



UNIVERSIDADE FEDERAL DO CEARÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

RELATÓRIO  
DISCIPLINA: INTELIGÊNCIA COMPUTACIONAL APLICADA

---

**RADIAL BASIS FUNCTION E MULTILAYER PERCEPTRON  
APLICADOS AO JOGO DE POKER**

---

SOBRAL  
2021

# SUMÁRIO

	Página
<b>1 INTRODUÇÃO</b>	<b>2</b>
1.1 MULTILAYER PERCEPTRON (MLP) . . . . .	2
1.2 RADIAL BASIS FUNCTION (RBF) . . . . .	3
<b>2 BANCO DE DADOS UTILIZADOS</b>	<b>5</b>
<b>3 RESULTADOS</b>	<b>6</b>
3.1 OUTROS TESTES . . . . .	7
<b>4 CÓDIGO UTILIZADO</b>	<b>9</b>
4.1 CÓDIGO DO MLP . . . . .	9
4.2 CÓDIGO DO RBF . . . . .	11
<b>5 CONCLUSÃO</b>	<b>12</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>13</b>

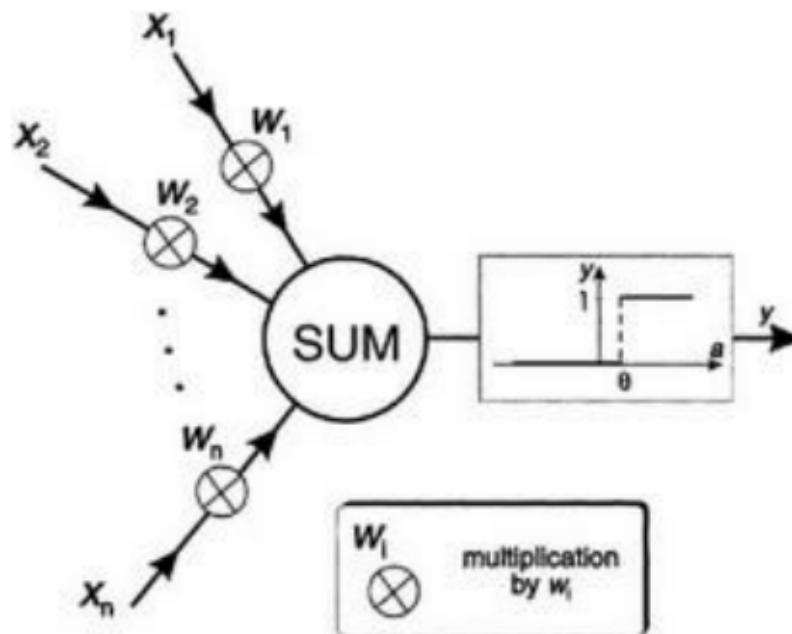
# 1 INTRODUÇÃO

---

## 1.1 Multilayer Perceptron (MLP)

No que se refere ao estudo de Reconhecimento de padrões, as RNA (Redes Neurais Artificiais) são um conjunto de técnicas de classificação que se deram com o surgimento da proposta do neurônio artificial em 1943.<sup>1</sup> O neurônio artificial se espelha em um neurônio real, algumas de suas características podem ser comparadas ao que acontece de forma biológica em um neurônio humano. Um exemplo desta relação são as sinapses neuronais (sinais eletroquímicos recebidos pelo neurônio) que, nas RNA, são as amostras de entrada a serem classificadas ou usadas para treino.<sup>2</sup> Tal simulação de neurônio é ilustrada na Figura 1.1.

Figura 1.1: Neurônio artificial ou Perceptron



Fonte: (GURNEY, 1997)

