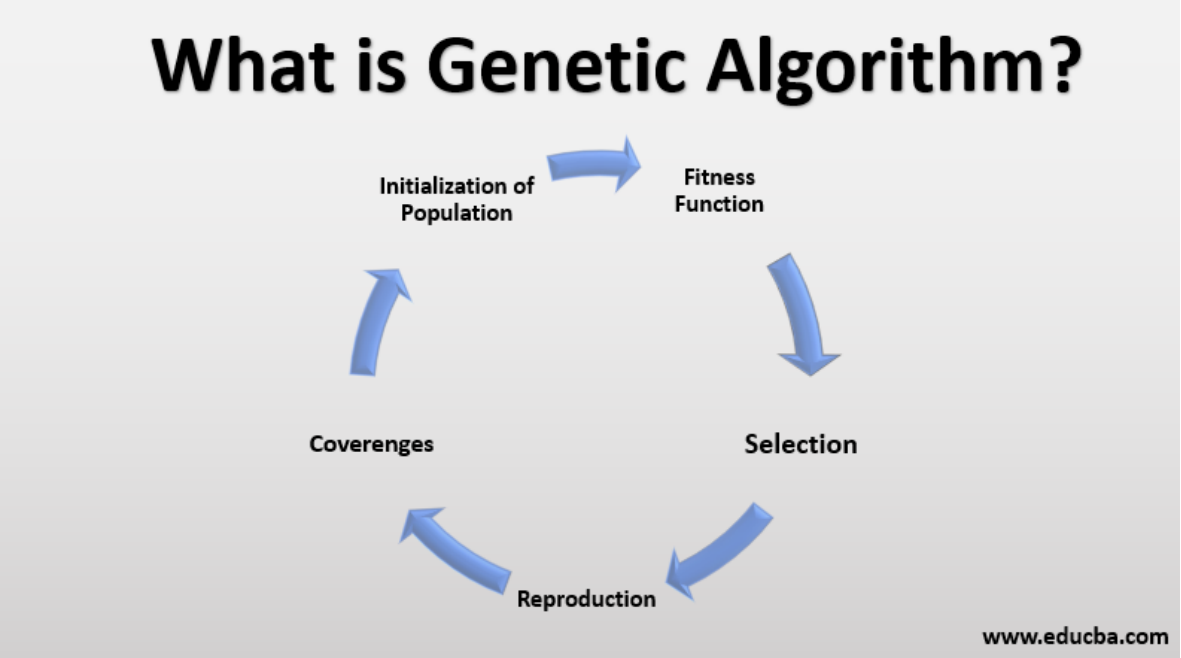
TEAM 88

REPORT

MDL ASSIGNMENT-3

The base genetic algorithm contains

- Fitness function
- Parents selection
- Cross over function
- Mutation Function



Summary:-

My genetic algorithm follows the same basic genetic algorithm path with some modifications in the steps

The Modifications

In the fitness function the combined error of the validation and the test set with 0.6 and 0.4 probability respectively.
Combined error=(0.4)*test error + (0.6)* validation error
In the selection of the mating pools for the new offspring the vectors with the less fitness values are selected and returned
In the cross over function the cross over is done using uniform cross over in which Each gene (bit) is selected randomly from one of the corresponding vector of the parent Vectors.Use tossing of a coin as an example technique.
In Mutation the index of the vector is randomly selected and 10% to -10% is added randomly to that selected element above and adding a random value it is also made sure that every element not to be above 10 or below -10.

Fitness function:

- The Fitness function returns the combined error for both the test and validation set.
  - The combined error is (0.4)*test error + (0.6)*validation error.
- In the fitness function it submits the request for the train and validation error and also submits the vectors and returns the combined error in a list.

```
def cal_pop_fitness(pop):
    error=[]
    for x in range(len(pop)):
        err = get_errors('fiP6JijQ2GF3yWtdOuTfzQkibcpBC4azI3ad0cJaLRqW7MyW3m',list(pop[x]))
        submit_status = submit('fiP6JijQ2GF3yWtdOuTfzQkibcpBC4azI3ad0cJaLRqW7MyW3m', list(pop[x]))
        combined=(0.4)* err[0]+(0.6)* err[1]
        error.append(combined)
        # print(err[1])
    return error
```

Parents Selection:-

- The parents are selected in such a way according to the fitness values which are returned from the fitness function above.
- The first 4 best from the population are selected which has less fitness values and returned to the main function for cross over of the parents for offspring's.

```
def select_mating_pool(pop, fitness, num_parents):
    parents = numpy.empty((num_parents, pop.shape[1]))
    for parent_num in range(num_parents):
        max_fitness_idx = numpy.where(fitness == numpy.min(fitness))
        max_fitness_idx = max_fitness_idx[0][0]
        parents[parent_num, :] = pop[max_fitness_idx, :]
        fitness[max_fitness_idx] = infinity
    return parents
```

Cross over:-

- Uniform cross over is used in the genetic algorithm
- By doing between first two and second third and so on... by taking one of two parent elements in making offspring's

```
def crossover(parents,offspring_size):
    offspring=numpy.empty(offspring_size)
    for k in range(offspring_size[0]):
        for l in range(offspring_size[1]):
            value=random.choice([0,1])
            if value==0:
                value1=k%parents.shape[0]
                offspring[k][l] = parents[value1][l]
            else:
                value1=(k+1)%parents.shape[0]
                offspring[k][l] = parents[value1][l]
    return offspring
```

Mutation:-

- It is done by choosing the index randomly for each vector and then replacing it with the value in between 1.1 x and 0.9 x. Like adding between 10% to -10% to the element of the vector.
- The vectors are returned to the main function after mutation.

Hyper Parameters:-

Pool Size:- 8
Vector Size:-11
Splitting point is not mentioned as uniform cross over is used

The first input 8 vectors are selected by first Brute force method to the given overfit.txt mentioned in the assignment. By decreasing the validation and test error for the over fit vector given it is used in the genetic algorithm process.
The other 8 vectors are generated by mutating the over fit vector given.

Tricks:-

Brute force is done for the initial vectors and then the vector is used for the genetic algorithm and then the best vector from that is used to run the genetic algorithm again and again

output of the code:-

The output returns all the vectors for each generation 10 times and outputs the final best vector