How to Evolve Life in a Computer using Python

Bert Chan

Big Data Consultant @ ASL

PyCon HK / Code Conf 2018
Programming in the 1990’s

• PC: **80286** (8MHz, 8MB RAM)
• OS: MS-DOS (Win3.1 too slow!)
• **Pascal**
  • Simulate life
  • Simulate gravity, fractals
  • Hack & decode games
• **Assembly**
  • Main loop – very fast!
  • Direct write to video cache
If you did coding and hacked stuffs in the 90’s, you’re a...
Simulate Life

• **Conway’s Game of Life**
  • John Conway 1970
  • Cellular Automata – array of **cells** (0 or 1)
  • **Neighborhood** (8 cells, sum)

• Simple **if-then-else** rule
  for cell in cells:
  
  if cell==1 and sum in [2, 3]: cell = 1  #survive  
  elif cell==0 and sum in [3]:  cell = 1  #born  
  else: cell = 0  #die
Conway’s Game of Life

• The moving **glider**
• **Green** cells survive, **blue** cells are born, **red** cells die
Conway’s Game of Life

• Spaceships, glider gun...
• Logic gate, clock, computer...
• Hackers love it!
• Good way to learn programming!
Play with the rules

• What if we...

\[
\text{cell: 0 or 1}
\]

\[
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\end{array}
\]

\[
\text{neighborhood (n)}
\]

\[
\text{sum} = \sum n
\]

\[
\text{cell = (if sum ... then ... else ...)}
\]
Play with the rules

• What if we...
  • Use floating point?

\[
\sum_n \text{cell} = (\text{if sum} \ldots \text{then} \ldots \text{else} \ldots)
\]
Play with the rules

• What if we...
  • Use floating point?
  • Bigger neighborhood? Circular?

\[
\text{neighborhood } (n) \\
\text{sum} = \sum n \\
\text{cell} = \text{(if sum ... then ... else ...)}
\]
Play with the rules

• What if we...
  • Use floating point?
  • Bigger neighborhood? Circular?
  • Weighted sum?

\[
\text{cell} = \begin{cases} 
\text{if sum ... then ... else ...} 
\end{cases}
\]

\[
\text{sum} = \sum nk
\]
Play with the rules

- What if we...
  - Use floating point?
  - Bigger neighborhood? Circular?
  - Weighted sum?
  - Smooth update?

\[ \text{cell} = \text{cell} + 0.1 \times f(\text{sum}) \]
Play with the rules

• What if we...
  • Use floating point?
  • Bigger neighborhood? Circular?
  • Weighted sum?
  • Smooth update?

• Spooky things happened...
Lenia

• New kind of **Artificial Life**
  • Microorganism-like creatures
  • Discovered 400+ species
  • Study their anatomy, behavior, physiology...

• Good programming exercise
  • JavaScript, C#, MATLAB, Python
Video

- Python ➔ showcase video
- https://vimeo.com/277328815
Kyoto

• Won GECCO Virtual Creatures Contest, Kyoto
• Honorable Mention in ALIFE Art Award, Tokyo
• Meet my AI hero – @hardmaru
  • David Ha (Google Brain Tokyo)
Using Python
for PyCon HK
Why Python?

• Good performance
• Fast coding
• Nice syntax (indent, list comprehension, etc)
• Lots of useful libraries
• Vibrant community (PyCon, GitHub...)
Python Libraries

• “Rule 34” of Python
  • “If there is a need, there is a Python library for it.”

Array calculations
- NumPy
- PyOpenCL / PyCUDA
- Reikna

GPU acceleration
- SciPy / OpenCV

Image processing
- PIL/Pillow

Interactive UI
- Tkinter
- Matplotlib

Record video
- subprocess + ffmpeg
- ffmpeg-python
NumPy

• Fast array calculations
  ✓ Machine learning, deep learning
  ✓ Basis of image processing, time-series
  ✓ Cellular automata (weighted sum using FFT)

• Main loop of Lenia in 3 lines
  potential_fft = np.fft.fft2(cells) * kernel_fft
  potential = np.fft.fftshift(np.real(np.fft.ifft2(potential_fft)))
  cells_new = np.clip(cells + dt * g(potential, m, s), 0, 1)
PyOpenCL/PyCUDA + Reikna

• GPU acceleration
  • (NVIDIA) CUDA → PyCUDA
  • (Apple) OpenCL → PyOpenCL

![CUDA Toolkit Documentation](https://docs.nvidia.com/cuda/cufft/index.html)

Computing a number \( \text{BATCH} \) of one-dimensional DFTs of size \( \text{NX} \) using cuFFT will typically look like this:

```c
#define NX 256
#define BATCH 10
#define RANK 1
...
{
cufftHandle plan;
cufftcomplex *data;
...
cudaMalloc((void**)&data, sizeof(cufftComplex)*NX*BATCH);
cufftPlanMany(&plan, RANK, NX, &embed, istride, idist,
&embed, ostride, odist, CUFFT_C2C, BATCH);
...
cufftExecC2C(plan, data, data, CUFFT_FORWARD);
cudaDeviceSynchronize();
...
cufftDestroy(plan);
cudaFree(data);
...}
```
PyOpenCL/PyCUDA + Reikna

• Reikna
  • PyOpenCL/PyCUDA wrapper
  • Compiles the GPU code for you
  • GPU accelerated FFT in a few lines

```python
gpu_thr = reikna.cluda.any_api().Thread.create()
gpu_fft = reikna.fft.FFT(cells.astype(np.complex64)).compile(gpu_thr)
op_dev = gpu_thr.to_device(cells.astype(np.complex64))
gpu_fft(op_dev, op_dev, **kwargs)
cells = op_dev.get()
```
PIL/Pillow, SciPy, OpenCV

