CREST

14th TAROT Summer School 2018

UCL, 2-6th July 2018

Genetic Improvement

Demo: count blue pixels

Goal: reduced runtime

W. B. Langdon

Computer Science, University College, London

Assumes Linux, tcsh, gcc

http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/gp-code/opencv_gp.tar.gz

Demo count blue pixels

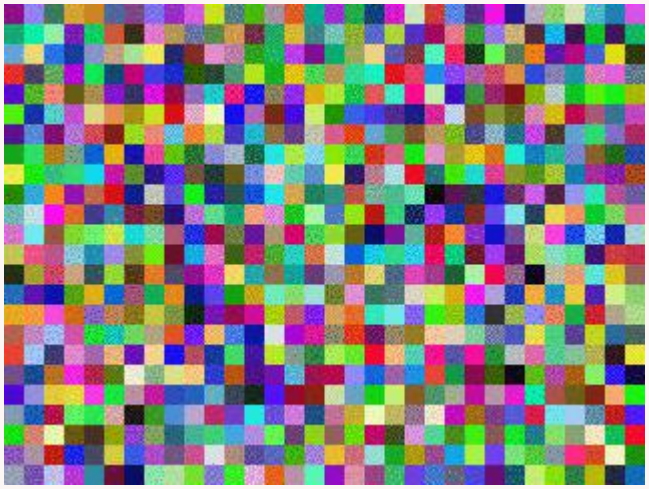
- Assumes Linux, tcsh, gcc.
- Download and unpack

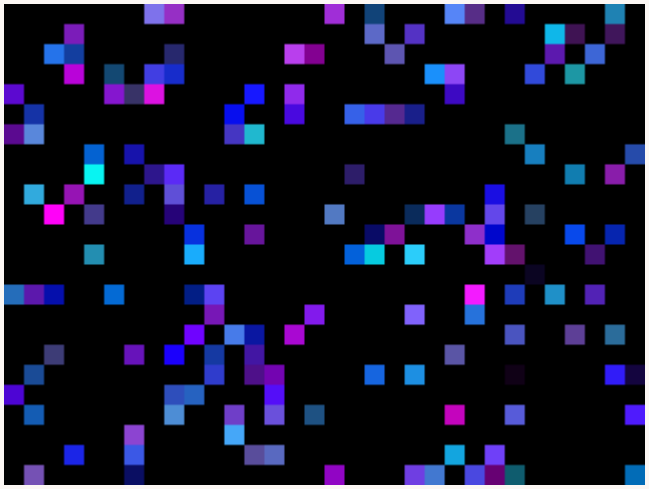
  http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/gp-code/opencv_gp.tar.gz

  gunzip -c opencv_gp.tar.gz | tar xvf –
- README.txt
  - Find part relating to blue.cpp (based on OpenCV SEEDS algorithm)
    - "API-Constrained Genetic Improvement", SSBSE-2016

README.txt

- Attempt only running evolution
- Use prepared grammar blue.bnf
- Fitness time taken to count blue pixels in fixed random image.





RE_gp.bat

- tcsh script creates many files, it must put them in per run directory.

  mkdir sources_blue

  ./RE_gp.bat 142605 100 10 blue 11 blue.bnf

  $1 seed, $2 popsize, $3 generations, sources_blue, $5 fitness repeats, $6 grammar

- Best of each generation displayed

Output of RE_gp.bat

- Open 2nd window

  display output in pop.000.fit

  cd sources_blue; tail –f pop.000.fit

Files in sources_blue/	
Population	pop.nnn
Raw output	pop.nnn.fit
Extract fitness	pop.nnn.fit2
Sort by fitness	pop.nnn.fit3
Parents of next generation	pop.nnn_select

Some Possible errors

- mkdir: cannot create directory ‘sources_blue’: File exists
  - Can use existing directory created by tar or give another name (must begin sources_)
- ./RE_gp.bat 142605 100 10 blue

No BNF given

  - Not enough parameters for RE_gp.bat
- Gen 000 ok 0 Zero!
  - Examine 000.fit for to see why everyone in generation 000 failed

What has evolved

- See best of last generation.
- Find it in pop.nnn.fit
- pop.nnn.fit shows both changes and fitness.
- Did mutant code pass all tests? How fast is the mutated code?

