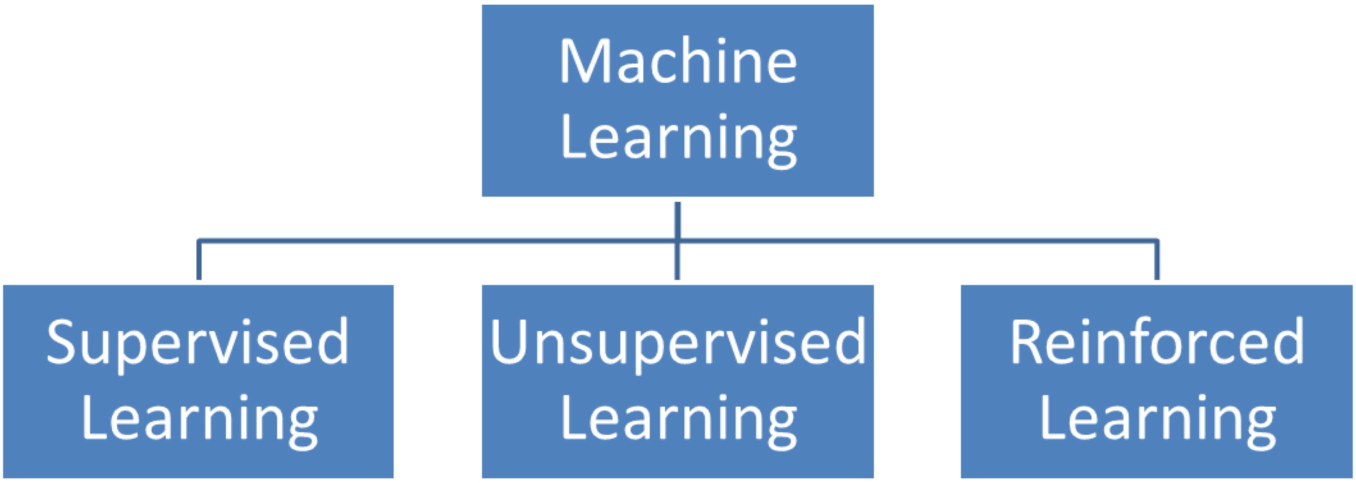
Machine Learning via Genetic Algorithm Demystified for Today's Era

Dr Vivek Venkobarao
Senior Member IEEE
Execom Member, CE,TEMS Bangalore Chapter
vivek.venkobarao@gmail.com

In last couple of years Machine learning and Artificial intelligence is gaining prominence and we keep hearing these terms daily. It offers a great career option today. This promises to solve problems and also benefit companies and individuals by making predictions for helping them make better decisions. The objective of machine learning is to allow the computers learn automatically without human intervention.

There are 3 major learning algorithms to make the computers learn



Supervised Learning: For learning the pattern the data scientist has ground truth ie the expected outcome is always associated with data for learning. The learned network is then compared with the known outcome to check accuracy. These sort of learning is used in handwriting recognition, translation engines etc

Unsupervised Learning: For this sort of learning the data scientist has only specific signature for learning. Usually the network is learned via clustering. User can see applications in recommendation systems like Amazon and Netflix

Reinforced Learning: For reinforcement leaning we choose the learning via actions. The action is associated with a reward. Based on action and reward the network is learned. Typically used for games.

One of the important supervised algorithms is genetic algorithms.

Introduction

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. Genetic algorithm exhibits implicit parallelism and can retain useful redundant information about what is learned from previous searches by its representation in individuals in the population, but GA may lose solutions and substructures due to the disruptive effects of genetic operators and is not easy to regulate GA's convergence.

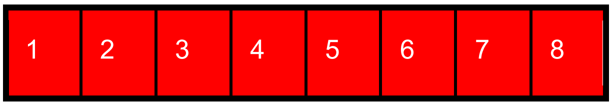
Genetic algorithms, as a member of the board class of evolutionary algorithms, are known as popular methods for global search or optimization. It is well established that it is hard and time consuming for a simple problem to find the best solution in case of complicated search spaces. To solve this problem, the members of population can grow up during their life time. For this purpose, the individuals should look for the best point in a relatively small neighbourhood using local search methods. However, using memetic algorithm has similar problem as side effects. The first one which is considered has the convergence of the population to local optimum points which decreases the efficiency of GA. The Monte Carlo method is combined with GA as a local search and it also diversifies the population when some individuals overlap on a local optimum is done in the past. But it does not perform any parameter optimization for the underlying local search which is another important challenge in memetic algorithm. As it's seen from above discussion GA with local search algorithms doesn't introduce significant difference in efficiency of prediction. However, the efficiency of prediction can be improved by dynamic adaptive techniques. However, another problem arises with cost and computational load when an advanced local search method is used to get a better efficient solution.

