15.10.19

Genetic programming

It is a randomized algorithm which is used to design function for a given problem
It has nothing to do with optimization

Two types of problems:

P: Algorithm for which deterministic solution can be designed. It has to process all the inputs.

NP problems: Those problems which can't be solved in polynomial time using traditional algorithms. (in N^p time)

Eg. Optimisation problems – max or min for which ML is applied like in function prediction.

Function prediction problem

Suppose we have a dataset with input x and output f(x). We need to design a function which gives values of f(x) for unknown value of x.

x	f(x)
1	1
2	4
3	9
4	16
5	

⟶ Sol: x^2 or $x^2+(x)(x-2)(x-3)(x-4)$

There can be infinite no. of solutions for a given problem and we need to pick up the best one. ML is applied to find a model which produces best solution.
It is hard problem because many methods can be applied resulting in different solutions.

We need to identify the input parameters, whichever are relevant.

Optimality: what things we want to increase and decrease.

For genetic programming, we first initialise a random function

Suppose the population size is 4.

Assuming four random functions:

i) $x^2 - x + 1$

ii) $2x^2 - 3x + 4$

iii) $x^3 - x^2 + x + 1$

iv) $x^2 + 2x + 1$

x	$f(x)$
1	1
2	4
3	9
4	16
5	25

For making functions, we can use linear regression or multi-regression (curve fitting), both of which give different functions. Other way is neural network.

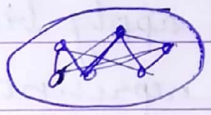
There are two types of model: -Mathematical models which always give some mathematical function.

- Biological models which don't give exact function or relationship. Here, this function is inherited in the shape of the model.

Neural network is a biological model where neurons are arranged. We can never get exact function.

Computer → used for problem solving.
There are computational requirements

Processing capability of humans and computers is different. Computer does it step by step while in humans, brain does it where there are several interconnected neurons which process input and produce output.
This was artificially implemented as a model



18.10.19
We can apply genetic algorithm and genetic programming in ML.

Iris data set

There are four attributes on the basis of which we will decide whether to play tennis or not.
They are : Outlook, Temperature, Humidity, Wind.
Outlook : can be sunny, rainy windy
If outlook is sunny, temp. is high, humidity is high, wind is weak.

	Outlook	Temp. (Hot)	Humidity	Wind	Play tennis
Input	Sunny	High	High	Weak	No
	Sunny	Hot	High	Strong	No
	Cloudy	Cool	Normal	Weak	Yes
	Rainy	Hot	Normal	Weak	Yes
	Cloudy	Mild	High	Strong	No

We already have this dataset and we have to form a correct function so that we can predict yes or no if we tell it the parameters for 6th day.

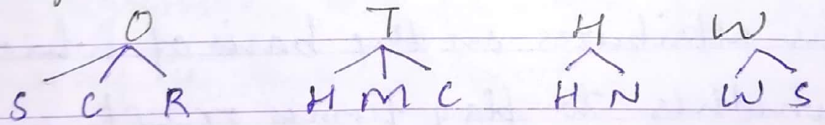
Play Tennis PT = f(O, T, H, W).

Fitness function is a hypothesis that satisfies the maximum no. of dataset.

We apply binary genetic algorithm. For this, we need to represent all this data in binary format.

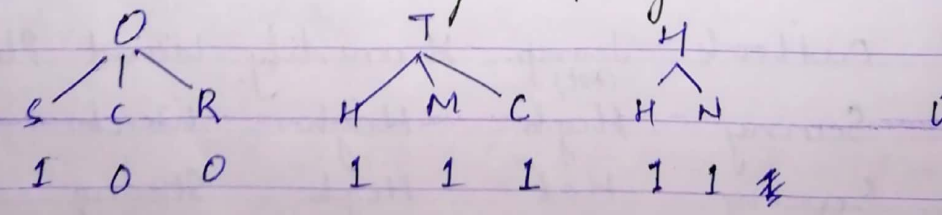
Outlook can have three values — so we use three bits. Similarly, we find for other parameters.

Binary GA —



Output will be yes or no. In input, there can be don't care values.

If outlook is sunny ⇒ play tennis is no.



∴ Chromose = 100111111∅10.

For chromosome 110 001 01 10 1.

110 ⇒ sunny or cloudy.

001 ⇒ temp is cool.

01 ⇒ humidity is normal.

10 ⇒ wind is weak.

∴ Play tennis is Yes.

In output variable, we can't do like this:

PT	
Yes	No
1	0 → Yes
0	1 → No
1	1 → Don't care

We need to represent by one bit only.

First we randomly take certain chromosomes - initial population. Thereafter, we find fitness of each solution and then apply selection, mutation and crossover. GA itself processes and gives optimal solution.

