Genetic Algorithms

By : Jamuna S Murthy

Define genetic algorithms and describe their components, including
genetic operators, fitness functions, and selection. Provide an illustrative example of genetic algorithm application.

Overview

Genetic algorithms (GAs) simulate evolution for learning and optimization tasks. Inspired by biological processes like mutation, crossover, and natural selection.

- **Hypothesis Representation**: Typically represented as bit strings, symbolic expressions, or computer programs. Interpretation varies depending on the application.
- **Population-Based Search**: Begins with an initial population of hypotheses. Hypotheses evolve through mutation and crossover to create new generations.
- **Fitness Evaluation**: Hypotheses are assessed using a fitness function. Most fit hypotheses are probabilistically selected for reproduction.
- **Applications**: Learning tasks (e.g., rule sets for robot control).Optimization (e.g., neural network topology and learning parameters).

Key Variants: **Genetic Algorithms**: Hypotheses as bit strings.

Genetic Programming: Hypotheses as computer programs.

Basic Structure of Genetic Algorithm

1. Each iteration in the cycle produces a new "generation" of chromosomes.
2. The entire set of generations is called a run.
3. Typical GA run is from 50 to 500 or more generations.
4. At the end of a run often there is at least one highly fit chromosome in the population.

Basic terminology of GA

1. **Population:** subset of all the possible solutions to the given problem.
2. **Chromosomes:** one such solution to given problem.
3. **Gene:** one element position of a chromosome.
4. **Allele:** value a gene takes for particular chromosome.
5. **Genotype:** population in the computation space.
6. **Phenotype:** population in the actual real world solution space.
7. **Decoding:** transforming a solution from the genotype to the phenotype space.
8. **Encoding:** transforming from the phenotype to genotype space.

Genetic Algorithms

GA(*Fitness*, *Fitness_threshold*, p, r, m)

Fitness: A function that assigns an evaluation score, given a hypothesis.

Fitness_threshold: A threshold specifying the termination criterion.

p: The number of hypotheses to be included in the population.

r: The fraction of the population to be replaced by Crossover at each step.

m: The mutation rate.

- *Initialize population:* $P \leftarrow$ Generate p hypotheses at random
- *Evaluate:* For each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$ do

Create a new generation, P_S*:*

1. *Select:* Probabilistically select $(1 - r)p$ members of P to add to P_S. The probability $\Pr(h_i)$ of selecting hypothesis h_i from P is given by

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P, according to $\Pr(h_i)$ given above. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
3. *Mutate:* Choose m percent of the members of P_s with uniform probability. For each, invert one randomly selected bit in its representation.
4. *Update:* $P \leftarrow P_s$.
5. *Evaluate:* for each h in P, compute $Fitness(h)$

- Return the hypothesis from P that has the highest fitness.

Representing Hypothesis

Hypotheses Representation:

- Encoded as **bit strings**, symbolic expressions, or other structures depending on the problem.
- Example:
  - For rule sets: bit strings encoding rules.
  - For neural networks: parameters like topology and weights.
- **Adaptability**: Flexible to represent various types of problems.

Example 1: Rule for a Boolean function.

- Target: $f(x1,x2,x3)=x1 \wedge \neg x2$
- Bit string representation: 101, where:
  - x1=1(True), x2=0(False), x3=1(Don't care).

Genetic Operators

Core Operators:

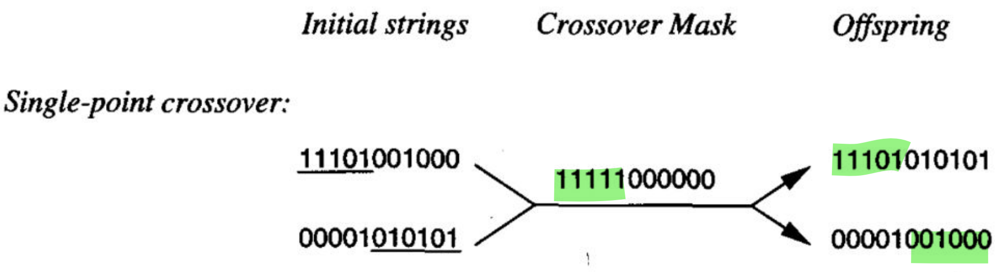
1. **Crossover (Recombination)**:
   - Combines parts of two parent hypotheses to create offspring.
   - Mimics biological reproduction.
   - Example: Split and recombine bit strings at a crossover point.
2. **Mutation**:
   - Randomly alters one or more bits in the hypothesis.
   - Ensures diversity and prevents premature convergence.
   - Example: Flipping a 0 to 1 or vice versa in a bit string.

Role: These operators introduce variability and explore the hypothesis space.

Genetic Operators

Single-Point Crossover

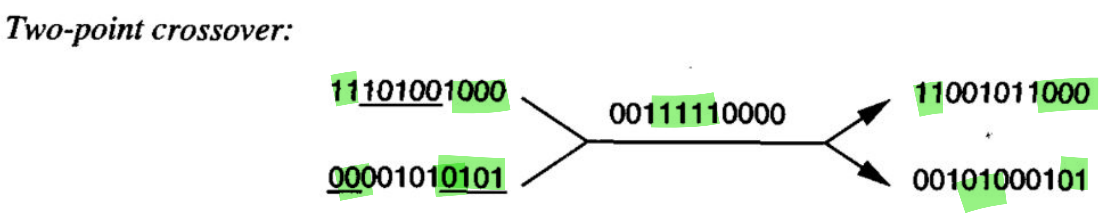
- **Description**: A single crossover point is selected, and the offspring inherit the genes from the parents up to the crossover point, then swap the remaining genes.



Genetic Operators

Two-Point Crossover

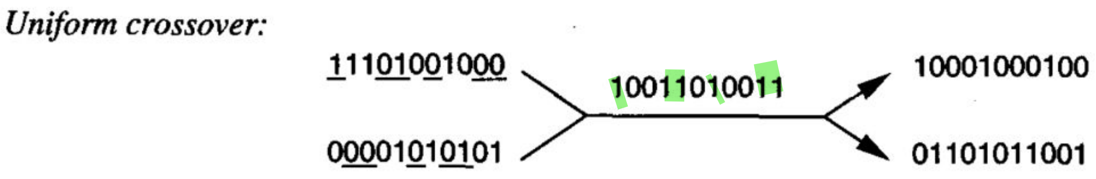
- **Description**: Two crossover points are selected. The genes between these points are swapped between the two parents.



Genetic Operators

Uniform Crossover

- **Description**: Each gene is selected randomly from one of the parents with equal probability.



Genetic Operators

Point Mutation

- Point mutation is a genetic operator used in Genetic Algorithms (GAs) to introduce diversity by altering a single bit or gene in a chromosome.

Point mutation: 11101001000 ⟶ 11101011000

How It Works:

1. **Selection**: A single individual (bit string) from the population is chosen for mutation.
2. **Alteration**:
   - A specific position (gene) in the chromosome is randomly selected.
   - The value of the selected position is flipped or changed.

Genetic Operators

Crossover Example:

- Parent 1: 11001
- Parent 2: 10111
- Single-point crossover at position 3:
  - Offspring 1: 11011
  - Offspring 2: 10101

Mutation Example:

- Original string: 11001
- Mutated string (flipping 3rd bit): 11101

Fitness Function

Purpose:

- Quantifies how well a hypothesis performs for the task.

- **Examples**:
  - **Function approximation**: Accuracy over training data.
  - **Game strategy**: Wins against other hypotheses.
  - **Optimization tasks**: Error minimization or performance metrics.
- **Dynamic Adaptability**: Fitness function adapts to the problem at hand.

Fitness Function

Measures the performance of a hypothesis.

Example 1: Boolean function learning.

