GRAVITATIONAL SEARCH ALGORITHM

Bruno Oliveira nº2019136478
Micael Eid nº2019112744

Indíce

Computação Swarm

- É inspirado no comportamento coletivo de organismos sociais, como insetos, pássaros e peixes, que interagem uns com os outros para realizar tarefas complexas de forma coordenada.
- Um grande número de agentes simples, chamados de "partículas" ou "indivíduos", interagem localmente entre si para atingir um objetivo global

- Pode ser aplicado:
  - Otimização de Hiperparâmetros
  - Treino Distribuído
  - Otimização de Topologia da Rede
  - Deteção de Anomalias e Adaptação Dinâmica
  - Seleção de Conjunto de Dados:

GSA

Vantagens

- É relativamente simples de entender e implementar
- Possui menos parâmetros para ajustar
- GSA tende a ser eficaz na exploração global do espaço de busca, graças à influência gravitacional entre as partículas
- Inspiração nas leis físicas da gravidade

Desvantagens

- Pode convergir para a solução ótima de maneira mais lenta
- Escolha dos mesmos pode influenciar significativamente o desempenho do algoritmo
- A eficácia do GSA pode variar dependendo das características específicas do problema

GSA e PSO

Semelhanças

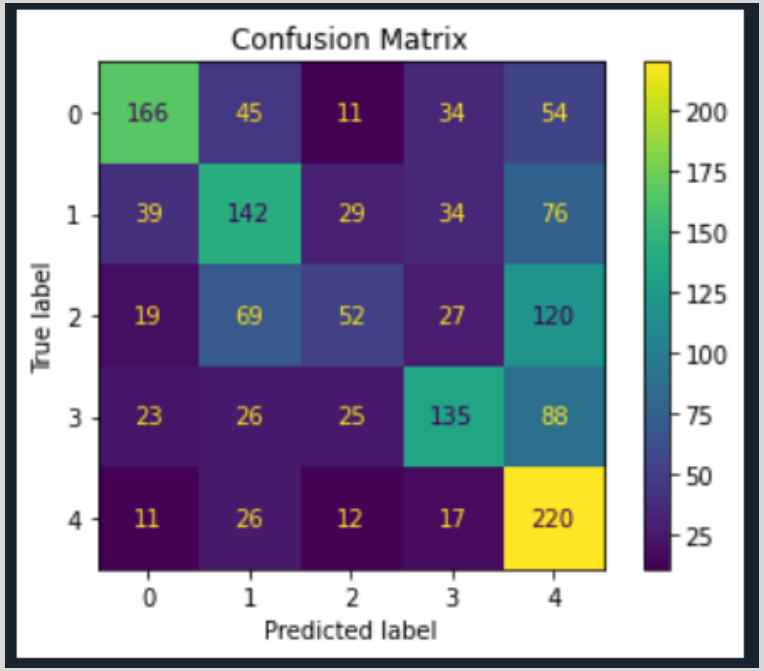
- Ambos os algoritmos são inspirados em comportamentos coletivos de entidades naturais (partículas no PSO e corpos celestes no GSA).
- Ambos procuram um equilíbrio entre a exploração do espaço de busca em busca de novas soluções e a exploração de regiões promissoras.

Diferenças

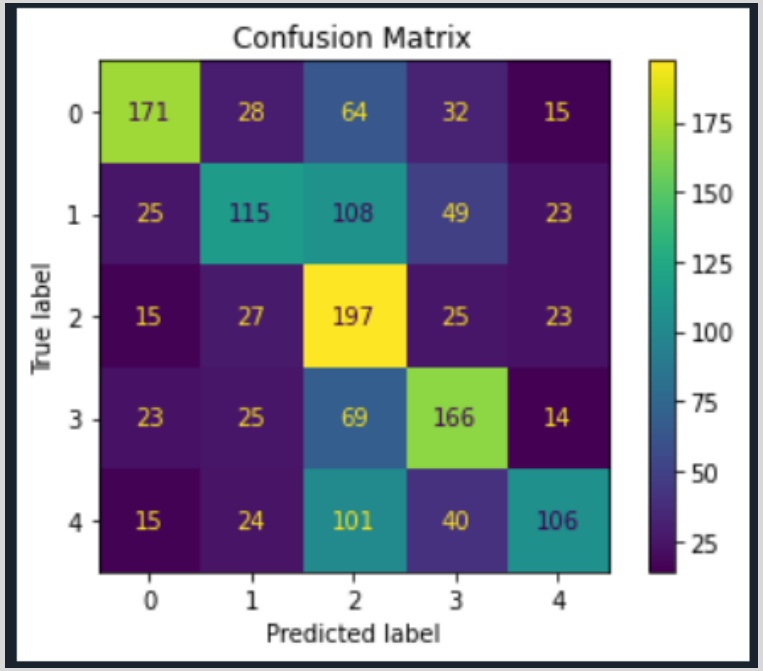
- PSO usa uma abordagem de atualização contínua das posições, enquanto o GSA emprega uma abordagem baseada em leis físicas para calcular as posições em cada iteração.
- No GSA, devido à sua influência gravitacional, tende a ser mais eficiente na exploração global do espaço de busca, enquanto o PSO pode ser mais sensível a ficar preso em ótimos locais.
- PSO requer a configuração de parâmetros como coeficientes de inércia, fatores de aprendizagem. O GSA tem menos parâmetros para ajustar, o que simplifica a implementação.