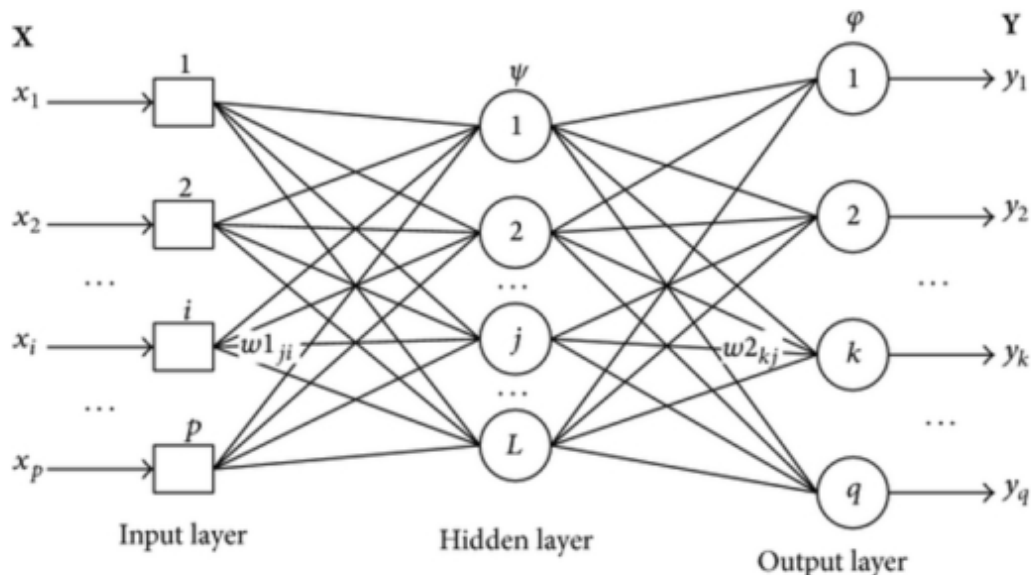
Nas MLP, algoritmo de cálculo dos pesos de cada atributo de entrada possibilita o uso da técnica *backpropagation* que permite regular mais rapidamente os pesos de cada

atributo de acordo com o erro informado pela função de ativação. Essa técnica revolucionou e reativou o interesse no uso de RNAs, removendo a impossibilitação de tempo que antes haveria deixado esta tecnologia de lado por parte da comunidade científica.

O *Bias* ( $w_0$ ) é um peso extra que permite um *offset* na amostra, deslocando a superfície de decisão e melhorando a classificação.<sup>3</sup> Por fim, existe a função de ativação que aplica uma operação não linear na saída do sistema. Existem funções de ativação linear, sigmoide, quadrada, entre outras. A Figura 1.1 mostra uma função de ativação quadrada entretanto esse tipo de função de ativação não é utilizada. Preferencialmente são utilizadas funções de ativação que podem gerar valores não inteiros entre 0 e 1, permitindo assim uma análise mais específica do erro por parte da rede.

Esse tipo de esquema mostrado na Figura 1.1 é chamado de Perceptron, um classificador simples de apenas uma camada. Para problemas de maior complexidade, existem as Perceptron de Multi-Camadas ou MLP, que possuem camadas ocultas de neurônios (entre a camada de entrada e saída). Esse emaranhado de neurônios aumenta a complexidade do classificador, permitindo capacidade maior de percepção de erros, entretanto, reduzindo a velocidade de processamento conforme o aumento de camadas ocultas. A Figura 1.3 mostra como a camada oculta interage com as duas camadas principais (saída e entrada).

Figura 1.2: Perceptron de multi-camadas.



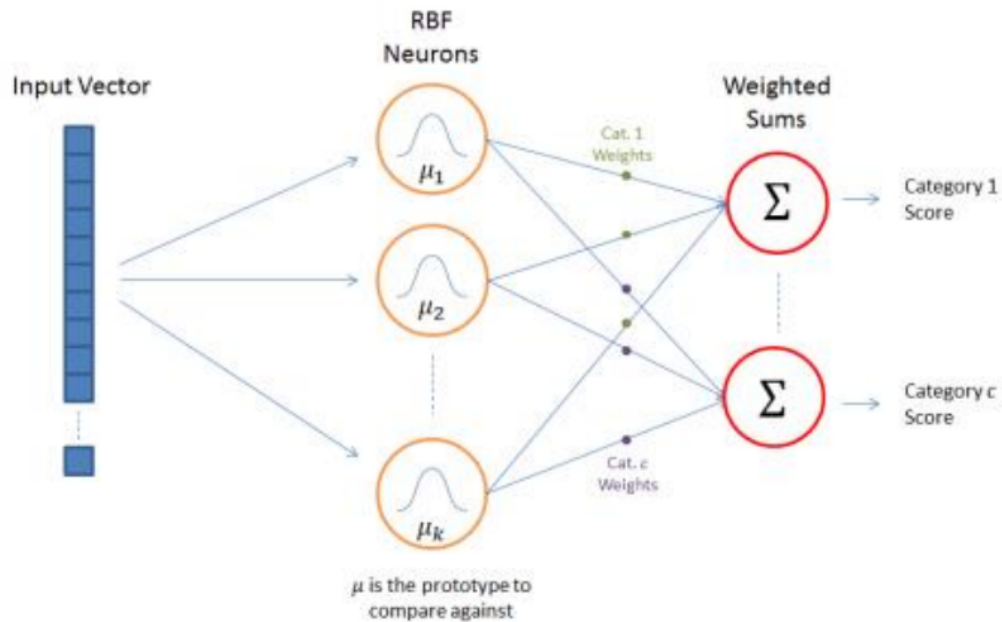
Fonte: [beatthebookie2017.files.wordpress.com](http://beatthebookie2017.files.wordpress.com)

