The “Magic” of Machine Learning through Genetic Algorithms

Phil Corbett

@PhilCorbettLive

corbett.phil@gmail.com

http://philcorbett.net

What will you get out of this talk?

- Understanding of the fundamentals of Genetic Algorithms
  - Generations
  - Fitness
  - Populations
  - Solutions

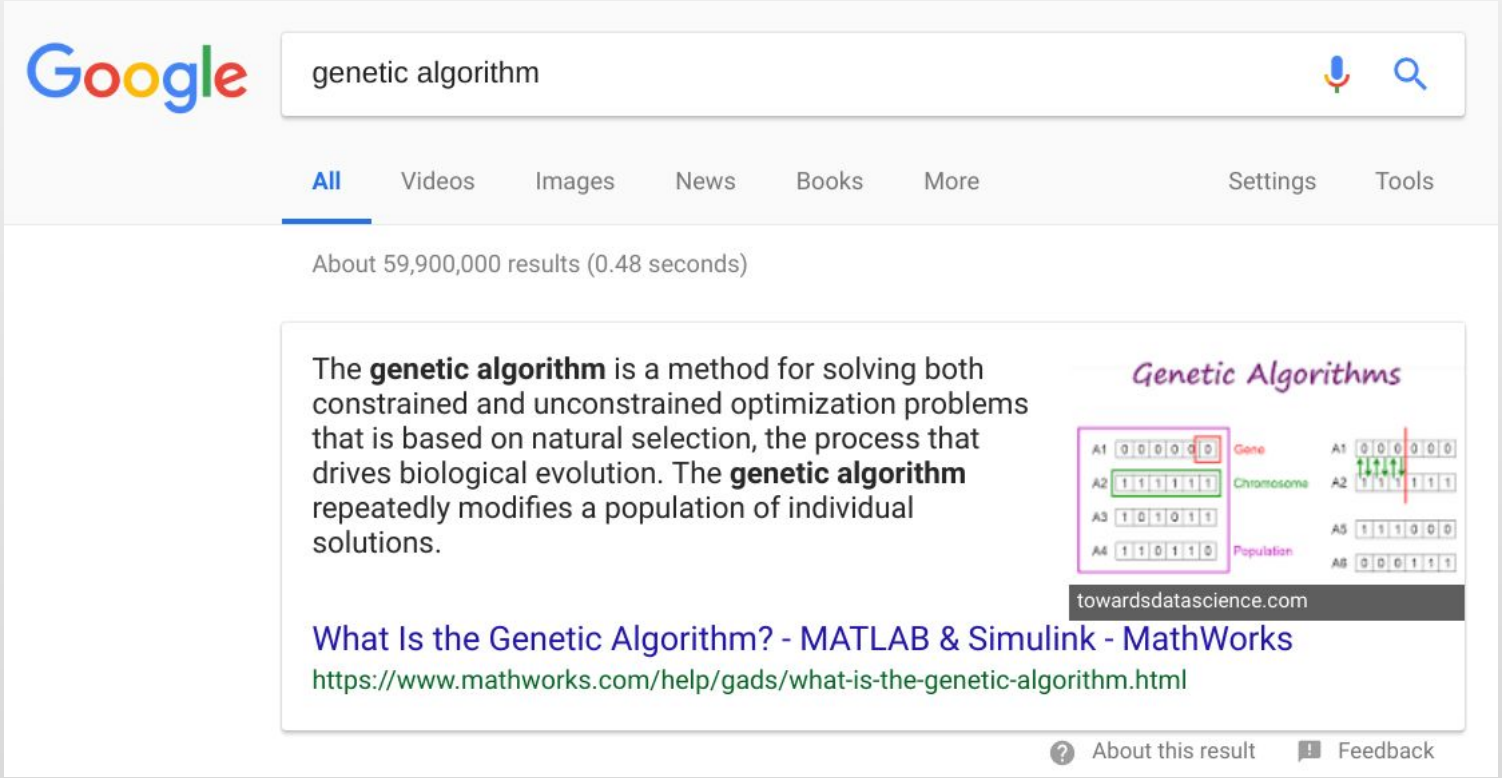
The Knapsack Problem

The Knapsack Problem

I’m going on a camping trip and have a list of items, each with a weight and priority, that I would like to take with me but I’ve only got room in by backpack for 100 lbs of things.

What can I take with me?

What is a Genetic Algorithm?



Softball #1

1

Softball #2

0

Softball #3

[0,1,1,0,1]

What is a solution?

What is a solution?

- A bit array [1,0,0,1,0,1,1,1,0,1,0,0]
- Represents a possible solution to a problem.
- Can indicate raw data or multiple Boolean values.

Solution

[0,0,0,0,0]

[0,1]

