


Redes Neurais e Algoritmos Genéticos na Análise de Nódulos em exames de Mamografia



Gabriel Nobrega de Lima
Andrea Laterza Wingerter

Tópicos

1

Conhecendo o Problema

2

Ferramentas

3

Rede Neural

4

Taxa de Aprendizado

5

Algoritmos Genéticos

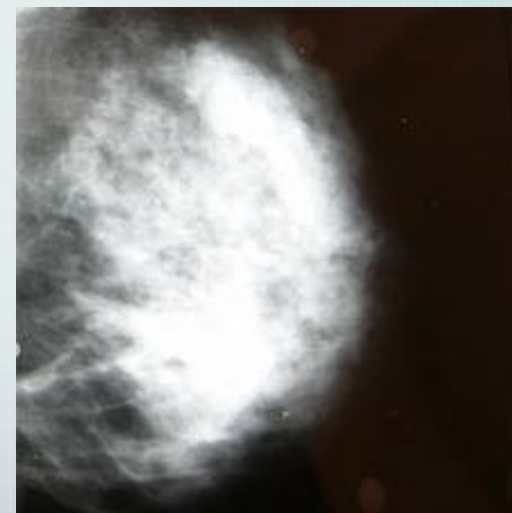
6

Projeto Implementado

Conhecendo o Problema

Mamografia:

- É o método mais utilizado para verificar a presença de nódulos nos seios;
- Benefícios da mamografia quanto a uma descoberta precoce e tratamento do câncer mamário são muito significativos, sendo maiores do que o risco da radiação e o desconforto durante o exame;
- 70% desses exames resultam em nódulos benignos.



Conhecendo o Problema

Biópsia mamária:

- Compreende a remoção de uma amostra do tecido da mama para avaliação laboratorial;
- Utilizada para determinar se uma amostra é maligna ou benigna.

Existe alguma forma de prever o tipo de nódulo através da análise das imagens da mamografia?

Para reduzir o número de intervenções cirúrgicas o uso de **redes neurais contendo uma base de dados** baseada em informações relativas a este exame pode ser aplicada no auxílio de classificação de nódulos encontrados.

Base de Dados

A base de Dados foi retirada do site UCI Machine Learning Repository contendo os resultados da biopsia de 961 pessoas. Pode ser acessada em:

<http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass>

Descrição da Base de Dados:

- 1) Bi-rads (Breast Imaging-Reporting and Data System): Controle quantitativo fornecida pelo médico em relação ao nódulo variando de 1 (benigno) a 5 (maligno);
- 2) Forma: Formato do nódulo;
- 3) Margem: Caracteriza o tipo de margem identificada na massa lesada;
- 4) Densidade: Caracteriza a densidade observada no nódulo;
- 5) Idade do paciente;
- 6) Severidade: Classifica o nódulo como maligno ou benigno;

Classificação:

- Benigno ou maligno;

