Lab 4.3: Áp dụng giải thuật di truyền giải quyết bài toán dự đoán doanh thu

ThS. Lê Thị Thùy Trang

2025-1-19

1. Giới thiệu về bài thực hành

Bài toán dự đoán doanh thu bán hàng dựa vào các kênh Marketing khác nhau là một dạng bài toán kinh điển trong trí tuệ nhân tạo. Trong bài thực hành này, chúng ta sẽ xây dựng chương trình áp dụng giải thuật di truyền để dự đoán doanh thu bằng thư viện `pyGAD`.

Bài thực hành được xây dựng dựa trên tập dữ liệu gồm 200 mẫu quan sát, mỗi mẫu có 3 đặc trưng, tương ứng với số tiền quảng cáo trên TV, Radio và Newspaper. Cột cuối cùng của bộ dữ liệu mang thông tin về doanh thu tương ứng với chi phí bỏ ra cho 3 kênh marketing.

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
199.8	2.6	21.2	15.6

Figure 1: Một vài mẫu từ dữ liệu quảng cáo advertising.csv

Giả sử chi phí bỏ ra cho các kênh Marketing tuyến tính với doanh thu, vì vậy doanh thu có thể được tính theo công thức:

$$\text{sale} = \theta_1 \text{ TV} + \theta_2 \text{ Radio} + \theta_3 \text{ Newspaper} + \theta_4$$

Chúng ta sẽ áp dụng giải thuật di truyền để tìm ra giá trị các tham số $\theta_1, \theta_2, \theta_3, \theta_4$.

Từ yêu cầu bài toán, chúng ta xác định một số thông tin cho GA như chiều dài mỗi chromosome.

Q1: Hãy cho biết chiều dài của chromosome trong trường hợp này là bao nhiêu?

A. 4

B. 5

C. 6

D. 7

2. Chuẩn bị môi trường thực hành

Trong bài thực hành này, ta sẽ sử dụng thư viện `pyGAD` - một thư viện `Python` hỗ trợ việc triển khai thuật toán di truyền (Genetic Algorithm). `pyGAD` hỗ trợ nhiều loại toán tử như: crossover, mutation, parent selection. `pyGAD` cho phép tối ưu hoá nhiều loại vấn đề như tối ưu hoá đơn mục tiêu hay tối ưu hoá đa mục tiêu khác nhau bằng thuật toán di truyền bằng cách tuỳ chỉnh hàm `fitness`.

Figure 2: pyGAD

`pyGAD` có thể được cài đặt thông qua `pip`

```
pip install pygad
```

Sau khi cài đặt, xác minh xem `pyGAD` đã được cài đặt thành công hay chưa bằng cách mở `Python` và chạy:

```
import pygad
print(pygad.__version__)
```

Bên cạnh đó, trong bài thực hành này, sinh viên sẽ làm quen với việc đọc và xử lý dữ liệu bằng thư viện `pandas`. Sinh viên cần cài thư viện này bằng `pip` như sau:

```
pip install pandas
```

3. Cài đặt chương trình

Chương trình sử dụng thư viện `pyGAD` để tối ưu hoá các trọng số của mô hình nhằm dự đoán doanh thu dựa trên các chi phí quảng cáo. Các thành phần của chương trình bao gồm:

3.1 Import các thư viện cần thiết

```
import numpy as np
import pandas as pd
import pygad
```

3.2 Đọc và chuẩn bị dữ liệu

pandas cung cấp cho chúng ta phương pháp đọc dữ liệu từ file csv, chuyển đổi nó thành dataframe một cách đơn giản như sau:

```
data = pd.read_csv('advertising.csv')
X = data[['TV', 'Radio', 'Newspaper']].values
y = data['Sales'].values
```

Dữ liệu được đọc từ file csv và chia thành:

- X: các đặc trưng đầu vào (TV, Radio, Newspaper)
- y: biến mục tiêu (Sales)

3.2 Xây dựng hàm `fitness`

Hàm compute_fitness tính fitness values của một chromosome bằng cách nghịch đảo của giá trị loss được tính bằng hàm compute_loss theo công thức:

$$\text{fitness} = \frac{1}{\text{loss} + \epsilon}$$

Trong đó $□= 10^{-6} $ để tránh chia cho 0.

