

Radial Basis Function (RBFs)

2º Semestre de 2009

Cleber Zanchettin
UFPE - Universidade Federal de Pernambuco
CIn - Centro de Informática

Introdução

- Tratam-se de redes neurais artificiais com múltiplas camadas que **não são treinadas por retropropagação** (*backpropagation*), e que não têm unidades de processamento com função de ativação do tipo sigmoidal.
- Estas redes utilizam funções de ativação com **campos receptivos locais** (*local receptive fields*).
- Esta abordagem emprega, na maioria dos casos, treinamento **supervisionado** e **não-supervisionado**.
- As redes são muito usadas como **interpoladores / aproximadores** e em tarefas de classificação.

Introdução



- Esta abordagem é inspirada na propriedade de alguns **neurônios biológicos** chamada de **resposta localmente sintonizada** (*locally tuned response*).
- Tais células nervosas respondem **seletivamente** a um intervalo finito do espaço de sinais de entrada.
- Os modelos atuais de RBF se diferenciam dos primeiros pois são de **natureza adaptativa** que permite a utilização, em muitas situações, de um número relativamente **menor** de unidades de processamento localmente sintonizadas.

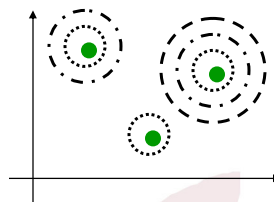
Radial-Basis Function Networks



- A função é aproximada como uma combinação linear de Funções de Base Radial (*Radial Basis Functions - RBF*). As RBFs capturam o comportamento local das funções.

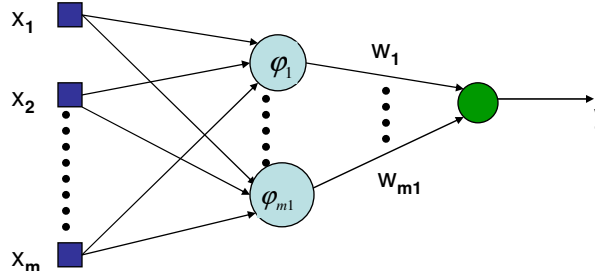
Motivação Biológica:

- RBF's representam receptores locais:



Arquitetura

- **Camada de entrada:** nodos que conectam a rede ao ambiente.



- **Camada Escondida:** aplica uma transformação não-linear do espaço de entrada para uma representação no espaço gerado pelas ativações dos neurônios da camada escondida
- **Camada de Saída:** aplica uma transformação linear do espaço escondido para o espaço de saída

Idéia principal

Quando uma rede de função de base radial (RBF) é usada para realizar uma tarefa *complexa* de classificação de padrões, o problema é resolvido basicamente pela **transformação do espaço de entrada para um espaço de alta dimensionalidade, de forma não-linear**.

POR QUE ?

Pela justificativa baseada no **Teorema de Cover** (1965) de que:

“Um problema complexo de classificação de padrões disposto não-linearmente em um espaço de alta dimensão tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade”.

Separabilidade por φ de padrões



Do estudo dos *perceptrons*, sabe-se que, uma vez que tenhamos padrões linearmente separáveis, o problema de classificação é relativamente simples de ser solucionado.

Assim, a compreensão sobre a operação de uma rede RBF como um classificador de padrões pode ser iniciada ao se estudar a separabilidade de padrões.

Para tal, considere uma família de superfícies em que cada uma divide naturalmente um espaço de entrada em duas regiões.

Considere que \mathcal{S} represente um conjunto de N padrões (vetores) $x_1, x_2, x_3, \dots, x_N$, cada um dos quais pertencente a uma de duas classes \mathcal{S}_1 e \mathcal{S}_2 . Dizemos que esta dicotomia (partição binária) dos pontos é separável, em relação à família de superfícies, se existir um hiperplano que separe os pontos da classe \mathcal{S}_1 daqueles da classe \mathcal{S}_2 .

Separabilidade por φ de padrões



Para cada padrão $x \in \mathcal{S}$, defina um vetor constituído por:

$$\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_{m_1}(x)]^T$$

sendo:

φ_i Função Escondida

$\{\varphi_i(x)\}_{i=1}^{m_1}$ Espaço Escondido

Uma dicotomia $\{\mathcal{S}_1, \mathcal{S}_2\}$ de \mathcal{S} é dita *separável por φ* , se existir um vetor w de dimensão m_1 tal que:

$$w^T \varphi(x) > 0, \quad x \in \mathcal{S}_1$$

$$w^T \varphi(x) < 0, \quad x \in \mathcal{S}_2$$

Exemplos de separabilidade por ϕ

O hiperplano definido pela equação:

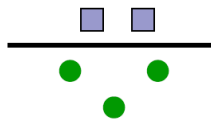
$$w^T \phi(x) = 0$$

descreve a superfície de separação no **espaço ϕ** (espaço escondido).