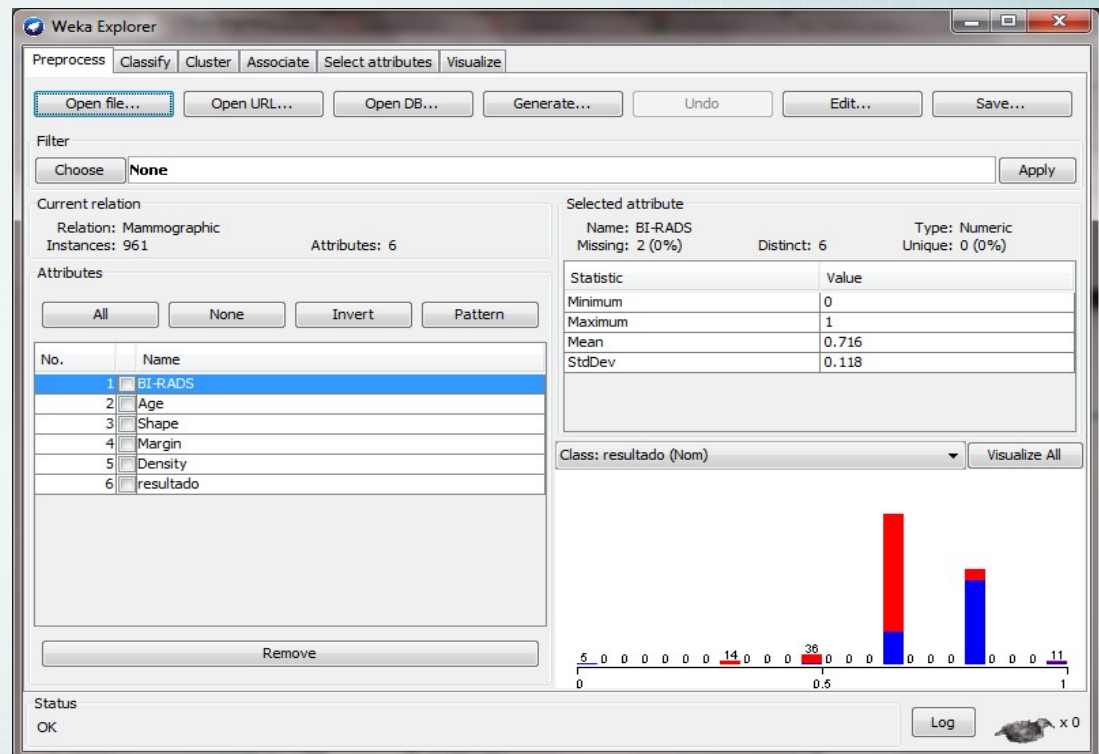
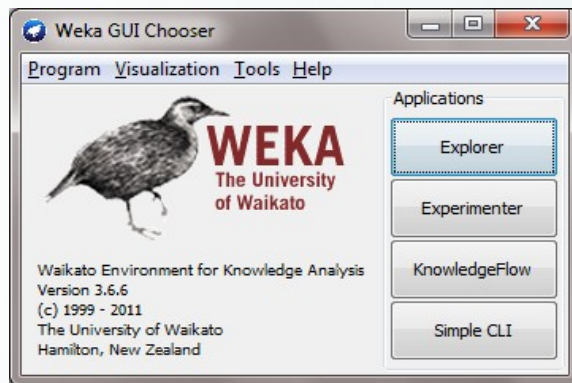
Ferramentas

Para a realização do projeto utilizamos os conceitos de redes Neurais e Algoritmos Genéticos, o Software WEKA (escrito em JAVA) e codificação na linguagem JAVA para classificação das redes e ajustes de taxas de aprendizado das redes neurais.

Weka 3.4.11 Data Mining

- Escrito em Java
 - Universidade de Waikato, Nova Zelândia em 1993
- Métodos de avaliação
 - Comparações entre resultados
 - Tempo de execução

Software Weka



Biblioteca Weka - Exemplo de Implementação de um MLP

```

FileReader reader = new FileReader("Mammographic.arff");
Instances conteudo = new Instances(reader);
MultilayerPerceptron mlp = new MultilayerPerceptron();

conteudo.setClassIndex(conteudo.numAttributes() );
mlp.setHiddenLayers("1"); // numero de camadas escondidas
mlp.setMomentum(0.200); // Momentum
mlp.setLearningRate(alpha); // Taxa de aprendizado
mlp.setTrainingTime(500); // tempo limite de treinamento
mlp.buildClassifier(conteudo); // treina
Evaluation eval = new Evaluation(conteudo);
eval.crossValidateModel(mlp, conteudo, 10,
    conteudo.getRandomNumberGenerator(1));
    
```

