

UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS - GRADUAÇÃO EM ENGENHARIA ELÉTRICA



EXERCÍCIO VII - REDES RBF

Matheus Barros Oliveira

20 de novembro de 2024

Relatório - Redes RBF

Relatório sobre RBF

20 de novembro de 2024

1 Introdução

As Redes Neurais Baseadas em Funções Radiais (RBF) são ferramentas versáteis e eficazes para problemas de classificação, regressão e aproximação de funções, devido à sua capacidade de modelar relações não lineares nos dados. Com uma estrutura de três camadas, a RBF transforma os dados de entrada em um espaço de características por meio de funções radiais, como a gaussiana, cuja parametrização depende de centros definidos frequentemente por algoritmos de k -médias. Neste trabalho, implementamos e avaliamos uma Rede Neural RBF em dois cenários distintos: a classificação de dados simulados e a aproximação da função $\text{sinc}(x)$ acrescida de ruído gaussiano, com foco na seleção de hiperparâmetros como o número de centros (p) e o spread (r). De acordo com [BISHOP, 1995], redes RBF podem ser vistas como extensões não lineares de modelos lineares, o que as torna ferramentas amplamente aplicáveis no aprendizado supervisionado.

2 Objetivo

O objetivo deste trabalho é implementar e avaliar uma Rede Neural Baseada em Funções Radiais (RBF) em dois cenários distintos: a classificação de dados simulados com distribuições não lineares e a aproximação da função $\text{sinc}(x)$ acrescida de ruído gaussiano. Busca-se explorar o impacto da seleção de hiperparâmetros, como o número de centros (p) e o spread (r), na capacidade da rede de modelar padrões complexos e de generalizar para novos dados, utilizando métricas de desempenho adequadas para classificação e regressão.

3 Materiais e Métodos

Para a execução deste exercício foram utilizados as seguintes ferramentas e base de dados descritas abaixo.

3.1 Materiais

Utilizou-se o RStudio (versão 4.4.1 (2024-06-14 ucrt) instalada em um Windows 11 (versão 23H2) com um processador 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz e 32GB de memória RAM.

3.2 Descrição da Base de Dados

Foram utilizadas as seguintes bases de dados no estudo:

- **mlbench.2dnormals(200)**: Conjunto de dados com duas classes normais bidimensionais, gerado com 200 pontos.
- **mlbench.xor(100)**: Problema clássico de separação não linear no formato "XOR", contendo 100 pontos.
- **mlbench.circle(100)**: Conjunto de dados com dois círculos concêntricos, gerado com 100 pontos.
- **mlbench.spirals(100, sd = 0.05)**: Duas espirais entrelaçadas, geradas com 100 pontos e desvio padrão de 0.05.

3.3 Metodologia

Segue abaixo os passos executados durante o experimento:

3.3.1 Passo 1: Carregamento da Base de Dados

Iniciamente realizou-se o carregamento do conjunto de dados para o ambiente R.

3.3.2 Passo 2: Separação do Conjunto de Testes e Treino

O conjunto de dados foi separado em 70% (conjunto de treino) e 30% (conjunto de validação) de forma aleatória.

3.3.3 Passo 3: Conversão das Classes

Quando necessário, foi realizado a conversão das classes de predição das bases de dados para as classes -1 e 1.

3.3.4 Passo 4: Visualização dos Dados

Com os dados carregados e tratados, foi necessário realizar a visualização dos conjuntos de treino e validação para se observar a distribuição dos mesmos.

3.3.5 Passo 5: Treinamento e Validação da Rede RBF

Após a avaliação dos conjuntos de treino e teste, foi realizado o treinamento da rede RBF. Para isso manteve-se fixo o parâmetro r em 10 e o parâmetro p foi variado entre os valores: 5,10,15 e 20.

3.3.6 Passo 7: Visualização de Superfície de Separação e Gráfico de Aproximação da Função

No caso dos problemas de classificação para cada modelo da rede RBF foi gerado um gráfico da superfície de separação do modelo e também analisado a acurácia do modelo. Como o valor de \hat{y}_i é um valor real, foi necessário convertê-lo para -1 e 1 para a comparação com os dados de treinamento.

