

# Tarea 5: Programación genética

Fernando Carrillo A01194204

12/10/20

## Descripción del problema

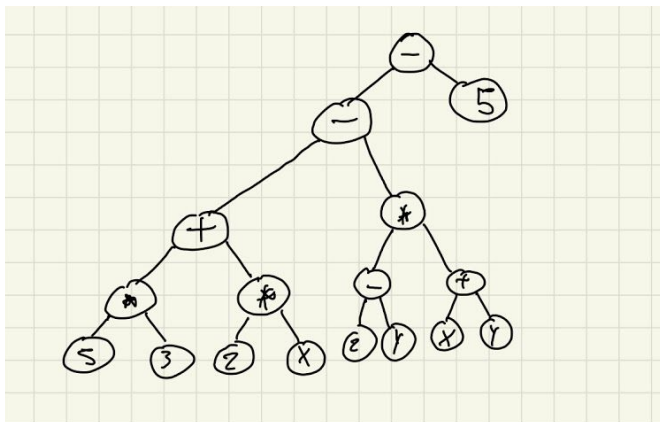
Dados los pares de valores  $(x, y)$  y sus evaluaciones en una función desconocida  $f(x, y)$ , se requiere encontrar la función de evaluación por medio de la programación genética tal que al evaluarla con las entradas  $(x, y)$ , el error cuadrático medio sea la menor posible comparado al resultado real. En otras palabras, queremos encontrar los componentes de una función que al ser evaluadas resultan en las evaluaciones dadas para  $f(x, y)$  partiendo de los valores  $(x, y)$ .

Valores dados:

$x, y$	0, 10	1, 9	2, 8	3, 7	4, 6	5, 5	6, 4	7, 3	8, 2	9, 1
$f(x, y)$	90	82	74	66	58	50	42	34	26	18

## Representación en LISP obtenida

`sub(sub(add(mul(5, 3), mul(2, x)), mul(sub(2, y), add(x, y))), 5)`



## Función que modela la representación

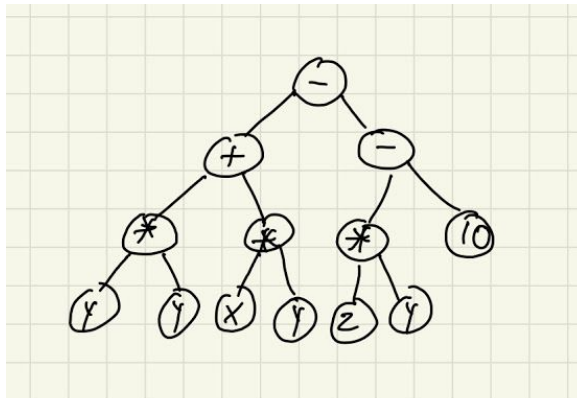
$$f(x, y) = (((5 * 3) + 2x) - ((2 - y) * (x+y))) - 5$$

## Simplificación de la función

$$f(x, y) = y^2 + xy - 2y + 10$$

## Representación en LISP de la simplificación

```
sub(add(mul(y, y), mul(x, y)), sub(mul(2, y), 10))
```



### Error cuadrático medio (MSE) de la representación

mse = 0.0

## Error en el mapea

[illegible]

No hay error en el mapeo

## Conclusiones y retos encontrados

Se me hizo muy interesante el concepto de la programación genética para encontrar soluciones por medio de la evolución, y esta actividad me ayudó a comprender mejor el proceso utilizando la librería de DEAP. En distintas ejecuciones del algoritmo tuve diferentes variaciones que eran válidas, pero la más concisa fue una con la representación  $f(x,y) = 9y + x$ .

Para completar la tarea utilicé parte del código que hicimos en clase para encontrar una función con una sola variable, y tuve que buscar en la documentación de DEAP como incluir una segunda variable. También tuve que pensar en cómo modificar la función de evaluación para que tome en cuenta las dos entradas.

Hace poco vi un video sobre la evolución de la vida orgánica y algo de teorías sobre vida inorgánica. Me gustaría saber los límites de estos algoritmos basados en la evolución para intentar hacer una simulación de la evolución de agentes inteligentes en un dado ambiente y las convergencias a las que llega en diferentes ambientes.