What has evolved: generation 000

- Best generation 000 0 11 1115045 SEEDS $Revision: 1.4 $ ./GP.exe 000 62 ...24 by 32 fake_image PROG <_blue.cpp_38>+<_blue.cpp_42>
- 0 diff in pixel count (zero so all tests ok)
- 1115045 average speed
- Insert copy of line 38 before line 42
- Find <_blue.cpp_38>+<_blue.cpp_42> in pop.000.fit id 62

<_blue.cpp_38>+<_blue.cpp_42>

- Insert line 38 before line 42 (pop.000.fit)

gip_fit1.bat PROG <_blue.cpp_38>+<_blue.cpp_42>

diff ./blue.cpp ../sources/blue.cpp

< Output_image(k) = GP_G(Input_image(k));i = 0;

> i = 0;

SEEDS $Revision: 1.4 $ ./GP.exe 000 62 ndiff 0 0 0 0 0 0 0 0 0 0 0 0 delta lld 1306354 1112934 1142468 1123509 1113430 1137594 1124671 1115045 1138384 1146760 1127792 1120842 tics, 24 by 32 fake_image

- Diff shows changes made by mutation
- Line with ./GP.exe is output of test case. Note run 11 times.

What has evolved

- See best of last generation.
- Find it in pop.nnn.fit
- pop.nnn.fit shows both changes and fitness.
- Did mutant code pass all tests? How long did it take?

What has evolved generation 010

- Best generation 010 0 11 641165 SEEDS $Revision: 1.4 $ ./GP.exe 010 88 ...24 by 32 fake_image PROG
  <_blue.cpp_98>+<_blue.cpp_100>
  <_blue.cpp_91>x<_blue.cpp_92> <_blue.cpp_89>x<_blue.cpp_95>
  <_blue.cpp_92> <_blue.cpp_113>+<_blue.cpp_98>
  <_blue.cpp_93>x<_blue.cpp_94> <_blue.cpp_95>+<_blue.cpp_82>
  <_blue.cpp_90>x<_blue.cpp_82> <_blue.cpp_72><_blue.cpp_80>
  <for1_blue.cpp_25><for1_blue.cpp_26> <_blue.cpp_75>
- 0 diff in pixel count (zero all tests ok)
- 641165 1115045 74% faster best gen 000
- Find id 88 in pop.010.fit

Best last generation

- Many changes, some ineffective?

gip_fit1.bat $Revision: 1.65 $ eden.cs.ucl.ac.uk 010 88

diff ./blue.cpp ../sources/blue.cpp Many changes including delete:

> GP_R2(width, height, input_image, red);

> GP_G2(width, height, input_image, green);

> GP_B2(width, height, input_image, blue);

- Diff shows changes made by mutation including deleting duplicate setting of red, green, blue arrays
- Optional (after demo) remove changes one at a time to decide which are really needed (see HC1.bat)

Check

- Evolution is good at finding weak spots
- Is code change found by evolution correct
- Will it work on other images

Conclusions

- Genetic Improvement (GI) applies Darwinian survival of the fitness to existing code
- GI for automatic bugfixing, software transplanting, performance improvement faster answers or better answers.

  BarraCUDA 3,095 sourceforge downloads (26 months).

  Commercial use by Lab7 (in BioBuilds Nov2015)

  IBM Power8.

  RNAfold par,SSE,CUDA (17,061 downloads)
- Future GI. Do impossible things
- Software is not fragile

  break it, bend it, Evolve it

The Genetic Programming Bibliography

http://www.cs.bham.ac.uk/~wbl/biblio/

12259 references, 10000 authors

Make sure it has all of your papers!
E.g. email W.Langdon@cs.ucl.ac.uk or use | Add to It | web link

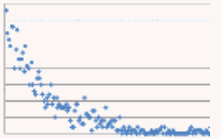

Co-authorships

RSS Support available through the Collection of CS Bibliographies.

Co-authorship community.
Downloads


Downloads by day

A personalised list of every author's GP publications.


Your papers

blog

Search the GP Bibliography at
http://liinwww.ira.uka.de/bibliography/Ai/genetic.programming.html