No caso dos teste de aproximação da função $\text{sinc}(x)$ foram gerados os gráficos para se verificar o quanto a curva geada pelo modelo se aproximava da curva em análise. Além disso foi utilizada a função abaixo para o cálculo do erro quadrático médio:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_{\text{validação},i} - \hat{y}_i)^2 \quad (1)$$

4 Resultados

Pode-se observar abaixo os resultados obtidos para os experimento:

4.1 Base de Dados: mlbench.2dnormals(200)

Pode-se observar na Figura 1 o conjunto de treinamento e o conjunto de validação da base de dados mlbench.2dnormals(200).

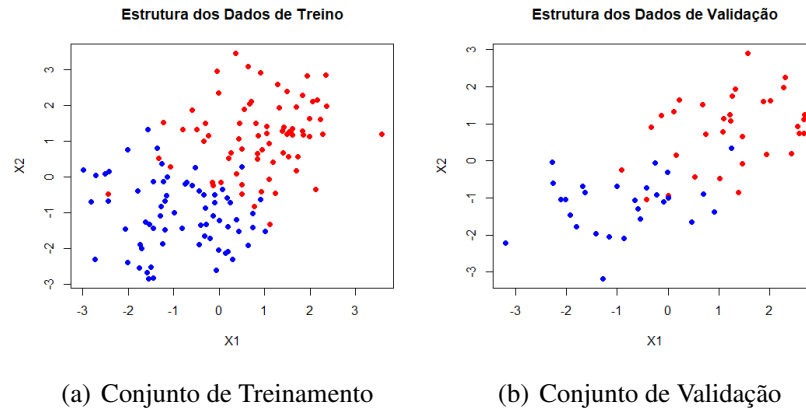


Figura 1: Conjunto de Treinamento e Validação do Conjunto de Dados: `ml-bench.2dnormals(200)`

Pode-se observar que a base de dados em análise é linearmente separável. Obversa-se na Figura 2 a superfície de separação calculada para a base de dados de validação.

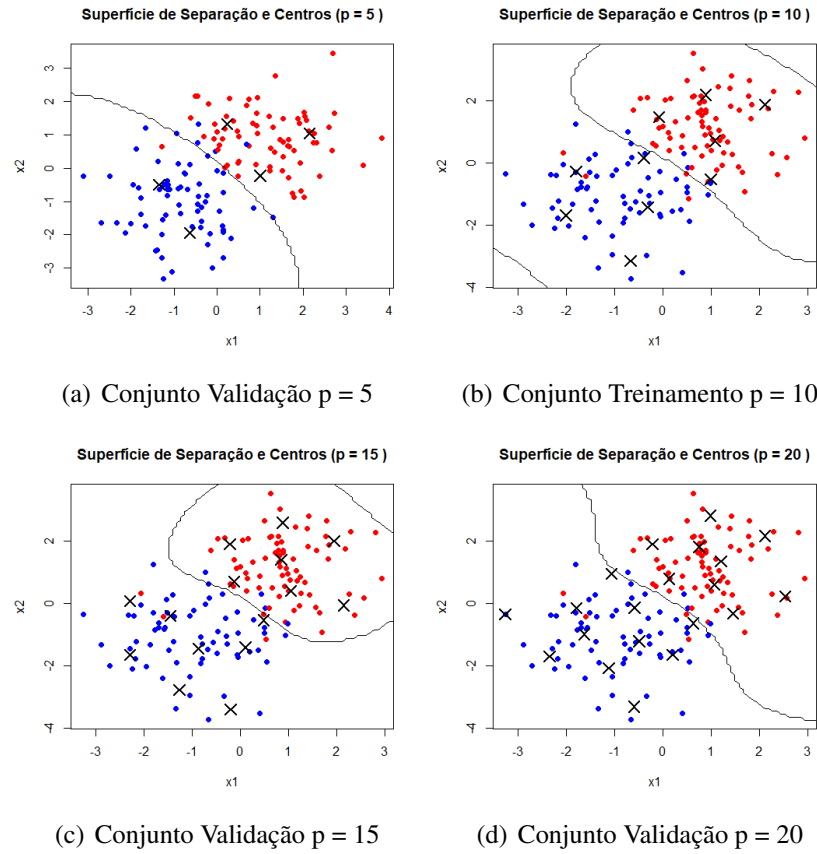


Figura 2: Gráficos das Superfícies de Separação RBF do Conjunto de Dados de Treinamento: `mlbench.2dnormals(200)`

Tabela 1: Acurácia do Modelo em função da variação de p

p	Acurácia (%)
5	93.33
10	98.33
15	95.00
20	96.67

4.2 Base de Dados: `mlbench.xor(100)`

Pode-se observar na Figura 3 o conjunto de treinamento e o conjunto de validação da base de dados `mlbench.xor(100)`.