Q2: Hãy hoàn thiện đoạn code sau đây, biết rằng loss được tính bằng công thức tính Mean Squared Error (MSE):

$$\text{MSE} = (y - \hat{y})^2$$

Trong đó:

- y: giá trị thực tế
- $\hat{y}$: giá trị dự đoán

```
# fitness function
def compute_loss(individual):
    theta = np.array(individual)
    y_hat = X.dot(theta)
    # code here
    return loss

def fitness_function(ga_instance, solution, solution_idx):
    loss = compute_loss(solution)
    epsilon = 1e-6
    # code here
    return fitness_value
```

2.3 Thiết lập các tham số

```
num_generations = 100
num_parents_mating = 10
sol_per_pop = 20
num_genes = X.shape[1]
init_range_low = -1.0
init_range_high = 1.0
parent_selection_type = "sss"
keep_parents = 5
crossover_type = "single_point"
mutation_type = "random"
mutation_percent_genes = 10
```

Các tham số này điều khiển cách thuật toán di truyền hoạt động

- `num_generations`: Số thế hệ.
- `num_parents_mating`: Số lượng cha mẹ được chọn để lai tạo.
- `sol_per_pop`: Số lượng giải pháp trong quần thể.
- `num_genes`: Số lượng gen, hay kích thước của chromosome.
- `init_range_low` và `init_range_high`: Phạm vi khởi tạo ban đầu của các trọng số.
- `parent_selection_type`: Phương pháp chọn cha mẹ (ở đây là "sss" - steady-state selection).
- `keep_parents`: Số lượng cha mẹ được giữ lại từ thế hệ trước.
- `crossover_type`: Phương pháp lai tạo (ở đây là "single_point").
- `mutation_type`: Phương pháp đột biến (ở đây là "random").
- `mutation_percent_genes`: Tỷ lệ đột biến (10%).

2.4 Khởi tạo và thực thi thuật toán

```
ga_instance = pygad.GA(num_generations=num_generations,
                       num_parents_mating=num_parents_mating,
                       fitness_func=fitness_function,
                       sol_per_pop=sol_per_pop,
                       num_genes=num_genes,
                       init_range_low=init_range_low,
                       init_range_high=init_range_high,
                       parent_selection_type=parent_selection_type,
                       keep_parents=keep_parents,
                       crossover_type=crossover_type,
                       mutation_type=mutation_type,
                       mutation_percent_genes=mutation_percent_genes)

ga_instance.run()
```

2.5 In ra kết quả

Đoạn code sau đây thực thi hai công việc chính:

- Lấy và in ra giải pháp tốt nhất
  - `solution`: các trọng số tối ưu của mô hình
  - `solution_fitness`: độ phù hợp của giải pháp tốt nhất
- Sử dụng các trọng số tối ưu để dự đoán doanh thu và in ra 5 dự đoán đầu tiên.

```
solution, solution_fitness, solution_idx = ga_instance.best_solution()
print("Các trọng số tối ưu: ", solution)
print("Độ phù hợp của giải pháp tốt nhất: ", solution_fitness)
y_pred = np.dot(X, solution)
print("Một số dự đoán doanh thu: ", y_pred[:5])
```

Q3: Hãy sửa lại chương trình tính `loss` bằng công thức tính Root Mean Squared Error (RMSE)? So sánh độ phù hợp của giải pháp tốt nhất với `loss` được tính bằng công thức tính MSE?

Q4: Thử thay đổi giá trị các tham số khi khởi tạo thuật toán như `num_generations` hay `mutation_percent_genes`. So sánh kết quả và nhận xét trong trường hợp giá trị khác nhau của hai tham số này?