Glossary of terms

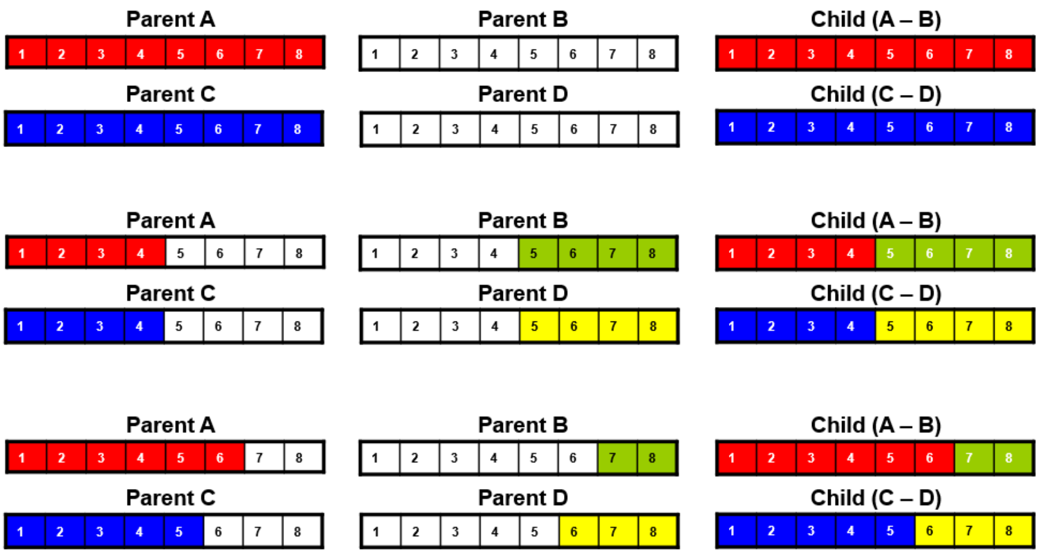
Population: Number of members in the sample space under consideration

Convergence: Tendency of members of the population to be the same.

Chromosome: Normally, in genetic algorithms the bit string which represents the individual as shown below.

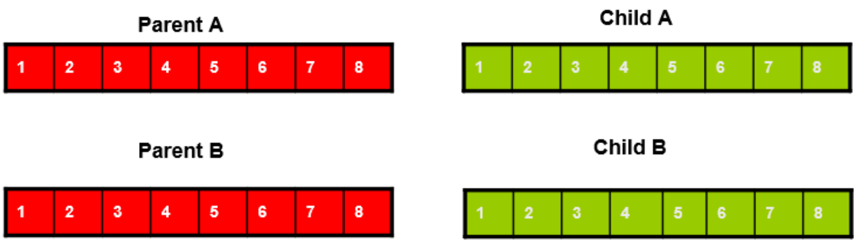


Crossover: Creating a new individual's representation from parts of its parent's representations. Illustration of crossover



Generation: When the children of one population replace their parents in that population.

Mutation: Arbitrary change to representation, often at random. Illustration of mutation



Fitness Function: A process which evaluates a member of a population and gives it a score or fitness.

Termination The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation).

Most of the times the termination occurs and still we don't reach the optimum. The ML scientist would use Simulated Annealing to reduce the energy of the individuals.

Five phases are considered in a genetic algorithm.
1.Initial population
2.Fitness function
3.Selection
4.Crossover
5.Mutation

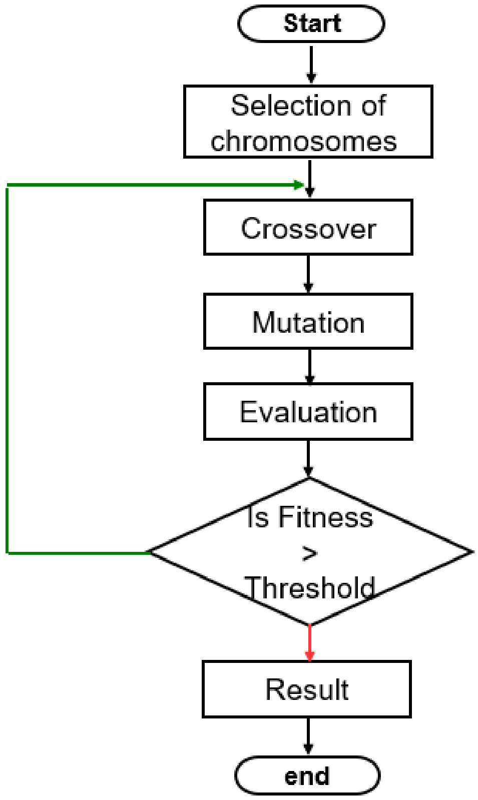


Figure 1 GA basic flow

Description of Genetic Algorithm in nutshell

An initial population is created containing a predefined number of individuals, each represented by a genetic string.