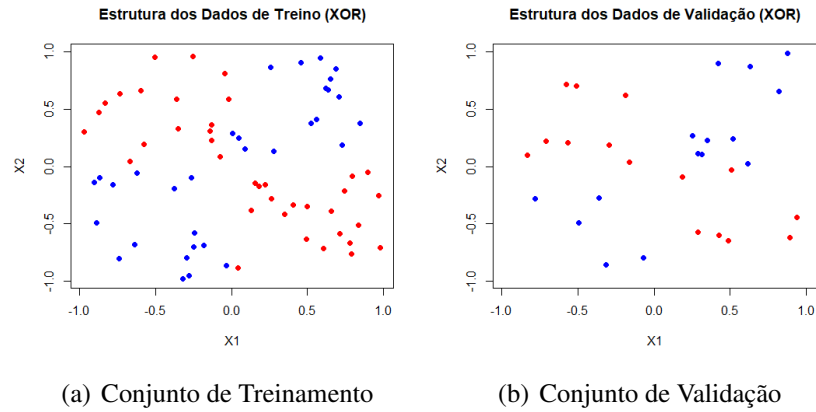


Figura 3: Conjunto de Treinamento e Validação do Conjunto de Dados: `mlbench.xor(100)`

Pode-se observar que a base de dados em análise não é linearmente separável. Observa-se na Figura 4 a superfície de separação calculada para a base de dados de validação.

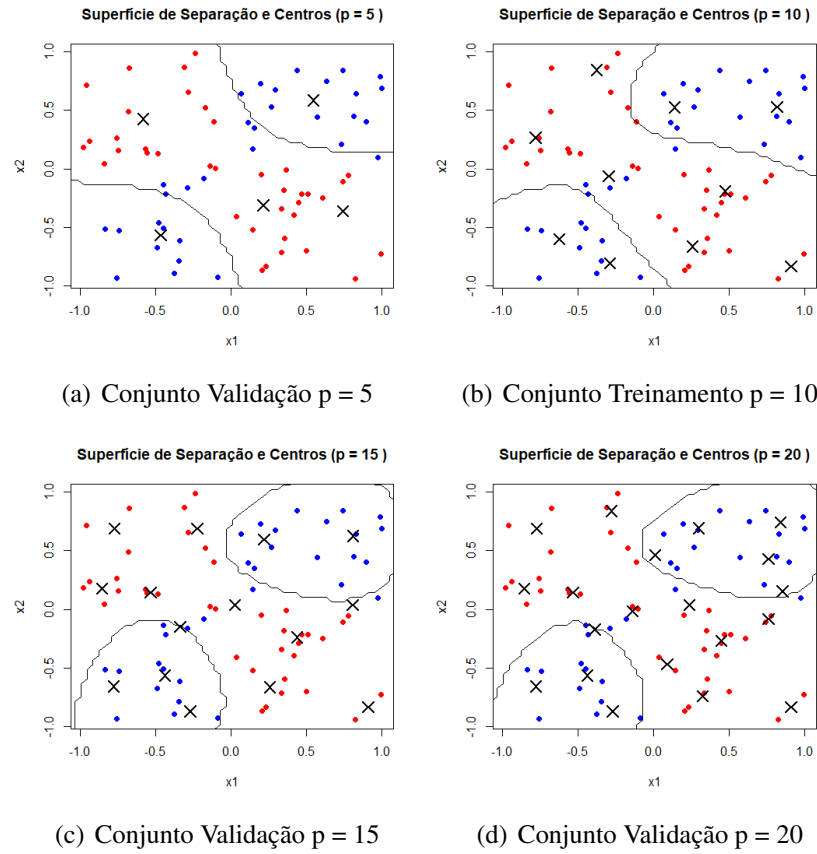


Figura 4: Gráficos das Superfícies de Separação RBF do Conjunto de Dados de Treinamento: `mlbench.xor(100)`

Tabela 2: Acurácia do Modelo em função da variação de p

p	Acurácia (%)
5	90.00
10	90.00
15	93.33
20	90.00

4.3 Base de Dados: `mlbench.circle(100)`

Pode-se observar na Figura 5 o conjunto de treinamento e o conjunto de validação da base de dados `mlbench.xor(100)`.

