

Trabalho Prático 1

Programação Genética

Yuri Diego Santos Niitsuma

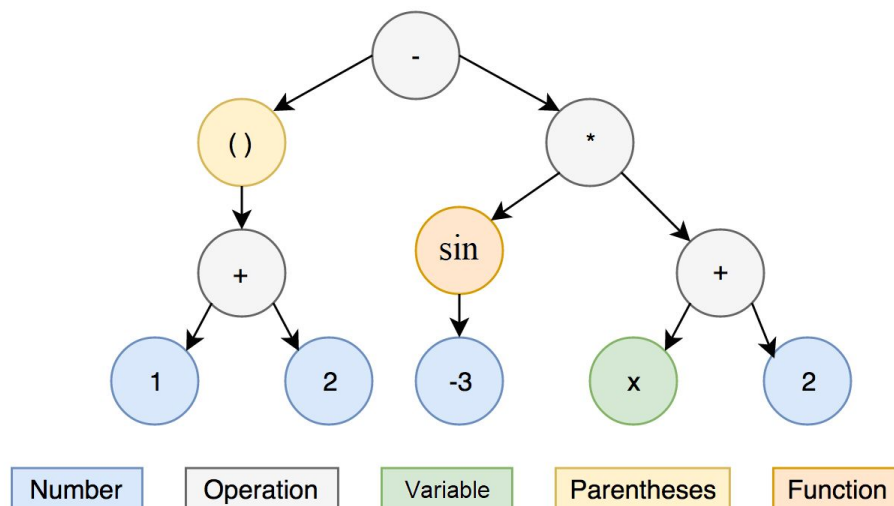
Introdução

O objetivo do trabalho é gerar uma população de expressões matemáticas e utilizando técnicas da programação genética para encontrar um conjunto de soluções que mais se aproximam de um conjunto de dados fornecidos.

Implementação

A linguagem utilizada foi o Python 3.6.2.

A estrutura das expressões matemáticas foi implementado utilizando árvore binárias contendo nós de vários tipos



O arquivo **tree.py** tem o papel de modelar os tipos descritos e manipular a árvore:

Terminais

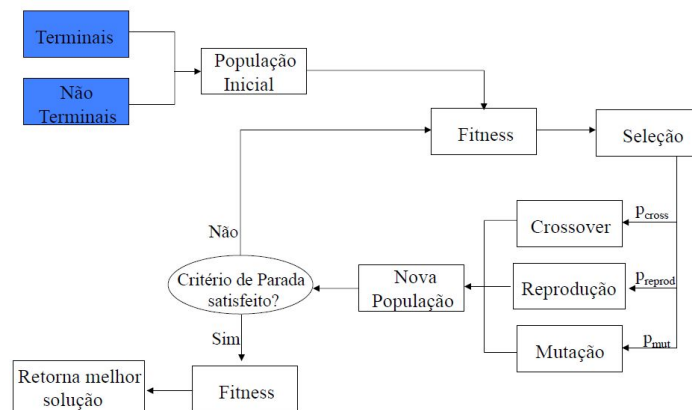
- Variable: $X_i \mid i \in \{1, \dots, n\}$
 - A variável em si.
- Constant:
 - Valor constante, na figura é representado pelo *Number*.

Não terminais

- Operator: Composto pelos operadores básicos $\{+, -, *, /\}$.
 - Na divisão por zero ficou definido que tem como resultado 1.0.
- Function: Utilizado para agrupar as seguintes funções:
 - $\sin(x) \mid x \in Real$
 - $\cos(x) \mid x \in Real$
 - $\ln(x)$
 - se $x > 0 \Rightarrow \ln(x)$
 - se $x \leq 0 \Rightarrow 0$
 - $\exp(x) \mid x \in Real$

O arquivo **indivudal.py** modela as propriedades do indivíduo: árvore, fitness, crossover e mutação.

O **gp.py** fica a cargo de seguir o fluxo da programação genética.