Everyone has an associated fitness measure, typically representing an objective value. The concept that fittest individuals in a population will produce fitter offspring is then implemented to reproduce the next population.

Selected individuals are chosen for reproduction at each generation, with an appropriate mutation factor to randomly modify the genes of an individual, to develop the new population.
The result is another set of individuals based on the original subjects leading to subsequent populations with better individual fitness. Those with lower fitness will get discarded from the population.

When to stop the Genetic Algorithm

This iterative process continues until one of the possible termination criteria is met:

if a known optimal or acceptable solution level is attained
if a maximum number of generations have been performed
if a given number of generations without fitness improvement occur

Basic concepts one must remember while solving Genetic Algorithm

The selection procedure randomly selects individuals of the current population for development of the next generation. The selection is based on the probability factor.

The crossover procedure takes two selected individuals and combines them about a crossover point thereby creating two new individuals.

The mutation procedure randomly modifies the genes of an individual subject to a small mutation factor, introducing further randomness into the population.

The evaluation procedure measures the fitness of each individual solution in the population and assigns it a relative value based on the defining optimization (or search) criteria

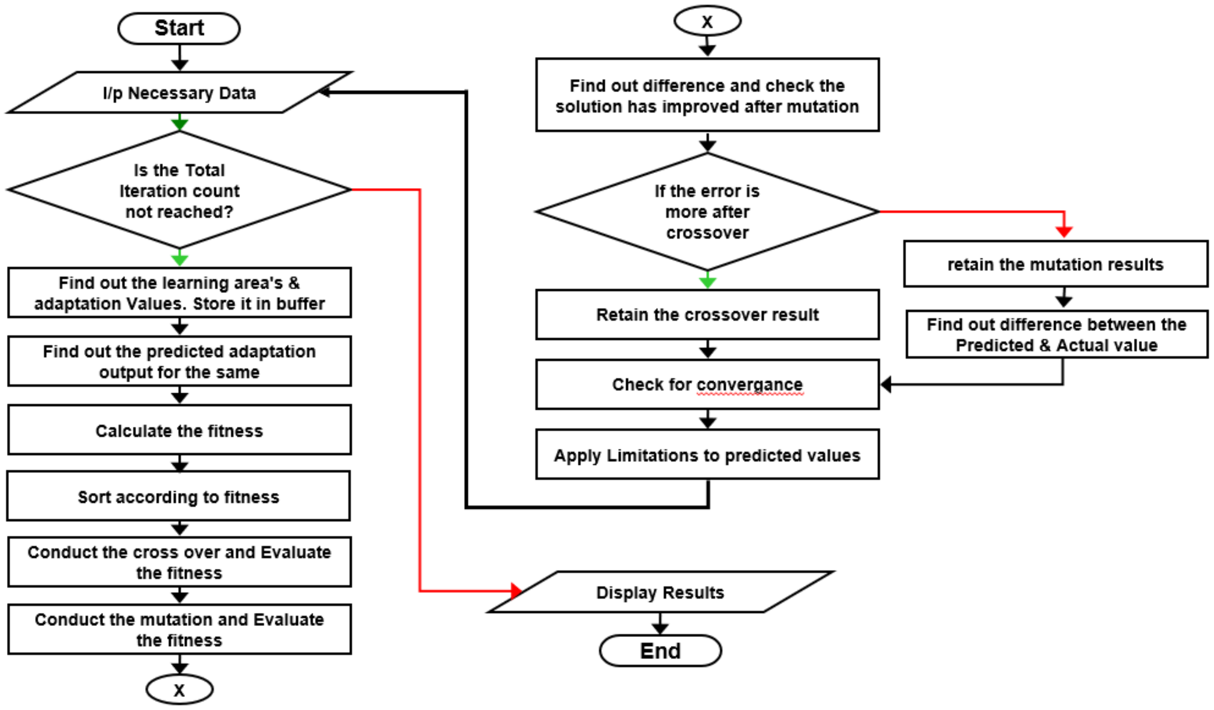


Figure 2: Detailed GA algorithm

GA Tradeoffs

Population size selection is probably the most important parameter, reflecting the size and complexity of the problem.

However, the trade-off between extra computational effort with respect to increased population size is a problem specific decision, as doubling the population size will approximately double the solution time for the same number of generations.