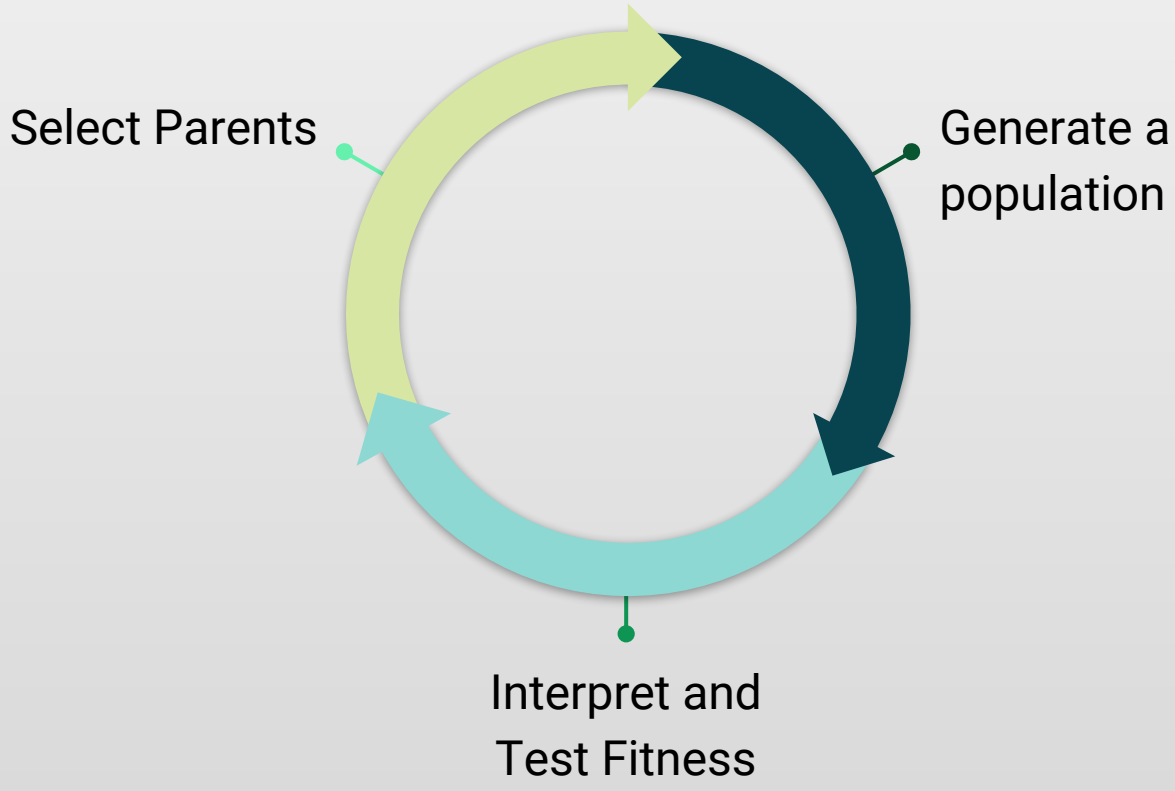
[1,1,0,0,1,0,1,1,1,1]

[1,1,0,0]

[.25, .17, .87, .90, .00]

Generation Cycle

Generation Cycle



Generate a Population

Initial Generation

Generate a random population of solutions.

Interpret and Test Fitness

Interpretation

- A problem specific function that translates a solution into something that is workable is typically needed.

Interpretation Knapsack Example

- Given 10 possible items that could be in a backpack.
- Each item is either in or out of the backpack.
- Our bit array represents multiple true/false values.
- [1,1,0,1,0,0,0,1,1,0]

Fitness

- The numeric representation of how close a specific solution is to solving the given problem.
- Fitness functions are specific to the problem you're trying to solve.
- A fitness function will evaluate the solution and return a number.
- The higher the number the more "fit" a solution is.

Fitness Example

```
function findFitness(solution) {
        var fitness = 0,
            i = 0,
            desiredBits = 2;

        for (var i = 0; i < solution.length; i++) {
            fitness += solution[i];
        }

        return {
            value: 1 / Math.abs(fitness - desiredBits),
            perfect: (fitness == desiredBits)
        };
    }
```

Select Parents

Select Parents

- Find the top solutions based on fitness score.
- Typically a higher fitness only provides a greater chance to be selected not a guarantee.

Generate a Population

Generate a Population

- Take the top selected parents and mate them to create a new population.
- Not constrained by traditional biological boundaries.

Generation Crossover

- Split both parents near the middle to mate them.
- [?,?,?,?,?,?,?,?] and [?,?,?,?,?,?,?,?]
  - [?,?,?,?,?,?,?,?]
  - [?,?,?,?,?,?,?,?]
  - [?,?,?,?,?,?,?,?]
  - [?,?,?,?,?,?,?,?]
  - etc.

Generation Mutation

- After mating, each bit has a small random chance to be flipped.
- The chance of mutation is called the mutation rate and it is typically around 1%.
- [0,0,0,0,0,0,0,0]
  - [0,0,1,0,0,0,0,0]
  - [0,0,0,0,0,0,1,0]
  - [1,0,0,0,0,0,0,0]
  - [0,0,0,1,0,1,0,0]
  - etc.

Termination Conditions

Termination Conditions

A termination condition tells our algorithm when to stop creating more generations.

There are three common ones:

1. Time
2. Generation Count
3. Fitness Threshold

Termination - Time

Stops processing after a certain interval of time has passed.

“Only create new solutions and generations for 3 minutes then stop.”

Termination - Generation Count

Stop looking for new solutions after you have reached a certain number of generations.

“Only create 1000 generations and then return the best solution found during that time.”

Termination - Fitness Threshold

When an ideal fitness score is known then it is good practice to stop processing once that threshold is found.

“A perfect fitness score is 1.0, if you find a solution with that fitness stop processing and return the solution with the 1.0 fitness score.”

Demos

Code

- Raw bit count
- Knapsack Problem
- JavaScript function generator

Summary

- Genetic Algorithms solve a problem.
- A solution is one possible way to solve the problem.
- A generation is a series of solutions from the same parents.
- Problem Specific
  - Fitness Function
  - Interpretation Function
- Termination Conditions indicate when to stop processing.
- Generational Cycle
  - Generate Population
  - Test Fitness
  - Find Parents

Thank you!

@PhilCorbettLive

corbett.phil@gmail.com

http://philcorbett.net