Lecture 9

Genetic algorithms. Example.

Genetic algorithms

The general scheme of the algorithm looks like:

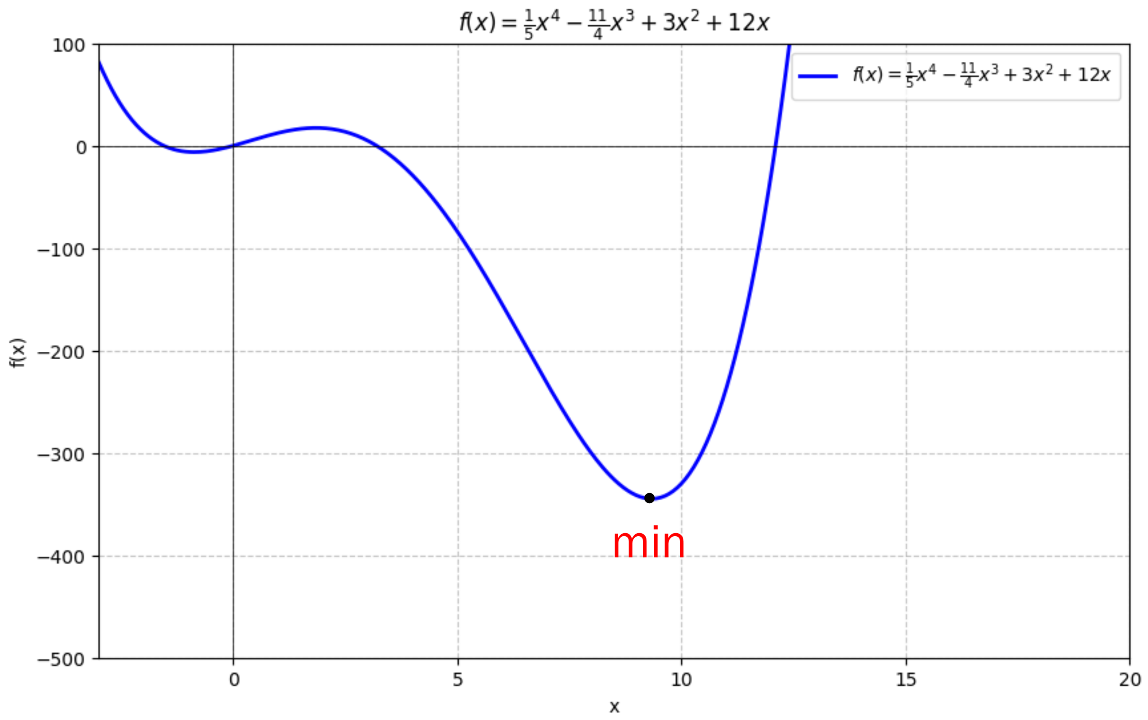
1. Creating an Initial Population
2. Selection (parent selection)
3. Crossbreeding (producing descendants)
4. Mutation (random changes in the genes of descendants)
5. Formation of a new population
6. Checking the termination condition

Genetic algorithms

Example. Find the global minimum of a function

$$f(x) = \frac{1}{5}x^4 - \frac{11}{4}x^3 + 3x^2 + 12x$$

on the interval $[0; 10]$



Genetic algorithms

For simplicity of presentation, we assume that x takes only integer values on the segment under consideration.

Let us randomly select several points from the segment under consideration: $\{0,2,5,7,9\}$.

We will consider these numbers as trial solutions to the problem.

Let us write the selected numbers in binary form: $\{0000,0010,0101,0111,1001\}$.

.

Genetic algorithms

In a competitive struggle, the fittest survives. The fitness of an individual in the problem under consideration is determined by the objective function, i.e. smaller values of the objective function correspond to more fit individuals. Therefore, we will calculate the value of the function at the trial points under consideration:

$$f(0) = 0$$
$$f(2) = 17{,}2$$
$$f(5) = -83{,}75$$
$$f(7) = -232{,}05$$
$$f(9) = -341{,}55$$

.

Genetic algorithms

Next, we move on to the reproduction process, that is, based on the original population, we create a new one in such a way that the trial solutions in the new population are located closer to the desired global minimum. To do this, we form mating pairs from the original population.

We assign each individual of the original population an integer from the range of 1 to 5, that is, we number them. The reproduction process consists of exchanging chromosome sections between the parents.

.

Genetic algorithms

Let the $i-th$ and $j-th$ individuals of the $k-th$ population be represented as $p_i^k = (a_1, a_2, \dots, a_n) \in P^k$ and $p_j^k = (b_1, b_2, \dots, b_n) \in P^k$. In a one-point crossover, two elements of population $k+1$: $p_i^{k+1} = (a_1, \dots, a_c, b_{c+1} \dots, b_n) \in P^{k+1}$, $p_j^{k+1} = (b_1, \dots, b_c, a_{c+1} \dots, a_n) \in P^{k+1}$, where the point c is chosen randomly.