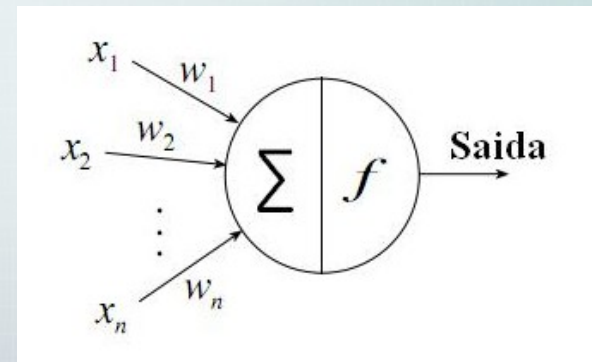

Neurônio Artificial

Modelo MCP (modelo de McCulloch e Pitts)

As entradas do neurônio correspondem ao vetor de entrada $X = [x_1, x_2, \dots, x_n]$ de dimensão n . Para cada uma das entradas x_i , há um peso correspondente w_i na entrada dos neurônios.

A soma das entradas x_i ponderada pelos pesos correspondentes w_i é chamada de saída linear u , onde $u = \text{Somatório}(w_i \cdot x_i)$. A saída do neurônio, chamada de saída de ativação, é obtida pela aplicação de uma função $f(\cdot)$ à saída linear, indicada por $y = f(u)$.

$f(\cdot)$ é chamada de função de ativação, e pode assumir várias formas, geralmente não lineares.



Redes Neurais analisadas

Perceptron

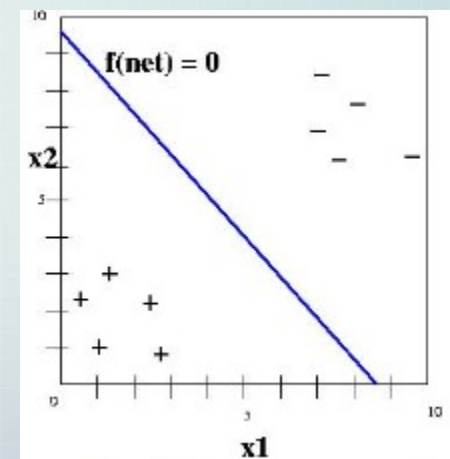
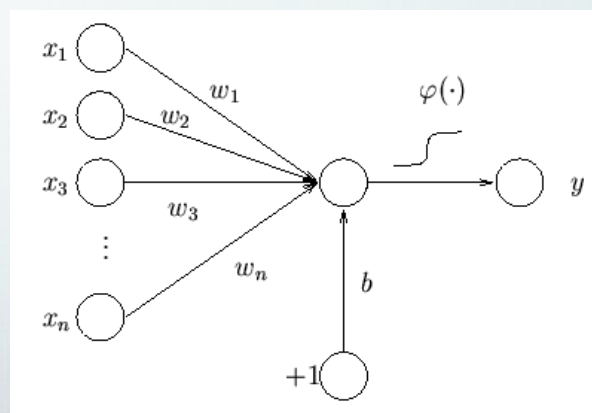
Criado no final da década de 50 por Rosenblatt na Universidade de Cornell

Primeira rede neural inventada.

Rede simples utilizada para classificar dados linearmente separáveis através de uma camada simples de neurônios (Cria um hiperplano de separação correspondendo a verdadeiro ou falso)

