

Disciplina: Computação Bioinspirada

Professor: Prof. Paulo Henrique Ribeiro Gabriel

Nome: Gabriel Teles **Matrícula:** 12211BSI274

Nome: Felipe Harrison Silva Cantarino **Matrícula:** 12011BSI226

Nome: João Victor Guizi Gerardi **Matrícula:** 12011BSI214

Projeto 3: Combinando Algoritmos Bioinspirados

Conjunto de dados

Para realização dessa parte do projeto foi utilizado, para treinamento e teste, a mesma base de dados do projeto passado, a base Iris, disponível em [\(FISHER, 1988\)](#). Para este projeto foi escolhido duas espécies de iris para fazer sua classificação: Iris-Setosa e Iris-Versicolour.

Pesos iniciais

Para gerar os pesos iniciais do Algoritmo Genético(AG), não foi definido nenhuma *seed* de aleatoriedade, dessa forma, cada teste iniciou com pesos diferentes para os cromossomos. Esta abordagem foi adotada, para melhor aproveitamento do AG, já que utiliza de aleatoriedade para seu funcionamento, sendo assim, evitaria que uma *seed* ruim seja escolhida prejudicando seu desempenho.

Os pesos podem variar de -1,0 até 1,0.

Parâmetros

No processo de criação do projeto, ficou definido quatro parâmetros customizáveis, sendo:

- **trainRate**: Proporção da base de dados destinada ao treino;
- **numberCromossomos**: Número de cromossomos que serão gerados no AG;
- **numInteracoes**: Número de execuções do AG;
- **taxaMutacao**: Taxa que os cromossomos vão sofrer mutação;

O Algoritmo Genético

O objetivo do trabalho era a utilização de um algoritmo genético para otimização dos pesos da rede neural perceptron, implementada no projeto passado. Dessa forma, foi adaptado o AG, desenvolvido no projeto 1, para chegar ao objetivo.

Estrutura dos Cromossomos

Os cromossomos em sua estrutura são compostos por uma lista de 4 números reais, correspondendo aos pesos.

Características

Para o projeto, o algoritmo genético foi desenvolvido com as seguintes técnicas na sua composição:

- **Método de Seleção**: Seleção por torneio para composição da população intermediária, utilizando elitismo para sempre adicionar o melhor cromossomo da geração;
- **Recombinação**: Foi utilizado o *cross-over* de 1 ponto;
- **Substituição**: Foi utilizado substituição dos 2 piores cromossomos da geração;

O processo de Mutação

Como neste projeto os cromossomos são de números reais, o processo de mutação é diferente do binário. No projeto foi definido que cada peso tem a chance de sofrer mutação, definido pelo parâmetro **taxaMutacao**. Caso ocorra a mutação de um peso, o processo foi definido pela seguinte lógica:

$$pesoMutado = pesoAnterior + (numAleatorio[0,1] * taxaMutacao) * (numAleatorio(-1,1))$$

A parte $(numAleatorio[0,1] * taxaMutacao)$, serve para definir em quanto o peso será alterado, nesse caso o peso será alterado com base numa porcentagem aleatória da **taxaMutacao**. Já a parte $(numAleatorio(-1,1))$ serve para definir de maneira aleatória se o peso será aumentado ou diminuído.

Aptidão

Para calcular a aptidão dos cromossomos foi escolhido a precisão. Para obter essa precisão foi aplicado a função de ativação, do perceptron, para cada dado da base de treino e em seguida verificado se a previsão foi correta ou errada. Dessa forma, foi calculado o número de acertos do cromossomo, definindo assim sua aptidão ou *fitness*.

Os melhores cromossomos serão aqueles com maior número de acertos.

Testes realizados

Para todos os testes realizados, foi analisada a variação de um dos parâmetros customizáveis em relação à assertividade do Perceptron. Essa assertividade foi obtida calculando a média de acertos, em 30 repetições do programa, em seguida, aplicando a seguinte fórmula de proporção:

$$Assertividade = \frac{mediaAcertos * 100}{maxAcertos}$$

Onde maxAcertos depende da proporção de dados destinados a teste, o parâmetro **trainRate**.