- Hypothesis: 101
- Training set:
  - Input: x_1=1, x_2=0, x_3=1, Output: True.
  - Input: x_1=1, x_2=1, x_3=0, Output: False.
- Fitness: Fraction of correctly classified examples.

Example 2: Maximizing Ones (a toy problem).

- Target: Maximize the number of 1s in the string.
- Bit string: 11001 → Fitness = 3 (3 ones).

Fitness Function

Measures the performance of a hypothesis.

Example 1: Boolean function learning.

- Hypothesis: 101
- Training set:
  - Input: x_1=1, x_2=0, x_3=1, Output: True.
  - Input: x_1=1, x_2=1, x_3=0, Output: False.
- Fitness: Fraction of correctly classified examples.

Example 2: Maximizing Ones (a toy problem).

- Target: Maximize the number of 1s in the string.
- Bit string: 11001 → Fitness = 3 (3 ones).

Selection

Mechanism:

- Hypotheses are **evaluated and ranked** based on their fitness.
- Probabilistic selection ensures higher fitness hypotheses are more likely to contribute to the next generation.
- Techniques:
  - **Roulette Wheel Selection**: Probability proportional to fitness.
  - **Tournament Selection**: Compete in small groups; best advances.
- **Goal**: Retain high-quality hypotheses and evolve them further.

Selection

Example: Roulette Wheel Selection

- Population:
  - Hypothesis 1 (11001): Fitness = 3
  - Hypothesis 2 (10111): Fitness = 4
  - Hypothesis 3 (11111): Fitness = 5
- Total fitness = 12.
- Probability of selection:
  - Hypothesis 1: 3/12=25%
  - Hypothesis 2: 4/12=33.3%
  - Hypothesis 3: 5/12=41.7%

Illustrative Example

Problem Statement:

Determine when a person should decide to play a sport based on environmental and personal factors.

Factors Considered:

1. Weather: Sunny (1) or Rainy (0).
2. Temperature: Warm (1) or Cold (0).
3. Fitness Level: High (1) or Low (0).
4. Time Availability: Ample (1) or Limited (0).

Illustrative Example

Bit String Representation:

Each hypothesis represents a decision rule in a **bit string** format:
Weather | Temperature | Fitness | Time → Decision

- Example:
  - 1101 → 1
  - "If the weather is sunny (1), temperature is warm (1), fitness level is high (0), and time is ample (1), then play sports (1)."

Initial Population:

Randomly generated hypotheses:

- Hypothesis 1: 1110 → 1
- Hypothesis 2: 1011 → 0
- Hypothesis 3: 0111 → 1
- Hypothesis 4: 0000 → 0

Illustrative Example

Fitness Function:

Evaluate hypotheses based on their accuracy in predicting decisions on a dataset of past scenarios:

- Training Data:
  - Input: 1011 → Play = 0.
  - Input: 1101 → Play = 1.
  - Input: 0111 → Play = 1.
- Fitness Calculation:
  - Hypothesis 1: Matches 2/3 scenarios → Fitness = 2/3
  - Hypothesis 2: Matches 1/3 scenarios → Fitness = 1/3.

Illustrative Example

Genetic Operators:

1. **Crossover**: Combine parts of two hypotheses to create offspring.
   - Parent 1: 1110 → 1
   - Parent 2: 1011 → 0
   - Crossover point: Between bits 2 and 3.
     - Offspring 1: 1111 → 1
     - Offspring 2: 1010 → 0
2. **Mutation**: Flip a random bit in a hypothesis.
   - Hypothesis: 1110 → 1
   - After mutation: 1100 → 1

Illustrative Example

Selection: Probabilistically select hypotheses based on fitness:

- Hypothesis 1: Higher chance due to better fitness.
- Hypothesis 2: Lower chance due to poor fitness.

Evolved Rule: After multiple generations, the algorithm might converge to an optimized hypothesis:

- Final Hypothesis: $1101 \rightarrow 1$.
- Meaning: "Play sports if the weather is sunny, temperature is warm, fitness level is low, and time is ample."

Questions ?