## 1.2 Radial Basis Function (RBF)

As RBF são RNAs cuja função de ativação da rede se dá por meio de uma função de base radial. Primeiramente descritas em 1988 por um artigo de Broomhead e Lowe,

são redes neurais baseada em funções radiais de ativação. Tipicamente construída por 3 camadas: Uma camada de entrada, uma camada escondida que utiliza funções não lineares de base radial e uma camada de saída linear. Seus principais usos são: Aproximação de funções, previsões em series temporais, controle de sistemas e classificação.<sup>4</sup>

Figura 1.3: Rede RBF.



Fonte: en.wikipedia.com

## 2 BANCO DE DADOS UTILIZADOS

---

O banco de dados utilizado foi retirado do site UCI,<sup>5</sup> segundo autor o banco de dados possui as seguintes características: Cada amostra é um exemplo de uma mão (de baralho) que consiste em cinco cartas de jogar tiradas de um baralho padrão de 52. Cada carta é descrita usando dois atributos (naipe e valor), para um total de 10 atributos. Os rótulos são chamados de “Mão de Poker”. A ordem das cartas é importante, é por isso que existem 480 mãos *Royal Flush* possíveis em comparação com 4.

O autor disponibiliza dois arquivos, um contendo uma base para treino com 1 milhão de amostras e uma segunda base para treino, contendo 25010 amostras. Ou seja, são disponibilizadas matrizes  $1000000 \times 11$  e  $25010 \times 11$ , onde as 10 primeiras colunas referem-se aos atributos e a coluna 11 aos rótulos.

### 3 RESULTADOS

---

Os resultados dos testes foram divididos em duas configurações Na Configuração 1, para o MLP, utilizou-se 25010 amostras para treino e todas as 1000000 amostras para teste, para o RBF foram utilizadas 25010 amostras de teste as 10000 primeiras amostras de treino, por limitações do MATLAB. Já na Configuração 2 foram utilizadas no máximo 230 amostras das classes mais numerosas para treino (classes 0 a 6) e foram utilizados todas as 25010 amostras para teste.

Os resultados da Configuração 1 são mostrados nas Tabelas 3.1 e Tabela 3.2. Já os resultados para a Configuração 2 são mostrados na Tabela 3.3 e na Tabela 3.4.

Tabela 3.1: Resultados para a Configuração 1 do MLP.

Rede de camadas	Épocas	MSE	Assertividade(%)
32 32	10	0.477	58.61
32 32	20	0.253	79.35
32 32	50	0.0908	99.28
32 32	100	0.0694	98.92

Tabela 3.2: Melhor caso para a Configuração 1 do RBF.

MSE	<i>Spread</i>	Épocas	Neurônios por época	Assertividade(%)
0.3502	1	300	20	52.99
0.2987	1	500	10	51.04
0.2186	2	1000	10	53.70

Tabela 3.3: Resultados para a Configuração 2 do MLP.

Rede de camadas	Épocas	MSE	Assertividade(%)
32 32	10	5.66	1.44
32 32	20	5.19	1.33
32 32	50	5.19	1.33
32 32	100	5.09	1.33

Tabela 3.4: Melhor caso para a Configuração 2 do RBF.

MSE	<i>Spread</i>	Épocas	Neurônios por época	Assertividade(%)
2.536	2	300	20	3.41
2.079	1	500	10	4.81
0.2775	2	1000	10	35.61

A quantidade de amostras de cada classe para a base de dados de treino é mostrada nas Figuras 3.5 e 3.6 para a Configuração 1 e Configuração 2 respectivamente.

Tabela 3.5: Quantidade de classes para a Configuração 1.

