AI VIET NAM – COURSE 2022

Genetic Algorithms - Exercises

Ngày 14 tháng 10 năm 2022

Phần I: Hướng dẫn chung

Problem 1 - GAs cho bài toán dự đoán

- **Mô tả bài toán**
  Bài toán dự đoán Doanh thu bán hàng (sale) dựa vào các kênh Marketing khác nhau là một dạng bài toán kinh điển thường được dùng trong giảng dạy Machine Learning. Bộ dữ liệu có thể download tại LINK

  Bộ dữ liệu bao gồm 200 mẫu, mỗi mẫu có 3 đặc trưng, là số tiền quảng cáo trên TV, Radio và Newspaper. Giá trị sale (doanh thu) được cung cấp tương ứng với chi phí của 3 kênh marketing được minh họa qua bảng sau:

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8

Hình 1: Một số sample đầu tiên của dataset advertising

  Để đơn giản vấn đề, chúng ta giả sử chi phí bỏ ra cho các kênh Marketing tuyến tính với doanh thu (mặc dù thực tế còn chịu ảnh hưởng bởi một số yếu tố khác nữa), vì vậy bộ dữ liệu này được mô tả lại theo công thức:

$$sale = c_1 * TV + c_2 * Radio + c_3 * Newspaper + c_4 \quad (1)$$

  Chúng ta sẽ dùng GA để tìm ra giá trị các tham số c_1, c_2, c_3, c_4. Từ yêu cầu bài toán, chúng ta đã xác định được một số thông tin cho GA:

  - Chiều dài của chromosome là 4
  - Gen có kiểu dữ liệu là floating-point
  - Miền giá trị của các tham số c_i: $a \leq c_i < b$ (a: giới hạn dưới (lower limit) và b: giới hạn trên (upper limit))

`Để thuận tiện cho việc cài đặt, phương trình (1) có thể viết lại như sau:`

$$sale = c_1 * TV + c_2 * Radio + c_3 * Newspaper + c_4 * 1.0 \quad (2)$$

Ta thêm giá trị 1.0 vào mỗi sample để đặc trưng là 4, bằng với số lượng tham số.