.

Genetic algorithms

Let us represent the process of one-point crossing over in the form of a table

№	Individual	Selected number	The second individual is the parent	Crossover point	Descendant individuals
1	0000	1	0000	1	0000
2	0010	4	0111	2	0011 0110
3	0101	5	1001	3	0101 1001
4	0111	4	0111	2	0111
5	1001	2	0010	2	1010 0001

Genetic algorithms

Next, we should consider possible mutations, i.e. random changes in the chromosomes obtained as a result of crossing. Let the probability of mutation be 0.2. For each descendant, we randomly generate a number from the interval [0,1]. If this number is less than 0.2, we invert the random gene. Mutations can improve or worsen the fitness of the descendant individual.

Genetic algorithms

Let's present the possible mutations in descendant individuals in the form of a table:

№	Descendant individuals	Random number	The gene selected for mutation	Descendant after mutation	Fitness before mutation	Fitness after mutation
1	0011	0,7	3	0011	4,95	4,95
2	0110	0,9	2	0110	-154,8	-154,8
3	0101	0,15	3	0111	-83,75	-232,05
4	1001	0,14	4	1000	-341,55	-300,8
5	1010	0,13	1	0010	-330	17,2
6	0001	0,5	2	0001	12,45	12,45

Genetic algorithms

Now, from the five parent individuals and the six offspring obtained, we will form a new population, into which we will select the five most adapted representatives. As a result, we will obtain a new generation, the representatives of which we will depict in the form of a table:

№	New population	Fitness of individuals in a new population
1	1001	-341,55
2	1000	-300,8
3	0111	-232,05
4	0110	-154,8
5	0101	-83,75

Genetic algorithms

Crossing over can be applied to the resulting population again, a mutation can be added, and the fittest individuals can be selected for the new generation. Thus, after several generations, a population of the fittest and most similar individuals will be obtained. The solution to the problem will be the best fitness value.

Ant colony optimization algorithm

Ant algorithm steps:

1. Initialization of the algorithm. That is, a certain set of ants is created and the parameters required by the algorithm are randomly assigned.

2. Search for a solution, which consists of determining the best path to the goal, which at each stage is determined by the probability of choosing one or another segment of the path. This probability has the form:

$$P_{ij,k}(t) = \frac{\left(\tau_{ij}\right)^{\alpha} \cdot \left(\eta_{ij}\right)^{\beta}}{\sum_{l \in J_{i,k}} (\tau_{il})^{\alpha} \cdot (\eta_{il})^{\beta}}$$

where $\tau_{ij} -$ pheromone level on the edge (i, j); $\eta_{ij} -$ the reciprocal of the distance between nodes $i\ and\ j$, α and $\beta -$ parameters (constants).

Ant colony optimization algorithm

3. The level of pheromones is given by the formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \frac{Q}{L_k(t)}$$

where ρ – is the evaporation rate, $L_k(t)$ – is the price of the current solution for ant *k*; *Q* is the order value of the price of the optimal solution. That is, the fraction $\frac{Q}{L_k(t)}$ is a pheromone deposited by ant *k*.

Ant colony optimization algorithm

Example. Solve the traveling salesman problem. Matrix of distances:

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 25 & 41 & 35 \\ 2 & 25 & - & 56 & 62 \\ 3 & 41 & 56 & - & 58 \\ 4 & 35 & 62 & 58 & - \end{pmatrix}$$

Ant colony optimization algorithm

The solution to the problem will consist of calculating the probabilities of moving from one city to another and constructing a route based on these probabilities. Let us recall that the probability of choosing one or another segment of the route is calculated using the formula:

$$P_{ij,k}(t) = \frac{(\tau_{ij})^{\alpha} \cdot (\eta_{ij})^{\beta}}{\sum_{l \in J_{i,k}} (\tau_{il})^{\alpha} \cdot (\eta_{il})^{\beta}}$$

where τ_{ij} – pheromone level on the edge (i, j); η_{ij} – the reciprocal of the distance between nodes i and j, α and β – parameters (constants).

Ant colony optimization algorithm

Therefore, we will start the solution by setting the parameters. For simplicity of presentation, we will assume that it is equally important for us to take into account the experience of other ants and the distances between cities, that is, $\alpha = 1 \, and \, \beta = 1$.

Next we move on to τ_{ij} – the pheromone level on the edge (i, j). This value can be generated either randomly or all edges can be set to the same numerical value (i.e. not favoring certain edges).

$$\tau(1) = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 1 & 1 & 1 \\ 2 & 1 & - & 1 & 1 \\ 3 & 1 & 1 & - & 1 \\ 4 & 1 & 1 & 1 & - \end{pmatrix}$$

Ant colony optimization algorithm

Let a salesman leave city number 1. His goal is to visit all the other cities and return to the beginning of the route. The length of the route should be minimal. Let's start by calculating the probabilities of moving to a particular city:

$$P_{12}(1) = \frac{(\tau_{12})^1 \cdot (\eta_{12})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{25}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0{,}43$$

$$P_{13}(1) = \frac{(\tau_{13})^1 \cdot (\eta_{13})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{41}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0{,}26$$

$$P_{14}(1) = \frac{(\tau_{14})^1 \cdot (\eta_{14})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{35}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0{,}31$$

Ant colony optimization algorithm

Now we should set the rule by which this or that transition will be selected.

