Savitribai Phule Pune University
Modern Education Society's College of Engineering, Pune
19, Bund Garden, V.K. Joag Path, Pune – 411001.

ACCREDITED BY NAAC WITH "A" GRADE (CGPA – 3.13)

DEPARTMENT OF COMPUTER ENGINEERING

A REPORT

ON

Genetic Algorithm Optimization on Mushroom Classification

B.E. (COMPUTER)

SUBMITTED BY

Mr. Yogen Ghodke (71818291L)
Mr. Sudesh Pawar (71818502B)
Miss. Shravani Kanade (71818357G)

UNDER THE GUIDANCE OF

Prof. Amol S. Kamble

(Academic Year: 2020-2021)

Savitribai Phule Pune University
Modern Education Society's College of Engineering, Pune
19, Bund Garden, V.K. Joag Path, Pune – 411001.

ACCREDITED BY NAAC WITH "A" GRADE (CGPA – 3.13)

DEPARTMENT OF COMPUTER ENGINEERING

Certificate

This is to certify that project entitled

Genetic Algorithm Optimization on Mushroom Classification

has been completed by

Mr. Yogen Ghodke (PRN. 71818291L)
Mr. Sudesh Pawar (PRN. 71818502B)
Miss. Shravani Kanade (PRN. 71818357G)

of BE COMP I in the Semester - II of academic year 2020-2021 in partial fulfillment of the Fourth Year of Bachelor degree in "Computer Engineering" as prescribed by the Savitribai Phule Pune University.

Prof. Amol S. Kamble
Project Guide

(Dr.(Mrs.) N. F. Shaikh)
H.O.D

Place: MESCOE, Pune.
Date: /05/2021

ACKNOWLEDGEMENT

It gives us great pleasure and satisfaction in presenting this mini project on "Genetic Algorithm Optimization on Mushroom Classification".

We would like to express my deep sense of gratitude towards all faculty for their support and advice during the development.

We have furthermore to thank Computer Department HOD **Dr.(Mrs.) N. F. Shaikh** *for her pearls of wisdom and our Project Guide* **Prof. Amol S. Kamble** *to encourage us to go ahead and for continuous guidance. His incisive and objective guidance and timely advice encouraged us with a constant flow of energy to continue the work. Finally, we must say that no height is ever achieved without some sacrifices made at some end and it us here we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period.*

We would like to thank all those, who have directly or indirectly helped us for the completion of the work during this mini project.

Yogen Ghodke (71818291L)
Sudesh Pawar (71818502B)
Shravani Kanade (71818357G)
B.E. Computer

Contents

List of Figures

Abstract

One of the most advanced algorithms in the field of computer science is Genetic Algorithm inspired by the Human genetic process of propogating genes from current generation to next.This is usually used for optimizations in algorithmic performance and is heuristic in nature and can be used at various places. The genetic algorithm is a search-based optimization technique.It is frequently used to find the optimal or nearest optimal solution. It was introduced by John Holland. It is based on Darwin's Natural Selection Theory. In this theory, he defined natural selection as the "principle by which each slight variation [of a trait], if useful, is preserved".

The concept was simple but powerful: individuals best adapted to their environments are more likely to survive and reproduce. Sometimes this theory is described as "survival of the fittest". Those who are fittest than others have the chance to survive in this evolution. The genetic algorithm is all about this. It mimics the process of natural selection to find the best solution. The genetic algorithm is a random-based classical evolutionary algorithm. It is a slow gradual process that works by making changes to the making slight and slow changes. Also, Genetic Algorithm makes slight changes to its solutions slowly until getting the best solution

Keywords-*Genetic Algorithm, Darwin's Natural Selection theory, Genetic Process*

Chapter 1

INTRODUCTION

1.1 Genetic Algorithm

Genetic algorithm (GA) as a computational intelligence method is a search technique used in computer science to find approximate solutions to combinatorial optimization problems. The genetic algorithms are more appropriately said to be an optimization technique based on natural evolution . They include the survival of the fittest idea algorithm. The idea is to first guess the solutions and then combining the fittest solution to create a new generation of solutions which should be better than the previous generation. We also include a random mutation element to account for the occasional mishap The genetic algorithm process consists of the following:

- **Encoding:** A suitable encoding is found for the solution to our problem so that each possible solution has unique encoding and the encoding is some form of a string.
- **Evaluation:** The initial population is then selected, usually at random though alternative techniques using heuristics have also been pro posed. The fitness of each individual in the population is then computed that is, how well the individual fits the problem and whether it is near the optimum compared to the other individuals in the population.
- **Crossover:** The fitness is used to find the individual's probability of crossover. Crossover is where the two individuals are recombined to create new individuals which are copied into the new generation.
- **Mutation:** Next mutation occurs. Some individuals are chosen randomly to be mutated and then a mutation point is randomly chosen. The character in the corresponding position of the string is changed.
- **Decoding:** Once this is done, a new generation has been formed and the process is repeated until some stopping criteria has been reached. At this point the individuals which is closest to the optimum is decoded and the process is complete.