Limitation of GA:

05.11.19

$\frac{x_1 + x_2}{2}$, $\frac{x_2 - x_2}{2}$

When we are changing the encoding, we have to change the operator as well.

Maximize $f(x_1, x_2) = x_1^2 + x_2^2$

$0 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 6$

We can apply real search GA method since the search space is continuous.

For initialisation, generate two real values like

(2.1, 2.2)
(3.6, 4.3)
(1.2, 2.4)
(0.7, 5.7)

We use a randomizer and generate 4 values for x_1 and x_2.

For crossover, we can apply the previous binary method but now these values are as vector.

P_1 (2.1, 2.2)
P_2 (3.6, 4.3)

Putting cutpoint and after crossover, we get:

↓

(2.1, 4.3)
(3.6, 2.2)

Dimension means how many variables are there. Like here, it is two dimensional.
GA is used to solve n dimensional problems. Such problems are big data problems where we are capturing complex data. For eg. in satellite communication

When we are taking n variables, the problem is that computational complexity is very high.
Dimensionality reduction: Reducing dimensions because of constraints (limited hardware and software)

Mutation: Can be applied in the same way by complementing (subtracting from the upper limit).

Mutation is mostly used for exploration to produce diversity in the solution space and crossover is used for exploitation.

Binary GA is applicable in:
- Software Engineering (Test case generation problem)
- Cloud computing

Software Engineering Problem

How is software different from program?
Software is a product. Program is educational method of solving a task using computer. Softwares are created for others and people pay for it. Requirements of the client are to be met. To create software, we have to go through a process called software engineering. One important step in this process is software testing.

The different steps are
Req. analysis, Design, Code, Testing, Maintenance.

Why is software testing hard?
Because we have to check for every input and output whether the program is giving correct results or not.

We can't do 100% testing but we have to check for those cases which can be used by the customer - very important test cases. Hence, this is an optimization problem.

8.11.19

```
int x, y, z;
z = x + y
```

Range of int is 2^{32}, so it might take at least 1-2 days.

There are two types of test cases:
Good test cases (which produce errors)
Bad test cases.

No company performs exhaustive test case checking.

Kinds of testing
Black box testing and White box testing.

Black box : Usually done in companies.
Software is assumed as a black box. We only pass input and check the output which is already known. We only verify. Matching output may be done by a human being or software and is called Test Oracle. (expected output with actual output)

Boundary testing.

White box : in academic institutions.
Black box is used when program is very complex.
In white box, it is assumed that we know everything about the program.

Program is made up of instructions. Instructions

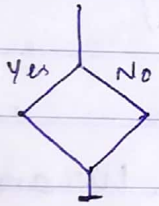
are of three types: Sequential, logical, looping.

Tester will check - whether every path has been traversed or not (each instruction) - statement coverage
- Whether each branch has been traversed or not - branch coverage
- every path — path coverage

```
{
  int x, y, z, d;
  if (x > y)
    z = x + y;
  else
    d = x - y;
}
```

We draw a graph.
Assignment instructions like $x = x+y$, $y = y-x$ are denoted by '|'.
Branch statements by — '^'

Thus, we finally get a flow graph.



Statements that can be grouped form a block

If $x = 2, y = 3$; then statement coverage = $\frac{1}{2}$
Because out of 2 statements only 1 is being executed

Our aim is to find those test cases which give maximum coverage and this is done using GA.

GA

(2, 3) = 50%.
(3, 2) = 50%.
(1, 1) = 50%.

There are several tools which can be used

Gcov (linux based tool) - tells block coverage.

Jcov

Lcov

```
if (x > y)
{   d = a - b;
    c = x - d;
}   e = f - e;
else
    m = n + 1;
```

if $x > y$ then coverage = 75%.

if we make this complex as:

if $((x > y) \&\& (x > (y+z)))$,

then coverage will decrease.

Test suite : Set of test cases which cover the entire program $= \{T_1, T_2, \ldots T_{10}\}$

Regression testing - used during maintenance.

Whenever we are maintaining the program, we have to create a new test suite for checking sections which have been modified.

Selecting those test cases from test suite which still work on modified program is called test case selection. For this also, GA is used.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
$\{T_2, T_4, T_6\}$	0	1	0	1	0	1	0	0	0	0

chromosome.

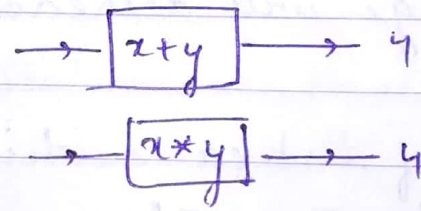
Generating a subset which covers maximum part of the program is our objective. Hence it is an optimization problem.

