Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 3, Issue 6; October-December, 2016, pp. 481-485

http://www.krishisanskriti.org/Publication.html

A Comparative Study in Soft Computing Approaches of (ANN & GA) Optimization

Kumar Shekhar Singh

Dept. of Commuter Science & Applications, Gaya College, Gaya, Bihar, India

Abstract—*Soft Computing refers to a variety of computational techniques which are applied in applications where the conventional methods of optimization are trapped in local minimums or they provide unsatisfactory solutions. Using the Soft Computing techniques, we find more precise and complete solutions in application areas such as Inventory Optimization, Supply Chain Management, Humanities, Medicine etc. The basic idea of Soft Computing in contrast of Hard computing is that it is more tolerant to imprecision, approximation, uncertainty and partial truth. The contemporary scenario in field of soft computing involves some important techniques such as ---*

(i) Fuzzy Logic (FL)

(ii) Genetic Algorithm (GA)

(iii)Artificial Neural Net Works (ANN)

In the proposed study I present a comparative modeling and analysis of two such computing techniques - GA and ANN for optimization.

Genetic Algorithm for Optimization

Genetic Algorithm is a powerful adaptive approach which applies the principle of genetics i. e. natural evolution of species, which is considered to be a natural from of optimization. It can be used in variety of optimization and search problems.

Optimization (Maximization or Minimization) using GA involves selection of best element with regard to some criteria from a set of available alternatives.

It involves selecting input values from a set (population) and then progressively the optimum value.

Finding the best available values of some objective function (called fitness function) given a defined domain

$f(x) : R^n = R$

Preference of GA over other Optimization methods such as linear programming, heuristics etc. is that it does not degenerate easily even if there are variations in the inputs due to justified factors.

Artificial Neural Net works for Optimization

ANN's are information systems that attempt to mimic the functioning of human brain. It is a computational functioning device modeled on human brain to perform variety of tasks such as pattern matching optimization, classification etc. These tasks are difficult for traditional optimization methods.

In the cases for ANN's the optimization function is called activation function.

$Y = f(Y_{in})$

Output = Function (net input)

The above function is similar to the calculation of output of pure linear equation.

$y = mx$

The conclusion of Abstract of the study finds two linear models of optimization.

1. INTRODUCTION

In 1975 Holland [1] put forward the idea of GA in his seminal paper Adaption in natural and artificial systems. It laid the foundation of for further studies and research in the field of Soft Computing in which principles of natural evolution are applied in problems related to optimization.

GA are adaptive random search algorithm based on evolutionary ideas of natural selection and genetics. They exploit the idea of random search for solving complex optimization problems. Important is to mention that GAs use data to direct the search into the region, where the probability of finding the solution is better. Robustness of GAs comes from the fact that they do not break or regenerate even if inputs are varied due to some noise.

In 1982 John Hopfield [2] introduced the concept of Neural Networks or Artificial Neural Networks (ANN), a model based on human brain demonstrating its computational abilities. The basic feature of the paper was the introduction of energy function of the network which is used for optimization. In 1985 Hopfield and Tank [3] and again in 1987 Hopfield & Tank provided some results in the context of optimization problems. ANNs processes the information in the way human

Section 1

Optimization refers to mechanism of finding decisions that satisfy certain neurons system or brain functions for optimum decision making.

constraints to meet a specification solution which should have optimum value.

1.1 Optimization problem & Solutions

Optimization problems are a large class of complex situations which seek solutions. We try to maximize or minimize some equation at the same time satisfying the constraints related to the problem.

Formally an optimization problem has objective function.

$f(x) : R^n R$ (Euclidean Plane)

and a set of constraints C i. e. equalities & inequalities

Usually the variables x_1, x_2, x_3 ..... x_n

are constrained to discrete values. Depending upon linearity or non linearity of the objective function We can classify between non linear and linear optimization.

Optimization problem can be further classified on the basis whether variable are continuous on discrete. Later is also called as combinatorial optimization problem.

The conventional models of optimization include -

Linear programming

Integer programming

Dynamic programming

Quadratic programming

1.2 Computational optimization

Algorithms terminate in a finite no. of steps.

Iteration based model result in a solution after there is a convergence after a certain no. of loops.

Heuristics provide approximate solution to some problems.

Simplex Algorithm [4] by George Dantiz was designed for linear programming John Von Neumann [5] developed the theory of duality in the same year 1947.

Iterative computational optimization generates a series of approximate solutions for a class of problems . A termination criterion or the convergent gives the solutions. Heuristic - iteration based solutions are also common in the domain.

There are some popular heuristic based optimization methods such as Artificial Neural Networks

Genetic Algorithm

Hill Climbing

Particle Swarm optimization

Ant Colony / Bee Colony optimization

Simulated Annealing

Tabu Search

Soft Computing & Optimization

Soft Computing deals with impression uncertainty partial truth and approximation to achieve tractability, robustness & low solution cost.

Complex system in biology medicine, the humanities, management sciences often remain intractable to conventional mathematical & analytical methods.