A árvore é gerada aleatoriamente com profundidade aproximada de 3. Um esforço de eliminar alguns “intros” é somando duas constantes entre um operador.

Experimentos

Os logs do resultado dos experimentos estão no diretório **output**.

keijzer-7

Segundo o artigo [Genetic Programming Needs Better Benchmarks](#), a função objetivo é $\ln(x)$.

As instâncias que obtiveram os melhores resultados são:

- 12.txt
 $(\ln((X1 * 1.133836227858187)) + \ln(\cos(\exp(7.859096223899583))))$
RMSE: 0.16038689504251544
- 14.txt
 $(\cos(\ln((X1 / X1))) + \ln((X1 / 2.900185344196262)))$
RMSE: 1.1896973429403954
- 18.txt
 $(\cos(\ln(-2.2314588880200366)) - \ln((2.4938123996942174 / X1)))$
RMSE: 1.3173450836810257
- 23.txt
 $(\ln((X1 + X1)) - \cos(\cos(0.5341914244103876)))$
RMSE: 0.7702321063637263
- 29.txt
 $\ln(((X1 + X1) / 1.9446354701999558))$
RMSE: 0.2793192126171043

Observe que todos contém logaritmo com algumas variações de constantes e funções trigonométricas, já que elas apenas influenciam em algumas flutuações se estiver no numerador.

Com um aumento da probabilidade de crossover para a margem de 50% foi obtido melhores resultados.

keijzer-10

No mesmo artigo a função objetivo é x^y contendo duas variáveis.

Com exceção do 1.txt primeiro, todos obtiveram erro abaixo de 1. Talvez pelo fato que a curva da função é mais comportada. Por exemplo a função constante 1.56 contém um erro RMSE aproximadamente de 0.89.

house

Estes dados possuem 10 variáveis e a função objetivo é desconhecida. Obtive muitos erros numéricos e grandes (na casa de 48 milhões). Talvez com tempo maior de execução ou refinamento no processo de mutação das árvores possa ajustar para menor.

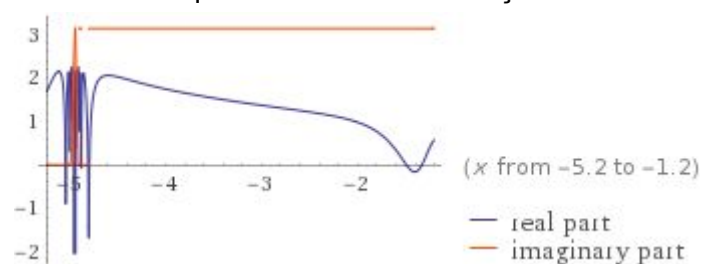
Conclusões

O código consiste de alguns problemas. Como a utilização de exponenciais e funções no denominador devem ser bem controladas ou eliminadas. Pois muitas funções compostas, principalmente no denominador, torna a função bem complexa, o que

Por exemplo a função

$$\ln(\cos(\exp((-2.627176889768328 / x))) - \ln(\exp((x / -0.5995299135427423)))) * \cos(((-4.575619132309248 - x) / (x + 4.938891699281122))))$$

Tedo uma frequência alta na vizinhança do -5.



Isto ocorre pois torna as suas derivadas sucessivas não limitadas formando funções “mal comportadas”, gerando alguns overfittings em que os indivíduos perdem o ranqueamento no cálculo da fitness sobre os dados de testes.

Outro ponto a destacar é a repetição de indivíduos gerados pelo processo de elitismo e seleção. Não encontrei uma forma de eliminar isto pois descartando alguns indivíduos acabam divergindo ainda mais da solução na iteração.

Referência

McDermott, James, Kenneth De Jong, Una-May O'Reilly, David R. White, Sean Luke, Luca Manzoni, Mauro Castelli, et al. 2012. “Genetic Programming Needs Better Benchmarks.” In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference - GECCO '12*. doi:10.1145/2330163.2330273.