Machine, Data & Learning

Project | Genetic Algorithm

Team 30 | Tribrid

- Nitin Chandak (2019101024)
- Ayush Sharma (2019101004)

Summary

Genetic Algorithm is totally inspired by Charles Darwin Theory of Natural Selection.

The genetic information in a genetic algorithm corresponds to a DNA chain in an individual. The DNA chain describes the solution or more correctly the parameters of the solution. The genetic information is built by chromosomes and is in general coded in a binary form (but can also be in another form of data). A solution is generally called an individual and the set of individuals that are present in a generation of a genetic algorithm are referred to as the population.

Implementation

- Setup the initial Population.
- Calculate Fitness of initial population.
- Until convergence repeat the evolution process as follows:
  - Calculate Selection Percentage of all individual in population.
  - Select parents from the population based on their selection percentage.
  - Perform Crossover to generate children.
  - Perform Mutation on new population i.e. children.
  - Calculate Fitness for new mutated population.

```
def GA():
    '''
    Main function to be called to run
    the implemented genetic algorithm.
    '''
    generation_loop = GENERATION_LOOP
    pop = []
    pop_fit = []

    if not is_valid_file(TRACE):
        pop = create_start_population()
```

```
        pop_fit = create_fitness(pop)
        data = {"Trace": []}
        write_file(TRACE,data)
    else:
        pop, pop_fit = data_breach()

    for generation in range(generation_loop):

        selected_pop = select_population(pop_fit)
        crossed_pop, cross_detail = simulate_cross_over(selected_pop)
        mutated_pop = mutation(crossed_pop)
        new_pop = mutated_pop.tolist()
        new_pop_fit = create_fitness(new_pop)

        save_stuff(pop_fit,selected_pop,crossed_pop,cross_detail,mutated_pop,new_pop_fit)

        pop = new_pop
        pop_fit = new_pop_fit
```

Iteration Diagrams