Artificial Intelligence

Enrique Peláez Ph.D.

Agenda

Machine Learning
- Genetic Algorithms

Applications

Genetic Algorithms

What are they?

Genetic Algorithms

- A brief history
  - The first examples known about genetic algorithms date back in the late 50's and beginning of the 60's; they were proposed by biologists trying to explicitly modeled natural evolution.
  - In 1962, researchers such as: G.E.P. Box, G.J. Friedman, W.W. Bledsoe and H.J. Bremermann had independently developed a version of an algorithm inspired in evolution, with the idea of optimizing functions and automatic learning, but their work produce almost not reaction.

Genetic Algorithms

What are they?

- Genetic Algorithms model the evolution process as a succession of frequent changes in the genes, with solutions similar to the concept of chromosomes
- The space of possible solutions is explored by applying transformations to those solutions, just like we observe in the living organisms: ***crossover, inversion, mutation.***

Genetic Algorithms

- ***What are they?***
  - A genetic algorithm (GA) is a programming technique that imitates de biological evolution as a strategy for solving problems.
  - Giving a specific problem, the input of a GA is a set of potential solutions to that problem, coded in some way and a metrics called aptitude function, which is used to quantitatively evaluate each potential solution.
  - These can be solutions that we already know they work, but want to improve them using GA.
  - The GA evaluates each potential solution according to the aptitude function.

Genetic Algorithms

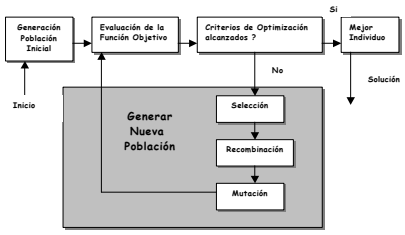
- ***What are they?***
  - From the randomly generated solutions, most of them will not work at all and will be eliminated. However, just by chance, some could be promising – they show some better activity, that might be imperfect or weak activity, but is closing the expected solution.
  - These promising solutions are kept and used to reproduce.
  - Multiple copies of them are used; not identical copies, but with some random changes introduced during the copy process.
  - Then, these digital descendents continue the process of reproducing themselves, creating new generations as solutions, and being evaluated with the aptitude function.

Genetic Algorithms

- ***What are they?***
  - Those solutions that have worsen or have not improved with the changes introduced in their code, are eliminated.
  - And, those solutions that have improved, the winners, are selected to be copied to produce the next generation with random changes, and the process continues.
  - The expectation is that the average aptitude of the population could increase after each round; hence, repeating this process thousands of times, until we discover good or better solutions to the problem.

Genetic Algorithms

What are they?



Genetic Algorithms

Basic concepts:

Genotype: genetic constituent of a living organism.

Phenotype: external observable characteristic of an organism.

Fitness: aptitude value of an individual.

Chromosome: the representation of the genotype through a *string*. Ex: a chain of bits.

The chromosome are composed by **genes** which can take different values named **alleles**.

Genetic Algorithms

How it works?

1. Randomly generate an initial population, P(0) and evaluate each individual, according to the value of ***fitness***.
2. Repeat until the end condition is reached
   a. Select some individuals for mating. Copy them in a mating buffer C(t).
   b. Apply the genetic operators, such as **mutation** or **crossover**, to the individuals placed in the mating buffer, and reproduce them as C'(t). Evaluate each children with the fitness value.
   c. Change the parents and descendents population to create a new population.

Genetic Algorithms

Example: maximize the function $f(x) = x^2$ over [0..31]

Genetic Algorithms

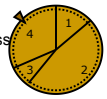
Example: maximize the function $f(x) = x^2$ over [0..31]

Generation 1

Aptitude value of an individual in the problem solution

N°	String	Valor X	Fitness X^2	% del total de fitness
1	01101	13	169	14.4
2	11000	24	576	49.2
3	01000	8	64	5.5
4	10011	19	361	30.9
Total			1170	100.0

Selection based on the proportional value of the fitness



Genetic Algorithms

Example: maximize the function $f(x) = x^2$ over [0..31]

Create the next generation

Move the roulette twice and based on the random number obtained, select two individuals from the first population.

Generate two children from the pair of parents selected.

Repeat the process until there are enough children and parents for replacing the old generation with new descendants.

Genetic Algorithms

Example: maximize the function $f(x) = x^2$ over [0..31]

Suppose that after moving the roulette 4 times, we get the following results, in this order: 2,1,4,2.

FatherID	Chromosome	Point of crossover	Children
2	11000	4	11001
1	01101		01100

FatherID	Chromosome	Point of crossover	Children
4	10011	2	10000
2	11000		11011

Genetic Algorithms

Example: maximize the function $f(x) = x^2$ over [0..31]

Generation 2

N°	String	Value X	Fitness X^2	% fitness from total
1	11001	25	625	35.6
2	01100	12	144	8.2
3	10000	16	256	14.6
4	11011	27	729	41.6
Total			1754	100.0

Repeat until the end condition is reached.

Genetic Algorithms

- Representation methods
  - Before a genetic algorithm is applied to solve a problem, a method for coding the potential solutions is needed, in such a way that a computer can process them.
  - A common coding method is to represent the solutions as binary chains; that is, sequences of 1s and 0s, where the position of each digit represents the value of some aspect of the solution.
  - Another method is coding the solutions as chains of integers or real numbers, where each position represent some aspect of a particular solution.
  - This method allows more precision and facilitates a representation of more complex situations, as the comparatively restrictive method of using just binary numbers..

Genetic Algorithms

- Representation methods
  - A third method consists of representing the individuals as a chain of letters, where each letter represent some specific aspect of the solution.
  - These 3 methods facilitate the definition of operators that can cause the changes expected in the selected candidate solutions; that is, change a 0 for 1 or vice versa, add or take a randomly generated number from the chain of real numbers, or change a letter for another.
  - Other strategy, developed by John Koza, from the Univ. Of Stanford, and named genetic programming, is to represent the problem and the solutions as data structures in trees.
  - In this method the random changes can be generated by changing the operator or the value of a node n the tree or exchanging a sub tree for another.