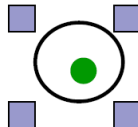
Assim, a **Superfície de separação** é formada por todos os pontos x tais que $w^T \phi(x) = 0$.

Exemplos de dicotomias separáveis:

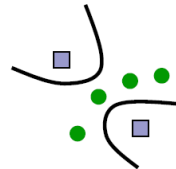
Linearmente separável



Esfericamente separável



Quadraticamente separável

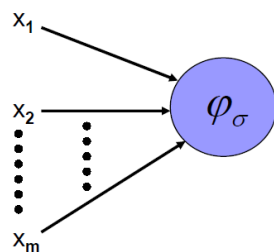


Modelo do neurônio escondido

- **Unidades escondidas:** use uma função de base radial

$$\phi_{\sigma}(\|x - t\|^2)$$

A saída depende da distância entre a entrada x e o centro t



$$\phi_{\sigma}(\|x - t\|^2)$$

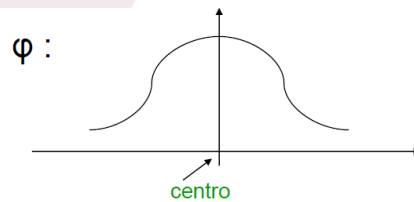
t é denominado **centro**
 σ é denominado **espalhamento**
centro e espalhamento são parâmetros

Neurônios escondidos

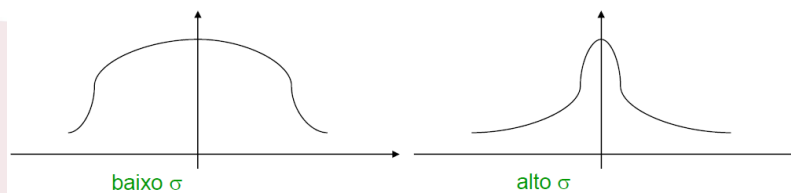


- Os neurônios da camada escondida são mais sensíveis às entradas próximas aos seus centros. Esta sensibilidade pode se ajustar através do espalhamento σ .
- **Espalhamento grande** \Rightarrow **menos sensibilidade**

Função de base radial gaussiana ϕ



σ é a medida do espalhamento da curva:



Tipos de ϕ



■ Multiquadráticas:

$$\phi(r) = (r^2 + c^2)^{1/2} \quad \begin{array}{l} c > 0 \\ r = \|x - t\| \end{array}$$

■ Multiquadráticas inversas:

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

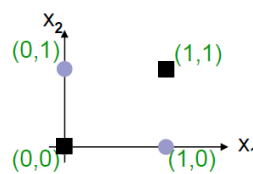
■ Funções gaussianas:

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

Exemplo: o problema XOR



■ Espaço de entrada:



■ Espaço de saída:



■ Construção de um classificador de padrões RBF tal que:

(0,0) e (1,1) são mapeados para 0, classe C1

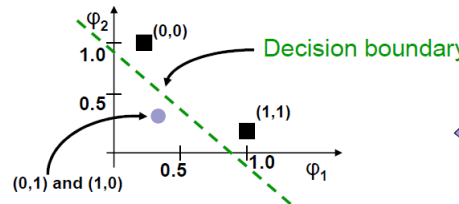
(1,0) e (0,1) são mapeados para 1, classe C2

Exemplo: o problema XOR

- No espaço escondido:

$$\varphi_1(x_1, x_2) = e^{-\|x-t_1\|^2} \quad t_1 = [1,1]^T$$

$$\varphi_2(x_1, x_2) = e^{-\|x-t_2\|^2} \quad t_2 = [0,0]^T$$



- Quando realizamos o mapeamento para o espaço escondido $\langle \varphi_1, \varphi_2 \rangle$, C1 e C2 se tornam *linearmente separáveis*. Assim, um classificador linear pode ser usado no espaço de $\varphi_1(x)$ e $\varphi_2(x)$ para solucionar o problema XOR.