Other parameters include the maximum number of generations to be performed, a crossover probability, a mutation probability, a selection method and possibly an elitist strategy, where the best is retained in the next generation's population.

Advantages of Genetic Algorithm

When minimal region is identified during the search process, the GA method is not efficient, even sometimes impossible, in reaching its minimum. This is because GA is opportunistic not deterministic.

Dynamic adaptive methods are very efficient in this regard and can guarantee a local minimum, but not a global one.

Conclusion:

It requires a large computational time

There is always a tradeoff between the computational time and population.

A drawback of this algorithm is that a solution is "better" only in comparison to other, presently known solutions for solving nonlinear equations. A better solution may be obtained by using adaptive estimators like Kalman Estimators, Neural Networks etc

It can also seen that the error decreases substantially with the increase in mutation probability. But nature does not allow higher mutation probability with maximum probability can be 5%.

It is also seen that the error decreases marginally when there is a crossover. Typically, nature allows up to 90 to 95% of crossover.

References

https://data-flair.training/blogs/

https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3?gi=eb9f41518b34

2008E19407 IN Vivek Venkobarao - Hybridizing Genetic Algorithms with simulated annealing and Dynamic adaptive methods for global optimization – Patent Application published

IEEE Talk Vivek Venkobarao - https://www.youtube.com/watch?v=7EMq4IqjS-k

About the author

Vivek Venkobarao has 13 years' experience in leading Automotive tier 1 supplier in Bangalore India. He has a Ph.D in Electrical Engineering and Certificate(I&E) Stanford University California. Has 20 papers published in International journals and conferences as first author. He is also Co-Author "Handbook of Research on Emerging Technologies for Electrical Power Planning, Analysis, and Optimization" from international publisher. He is also reviewer in various SAE and IEEE conferences. He has 10 patents. His research interests include Control Systems, Mathematical model of real time systems, Neural Networks, Fuzzy logic, Bio Inspired Computing.

The 10 Algorithms Machine Learning Engineers Need to Know

It is no doubt that the sub-field of machine learning / artificial intelligence has increasingly gained more popularity in the past couple of years. As Big Data is the hottest trend in the tech industry at the moment, machine learning is incredibly powerful to make predictions or calculated suggestions based on large amounts of data. Some of the most common examples of machine learning are Netflix's algorithms to make movie suggestions based on movies you have watched in the past or Amazon's algorithms that recommend books based on books you have bought before.

So if you want to learn more about machine learning, how do you start? For me, my first introduction is when I took an Artificial Intelligence class when I was studying abroad in Copenhagen. My lecturer is a full-time Applied Math and CS professor at the Technical University of Denmark, in which his research areas are logic and artificial, focusing primarily on the use of logic to model human-like planning, reasoning and problem solving. The class was a mix of discussion of theory/core concepts and hands-on problem solving. The textbook that we used is one of the AI classics: Peter Norvig's Artificial Intelligence—A Modern Approach, in which we covered major topics including intelligent agents, problem-solving by searching, adversarial search, probability theory, multi-agent systems, social AI, philosophy/ethics/future of AI. At the end of the class, in a team of 3, we implemented simple search-based agents solving transportation tasks in a virtual environment as a programming project.

I have learned a tremendous amount of knowledge thanks to that class, and decided to keep learning about this specialized topic. In the last few weeks, I have been multiple tech talks in San Francisco on deep learning, neural networks, data architecture—and a Machine Learning conference with a lot of well-known professionals in the field. Most importantly, I enrolled in Udacity's Intro to Machine Learning online course in the beginning of June and has just finished it a few days ago. In this post, I want to share some of the most common machine learning algorithms that I learned from the course.

Machine learning algorithms can be divided into 3 broad categories—supervised learning, unsupervised learning, and reinforcement learning.Supervised learning is useful in cases where a property (label) is available for a certain dataset (training set), but is missing and needs to be predicted for other instances. Unsupervised learning is useful in cases where the challenge is to discover implicit relationships in a given unlabeled dataset (items are not pre-assigned). Reinforcement learning falls between these 2 extremes—there is some form of feedback available for each predictive step or action, but no precise label or error message. Since this is an intro class, I didn't learn about reinforcement learning, but I hope that 10 algorithms on supervised and unsupervised learning will be enough to keep you interested.
https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html

Machine Learning Algorithms Tutorial: This Machine Learning Algorithms Tutorial video will help you learn you what is Machine Learning, various Machine Learning problems and the algorithms, key Machine Learning algorithms with simple examples and use cases implemented in Python.
https://www.youtube.com/watch?v=I7NrVwm3apg