1.2 Motivation

Most of the state-of-the-art algorithms have similar performance measures in the field of Machine Learning. The deciding factor depends on two major pillars :
Data Pre-processing and Optimizations done on the algorithms.

Due to the widescale availability of ETL Tools and significant emphasis on clean data these days, the first factor is more or less same across different approaches in terms of classification.Therefore, the major factor is the Optimization performed on the model to tune its performance.
Therefore, our mini project aims to use Genetic Algorithm Optimization to avail maximum benefits and ensure the best performance of our model.

Chapter 2

Optimization

2.1 Problem Statement

This project intends to develop a neural network to accurately classify mushrooms based on the features present in dataset and then optimize it using Bioalgorithm such as Genetic Algorithm.

2.2 Optimization using Genetic Algorithm

GAs are a heuristic solution-search or optimisation technique, originally motivated by the Darwinian principle of evolution through (genetic) selection. A GA uses a highly abstract version of evolutionary processes to evolve solutions to given problems. Each GA operates on a population of artificial chromosomes. These are strings in a finite alphabet (usually binary). Each chromosome represents a solution to a problem and has a fitness, a real number which is a measure of how good a solution it is to the particular problem.

Starting with a randomly generated population of chromosomes, a GA carries out a process of fitness-based selection and recombination to produce a successor population, the next generation. During recombination, parent chromosomes are selected and their genetic material is recombined to produce child chromosomes. These then pass into the successor population. As this process is iterated, a sequence of successive generations evolves and the average fitness of the chromosomes tends to increase until some stopping criterion is reached. In this way, a GA "evolves" a best solution to a given problem.

Chapter 3

RELATED WORK

Metaheuristic algorithms are used to solve real-life complex problems arising from different fields such as economics, engineering, politics, management, and engineering. Intensification and diversification are the key elements of metaheuristic algorithm. The proper balance between these elements are required to solve the real-life problem in an effective manner. Most of metaheuristic algorithms are inspired from biological evolution process, swarm behavior, and physics' law. These algorithms are broadly classified into two categories namely single solution and population based metaheuristic algorithm. Single-solution based metaheuristic algorithms utilize single candidate solution and improve this solution by using local search. However, the solution obtained from single-solution based metaheuristics may stuck in local optima. The well-known singlesolution based metaheuristics are simulated annealing, tabu search (TS), microcanonical annealing (MA), and guided local search (GLS). Population-based metaheuristics utilizes multiple candidate solutions during the search process. These metaheuristics maintain the diversity in population and avoid the solutions are being stuck in local optima. Some of well-known population-based metaheuristic algorithms are genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), spotted hyena optimizer (SHO), emperor penguin optimizer (EPO), and seagull optimization (SOA).

Genetic Algorithms can be easily hybridized with other optimization methods for improving their performance such as image denoising methods, chemical reaction optimization, and many more. The main advantages of hybridized GA with other methods are better solution quality, better efficiency, guarantee of feasible solutions, and optimized control parameters. It is observed from literature that the sampling capability of GAs is greatly affected from population size. To resolve this problem, local search algorithms such as memetic algorithm, Baldwinian, Lamarckian, and local search have been integrated with GAs. This integration provides proper balance between intensification and diversification. Another problem in GA is parameter setting. Finding appropriate control parameters is a tedious task. The other metaheuristic techniques can be used with GA to resolve this problem. Hybrid GAs have been used to solve the issues mentioned in the preceding subsections.

GAs have been integrated with local search algorithms to reduce the genetic drift. The explicit refinement operator was introduced in local search for producing better solutions. El-Mihoub et al. established the effect of probability of local search on the

population size of GA. Espinoza et al. investigated the effect of local search for reducing the population size of GA. Different search algorithms have been integrated with GAs for solving real-life applications.

GA shows the superior performance for solving the scheduling problems such as job-shop scheduling (JSS), integrated process planning and scheduling (IPPS), etc. To improve the performance in the above-mentioned areas of scheduling, researchers developed various genetic representation, genetic operators, and hybridized GA with other methods.

Due to development in multimedia applications, images, videos and audios are transferred from one place to another over Internet. It has been found in literature that the images are more error prone during the transmission. Therefore, image protection techniques such as encryption, watermarking and cryptography are required. The classical image encryption techniques require the input parameters for encryption. The wrong selection of input parameters will generate inadequate encryption results. GA and its variants have been used to select the appropriate control parameters. Kaur and Kumar developed a multi-objective genetic algorithm to optimize the control parameters of chaotic map. The secret key was generated using beta chaotic map. The generated key was use to encrypt the image. Parallel GAs were also used to encrypt the image.

Genetic algorithms have been applied in medical imaging such as edge detection in MRI and pulmonary nodules detection in CT scan images. A template matching technique with GA for detecting nodules in CT images. Kavitha and Chellamuthu used GA based region growing method for detecting the brain tumor. GAs have been applied on medical prediction problems captured from pathological subjects. Sari and Tuna used GA used to solve issues arises in biomechanics. It is used to predict pathologies during examination. Ghosh and Bhattachrya implemented sequential GA with cellular automata for modelling the coronavirus disease 19 (COVID-19) data. GAs can be applied in parallel mode to find rules in biological datasets. The authors proposed a parallel GA that runs by dividing the process into small sub-generations and evaluating the fitness of each individual solution in parallel. Genetic algorithms are used in medicine and other related fields. Koh et al. proposed a genetic algorithm based method for evaluation of adverse effects of a given drug.