Teste 1: Diferentes proporções

Para este teste, foi definido como parâmetros: taxaMutacao = 0,5, numInteracoes = 200 e numberCromossomos = 20. Este teste analisou a assertividade em relação à variação do parâmetro **trainRate**, entre o intervalo 0,1 a 0,9.

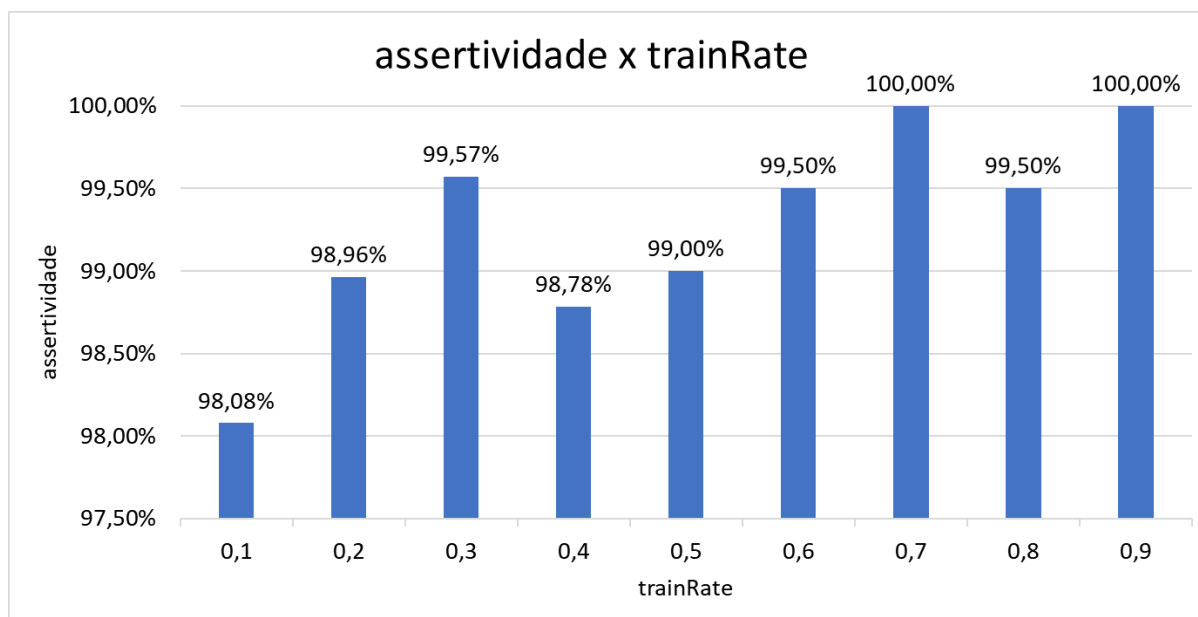


Figura 1 - Gráfico do projeto atual assertividade em relação à variação do parâmetro trainRate

Pelo gráfico obtido com o teste, é possível notar que o Algoritmo genético, com 20 cromossomos, conseguiu gerar pesos bons para fornecer ao Perceptron uma assertividade de mais de 99,5%, utilizando apenas 30% dos dados para treinamento.

Com base no gráfico do projeto passado, que analisa a assertividade em relação ao parâmetro **trainRate**, é possível notar que o Perceptron com essa mesma proporção de treino chegou em apenas 98%.