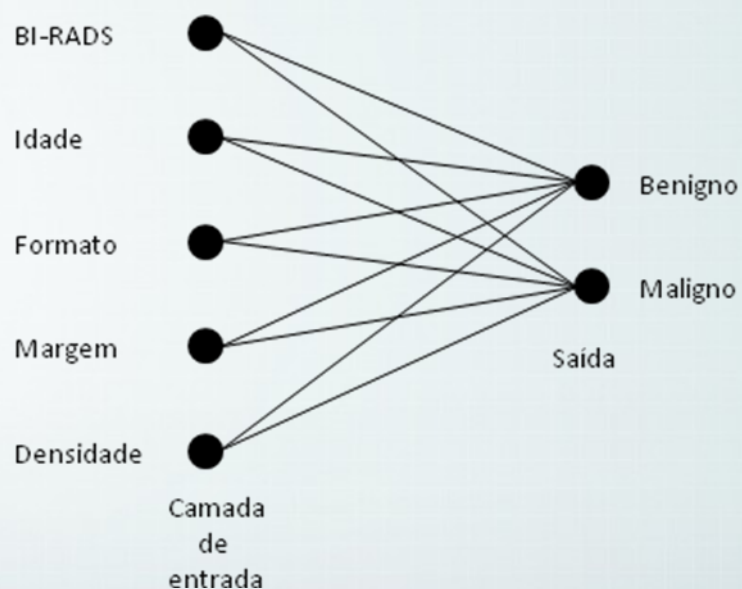
Função de ativação linear, ou threshold.

$$f(x) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases}$$



Redes Neurais analisadas

Perceptron

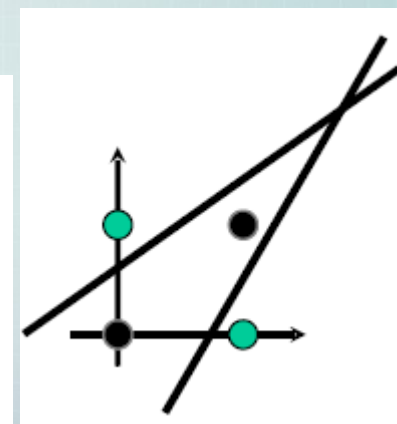
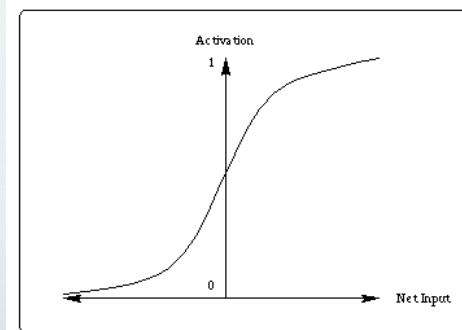
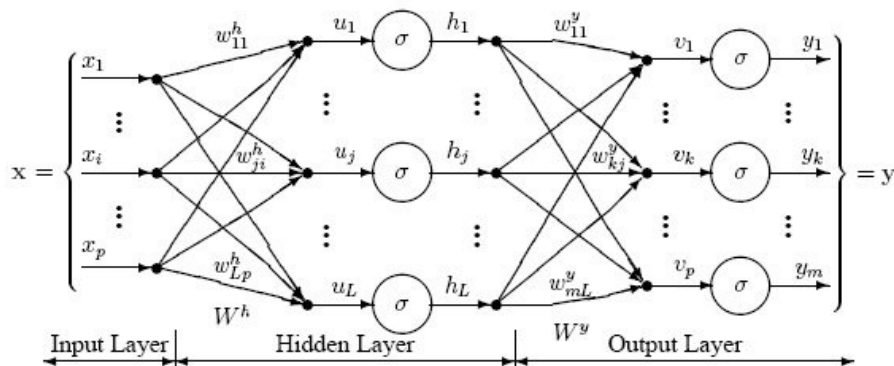