- **Triển khai thuật toán**

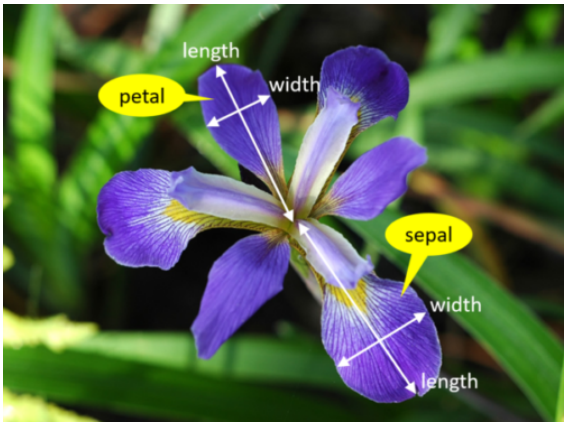
1. **Khai báo các tham số cần thiết:**
   - **n**: kích thước của một cá thể (individual)
   - **m**: kích thước của (quần thể) population
   - **n_generations**: Số lần lặp để tạo ra các thế hệ mới. Mỗi một lần lặp là một thế hệ mới được sinh ra
   - **losses**: lưu giá trị loss để vẽ biểu đồ
2. **Khởi tạo các hàm tính toán cần thiết:** các bạn có thể đặt tên khác, nhưng phải rõ nghĩa, và tuân theo quy tắc đặt tên hàm
   - **generate_random_value(bound)**: hàm tạo giá trị ngẫu nhiên $[\frac{-bound}{2}, \frac{bound}{2})$ cho một gene
   - **compute_fitness(individual)**: hàm đánh giá độ tốt của cá thể, và được dùng trong bước selection.
   - **compute_loss(individual)**: Là hàm nghịch đảo của hàm fitness
   - **create_individual()**: tạo ra các giá trị ngẫu nhiên cho một cá thể (tham số theta của mô hình)
   - **crossover(individual1, individual2, crossover_rate = 0.9)**: bước thực hiện việc lai tạo (trao đổi gen) giữa 2 individual với tỉ lệ crossover_rate.
   - **mutate(individual, mutation_rate = 0.05)**: bước thực hiện việc đột biến cho một cá thể với tỉ lệ đột biến là mutation_rate.
   - **selection(sorted_old_population)**: hàm chọn lọc những cá thể tốt trong một quần thể
   - **create_new_population(old_population, elitism=2, gen=1)**: Bước tạo ra quần thể mới
3. **Thực hiện huấn luyện:**
   (a) **Initial Population**: Khởi tạo ngẫu nhiên một quần thể
   (b) **Fitness function**: Tìm fitness value (giá trị đánh giá độ phù hợp) của mỗi chromosomes (mỗi **individual** có một **chromosome** là một list các **gene**. Mỗi chromosome là một tập hợp các tham số giúp xác định một solution được đề xuất cho vấn đề mà thuật toán di truyền đang cố gắng giải quyết)
   (c) **Selection**: Lựa chọn chromosomes trong population hiện tại
   (d) **Cross-over**: Tạo ra các chromosomes mới bằng cách kết hợp và trao đổi gene giữa các chromosomes cũ (parents)
   (e) **Mutation**: Thực hiện đột biến làm thay đổi một hoặc nhiều giá trị gene trên một chromosome. Mutation giúp tạo ra sự đa dạng hơn trong quần thể. Population thu được sẽ được sử dụng trong thế hệ tiếp theo

   Giải thuật GAs thực hiện bước (a) một lần ban đầu, và các bướ từ (b) đến (e) được thực hiện cho mỗi generation.

Problem 2 - GAs cho bài toán phân loại (1D)

- **Mô tả bài toán**

Tiếp nối problem 1, ở bài này chúng ta mở rộng bằng cách kết hợp GA với Logistic Regression. Bộ dữ liệu được dùng là IRIS 1D, có thể download tại LINK



Hình 2: Cánh và đài hoa Iris

Iris setosa Iris versicolor Iris virginica

Hình 3: 3 loại hoa Iris

Hoa Iris là một loại hoa Lan, bộ dữ liệu bao gồm 6 mẫu, mỗi mẫu có 1 đặc trưng, là chiều dài cánh hoa (petal length), tương ứng với các loại hoa, lần lượt có giá trị là 0 và 1, được minh họa qua bảng sau:

Petal_Length	Label
1.4	0
1	0
1.5	0
3	1
3.8	1
4.1	1

Hình 4: Một số sample của dataset iris 1D

Trong bài toán này, ta dùng GA kết hợp Logistic Regression để phân loại hoa dựa vào chiều dài cánh hoa:

$$\textit{Hàm Sigmoid: } f(x) = \frac{1}{1+e^{-x}}$$

$$\textit{Hàm loss: } L = ylog(y\_hat) - (1-y)log(1-y\_hat)$$

- **Triển khai thuật toán**
  Thực hiện tương tự như Problem 1, chỉ khác ở chỗ khi tính hàm compute_loss(individual) có sử dụng tới loss_function(y_hat, y), predict(X, theta) và sigmoid_function(z).