Aprendizagem em Redes RBF

- O método de treinamento, como os demais modelos supervisionados, deve **reduzir o erro na saída** da rede a valores aceitáveis através da adaptação dos parâmetros livres na rede RBF: **centros**, **larguras dos campos receptivos** e **pesos** entre a camada escondida e a de saída. A aprendizagem pode ser **supervisionada**, **não-supervisionada** ou **híbrida**.
- O treinamento híbrido combina aprendizagem não-supervisionada (ANS) com supervisionada (AS). Este é o tipo de treinamento **mais empregado** pois, em geral, não se sabe que saídas se desejam para a camada escondida.
 - ANS: treina a camada escondida definindo seus parâmetros livres.
 - AS: determina os valores dos pesos entre as camadas escondida e de saída, considerando constantes os parâmetros já definidos.

Aprendizagem em Redes RBF



▪ Treinamento não-supervisionado

- Determinar os valores dos centros por: (a) seleção aleatória; (b) distribuição sobre uma grade regular; (c) técnica de agrupamento (*clustering*); ou (d) outro algoritmo.
- Determinar a largura do campo receptivo através de uma heurística que emprega (a) distância euclidiana média entre centros; (b) distância euclidiana entre centro e vetor de entrada; (c) distância euclidiana entre centros; ou (d) distância euclidiana entre os centros determinados pelo método *k*-médias.

▪ Treinamento supervisionado

- Determinar os pesos por um método que resolva o problema de minimização do erro: (a) método dos mínimos quadrados; (b) método da regra delta; ou (c) matriz pseudo-inversa.

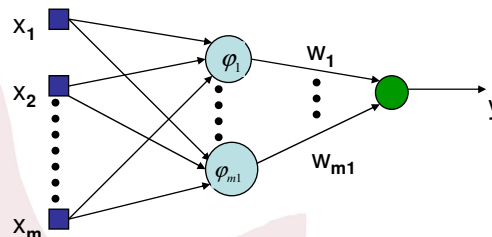
Algoritmos de aprendizado



▪ Os **parâmetros** a serem ajustados são:

- **centros**
- **espalhamentos**
- **Pesos**

▪ Há diferentes algoritmos de aprendizagem



Algoritmo de aprendizagem 1



- **Centros são selecionados aleatoriamente**

- As localizações dos **centros** são escolhidas aleatoriamente a partir do **conjunto de treinamento**

- **Espalhamentos** escolhidos por **normalização**:

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

$$\varphi_i(\|x - t_i\|^2) = \exp\left(-\frac{m_1}{d_{\max}^2} \|x - t_i\|^2\right)$$
$$i \in [1, m_1]$$

Algoritmo de aprendizagem 1



- **Pesos** são encontrados através de **métodos de pseudo-inversa**

$$w = \varphi^+ d \leftarrow \text{Resposta desejada}$$

Pseudo-inversa de φ

$$\varphi = \{\varphi_{ji}\} \quad \varphi_{ji} = \exp\left(-\frac{m_1}{d_{\max}^2} \|x_j - t_i\|^2\right)$$
$$j = 1, 2, \dots, N, \quad i = 1, 2, \dots, m_1$$

Algoritmo de aprendizagem 2



- **Processo de aprendizagem Híbrido:**
 - **Etapa de aprendizagem auto-organizada** para encontrar os centros
 - **Espalhamento** escolhido por normalização
 - **Etapa de aprendizagem supervisionada** para encontrar os pesos, usando algoritmo LMS (*Least Mean Square*)

Algoritmo de aprendizagem 2: Centros



- **Algoritmo de agrupamento de K-médias para os centros**

- 1 **Inicialização:** $t_k(0)$ random $k = 1, \dots, m_1$
- 2 **Amostragem:** retire um vetor x do espaço de entrada C
- 3 **Casamento de similaridade:** encontrar o centro mais próximo

$$k(x) = \arg \min_k \|x(n) - t_k(n)\|$$

- 4 **Atualização:** ajuste dos centros

$$t_k(n+1) = \begin{cases} t_k(n) + \eta [x(n) - t_k(n)] & \text{se } k = k(x) \\ t_k(n) & \text{caso contrário} \end{cases}$$

5. **Continuação:** incremente n de 1, volte para o passo 2 e continue o procedimento até que não sejam mais observadas mudanças

Algoritmo de aprendizagem 3

- Aprendizagem supervisionada para todos os parâmetros utilizando o método do gradiente descendente

- **Modificando os centros**

$$\Delta t_j = -\eta_{t_j} \frac{\partial E}{\partial t_j}$$

Taxa de aprendizagem para t_j

Função de erro

$$\frac{1}{2} \sum_{i=1}^N e_i^2$$

Algoritmo de aprendizagem 3

- **Modificando os espalhamentos**

$$\Delta \sigma_j = -\eta_{\sigma_j} \frac{\partial E}{\partial \sigma_j}$$

- **Modificando os pesos da saída**

$$\Delta w_{ij} = -\eta_{ij} \frac{\partial E}{\partial w_{ij}}$$

Estratégias de Treinamento



Busca-se o compromisso entre **precisão** e **generalização**. Para tal, duas estratégias de treinamento podem ser empregadas: **hold-out** e **validação cruzada** (*crossvalidation*).