Mutation testing : Used in very sensitive programs Eg. in a nuclear reactor

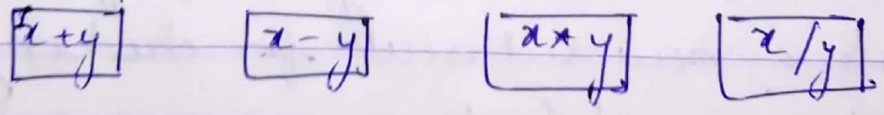
These programs are not big but the sensitivity is very high.

In a program, we have data and operator. We always check the effect of change in data, but not the operator.

Suppose 2, 2



Thus, this test case can't find out error.
What we do is we create several programs with different operators, like



We check the 'mutancy error' of each test case.
We need such test cases that give different output for each mutant.

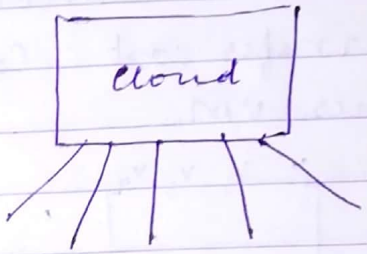
Cloud computing.

Cloud is like a network which provides services to the users.

Cloud model is service model.

We can give our infrastructure as a service, software as a service, application as a service, etc.

When we want to run an application



Large programs are divided into sections, and a workflow is created in which order the tasks are to be executed. We need to run these tasks on virtual machines present on the cloud in such a way that the program executes in minimum time. This problem is called work flow scheduling problem and is solved using GA.

How the cloud services must be distributed to the users such that maximum profit is gained.

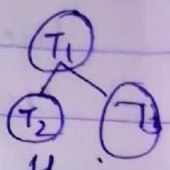
Solution of first optimization problem:

Suppose there are 10 tasks. Then we create an array of size 10. Suppose there are 4 virtual machines. So, all the entries of the array are filled with values from 1 to 4 using a randomizer. This is a workflow

1	2	1	2	3	4	1	2	3	4
0	1	2	3	4	5	6	7	8	9

What is the typicality ?

These tasks are not independent. There might be some dependency like

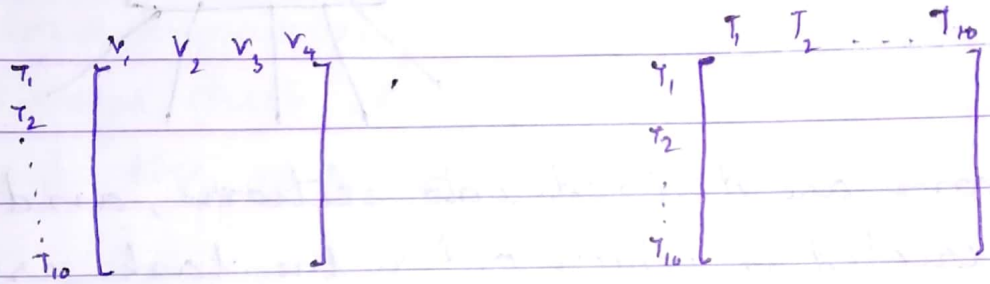


T_2 must be executed after T_1, T_3 after T_1.

So, we have a dependency graph which makes this typical.

We have execution cost and transfer cost.

Transfer cost : Cost of transferring from one VM to other VM.



Execution cost.

Transfer cost.

Total fitness : Cost + time taken.

Q. Where do we use GA in regression testing.

Q. Which other problems belong to cloud ?

Regression testing — Binary GA.

White box testing — Real no. GA.

PSO can't be applied in white box testing.

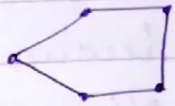
Regression testing can be used in white box testing.

5.11.19

How should we design selection/crossover/mutation operators ?

Other kinds of operators can be formulated based on our problem.

Eg. in TSP, the chromosome can be



1 2 3 4 5 or 1 3 4 2 5

Crossover and mutation won't work because same city may appear more than once. Either we need to change the coding scheme like we can take edges or we need new crossover and mutation operator and research is going over this.

We can define a crossover which works like previous crossover and then make the solution feasible.

like if we get 1 2 4 2 5,

then we traverse and find which city is repeating and replace it by those which are missing. This is called update of previous operator.

Optimization algorithms are

→ Population based : PSO, GA (There are many points)

→ Point to point : Simulated annealing

Only one point and we define operators which update this.

For improving diversity : exploration

convergence rate : exploitation

Encoding is of : Binary and Real

Another coding that is used is Quantum coding.

Minimum Cost Spanning Tree.
GA can be applied because it is an optimization problem
However, there is no advantage of applying GA because we have an algorithm that provides global optimum solution.

Single Source Shortest Path. Find ?

