

Tarea 4

Javier Espinoza
November 27, 2017

CC5114-1 Redes Neuronales y Programación Genética
Departamento de Ciencias de la Computación
Facultad de Ciencias Físicas y Matemáticas

1 Cómo ejecutar

Para ejecutar el script necesario, se debe correr el script *Run.py* que contiene una sola idea. Se crean funciones para poder ser introducidas en nodos, se crea un set con números enteros que servirán para la ecuación, y luego se comienza a construir el árbol correspondiente, evaluando los fitness de cada uno para luego encontrar el mejor.

2 Árbol

Se creó una clase Node que almacena clases u objetos del tipo FunWrapper, Variable o Number que luego se irán concatenando para luego obtener un árbol sintáctico. Ejemplo:

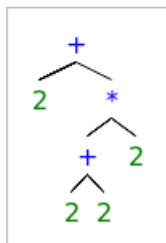


Figure 1: Ejemplo de syntax tree creado.

3 Ejemplo

Para que funcione esta implementación, se requiere de una función definida a la cuál queramos aproximarnos. En este caso, la función será: $x^2 + x + 1$. Se quiere encontrar un syntax tree que, con las funciones dadas y las constantes dadas, sus valores al evaluar la variable X sean similares a las evaluaciones de la ecuación dada.

4 Resultado

Se definieron 200 generaciones como máximo, con una profundidad de árbol de a lo más 10, mutation rate de 0.1 y probabilidad de crossover como 0.9. Ahora, para la función requerida, los árboles que se dieron al hacer los crossovers son los siguientes (se hicieron 3 pruebas para el caso. Se pueden hacer más pero el tiempo es corto).

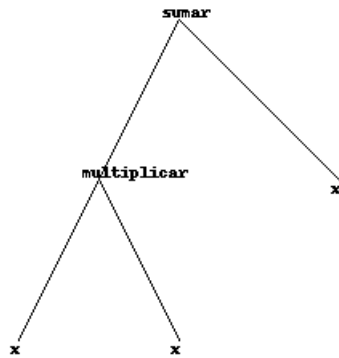


Figure 2: Primera aproximación para la función requerida: $x * x + x$

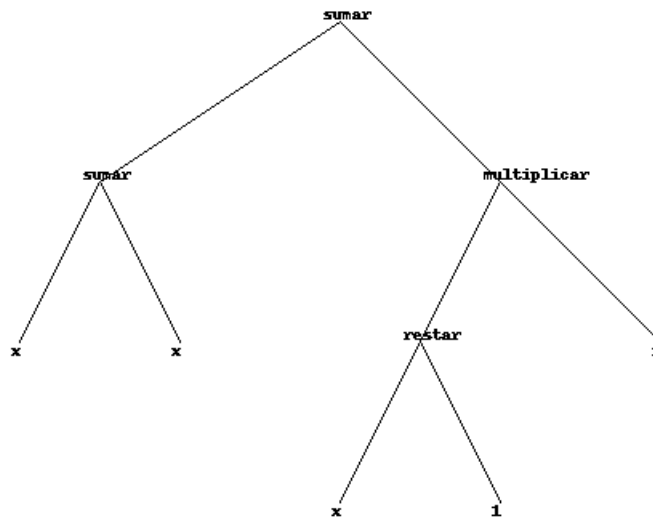


Figure 3: Segunda aproximación para la función requerida: $x + x + (x - 1) * x$

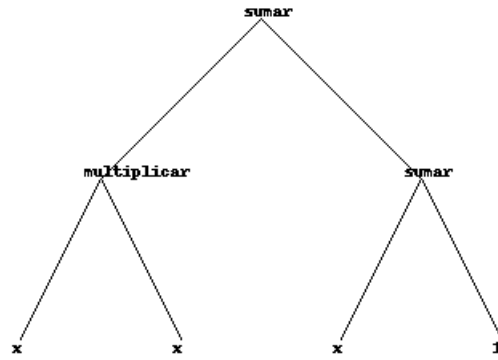


Figure 4: Tercera aproximación para la función requerida: $x * x + x + 1$

Como puede verse en los resultados, las funciones aproximadas por parte de los árboles sintácticos se asemejan o incluso son la misma función dada, pues el experimento 3 corresponde a la misma función. Entonces fue un éxito.