Genetic Programming

Sanjeev Shrestha

March 24, 2014

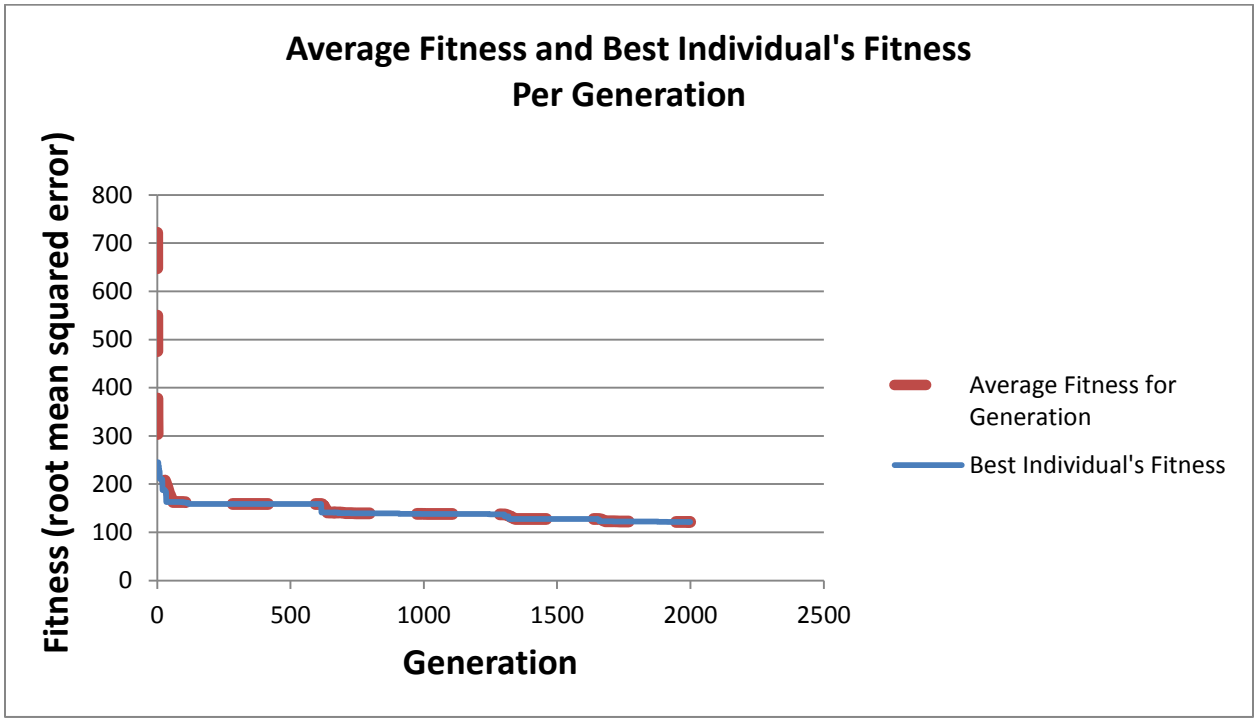
Abstract

Artificial intelligence requires the machine to have a learning approach to any given problem. Genetic Programming, defines a field in which we generate a solution in the form of a program or a formula which best fits the given dataset. The generated solution is a result of the evolution process occurring in between Individuals represented by trees. The Individuals are subjected to normal evolutionary process such as selection, crossover and mutation. Symbolic regression is a particular aspect of genetic programming in which we generate a particular formula on the basis of the given dataset. The report is a summary regarding how experiment was carried out over the test data set and how a solution was evolved to best fit the given data points. The fitness values of the generated formulae were calculated on the basis of statistical principle of root mean squared error; which describes how far a data point is from its actual value as error & root mean squared of the corresponding error values i.e. $\sqrt{\sum e_i^2}$. Selection operation performed was tournament selection; the best individual was selected from a pool of random individuals having tournament size of 10. The crossover operation performed was sub tree crossover where random node in between two individual trees were selected and swapped. The concept of Node mutation, which suggests that a Node in the tree be changed from one type to the next (having same arity), was used. Even though the experiment was not able to find the optimal solution, it did a pretty good job in finding the best individual. During, the experiment, I noticed that the solution could not perform better after a certain threshold. The trend in average fitness and best fitness of the individual in the case of symbolic regression suggested that the average fitness deviated by a higher value than best fitness. This was overcome in successive generations however. The report also reflects the difference in between actual output over the test output generated by the evolved function.