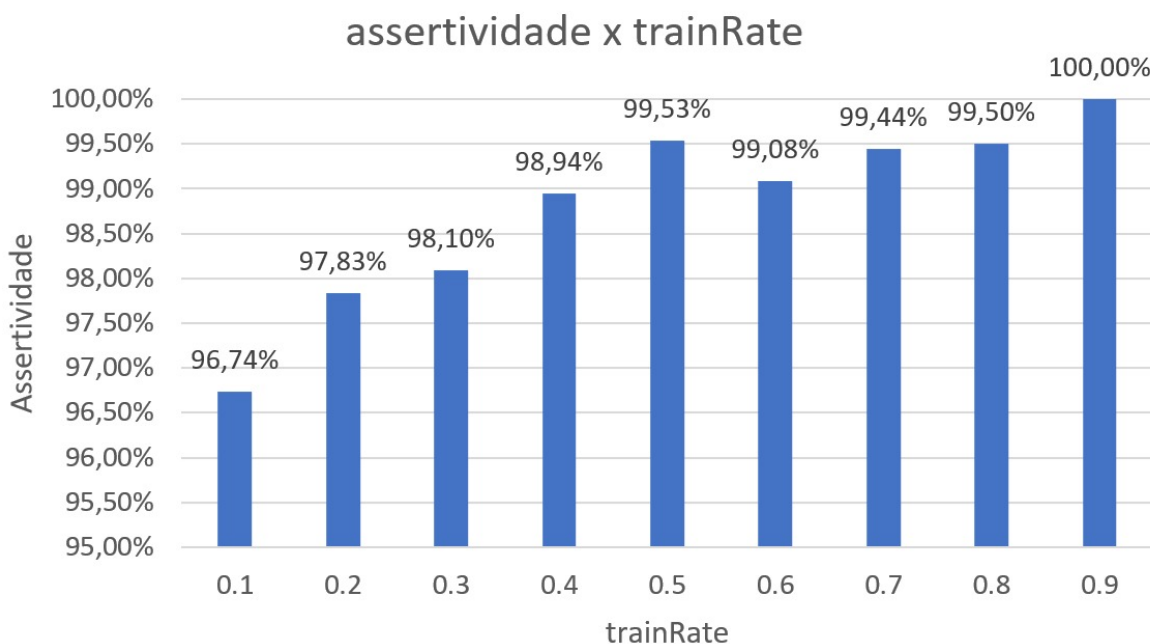


Figura 2 - Gráfico do projeto anterior assertividade em relação à variação do parâmetro trainRate

Teste 2: Diferentes números de interações

Para este teste, foi definido como parâmetros: taxaMutacao = 0,5, trainRate = 0,5 e numberCromossomos = 20. Este teste analisou a assertividade em relação à variação do parâmetro **numInteracoes**, entre o intervalo 100 a 1000.

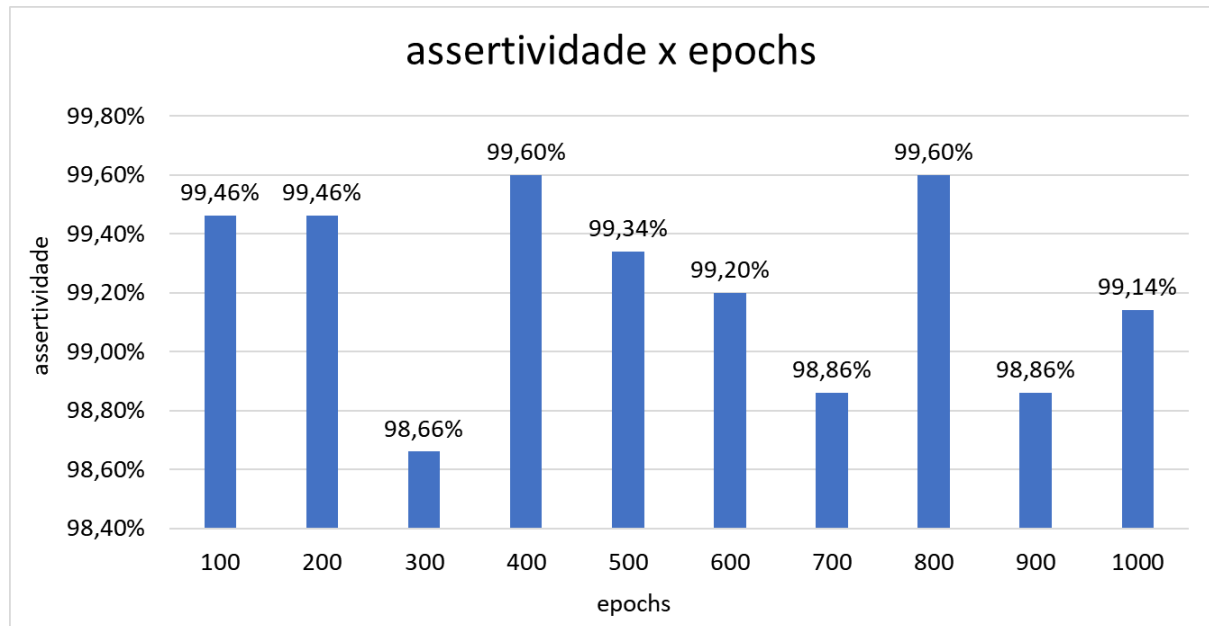


Figura 3 - Gráfico do projeto atual assertividade em relação à variação do parâmetro numInteracoes