- **Hold-out**: Neste procedimento o conjunto de padrões é dividido em três grupos: treinamento, validação e teste. Cada topologia com seus centros tem seu desempenho testado com respeito aos três conjuntos.

Estratégias de Treinamento



Cross-validation: Nesta estratégia, todos os padrões são considerados para o treinamento (em geral são poucos). Neste caso, divide-se os padrões em n grupos. Seleciona-se aleatoriamente $n-1$ conjuntos para treinamento e testa-se a rede com aquele conjunto que não foi selecionado.

Este processo deve prosseguir até que todos os conjuntos tenham sido usados para testes. A partir daí, calcula-se o erro (E).

Comparação com as Redes MLP



As redes RBF são utilizadas em tarefas complexas (não-lineares) de classificação de padrões.

Comparação entre redes RBF e redes MLP:

- Ambas são exemplos de redes *non-linear layered feed-forward*.
- Ambas são **aproximadores universais**.
- Quanto às camadas escondidas:
 - Na rede RBF há uma única camada escondida.
 - Na rede MLP pode haver diversas camadas escondidas.

Comparação com as Redes MLP



- Modelos de Neurônios:
 - Os neurônios da camada escondida de uma RBF são diferentes. Eles têm um propósito diferentes daquele dos neurônios da camada de saída.
 - Os neurônios típicos na camada escondida ou de saída em uma MLP comungam de um *modelo comum de neurônio*.
- Linearidade:
 - A camada escondida de uma rede RBF é *não-linear*, a camada de saída da RBF é *linear*.
 - As camadas escondida e de saída em uma MLP são geralmente *não-lineares*.

Comparação com as Redes MLP



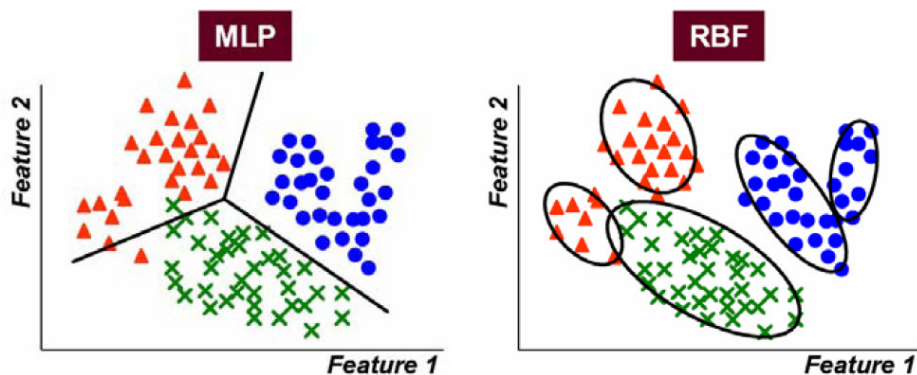
■ Funções de Ativação:

- O argumento da função de ativação de cada neurônio da camada escondida de uma rede RBF é a *distância Euclidiana* entre o vetor de entrada e o centro daquela unidade.
- O argumento da função de ativação de cada unidade escondida em uma MLP é o *produto interno* do vetor de entrada e o vetor com os pesos sinápticos daquela unidade.

■ Aproximações:

- A rede RBF ao usar funções Gaussianas constrói aproximações *locais* para o mapeamento não-linear entrada-saída.
- A rede MLP constrói aproximações *globais* para o mapeamento não-linear entrada-saída.

Comparação com as Redes MLP



Bibliografia básica



HASSOUN, M. H. (1995). *Fundamentals of Artificial Neural Networks*. Cambridge: The MIT Press.

Fonte



Site do Prof. **Aluizio Araújo** / CIN - UFPE:
<http://www.cin.ufpe.br/~aluizioa>

Site do Profa. **Elena Marchiori** / Vrije University:
<http://www.cs.ru.nl/~elenam/>

Site do Prof. **Arthur Braga** / DEE - UFC:
<http://www.dee.ufc.br/~arthurp>

Applet de simulação

<http://lcn.epfl.ch/tutorial/english/rbf/html/index.html>