Redes Neurais analisadas

Multi Layer Perceptron

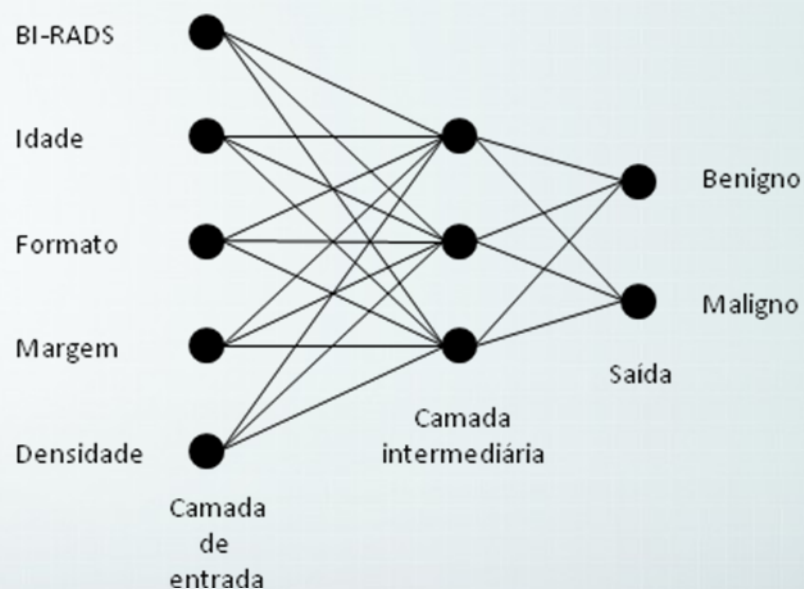
Perceptron de várias camadas, utiliza estas camadas para transformar problemas linearmente não separáveis em problemas linearmente separáveis.

As camadas intermediárias simplificam o problema para a camada de saída, utiliza a função sigmoid (unipolar ou bipolar) para ativação dos neurônios.



Redes Neurais analisadas

Multi Layer Perceptron



Pré-Processamento dos Dados

- o Pré-processamento
 - o Retirar atributos fora de intervalo
 - o Normalização dos dados

Atributo	Média	Desvio Padrão	Faltantes	Únicos
Bi-Rads	0,716	0,118	2 (0%)	0 (0%)
Idade	0,578	0,151	5 (1%)	6 (1%)
Formato	0,680	0,311	31 (3%)	0 (0%)
Margem	0,559	0,313	48 (5%)	0 (0%)
Densidade	0,728	0,095	76 (8%)	0 (0%)

Procedimentos de Testes

- Testes comparativos divididos em duas categorias
 - ✓ Métodos de treinamento
 - 10 Fold - Cross Validation
 - Divisão percentual
 - Treinamento: 70%
 - Teste: 30%
 - ✓ Modificação da taxa de aprendizado
 - Será modificada a taxa de aprendizado da rede para verificação de convergência (0,2; 0,5; 0,8)

Resultados

Resultados com Perceptron utilizando validação cruzada

Taxa de aprendizado	Classificações corretas (%)	Erro geral	Erro quadrático médio	Tempo de execução
0.2	78.564	0.1602	0.4003	0.69 s
0.5	77.0031	0.1613	0.4016	0.69 s
0.8	78.564	0.155	0.3938	0,53 s

Resultados com Perceptron com divisão percentual dos registros

Taxa de aprendizado	Classificações corretas (%)	Erro geral	Erro quadrático médio	Tempo de execução
0.2	75.7785	0.1626	0.4033	0,55 s
0.5	75.0865	0.1419	0.3767	0,55 s
0.8	80.2768	0.1488	0.3857	0,56 s

Resultados

Matrizes de Confusão - Perceptron

10 Fold Cross Validation			Divisão Percentual	
Taxa de Aprendizado = 0,2				
A	B	Classificado Como	A	B
319	101	A = Maligno	111	17
53	436	B = Benigno	30	108
Taxa de Aprendizado = 0,5				
A	B	Classificado Como	A	B
318	92	A = Maligno	106	17
63	422	B = Benigno	24	111
Taxa de Aprendizado = 0,8				
A	B	Classificado Como	A	B
313	100	A = Maligno	95	26
49	442	B = Benigno	17	137

Resultados

Resultados MLP utilizando validação cruzada

Taxa de aprendizado	Classificações corretas (%)	Erro geral	Erro quadrático médio	Tempo de execução
0.2	83.5588	0.1644	0.4055	1,69 s
0.5	83,6629	0,1634	0.4042	1,69 s
0.8	82,8304	0.1717	0.4144	2,02 s