Função Ackley

Função Ackley: 45%



Função Fitness do GSA: 50%



Optimização de Hiperparâmetros

Mudanças no código

- Reshape das imagens – As imagens passaram de uma resolução de 28x28 para 100x100
- Cor das Imagens – As imagens passaram do tom acinzentado para RGB

Hiperparâmetros

- Nº de neurónios:
  - 50, 100, 150, 200, 250, 500
- Nº de camadas:
  - 1 a 5
- Funções de otimização:
  - Softmax, sigmoide, reLU

Optimização de Hiperparâmetros

- Parâmetros do Gravitational Search Algorithm utilizados para otimização:
- Nº de agentes - 2, 3, 5, 10 e 50
- lb - [1, 50]
- Ub – [5, 500]
- GO - 2 e 3
- Iterações - 2 e 3
- Dimensions - 5, 10, 20 e 100

- Função Fitness da GSA

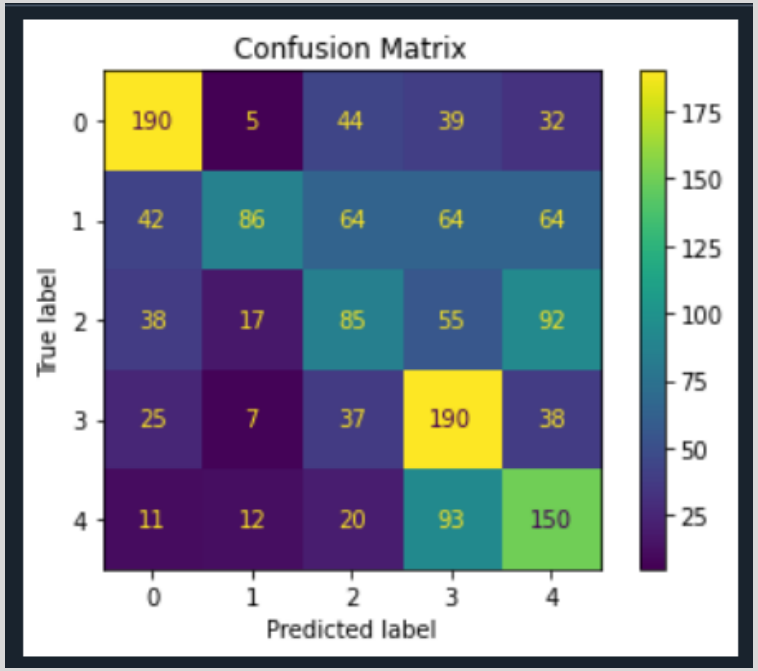
```python
dimension = 2

def gravitational_force(m1, m2, r, G):
    return G * (m1 * m2) / r**2

def fitness_function(position):
    """
    Evaluate the fitness of a solution for the GSA optimization based on gravitational forces.

    Parameters:
    - position (numpy array): The position of a solution in the search space.
    - dimensionality (int): The dimensionality of the search space.

    Returns:
    - fitness (float): The fitness value of the solution.
    """
    G = 6.674 * 10**-11  # gravitational constant

    # Assuming each element of the position array represents a mass
    masses = position

    # Calculate total gravitational force among masses
    total_force = 0.0
    for i in range(dimension):
        for j in range(dimension):
            if i != j:
                r = np.abs(i - j)  # Simple distance assumption for illustration
                total_force += gravitational_force(masses[i], masses[j], r, G)

    # Fitness is the negative of the total gravitational force (since we typically maximize fitness)
    fitness = -total_force

    return fitness
```

Conclusão e discussão de resultados

- Contrariamente ao trabalho realizado na meta 1, esta meta foi bastante mais trabalhosa tendo em conta a complexidade e a valorização da mesma.
- Devido a uma falta de documentação do nosso algoritmo, a nossa implementação do mesmo demorou muito mais tempo do que era suposto, comprometendo o nosso desenvolvimento do PSO pedido para esta meta.
- Contudo e tendo em conta estas adversidades conseguimos implemetar o nosso algoritmo obtendo um aumento relativamente à meta passada.
- Na meta passada a nossa accuracy era de 45% (média) e aumentou para 56% (em média) em praticamnete todos os testes sendo que o nosso melhor resultado foi de 58%.
- Esperamos que na meta final o nosso valor de accuracy consiga estar em valores acima de 90%.

Conclusão e discussão de resultados

Valores base: 47%



Valores com SGA: 58%