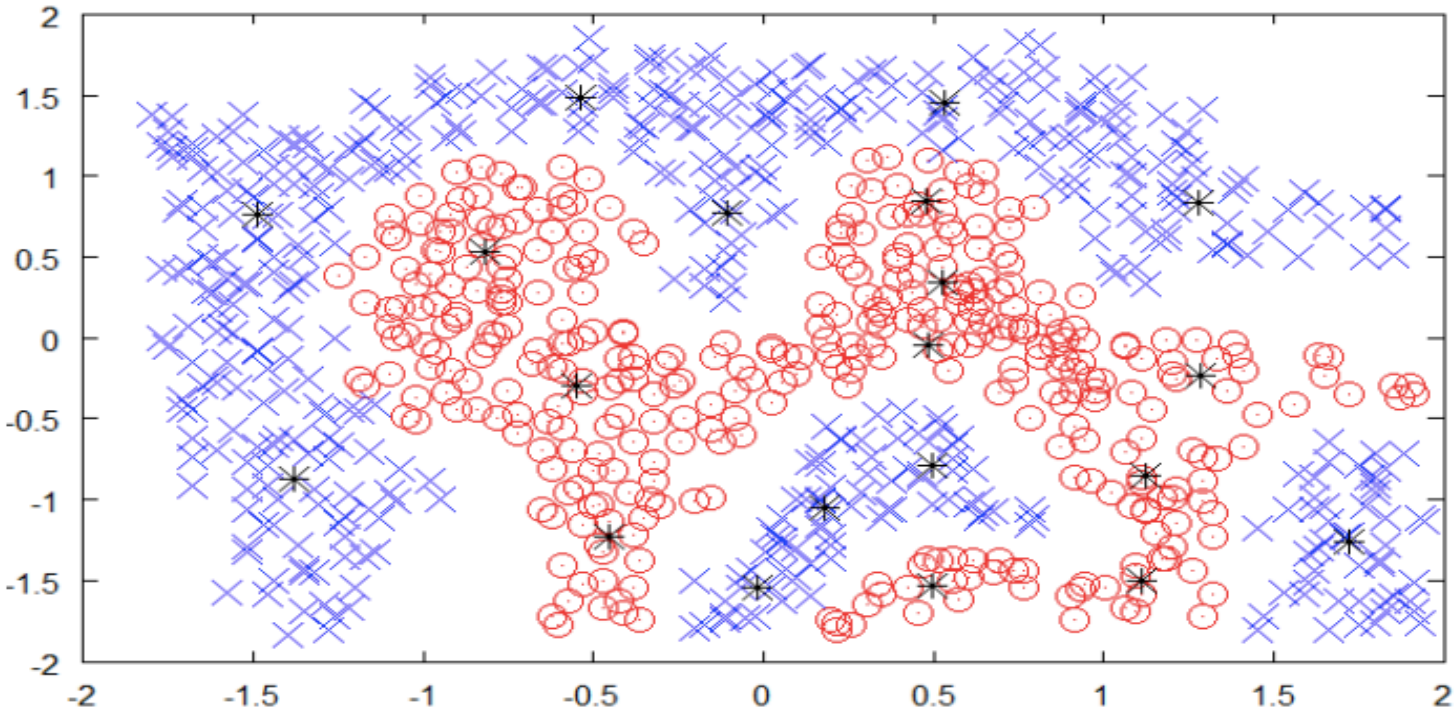
Input x	$\varphi_1(x)$	$\varphi_2(x)$
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(1,0)	0.3678	0.3678
(0,0)	0.1353	1



The nonlinear φ function transformed a nonlinearly separable problem into a linearly separable one !!!