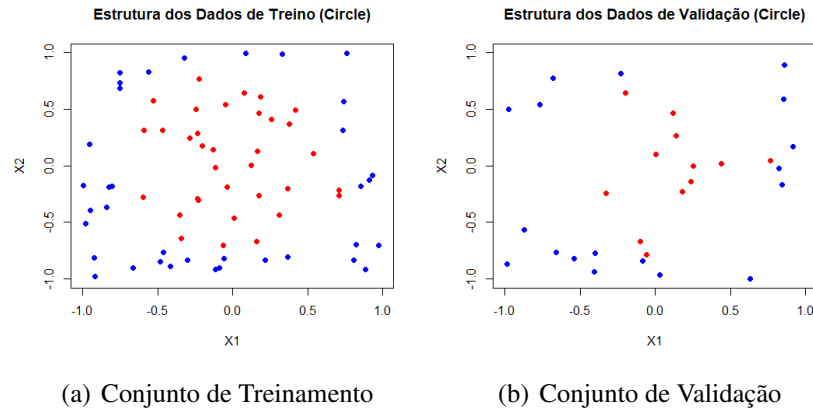


Figura 5: Conjunto de Treinamento e Validação do Conjunto de Dados: `mlbench.circle(100)`

Pode-se observar que a base de dados em análise não é linearmente separável. Obversa-se na Figura 6 a superfície de separação calculada para a base de dados de validação.

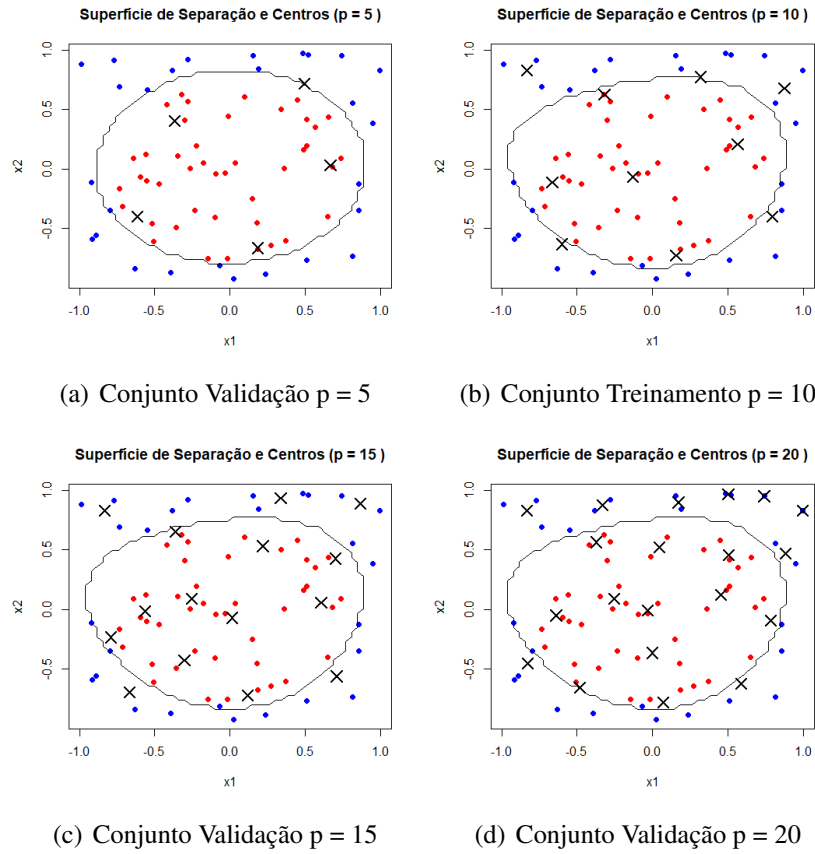


Figura 6: Gráficos das Superfícies de Separação RBF do Conjunto de Dados de Treinamento: `mlbench.circle(100)`

Tabela 3: Acurácia do Modelo em função da variação de p

p	Acurácia (%)
5	100
10	100
15	100
20	100

4.4 Base de Dados: `mlbench.spirals(100,sd = 0.05)`

Pode-se observar na Figura 7 o conjunto de treinamento e o conjunto de validação da base de dados `mlbench.xor(100)`.