- Randomly generate a number $a \in [0,1]$
- If the generated number a does not exceed $P_{12}(1)$, then we should select the route $1 \rightarrow 2$.
- If the generated number a is greater than $P_{12}(1)$, but less than or equal to $P_{12}(1) + P_{13}(1)$, then we select the route $1 \rightarrow 3$.
- If the generated number a is greater than $P_{12}(1) + P_{13}(1)$, but less than or equal to $P_{12}(1) + P_{13}(1) + P_{14}(1)$, then we select the route $1 \rightarrow 4$.

Let, for example, the randomly generated number $a = 0.2$. According to the rule described above, we get the transition $1 \rightarrow 2$.

Ant colony optimization algorithm

Thus, city 1 has already been considered and can be crossed off the list of cities. The salesman is in city 2 and the next city to visit should be determined. To do this, we calculate the probabilities of moving from city 2 to the other cities (except for the first, since it has already been considered):

$$P_{23}(1) = \frac{(\tau_{23})^1 \cdot (\eta_{23})^1}{\tau_{23}\eta_{23} + \tau_{24}\eta_{24}} = \frac{1 \cdot \frac{1}{56}}{1 \cdot \frac{1}{56} + 1 \cdot \frac{1}{62}} = 0{,}53$$

$$P_{24}(1) = \frac{(\tau_{24})^1 \cdot (\eta_{24})^1}{\tau_{23}\eta_{23} + \tau_{24}\eta_{24}} = \frac{1 \cdot \frac{1}{62}}{1 \cdot \frac{1}{56} + 1 \cdot \frac{1}{62}} = 0{,}47$$

Ant colony optimization algorithm

Similar to the previous step, we will set the rule by which this or that transition will be selected.

- Randomly generate a number $a \in [0,1]$
- If the generated number a does not exceed $P_{23}(1)$, then the route $2 \rightarrow 3$ should be selected.
- If the generated number a is greater than $P_{23}(1)$, but less than or equal to $P_{23}(1) + P_{24}(1)$, then we select the route $2 \rightarrow 4$.

Let, for example, the randomly generated number $a = 0.72$. According to the rule described above, we get the transition $2 \rightarrow 4$.

Ant colony optimization algorithm

Thus, the following route was formed: $1 \to 2 \to 4 \to 3 \to 1$.

Let's calculate its length: $L_0(1) = 25 + 62 + 58 + 41 = 186$

Let's recalculate the pheromone level on the edges. For the edges used above ($1 \to 2 \to 4 \to 3 \to 1$) let's recalculate the pheromone level using the formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \frac{Q}{L_k(t)}$$

$\rho = 0{,}2, Q = 200$

Ant colony optimization algorithm

$$\tau_{12}(2) = (1 - 0{,}2) \cdot 1 + \frac{200}{186} = 1{,}88$$

$$\tau_{24}(2) = (1 - 0{,}2) \cdot 1 + \frac{200}{186} = 1{,}88$$

$$\tau_{43}(2) = (1 - 0{,}2) \cdot 1 + \frac{200}{186} = 1{,}88$$

$$\tau_{31}(2) = (1 - 0{,}2) \cdot 1 + \frac{200}{186} = 1{,}88$$

Ant colony optimization algorithm

For edges not used in the first step, the pheromone values are reduced as follows:

$$\tau_{ij}(2) = (1 - \rho) \cdot \tau_{ij}(1)$$

$$\tau_{14}(2) = \tau_{23}(2) = \tau_{32}(2) = \tau_{41}(2) = (1 - 0{,}2) \cdot 1 = 0{,}8$$

Then the pheromone level matrix will look like:

$$\tau(2) = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 1{,}88 & 1{,}88 & 0{,}8 \\ 2 & 1{,}88 & - & 0{,}8 & 1{,}88 \\ 3 & 1{,}88 & 0{,}8 & - & 1{,}88 \\ 4 & 0{,}8 & 1{,}88 & 1{,}88 & - \end{pmatrix}$$

Ant colony optimization algorithm

This completes the first step of the cycle. Repeat these steps until the specified number of iterations is reached.

Then the solution continues according to the following scheme: if for some route the value of its length becomes less than the value of the route length for the previous step, then this value should be saved, using it when recalculating the pheromone level. If the value of the route length has not decreased, then the length of this path should not be saved - the pheromone for it simply evaporates with a given intensity, and is not updated. We continue the solution until the value of the shortest route length at the current step differs little from the value at the previous step or until a sufficiently large number of iterations specified in advance are performed.