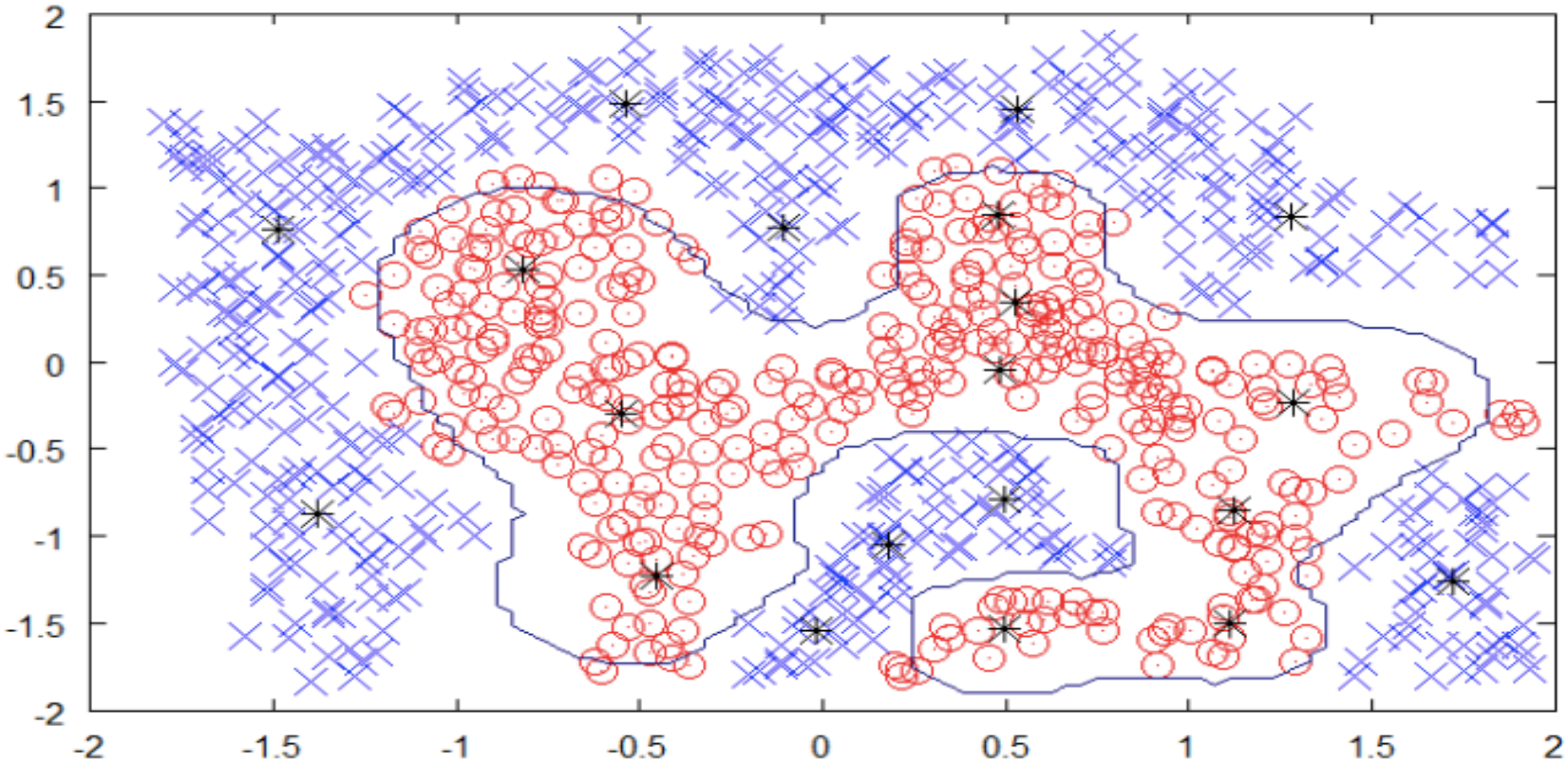
UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

RBFs



DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

RBF



UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Definição do Problema

- Dado um vetor de entrada de D dimensões $x^p = \{x^p_i: i = 1,..,D\}$, queremos encontrar um vetor de saída correspondente no espaço de c dimensões, $t^p = \{t^p_k: k = 1,..,c\}$
- Essas saída serão geradas por um conjunto de funções $g_k(x)$. A ideia é aproximar $g_k(x)$ com funções $y_k(x)$ da seguinte forma

$$y_k(\mathbf{x}) = \sum_{j=0}^{M} w_{kj}\phi_j(\mathbf{x})$$

M : número de neurônios na camada escondida

Definição do Problema

- O problema se transforma em encontrar valores apropriados para:
  - M : número de neurônios na camada escondida
  - U_{ij} : centros (valores dos neurônios da camada escondida)
  - σ_j, *spread* da função
  - W_{kj}: pesos da camada escondida para camada de saída

Treinamento

- Neurônios da camada de entrada para escondida tem funções bem diferentes dos da camada escondida para de saída
- Em um passo inicial, otimizamos os parâmetros das funções RBF
  - Os centros (protótipos dos dados)
  - Os *spreads*
- Em um segundo passo, deixamos esses parâmetros fixos e treinamos a segunda parte da rede

Algoritmos de Aprendizado

- 3 mais comuns:
  - Centros aleatórios + método da pseudo-inversa
  - K-means + Least Mean Square (RMS)
  - Descida do gradiente (funciona em uma única fase)

Algoritmos de Aprendizado

- 3 mais comuns:
  - Centros aleatórios + método da pseudo-inversa
    - Centros U são escolhidos como exemplos do conjunto de treinamento e spreads por normalização

$$\sigma = \frac{\text{Distância máxima entre 2 centros}}{\sqrt{\text{número de centros}}} = \frac{d_{max}}{\sqrt{m_1}}$$

    - Pesos para uma rede de uma camada com função linear são calculados usando o método da pseudo-inversa (pseudo-inversa pode ser calculada usando, por exemplo, SVD)

$$y_k(\mathbf{x}^p) = \sum_{j=0}^{M} w_{kj}\phi_j(\mathbf{x}^p)$$

Algoritmos de Aprendizado

- 3 mais comuns:
  - K-means + LMS
    - Centros U são escolhidos utilizando o algoritmo K-means e spreads por normalização
    - Pesos da camada de saída aprendidos usando a regra delta

$$\mathbf{w}(t) = \mathbf{w}(t) + \eta\, e_i\, \mathrm{x} \quad \text{(peso)}$$

η é a taxa de aprendizagem

e_i é o erro entre a saída encontrada e a desejada

Algoritmos de Aprendizado

- 3 mais comuns:
  - Centros aleatórios + método da pseudo-inversa
  - K-means + Least Mean Square (RMS)
  - Descida do gradiente (funciona em uma única fase)

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Redes RBF
Radial Basis Function

Gisele L. Pappa

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO