



UNIVERSIDADE FEDERAL DE UBERLÂNDIA – Campus Santa Mônica
SISTEMAS DE INFORMAÇÃO
Computação Bioinspirada

Ellen Christina - ???

Nicolli Freitas - 11911BSI241

Victor Hugo Buiatti Marçal - 11911BSI216

Combinando algoritmos bioinspirados:

**Utilizando algoritmo genético (AG) para otimização de pesos em
uma rede neural do tipo Perceptron**

UBERLÂNDIA - MG

MARÇO 2022

O código foi executado no ambiente Google Colab e utilizando duas classes de Iris para análise (Iris-setosa e Iris-versicolor), no gráfico de resultados os pontos azuis representam a classe Iris-versicolor e os pontos laranjas Iris-setosa. Além disso, utilizamos apenas 2 características da Iris, o tamanho da pétala e o tamanho da sépala.

Nos parâmetros, foram utilizados 50% da base para treinamento e o restante para predição, a taxa de aprendizado usada foi de 0.01, com 10 épocas para cada indivíduo da população e com um número de gerações igual a 20.

Após aplicação do perceptron para cada cromossomo é definido a sua aptidão (fit) e com isso inicia-se o processo evolutivo. Optamos pela recombinação aritmética simples de 1 ponto com alfa igual a 0.1 e a seleção baseada no Fitness, onde é verificado se os filhos possuem fitness maiores que os dois piores (com menor fit) da população. Ao fim do número de gerações pré-definido retornamos o indivíduo com maior fitness que será utilizado na predição dos dados.

Verificamos o conjunto de pesos iniciais gerados de forma aleatória.

```
Pesos Iniciais:
[ 0.38060166  0.46915048 -0.04781192]
[0.38659723  0.37758355  0.29419769]
[0.08128469  0.4056423   0.26511382]
[-0.24680909 -0.2880206   -0.44970357]
[-0.06319459  0.36505661 -0.22910063]
[-0.44520375 -0.14370292 -0.21233304]
[0.10404182  0.43653068  0.0979003 ]
[ 0.1757228   0.46981034 -0.28648649]
[-0.12907194 -0.07060207 -0.19008862]
[0.03408303  0.25415428  0.19052457]
[-0.34651621  0.1646207   -0.41871266]
[ 0.12738799 -0.00057239 -0.10207604]
[-0.09792838 -0.36227532  0.22384495]
[-0.37225136 -0.23227073  0.09576384]
[-0.43136759  0.45379833  0.0517556 ]
[ 0.13242448  0.44719449 -0.28493243]
[-0.46012026 -0.14269331 -0.38526209]
[ 0.05290969 -0.07131564  0.12325053]
[ 0.2882566   0.40060265  0.02120143]
```

A alteração desses pesos após a classificação com perceptron.

```
Pesos pós perceptron:
[[0.22060165508477153, -0.13084951824839958, 0.18818807763955325], [0.2865972336830048, -0.12041645354264695, 0.14019769083150713], [-0.01233304, 0.26511382, -0.44970357], [0.38659723, 0.37758355, 0.29419769], [0.08128469, 0.4056423, 0.26511382], [-0.24680909, -0.2880206, -0.44970357], [-0.06319459, 0.36505661, -0.22910063], [-0.44520375, -0.14370292, -0.21233304], [0.10404182, 0.43653068, 0.0979003], [0.1757228, 0.46981034, -0.28648649], [-0.12907194, -0.07060207, -0.19008862], [0.03408303, 0.25415428, 0.19052457], [-0.34651621, 0.1646207, -0.41871266], [0.12738799, -0.00057239, -0.10207604], [-0.09792838, -0.36227532, 0.22384495], [-0.37225136, -0.23227073, 0.09576384], [-0.43136759, 0.45379833, 0.0517556], [0.13242448, 0.44719449, -0.28493243], [-0.46012026, -0.14269331, -0.38526209], [0.05290969, -0.07131564, 0.12325053], [0.2882566, 0.40060265, 0.02120143]]
```

E por fim a seleção do melhor individuo (com maior fitness).

```
Pesos apos treinamento: [0.36653331561696434, -0.17312410768331357, 0.1806246519593701]
```

Foi obtido uma precisão de 100% nas predições.

```
Precisão do modelo: 1.0
```

E observando o gráfico podemos observar que os dados foram separados corretamente.