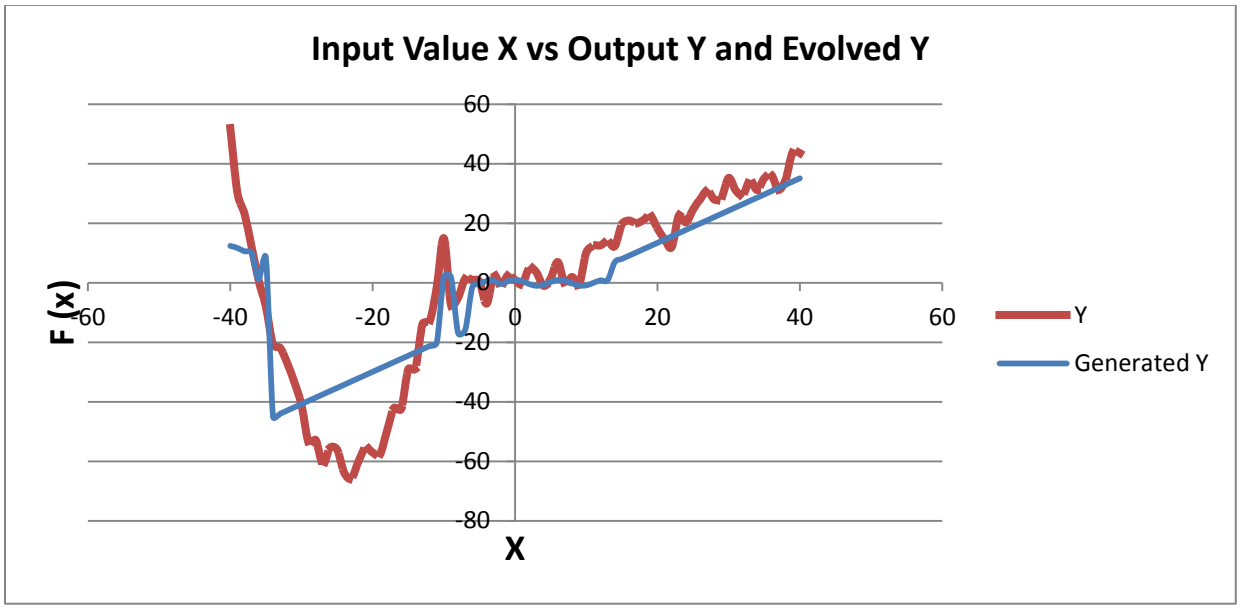
Project Details:-

Algorithm	Steady State Model
Population size	100
Tree Depth	6
Selection method	Tournament selection
Elitism (if used)	-
Crossover method	Sub tree Crossover
Crossover rate	10% of the population i.e., 10 out of 100 Individuals
Mutation method	Node Mutation
Mutation rate	50%
Operator/non-terminal set	{ ADD, DIVIDE, SUBTRACT, MULTIPLY, IF GREATER THAN, IF LESS THAN, SIN X, COS X}
Terminal set	{ VARIABLE X, CONSTANT}
Fitness function	ROOT MEAN SQUARED ERROR
Size control (if any)	None
Stop Condition	If solution has been found (fitness = 0) or Up to 2000 MAX_ITERATIONS
Random Number Range	[-5,15]

Graph showing average fitness and best fitness evolving over time



Graph showing test points and best evolved function



Discussion

In conclusion, Genetic programming is used to generate a curve that best fits the given data set. During the experiment, I experienced the evolutionary process to be slower and more tedious than in the case of Genetic Algorithms. The memory and performance issues were a major factor in working with the project. Also, the sub tree crossover was a major challenge to overcome in the project. Most intriguing aspect of the project was the ability of Genetic Programming to be able to survive destructive crossover and also have a relatively good fitness. While analyzing the best individual in the generated statistics, I found the best individual to remain constant for a number of generations. I think this is due to the high mutation rate and the destructive crossover taking place in the Individual trees. The experiment resulted in a fairly decent formula for the given problem. Finally, I conclude that Genetic Programming should be used when statistical methods to the problem are not applicable.