Problem 3 - GAs cho bài toán phân loại (IRIS Flowers)

- **Mô tả bài toán**
  `Cũng là hoa Iris, nhưng mở rộng cho Problem 2 thành nhiều đặc trưng, tức dạng bài toán multidimension, có thể download data tại` LINK

  `Một số sample của dữ liệu:`

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Label
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0
5	3.6	1.4	0.2	0
5.4	3.9	1.7	0.4	0
4.6	3.4	1.4	0.3	0
5	3.4	1.5	0.2	0
4.4	2.9	1.4	0.2	0
4.9	3.1	1.5	0.1	0
5.4	3.7	1.5	0.2	0
4.8	3.4	1.6	0.2	0

Hình 5: Một số sample của dataset Iris

- **Triển khai thuật toán**
  Thực hiện tương tự như Problem 2, chỉ khác ở chỗ tạo dữ liệu đầu vào cho X, thay vì là 1 đặc trưng (2 column, có bias) sẽ thành 4 đặc trưng (5 column, có bias).

Phần II: Trắc nghiệm

1. **Exploration (khám phá) tương ứng với bước nào trong GA**
   (a) Population Initialization
   (b) Cross-over
   (c) Mutation
   (d) Selection
2. **Exploitation (khai thác) tương ứng với bước nào trong GA**
   (a) Population Initialization

(b) Cross-over

(c) Mutation

(d) Selection

3. **Các bạn hãy sử dụng file GA-LogisticRegression-IRIS-1D-hint, cài hàm compute-loss() và cho chạy với 1 generation (Các bạn gán giá trị của n_generations=1 trong code gợi ý). Giá trị loss tốt nhất (giá trị loss của cá thể có giá trị loss nhỏ nhất) là**

   (Trước khi chạy chương trình, các bạn hãy clear hay restart các cell trước, để đảm bảo hàm random chạy lại từ đầu với seed=0)

   (a) 0.0171

   (b) 0.0925

   (c) 0.0383

   (d) 0.0676

4. **Các bạn hãy sử dụng file GA-LogisticRegression-IRIS-1D-hint, cài hàm compute-loss() và cho chạy với 100 generation (Các bạn gán giá trị của n_generations=100 trong code gợi ý). Giá trị theta tốt nhất là**

   (a) b = -9.84, w = 4.426

   (b) b = -9.95, w = 4.535

   (c) b = -9.91, w = 4.428

   (d) b = -9.86, w = 4.235

5. **Các bạn hãy sử dụng file GA-LogisticRegression-IRIS-Full-hint, cài hàm compute-loss() và cho chạy với 1 generation. Giá trị loss tốt nhất là**

   (a) 0.0000098

   (b) 0.0000095

   (c) 0.0000097

   (d) 0.0000096

6. **Các bạn hãy sử dụng file GA-LogisticRegression-IRIS-Full-hint, cài hàm compute-loss() và cho chạy với 100 generation. Giá trị theta tốt nhất là**

   (a) $b = -9.04, w_1 = 0.72, w_2 = -9.47, w_3 = 9.99, w_4 = 9.99$

   (b) $b = -8.90, w_1 = 0.74, w_2 = -9.59, w_3 = 9.99, w_4 = 9.99$

   (c) $b = -8.90, w_1 = 0.73, w_2 = -9.58, w_3 = 9.99, w_4 = 9.99$

   (d) $b = -8.90, w_1 = 0.75, w_2 = -9.57, w_3 = 9.99, w_4 = 9.99$

7. **Các bạn hãy sử dụng file GA-Advertising-hint, cài hàm compute-loss() và cho chạy với 1 generation. Giá trị loss tốt nhất là**

   (a) 499.42

   (b) 498.45

   (c) 499.19

   (d) 498.89

8. **Các bạn hãy sử dụng file GA-Advertising-hint, cài hàm compute-loss() và cho chạy với 100 generation. Giá trị theta tốt nhất là**

   (a) $b = 4.51, w_1 = 0.09, w_2 = 0.18, w_3 = -0.06$

   (b) $b = 4.97, w_1 = 0.02, w_2 = 0.16, w_3 = -0.02$

   (c) $b = 4.98, w_1 = 0.08, w_2 = 0.17, w_3 = -0.09$

   (d) $b = 4.99, w_1 = 0.04, w_2 = 0.17, w_3 = -0.01$