Soft Computing is the use of inexact solution to computationally hard tasks. Such as solution of N P - Complete problems for there is no known algorithm that can compute an exact solution in polynomial time. Some Soft Computing methods -

Fuzzy Logic

Evolutionary Computation

Neural Networks

G. A.

Machine learning

Probabilistic Reasoning

Technically Evolutionary Computation (Eg. GA) belong to the family of trial & error problem solvers and can be considered global optimization methods with met heuristic or stochastic optimization character, distinguished by the use of candidate solutions.

Evolutionary Computations uses iterative progress such as growth in a population. inwhich population is selected in a guided random such using parallel processing to achieve the desire end. Such progress are often inspired by biological mechanism for evolution.

1.3 Optimality

According to Fermat's theorem of optimality 'Optima of unconstraint problems are found at stationary points, where the first derivative or the gradient of the objective function is zero'.

Stationery point is apoint of domain of function where the derivative is zero. It is a point where the function stops increasing or decreasing.

Local Minimum - Where the decorative of the function changes from negative to positive.

Local Maximum - Where the derivative of the function changes from positive to negative.

Global Minimum or Global Maximum is that point that is globally maximum or globally minimum (Global optima)

A point x* is global optima (maximum or minimum) for a function.

f(x) : R^n R on the set S if it is feasible (x* S) and if no other feasible solution has higher or lower values respectively.

fx* > fx, x S Global Max.

fx* < fx, x S Global Min.

2. OPTIMIZATION MODEL & G.A.

In, 1975 Holland described ho to apply the principles of natural evolution to problems, Thus the foundation of Genetic optimization Algorithm (GA) as laid. GAs are adaptive random heuristic algorithms based on evolutionary ideas of natural selection and genetics.

2.1 GAs & Search Space

Gas offer great benefits in searching large state space, multimodal search space or even n-dimensional search space.

The area in which the feasible solutions can be found is called the search space. Each point in this area represents a possible/feasible solution. Thus each feasible solution can be marked by its optimum (fitness) value depending upon the nature of the problem maximization or minimization.

GAs search for the best solution among a number of feasible solution i. e. among the various optimal (fitness) points a single optimum point is selected. The difficulties in this situation is to be find the start point and entrapment in local minimum.

2.2 Basic Idea

The gene pool of a given population potentially contains solution to a given adaptive problem. The solution is to be found by association of Genomes. No subject has such a genome but during reproduction new genetic combination are found and finally a subject Inherits a better gene. This concept of GA is called crossover.

Holland's algorithm is effective because it not only considered the crossover but also the mutation. Mutation may improve the capability of the algorithm and may find better optimum solutions.

2.2.1 Biological Background

All living organization are made up of cells. Each cell consists of set of chromosomes which are strings of DNA and serve as model for whole organization.

Along the strings, Block of DNA are called GENES. Each gene encodes a particular protien i. e. TRAIT.

Each gene has its own position in chromosome which s called Locus. Complete set of genetic material (all chromosomes) is called GENEOME. A particular set of gens in genome is GENOTYPE.

2.2.2. GA Pseudo code

Step 1 Generate random population of n chromosomes.

Step 2 Select a pair from the population

Step 3 Cross Over

Step 4 Mutation

Step 5 Evaluate fitness

If fitness satisfied stop

Else go to step 2

2.3 Formulation of Fitness function for optimization

The principle of GAs is simple - Copy genetics and natural selection by a computational algorithm.

The variables of the problems are DNA (represented as Vectors) or 2D or 3D matrices. The set (population) of this problem dependent variable/parameter value are processed by the GA.

We start with a random population i. e. value of vectors/matrices are randomly generated.

To optimize we need a objective function called as FITNESS FUNCTION. which selects the best solution candidate from the population and deletes the rests.

Fitness function in GA.

Function we want to optimize. For classical methods it is called objective functions. There is no common recipe for specifying an appropriate fitness function which strongly depends upon the given problem. However it is necessary to provide enough information to guide the G.A. to solution.

In many applications the fitness function is based on comparison of desired and actually obtained output.

There are basically two well known transformation which allow to standardize the fitness functions that always minimize or maximize.

Let us consider a fitness function f. assuming that the no. of individuals in the populations not fixed i. e. m_t at time t.

Maximize/Minimize

$$f_s(b_i, t) = f(b_i, t) - \max_{j=1}^{mt} f(b_j, t)$$

3. OPTIMIZATION MODEL & ANN'S

ANN's are computational approach modeled on the basis of interconnection of human neurons (Biological neurons) for optimal solutions each neuron has enforcing on inhibitory tendency (activation) based on the summation of inputs. These include outputs coming from other inter connected neurons and a certain bias. There is a THRESOLD function on the basis of which neurons are activated (FIRED) and the output

is propagated to other neurons on the network as inputs. ANNs based optimization models are particularly suitable in situations were decision making becomes difficult using the conventional optimizing method. These system are said to be trained for decision making.

A neural network may consist of mutiple layers and signals are propagated from back to front. In certain cases there are back propagation in which forward movement is looped back as reset weights on frontal neurons.

$Y_{in} = x_1,w_1 + x_2w_2$

Where x_1 is input w1 is weight from the neuron X_1

The earliest working in this area was by Mc-Culloch and Pitts[6] in 1943 was called THRESOLD Logic.