Pelo gráfico obtido nesse teste, é possível notar que o Algoritmo genético não melhorou os pesos, para melhorar a assertividade do perceptron conforme o número de gerações foi aumentando, demonstrando assim que o parâmetro **numInteracoes** maior, não aumenta a assertividade.

Com base no gráfico do projeto passado, que analisa a assertividade em relação ao parâmetro **epochs**, analisando a assertividade para 100 e 200, é possível notar que com a ajuda do AG o perceptron melhorou um pouco sua assertividade quanto a repetições.

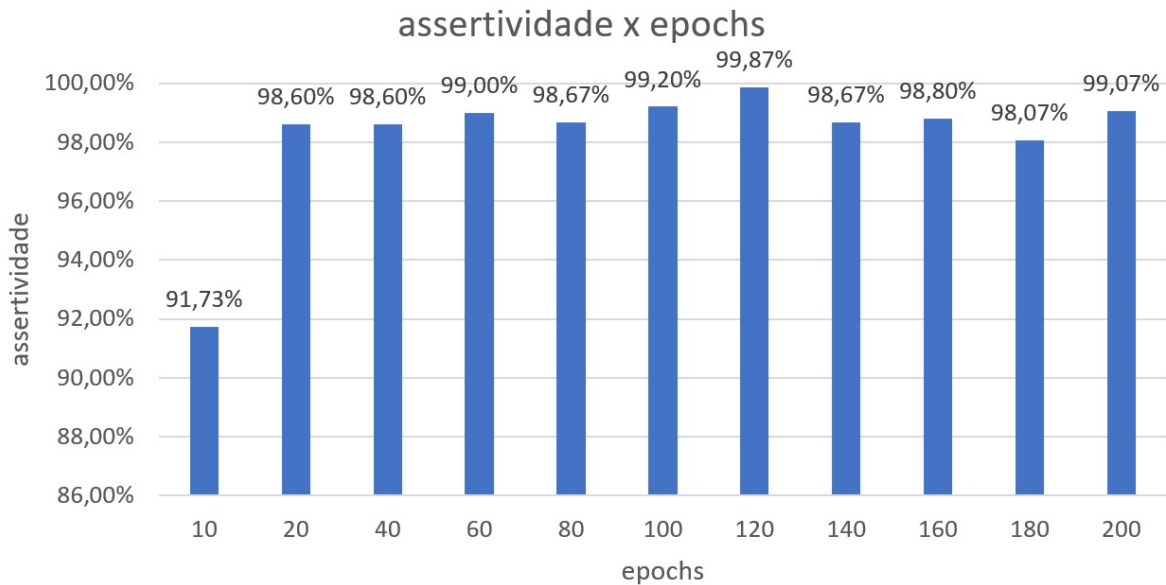


Figura 4 - Gráfico do projeto anterior assertividade em relação à variação do parâmetro epochs

Teste 3: Diferentes números de cromossomos

Para este teste, foi definido como parâmetros: $\text{trainRate} = 0,3$, $\text{numInteracoes} = 200$, $\text{taxaMutacao} = 0,5$. Este teste analisou a assertividade em relação à variação do parâmetro **numberCromossomos**, entre o intervalo 10 a 100 cromossomos.

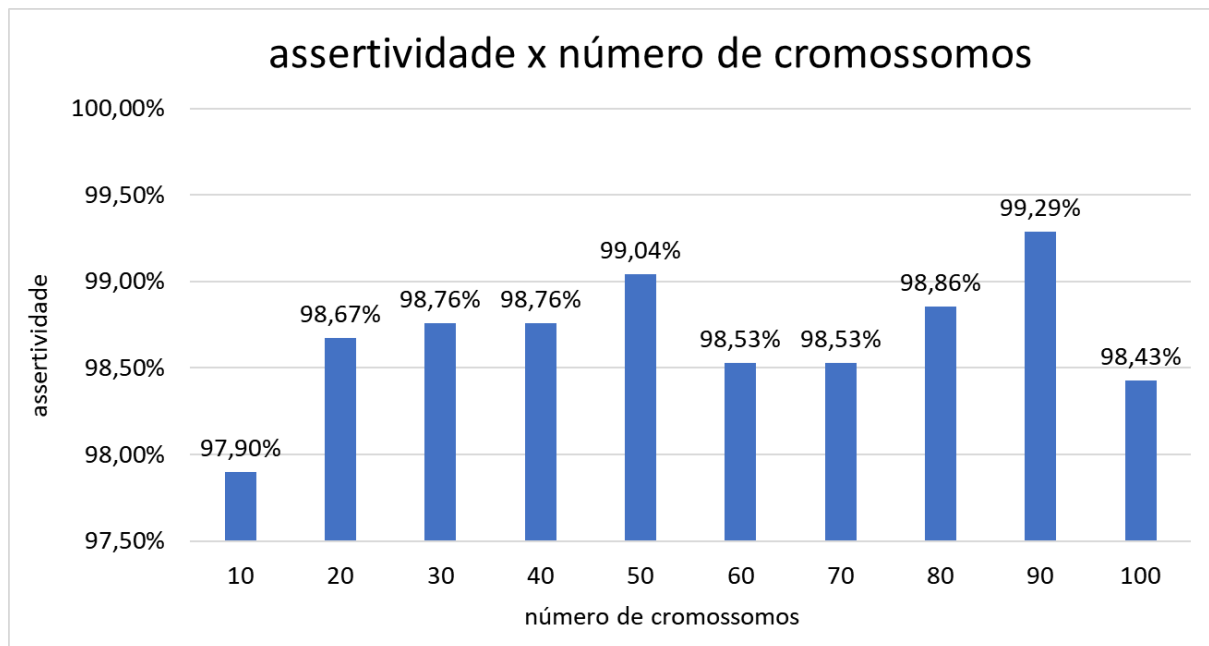


Figura 5 - Gráfico do projeto atual assertividade em relação à variação do parâmetro numberCromossomos

No gráfico acima, é possível observar que de 10 a 50 cromossomos, a assertividade foi crescente, mas não se pode afirmar que o número de cromossomos é o principal causador deste efeito, pois quando a execução foi feita com 60, 70 e até mesmo 100 cromossomos, a assertividade foi menor do que na execução com apenas 20.

Teste 4: Diferentes taxas de mutação

Para este teste, foi definido como parâmetros: `trainRate = 0,5`, `numInteracoes = 200`, `numberCromossomos = 20`. Este teste analisou a assertividade em relação à variação do parâmetro **taxaMutacao**, entre o intervalo 0,1 a 1,0 de taxa de mutação.