Resultados MLP com divisão percentual dos registros

Taxa de aprendizado	Classificações corretas (%)	Erro geral	Erro quadrático médio	Tempo de execução
0.2	82,699	0,173	0.4159	1,83 s
0.5	80.969	0.1903	0.4362	1,72 s
0.8	80,969	0.1903	0.4362	1,84 s

Resultados

Matrizes de Confusão - MLP

10 Fold Cross Validation			Divisão Percentual	
Taxa de Aprendizado = 0,2				
A	B	Classificado Como	A	B
369	76	A = Maligno	104	26
82	434	B = Benigno	24	135
Taxa de Aprendizado = 0,5				
A	B	Classificado Como	A	B
373	72	A = Maligno	106	24
85	431	B = Benigno	31	128
Taxa de Aprendizado = 0,8				
A	B	Classificado Como	A	B
370	75	A = Maligno	106	24
90	426	B = Benigno	31	128

Discussão

Através da análise dos resultados é possível observar o baixo desempenho apresentado pela rede Perceptron em relação à MLP. Isto pode ser devido à sua incapacidade de resolver problemas que não sejam linearmente separáveis. Por outro lado, pode-se perceber que a rede MLP apresentou o melhor desempenho quando levada em consideração a taxa de acerto, o que pode ser obtido devido à sua natureza, por ser composta de várias redes do tipo Perceptron interligadas entre si, esta apresenta a capacidade de resolver problemas não linearmente separáveis.

Levando em consideração a porcentagem de classificações corretas é possível notar que a rede MLP apresentou melhor desempenho (83,6629%) quando comparada à rede Perceptron (80.2768%) .

Taxa de Aprendizado

Como encontrar o valor da taxa de aprendizado que permita rede neural alcançar o menor erro quadrático médio em seu treinamento?



Algoritmos Genéticos

A teoria da evolução diz que na natureza todos os indivíduos dentro de um ecossistema competem entre si por recursos limitados, tais como comida e água.

Aqueles dentre os indivíduos (animais, vegetais, insetos, etc) de uma mesma espécie que não obtêm êxito tendem a ter uma prole menor.

Esta descendência reduzida faz com que a probabilidade de ter seus genes propagados ao longo de sucessivas gerações seja menor.

A combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente ao combinar características possivelmente positivas de cada um dos reprodutores

Algoritmos Genéticos

Aqueles dentre os indivíduos (animais, vegetais, insetos, etc) de uma mesma espécie que não obtêm êxito tendem a ter uma prole menor.

Esta descendência reduzida faz com que a probabilidade de ter seus genes propagados ao longo de sucessivas gerações seja menor.

A combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente ao combinar características possivelmente positivas de cada um dos reprodutores

Algoritmos Genéticos

Codificação do Cromossomo

Cromossomo A	101100101100101011100101
Cromossomo B	111111100000110000011111

Algoritmos Genéticos

Algoritmo:

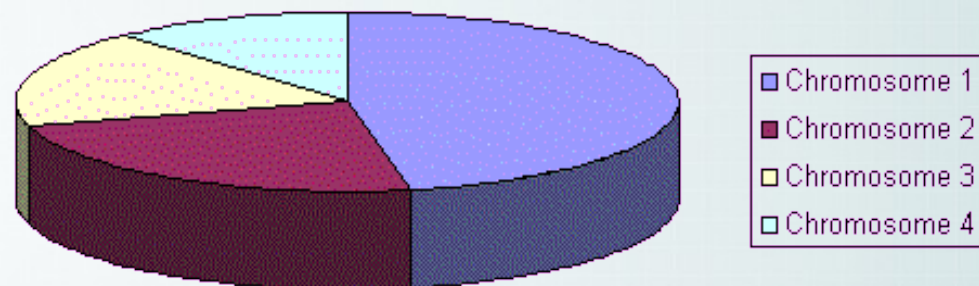
- 1) Gerar uma população inicial aleatória
- 2) Avaliar eficiência de cada cromossomo
- 3) Selecionar nova população
 - 3.1) Seleção
 - 3.2) Cruzamento
 - 3.3) Mutação

Os itens 3.1 a 3.3 são realizados até que a totalidade dos indivíduos da nova população sejam alcançados.