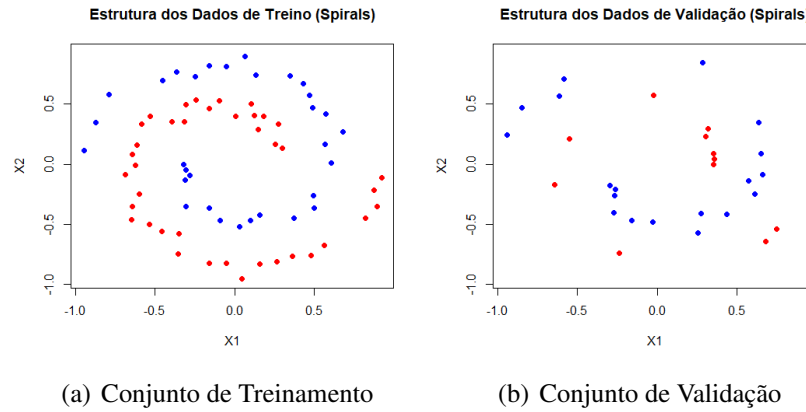


Figura 7: Conjunto de Treinamento e Validação do Conjunto de Dados: `mlbench.spirals(100,sd = 0.05)`

Pode-se observar que a base de dados em análise não é linearmente separável. Obversa-se na Figura 8 a superfície de separação calculada para a base de dados de validação.

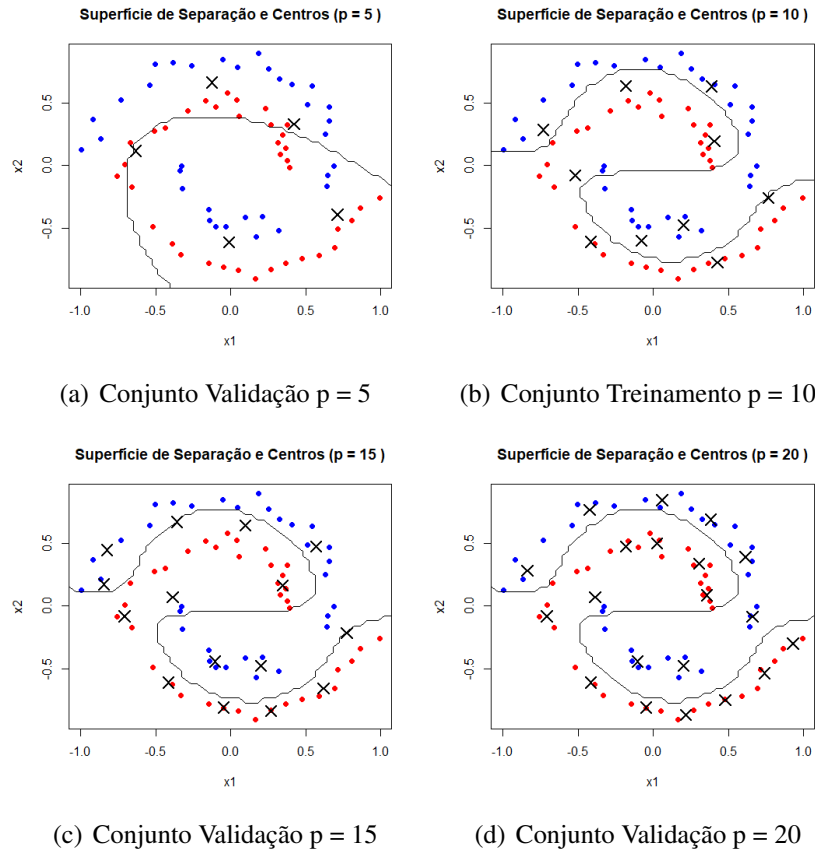


Figura 8: Gráficos das Superfícies de Separação RBF do Conjunto de Dados de Treinamento: `mlbench.spirals(100, sd = 0.05)`

Tabela 4: Acurácia do Modelo em função da variação de p

p	Acurácia (%)
5	36.67
10	100
15	100
20	100

4.5 Função de Aproximação: $\text{sinc}(x)$

Pode-se observar na Figura 9 a aproximação gerada da função RBF para a função $\text{sinc}(x)$ para os 100 pontos acrescidos com os ruídos gaussianos.