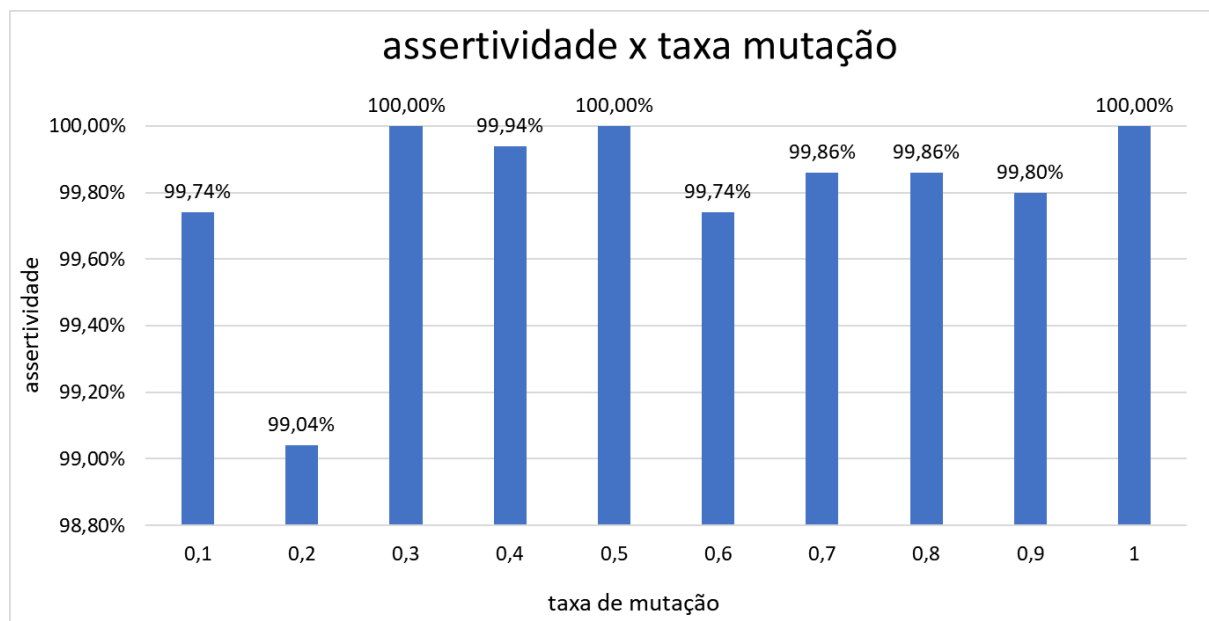


Figura 6 - Gráfico do projeto atual assertividade em relação à variação do parâmetro taxaMutacao

Comparando os resultados com diferentes taxas de mutação com a execução do projeto anterior que executava com diferentes taxas de aprendizagem, pode-se observar um equilíbrio entre os resultados, que por 5 vezes foi maior na execução do projeto anterior e 4 vezes mais no projeto atual.

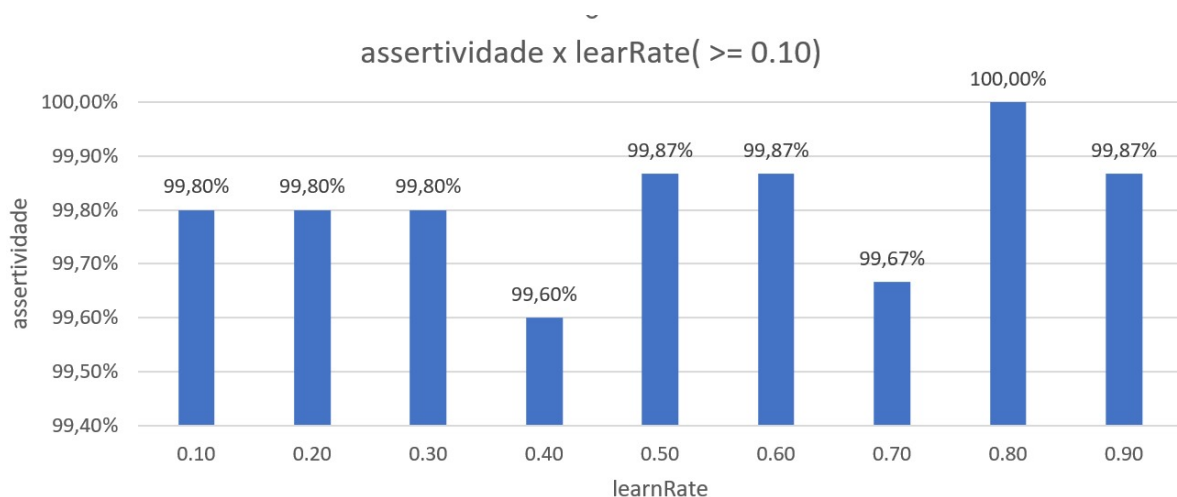


Figura 7 - Gráfico do projeto anterior assertividade em relação à variação do parâmetro learnRate

Resultados obtidos

De modo geral, o perceptron, com o auxílio do Algoritmo genético para otimização dos pesos da rede neural, teve um resultado acima do obtido no projeto anterior, no qual o perceptron melhorava seus pesos sozinho.

A melhora no resultado obtido pode ter ocorrido, por conta da presença de mais pesos variados, uma vez que, o algoritmo genético utiliza vários cromossomos, dessa forma aumentando a possibilidade de surgir bons valores para pesos. Outro fator que pode influenciar nessa melhora, pode ser por conta da distribuição dos dados, que são mais fáceis de serem linearmente separados.