Classe	Quantidade
0	501209
1	422498
2	47622
3	21121
4	3885
5	1996
6	1424
7	230
8	12
9	3
<b>TOTAL:</b>	1000000

Tabela 3.6: Quantidade de classes para a Configuração 2.

Classe	Quantidade
0	230
1	230
2	230
3	230
4	230
5	230
6	230
7	230
8	12
9	3
<b>TOTAL:</b>	1855

### 3.1 Outros testes

Os resultados seguintes são referentes a um teste complementar utilizando aproximadamente 100 mil amostras de treino para a MLP.



Tabela 3.7: Quantidade de classes para o teste extra.

Classe	Quantidade
0	14250
1	14250
2	14250
3	14250
4	14250
5	14250
6	14250
7	230
8	12
9	3
<b>TOTAL:</b>	99995

Tabela 3.8: Resultados para a Configuração 2 do MLP.

Rede de camadas	Épocas	MSE	Assertividade(%)
32 32	50	5.19	2.44
32 32	500	5.09	2.48

## 4 CÓDIGO UTILIZADO

---

Foram utilizados dois códigos distintos, em MATLAB, para o RBF e MLP, cada código possui o papel de treinar e testar em seguida, apresentando o resultado em forma de decimal.

### 4.1 Código do MLP

```
%% Cleaning
clc
clear all
close all

%% Loading data
rng(1)
Y1 = load('poker-hand-training-true.data'); %carrega o teste
Y2 = load('poker-hand-testing.data'); % carrega o treino
% Y2 = load('treino.mat'); %nova base de dados
% Y2=Y2.Y2; %nova base de dados
rotulosY1 = Y1(:,11);
rotulosY2 = Y2(:,11);
Y1(:,11) = [];
Y2(:,11) = [];

%% MLP
net = feedforwardnet([32, 32]);
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'logsig';
net.layers{3}.transferFcn = 'logsig';
net.trainParam.epochs = 100 ;
net.trainParam.max_fail =100;
net = train(net,Y2',rotulosY2');
y = net(Y1'); %matriz de teste
y = round(y);
Acc = length(find(y' == rotulosY1))/25010
```



## 4.2 Código do RBF

```
%% Cleaning
clc
clear all
close all
rng(1)

%% Loading data
Y1 = load('poker-hand-training-true.data'); %teste
%Y2 = load('poker-hand-testing.data'); %treino
Y2 = load('treino.mat'); %nova base de dados
Y2=Y2.Y2; %nova base de dados
rotulosY1 = Y1(:,11);
rotulosY2 = Y2(:,11);
Y1(:,11) = [];
Y2(:,11) = [];

%% RBF
net = newrb(Y2', rotulosY2', 0.01, 1, 500, 10);
y = sim(net,Y1'); %matriz de teste
y = round(y);
Acc = length(find(y' == rotulosY1))/25010
```

## 5 CONCLUSÃO

---

Pode-se então observar, pela análise dos dados obtidos, que para o problema de classificação em questão na configuração 1 o pior resultado da MLP atingiu assertividade de 58.61% enquanto a RBF atingiu, n melhor caso, 53.70% uma diferença de 4.91% a menos, e no melhor caso da MLP com 98.92% essa diferença sobe para 45.22%. Já na segunda classificação a vantagem vai para a RBF onde sua configuração atingiu assertividade de 35% enquanto a MLP no melhor caso somente atingiu 1.44% uma diferença de 33.56%. Com esses dados podemos concluir que as redes MLP possuem uma melhor assertividade em relação as redes RBF com uma maior quantidade de amostras de treino, logo rede MLP com a configuração 1, treino com 25010, com 98.92% seria a melhor escolha para a resolução do problema.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- [1] MCCULLOCH W.S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, v. 5, p. 115–133, 1943.
- [2] GURNEY, K. **An introduction to neural networks**. [S.l.]: UCL Press, 1997.
- [3] BACKES A. R.; JUNIOR, J. J. d. M. S..**Introdução à visão computacional usando Matlab**. [S.l.]: Alta Books Editora, 2019.
- [4] WIKIPEDIA. **Radial basis function**. Disponível em: [https://en.wikipedia.org/wiki/Radial\\_basis\\_function](https://en.wikipedia.org/wiki/Radial_basis_function). Acesso em: 8 de fevereiro de 2021.
- [5] UCI: Machine Learning Repository. **Poker Hand Data Set**. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Poker+Hand>. Acesso em: 24 de fevereiro de 2021.