Chapter 4

METHODOLOGY

A simple GA works by randomly generating an initial population of strings, which is referred as gene pool and then applying (possibly three) operators to create new, and hopefully, better populations as successive generations. The first operator is reproduction where strings are copied to the next generation with some probability based on their objective function value. The second operator is crossover where randomly selected pairs of strings are mated, creating new strings. The third operator, mutation, is the occasional random alteration of the value at a string position. The crossover operator together with reproduction is the most powerful process in the GA search. Mutation diversifies the search space and protects from loss of genetic material that can be caused by reproduction and crossover. So, the probability of applying mutation is set very low, whereas the probability of crossover is set very high.

4.1 Steps of Algorithms

- Randomly create the initial population of individual string of the given TSP problem and create a matrix representation of the cost of the path between two cities.
- Assign the fitness to each chromosome in the population using fitness criteria measure.

  $$F(x) = 1/x$$

  where, x represents the total cost of the string. The selection criteria depends upon the value of string if it is close to some threshold value.
- Create new off - spring population from two existing chromosomes in the parent population by applying crossover operator.
- Mutate the resultant off-springs if required. NOTE: After the crossover offspring population has the fitness value higher than the parents.
- Repeat step 3 and 4 until we get an optimal solution to the problem.

A genetic algorithm is considered completed if a certain number of iterations are passed (it is desirable to limit the number of iterations, since the genetic algorithm works on the method of trial and error, which is quite a long process), or if the

satisfactory value of the fitness function was obtained. Generally, the genetic algorithm solves the problem of maximizing or minimizing and the adequacy of each decision(chromosome) is assessed using the fitness function. Genetic algorithm works according by the following principle:

- **Initializing:** Establishing fitness function. Forming the initial population. Classically, the initial population creating by random filling of genes in the chromosomes. However, to increase the convergence rate, the initial population can be filling in specific way, there the values can be analysed in advance for exclusion of definitely unsuitable genes.

- **Evaluation of population:** Each of the chromosomes is evaluated by the fitness function. Based on specified requirements, chromosomes acquire a certain value in accordance with the solution of the problem

- **Crossover:** The first significant difference from conventional methods and one of the most important stages of the algorithm. After selection and retrieving the suitable chromosomes to solve the problem, they crossover with each other. Randomly selected chromosomes generate new chromosomes. Crossover occurs based on the selection of a specific position in the two chromosomes and mutual replacement of parts. After filling the required number of chromosomes to create a new population, the algorithm proceeds to the next step.

- **Mutation:** : This is also a step characteristic for GA only. In random order, a random gene can change values to a random one. The main purpose of the mutation is the same as in biology – the introduction of genetic diversity in the population. The main goal of mutation is to obtain solutions that could not be produced with existing genes. This will allow, firstly, to avoid falling into local extremes, since mutation may allow the algorithm to go a completely different path, and secondly, to "dilute" the population in order to avoid a situation where there are only identical chromosomes in the entire population that will not move towards a global solution. After all stages of the genetic algorithm have been completed, it is estimated whether the population has reached the desired accuracy of the solutions, or whether a certain number of populations have been reached. If these conditions have been met, algorithm stops working. Otherwise, the cycle is repeated with new population until the conditions are reached.

Chapter 5

RESULT

Using the genetic algorithm, we were able to optimize the accuracy of the Neural Network Model on the Mushroom Classification data set.

```
In [9]: weights_mat = vector_to_mat(weights_vector, weights_mat)
        best_weights = weights_mat [0, :]
        acc, predictions = predict(x_train, y_train, best_weights, sigmoid)
        print("Accuracy of the Initial solution is : ", acc)

        Accuracy of the Initial solution is :  0.5189940204009849
```

Figure 5.1: Accuracy of Model Before Genetic Algorithm

```
In [12]: weights_mat = vector_to_mat(weights_vector, weights_mat)
         best_weights = weights_mat [0, :]
         acc, predictions = predict(x_train, y_train, best_weights, sigmoid)
         print("Accuracy of the best solution is : ", acc)

         Accuracy of the best solution is :  0.9154062609919099
```

Figure 5.2: Accuracy of Model After Using Genetic Algorithm

Chapter 6

CONCLUSION

Genetic algorithm appear to find good solutions for the Optimisation purpose,however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. We have used UCI repository's dataset. UCI stands for University of California, Irvine. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. It is used by students, educators, and researchers all over the world as a primary source of machine learning data sets. As an indication of the impact of the archive, it has been cited over 1000 times.GAs have various advantages which have made them immensely popular. These include:

- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of "good" solutions and not just a single solution.
- Always gets an answer to the problem, which gets better over the time.
- Useful when the search space is very large and there are a large number of parameters involved.