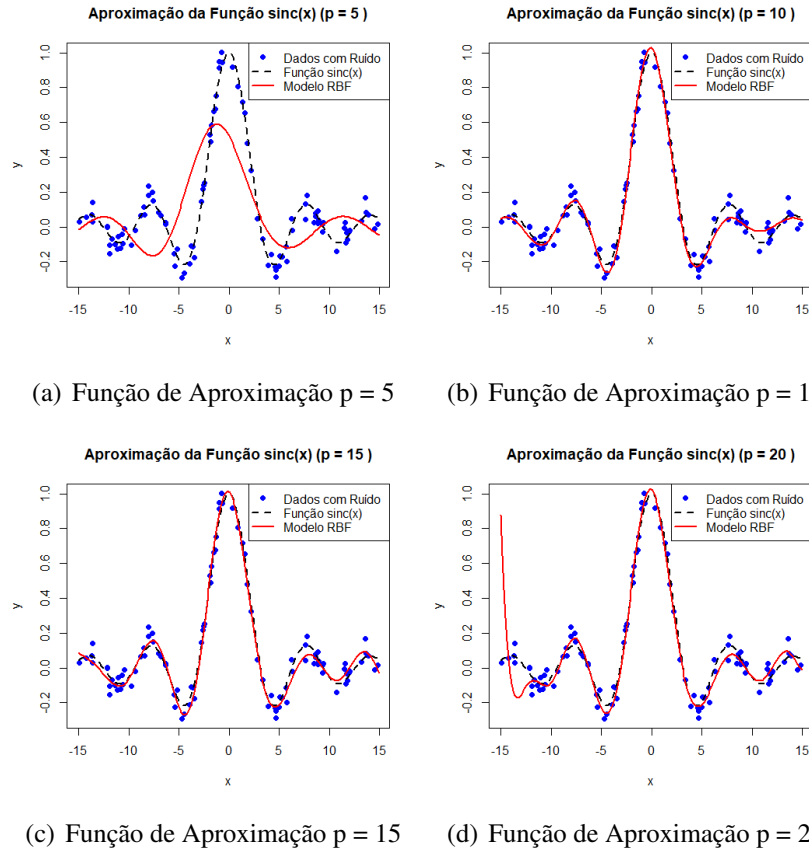


Figura 9: Gráficos das Superfícies da Função de Aproximação: $\text{sinc}(x)$

Pode-se observar na Figura 10 a aproximação gerada da função RBF para a função $\text{sinc}(x)$ para os 50 pontos acrescidos com os ruídos gaussianos.