GA is applied in 0/1 knapsack problem.

Deterministic : check each solution.
Randomized : check random solutions
Hence, randomized algorithms are faster.

Quantum GA.
We use quantum encoding. Solutions are represented in the form of quantum bits.

We assume that the state of atom is probabilistic

We use Q bits which can be converted into binary bit.
↳ $\{\alpha, \beta\}$
α^2 → represents probability of being in 0 state
β^2 → 1
$\therefore \alpha^2 + \beta^2 = 1$

The probability of 0 and 1 state is equal

$\therefore \alpha^2 = \beta^2 = \frac{1}{\sqrt{2}}$

$\{\alpha, \beta\} = \left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

We need to convert these quantum bits into binary bits and then apply different operators. This is because we don't have the technology that work on quantum bits.

$\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$ $\left\{\frac{1}{\sqrt{2}}, \frac{\sqrt{3}}{2}\right\}$ $\left\{\frac{\sqrt{3}}{2}, \frac{1}{2}\right\}$

$\begin{cases} \text{If } (r < \alpha) \Rightarrow 1 \\ \text{else } 0 \end{cases}$

Represents 8 states.

We generate random values for these.

suppose these are = ·4, ·7, ·8

Now $\alpha = \frac{1}{\sqrt{2}} = \cdot 7 < \cdot 4$

$\therefore$ It represents 0

$\alpha = \frac{1}{2} = 0.5 < 0.7 \Rightarrow 0$

$\alpha = \frac{\sqrt{3}}{2} = 0.86 > \cdot 8 \Rightarrow 1$

$\therefore$ State = 001 (in binary)

Probability of exploration is very high but there is problem in exploitation because 8 bits are generated randomly so these values can be either close or far from previous values. Hence, exploration is high.

In terms of exploitation, this is disadvantageous

If the probability of being in state 1 = $1/4$,
then $\alpha = \frac{1}{2}$, $\beta = \frac{\sqrt{3}}{2}$

Single objective optimisation
There are several problems where we have to deal with multiple conflicting objective problems. They are called multi objective optimisations. When we try to optimize one objective, other one will decrease.

Eg. Max $f_1(x) = \sin x$
$0 \leq x \leq \pi/2$

Max $f_2(x) = \cos x$
$0 \leq x \leq \pi/2$

This is multi objective optimization problem. There may be multiple solutions in such problems.

Pareto front
Pareto was a scientist in Economics.
There are many good solutions and these are non-comparable.

Max $f_1(x) = \sin x$, Min $f_2(x) = \cos x$ $\quad 0 < x \leq \pi/2$
Single objective optimization problem

Non dominated sorting
- Assigning a rank to the solutions

Suppose we have 6 solutions.
Their fitness values are as →
Now, we apply ranking method and make a front which forms a

	Min $f_1(x)$	Max $f_2(x)$
	1	2
	2	1
	3	4
	4	3
	5	5
	1	1

class — non dominated front of class 1.

Dominance check: Starting from the 1st solution compare with all others. We find if a solution is dominating others.

x_1 dominates x_2 if it is better in atleast one objective and is equal in other. x_2 then does not belong to that class.

From the table:

x_1 dominates x_2 because $1 < 2$ and $2 > 1$.

Dominating set — formed by solutions that dominate (x_1)

Dominated set — other solutions. (x_2)

$x_1 - x_3$: $1 < 3$, $2 < 4$.

$\therefore$ x_1 does not dominate x_3.

$\therefore$ Dominating set : x_1, x_3.

Dominated set - x_2

Similarly for x_4, x_5, x_6

Dominating set : x_1, x_3, x_4, x_5

Dominated set : x_2, x_6

Now we compare in dominating set.

x_1, x_3 : already compared.

x_3, x_4 : x_3 is dominating.

After x_5,

Dominating set : x_1, x_3, x_5.

Dominated set : x_2, x_6, x_4.

This is non dominated front class 1.

Now for x_2, x_4, x_6.

x_6 dominates x_4.

Optimal non dominated front is pareto front.

∴ Front 2 : x_6, x_4

Front 3 : x_2

Complexity : mn^3

m : no. of objectives

n : no. of solutions

Q.

	Min $f_1(x)$	Min $f_2(x)$	Max $f_3(x)$
x_1	1	1	3
x_2	2	2	1
x_3	3	4	4
x_4	4	3	5
x_5	5	2	6
x_6	1	2	1

Dominating

ND (x_1, x_3, x_4, x_5)

Front 1.

Dominated

D (x_2, x_6)

$x_6 \rightarrow$ Front 2.

$x_2 \rightarrow$ Front 3.

- If criteria is not given, assume objective to be minimum.

What will happen if we increase objective functions No. of front will decrease as probability of failure increases