- 4) O número de gerações limite foi atingido? Se não, ir para 2) caso contrário Terminar.

Algoritmos Genéticos

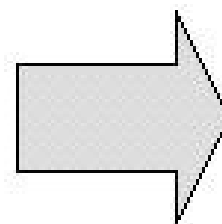
Seleção Por Roleta



Algoritmos Genéticos

Seleção Por Torneio

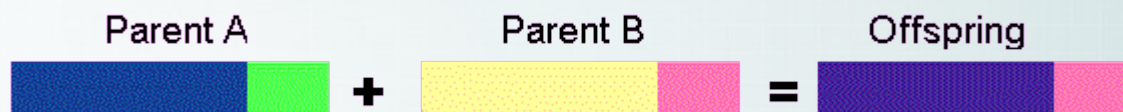
Indivíduo	Fitness
x ₁	200
x ₂	100
x ₃	9500
x ₄	100
x ₅	100
x ₆	10000
x ₇	1
x ₈	40



Torneios		
x ₁	x ₇	x ₈
x ₂	x ₃	x ₅
x ₆	x ₄	x ₄
x ₂	x ₇	x ₁
x ₅	x ₅	x ₅
x ₃	x ₄	x ₂
x ₄	x ₂	x ₆
x ₄	x ₆	x ₅

Algoritmos Genéticos

Cruzamento de Ponto Único



$$11001011 + 11011111 = 11001111$$

Algoritmos Genéticos

Mutação

Descendência Original

110**1**111000011110

Descendência Mutada

110**0**111000011110

Algoritmos Genéticos

Elitismo

Quando criamos uma nova população por cruzamento e mutação, nós temos uma grande chance de perder os melhores cromossomos.

O Elitismo copia primeiro os melhores cromossomos para a nova população. Isto previne a perda da melhor solução já encontrada otimizando o processo.

Projeto Implementado

Projeto Algoritmos Genéticos

Configuração

Seleção por:

☒ Roleta ☒ Torneio

☐ Utilizar Elitismo

Max iterações: 5

Tamanho pop.: 10

Iniciar

Resultados

Torneio (apt.)	Roleta (apt.)
0.6474669554386425	0.6443372723377248
0.6467569728150651	0.644272392728129
0.6467569728150651	0.6443372723377248
0.6467569728150651	0.6467210824672176
0.6467569728150651	0.6437354628666235
0.6467569728150651	0.6442955359151445
0.6363613825230954	0.6443142443072765

Executando Roleta: 100 %

Total: 100 %

Melhor resultado Torneio: 0.028 (Alfa)

Melhor resultado Roleta: 0.039 (Alfa)

Tempo de execução: 542 segundos

Tempo de execução: 539 segundos

Resetar

Base de dados: Mammographic.arff Log

Projeto Algoritmos Genéticos

Configuração

Seleção por:

☐ Roleta ☒ Torneio

☒ Utilizar Elitismo

Max iterações: 5

Tamanho pop.: 10

Iniciar

Resultados

Torneio (apt.)	Roleta (apt.)
0.64473389084591	
0.6429595078463296	
0.639753143061363	
0.639753143061363	
0.639753143061363	
0.6373990228011936	
0.6368536751465377	

Executando Torneio: 100 %

Total: 100 %

Melhor resultado Torneio: 0.146 (Alfa)