Donald Hebb [7] created the Law of Synaptic Strengthening a learning mechanism called unsupervised learning.

Frank Rosen blatt [8] created the PERCEPTRON a pattern matching algorithm used for pattern matching and optimization.

3.1 ANN's Model

ANN's are simple mathematical model defining function

$f(x) \rightarrow Y$

But sometimes the model are associated with particular type of learning or training.

Given a specific optimization task and function f or class of functions F learning uses a set of observation which solves the task with optimum. The cost function C.

C(F) {{{R} find a solution such that no solutions has less cost or the smallest possible cost.

3.2 Basic Termology

Weight

In ANNs each neuron is connected to other neurons by directed communication links carrying a weight. The weight is information about input signal. This information is used to solve a problem.

A weight matrix W contains all adaptive element of an ANN. Set of all W matrices provide the set of all information (possible) processed by an ANN.

Bias

Bias has the impact on calculating the net input of an ANN. The bias b in included by adding a components X_0=1 to the input vector X.

$X = [1, x_1, x_2, x_3 \ldots\ldots\ldots\ldots xn]$

$Y = mx+c$

Threshold - It is a set value based upon which the final output of the network can be calculated. This value is used in activation function.

A comparison to make between calculates net input and the threshold to obtain the ANN's output. For each application there is a threshold limit

F (net) = 1 if net =

-1 if net <

= threshold value

f = activation function

3.3 Important Types of ANN's

Single layered feed forward networks

Multi layered feed forward networks

MP Neurons

Hebbs Network

Perceptron

Back Propagation Network

Discrete Hopfield Network

Continuous Hopfield Network

4. CONCLUSION

GA's & ANN's are both optimization approaches based on biological systems. The purpose of both is to traverse the search space of possible answers to a problem.

Both methods of optimization are unconventional based on Random/Heuristic search/optimize methodology.

Both methods are away from the entrapment of local minimum and find a satisfactory solution for optimizing problems.

However condition of applicability may differ for both the approaches If we want to quantify a problem worth a solution GA is more convenient and in these cases number of items to be searched are in different classes. An ANN can learn to classify the items not observed before such as face recognitions of voice recognition.

A GA finds the combination of values of variables borrowing from technique from natural evolution. ANN’s follow simplistic model of brain and by changing number of weight between them, finds the output based on inputs.

GA's & ANN's are fundamentally different concept but solution class of problems they attempt to solve may overlap.

Both the approaches, however dynamically rearrange themselves as they approach the goal.

One of the characters of the GAs from the search space point of view is that they perform well in large search space and converge on one or more solution but not necessarily best solutions.

One of the important characteristics of GA & ANNs is that both correlate inputs outputs from a potentially noisy set of historical/training data.

ANNs are a model with setting of weights produces some output methods used are perceptrons, Back propagation nets, Hopfield nets etc. given the right weight they work wells are the problem.

GAs on the other, address the fitting models. They try to optimize overall problem of a given class. We write the model data in such a way that crossover operation is a reasonable thing to get out while producing a better than satisfactory result. if not the best result.

Problems from the class that require intuition such as pattern matching voice recognition writing recognition etc. are better suited for solution using ANNs.

However training a ANN may require variable amount of time. In contrast GAs calculate and ad-hoc solutions on the spot. There is a tradeoff between intuition and calculation. We some time guess faster than we calculate. However guessing may be wrong.

Concluding the comparative study from the problem domains addressed by these two soft Computing approaches viz GAs & ANNs we find that -

GAs deploys a randomized search on evolutionary operators developing individuals to be tested and compared by their fitness. These operators may be non deterministic and we can design them so they can find individuals in close proximity and look that and farther away in the parameter space to overcome the problem of getting stuck in local optima. However success of GA depends upon model we develop. Which is tradeoff between high expression potential and generality.

As ANN usually are multilayered and parameter space is not convex and contains local optima the gradient descent algorithm might stuck gradient descent is deterministic that searches through close proximity. That so the ANNs usually are randomly initialized and we train more than one model.

Lastly ANNs might be trained fast and get reasonable results with few efforts. In theory ANNs that is large enough is able to approximate any target functions. GA on the other hand make lot of design choices to get good results, that are able to search through very complex problem space and good results.

REFERENCES

[1] 'Adaption in Natural and Artificial Systems' John Holland (1975) MIT Press

[2] 'Neural Networks & Physical Systems' with emergent collective Computational abilities. J. J. Hopfield (1982)

[3] 'Neural Computations of Decision in Computational Problems' Hopfield & Tank (1985)

[4] 'Simplex Algorithm' George Dantzig' (1947) Princeton University Press.

[5] 'Theory of Duality in Linear Programming' John-Von-Neumann(1947)

[6] 'A logical calculus of the ideals immanent in neurons activity' Mc. Culloch & PItts (1943) Bulletin of Mathematical Biophysics

[7] 'The organization of Behaviors' Donald Hebb (1947)

[8] 'Perceptron Algorithm' Frank Rosenblatt (1957)