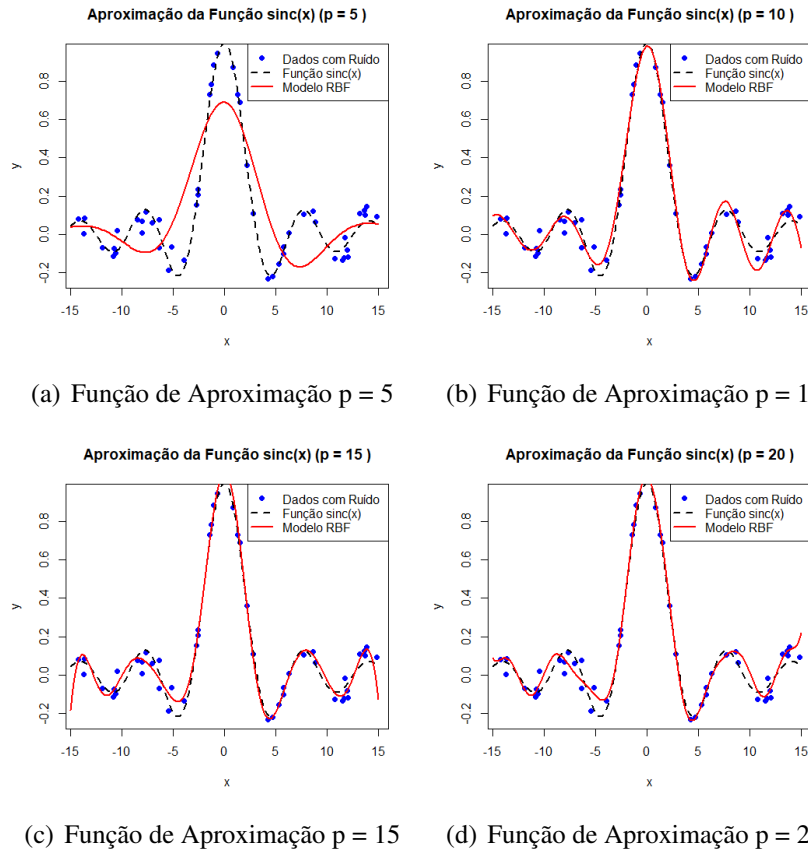


Figura 10: Gráficos das Superfícies da Função de Aproximação: $\text{sinc}(x)$

Tabela 5: Erro Médio Quadrático para cada variação de p

p	Número de amostras	Valor
5	100	0.0532
10	100	0.0031
15	100	0.0029
20	100	0.0304
5	50	0.0298
10	50	0.0037
15	50	0.0040
20	50	0.0036

4.6 Análise dos Resultados

Pode-se observar abaixo a análise dos resultados.

4.6.1 Modelo Classificador

Par a função RBF operando como separador percebe-se que a variação do fator p provoca uma mudança na superfície de separação do modelo. Quanto maior o valor de p obversa-se que mais *clusters* são gerados, a cruz no gráfico representa o centro do *cluster*, e a superfície de separação tende a seguir a dispersão dos *cluster* gerados.

A variação do fator p tende a melhorar a acurácia do modelo conforme pode-se obsvar nas tabelas. Para as bases de dados `mlbench.circle(100)` e `mlbench.spirals(100,sd = 0.05)` o modelo RBF apresentou um ótimo resultado, chegando até a 100% de acurácia. Já para as bases de dados `mlbench.2dnormals(200)` e `mlbench.xor(100)` o modelo também apresentou um bom desemepnho, mas não chegou a 100% de acurácia.

4.6.2 Modelo Aproximação da Função $\text{sinc}(x)$

Ao se analisar o gráfico e o valor do erro quadrado médio das funções de aproximação calculadas pela RBF também percebe-se que a alteração do valor p faz com que o modelo se aproxime cada vez mais dos dados e também provoque uma tendência de redução do erro médio quadrado.

Nos duas aproximações, tanto com 100 e 50 dados com ruídos gaussianos, o modelo apresentou praticamente o mesmo resultado, o que demonstra que o modelo também esta conseguindo generalizar bem com menos resultados.

5 Conclusão

Neste trabalho, foi analisado o impacto da variação do número de centros (p) e do número de amostras na performance de Redes Neurais Baseadas em Funções Radiais (RBF) aplicadas a problemas de classificação e regressão. Observou-se que, para bases de dados mais simples, como `mlbench.2dnormals`, o modelo apresentou alta acurácia independentemente de p , enquanto em bases mais complexas, como `mlbench.spirals`, o aumento de p foi essencial para alcançar um bom desempenho. Além disso, no problema de aproximação da função $\text{sinc}(x)$, foi constatado que não houve diferença significativa entre os

resultados obtidos com 50 e 100 amostras, o que sugere que o modelo RBF é robusto a variações no tamanho do conjunto de dados quando os exemplos são representativos.

Os resultados reforçam a capacidade das redes RBF de modelar padrões não lineares e aproximar funções em diferentes contextos. A utilização do algoritmo k -médias para determinar os centros das funções radiais desempenhou um papel crucial na formação dos clusters, permitindo que a superfície de separação do modelo acompanhasse a dispersão dos dados. Contudo, a escolha de p deve ser feita com cautela, já que valores muito altos podem levar a um ajuste excessivo aos dados de treinamento (overfitting), especialmente em bases mais simples. Como trabalho futuro, seria interessante explorar métodos automáticos para a escolha de p e do spread (r), além de investigar a influência de diferentes distribuições de ruído nos dados.

Referências

[BISHOP, 1995] BISHOP, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.