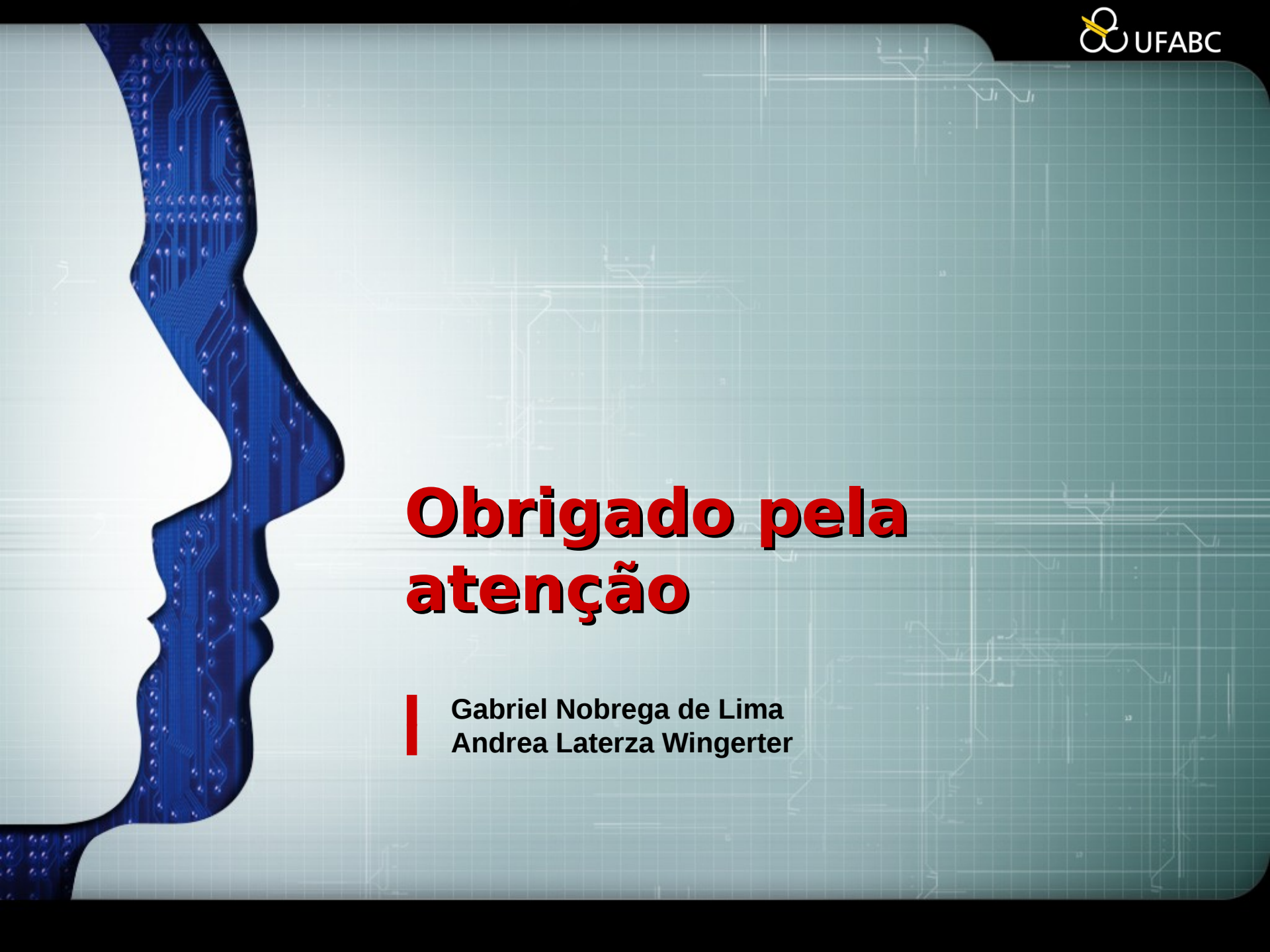
Melhor resultado Roleta: (Alfa)

Tempo de execução: 349 segundos

Tempo de execução:

Resetar

Base de dados: Mammographic.arff Log



Obrigado pela atenção

Gabriel Nobrega de Lima
Andrea Laterza Wingerter