• Image handling
  • PIL (Python Image Lib) → pillow
  • Create image, draw lines/texts, save GIF...
    ```python
    img = PIL.Image.frombuffer('P', buffer.shape, buffer, ...)
draw = PIL.ImageDraw.Draw(img)
    ```
    ```python
    img[0].save(path, format='GIF', append_images=self.gif[1:], loop=0 ...)
    ```

• Image processing
  • SciPy
    ```python
    scipy.ndimage.rotate(A, reshape=False, order=0, mode='wrap')
    ```
  • OpenCV-Python
Tkinter vs Matplotlib

• Interactive UI
  • Real-time 2D image display
  • Menu, keyboard binding, clipboard

• Matplotlib
  • For data visualization
  • Powerful but slow...

• Tkinter (Toolkit interface)
  • Basic and fast
  • Others: wxPython, PyQt, PyGTK...
Tkinter vs Matplotlib

• Interactive UI
  ```python
  win = tk.Tk()
tk.Canvas()
tk.Menu()
  win.bind('<Key>', key_press_event)
  win.clipboard_get()
  ```

• Python 3
  ```python
  import tkinter as tk
  ```
subprocess + ffmpeg

• Pythonic FFmpeg wrappers
  • ffmpeg-python, ffmpy, etc.

• Pipe video to ffmpeg
  
  ```python
  cmd = ['/usr/local/bin/ffmpeg', '-f','rawvideo', ...]
  video = subprocess.Popen(cmd, stdin=subprocess.PIPE)
  for img in imgs:
    video.stdin.write(img.convert('RGB').tobytes())
  video.stdin.close()
  ```
About ALife and AI

for HK Code Conf
Lenia

• Not just funny creatures
• Using AI to create ALife
About ALife

• Artificial Life
  • Simulate biological life or create new lifeforms
  • → Create a body

Wet ALife = biochemistry, synthetic life
  Artificial cell, expanded genetic code...

Hard ALife = hardware, robots & machines
  Humanoids, Strandbeest...

Soft ALife = software, simulations
  Cellular automata, virtual creatures...
About ALife

**Wet ALife** = biochemistry, synthetic life
Artificial cell, expanded genetic code...

**Hard ALife** = hardware, robots & machines
Humanoids, *Strandbeest*...

**Soft ALife** = software, simulations
Cellular automata, virtual creatures...

Synthetic cell (*JCVI*, 2010)

Expanded DNA (*TSRI*, 2014)
About ALife

**Wet ALife** = biochemistry, synthetic life
Artificial cell, expanded genetic code...

**Hard ALife** = hardware, robots & machines
Humanoids, *Strandbeest*...

**Soft ALife** = software, simulations
Cellular automata, virtual creatures...

*Strandbeest* (*Theo Jansen*, 1990)

About ALife

**Wet ALife** = biochemistry, synthetic life
Artificial cell, expanded genetic code...

**Hard ALife** = hardware, robots & machines
Humanoids, *Strandbeest*...

**Soft ALife** = software, simulations
Cellular automata, virtual creatures...

Virtual creatures (*Karl Sims*, 1994)

Soft robots (*Nick Cheney*, 2014)
About AI

• Artificial Intelligence
  • Machines do: learning, planning, vision, language, emotion, art
  • → Create a mind

GOFAI = Good old-fashioned AI
Symbolic, expert systems

ML = Machine Learning
Supervised, unsupervised, reinforced

EA = Evolutionary Algorithms
Neuro-evolution, novelty, etc

DL = Deep Learning
Deep neural nets + big data + many GPU
About AI

**GOFAI** = Good old-fashioned AI
Symbolic, expert systems

**ML** = Machine Learning
Supervised, unsupervised, reinforced

**DL** = Deep Learning
Deep neural nets + big data + many GPU

**EA** = Evolutionary Algorithms
Neuro-evolution, novelty, etc

Deep Blue vs. Kasparov *(IBM, 1997)*

Watson in Jeopardy! *(IBM, 2011)*
About AI

**GOFAI** = Good old-fashioned AI  
Symbolic, expert systems

**ML** = Machine Learning  
Supervised, unsupervised, reinforced

**DL** = Deep Learning  
Deep neural nets + big data + many GPU

**EA** = Evolutionary Algorithms  
Neuro-evolution, novelty, etc

AlphaGo vs. Lee Sedol (*DeepMind*, 2016)

BigGAN (*Andrew Brock*, 2018)

Autopilot (*Tesla*, 2014)
About AI

**GOFAI** = Good old-fashioned AI
Symbolic, expert systems

**ML** = Machine Learning
Supervised, unsupervised, reinforced

**DL** = Deep Learning
Deep neural nets + big data + many GPU

**EA** = Evolutionary Algorithms
Neuro-evolution, novelty, etc

PicBreeder (*Eplex, 2007*)

Evolutionary AutoML (*Google Brain, 2017*)
DL = Deep Learning  
(Gradient Descent)

EA = Evolutionary Algorithm  
(Natural Selection)
Towards AGI

• Artificial **General** Intelligence
  • Sapience, sentience, consciousness
  • When? How? Should we?
Towards AGI

- Consciousness? Mind? Soul?
- Learning, Planning (Reinforcement Learning?)
- Knowledge, Reasoning (Symbolic AI?)
- Senses, Communication (Deep Learning?)
- Curiosity, Creativity (Evolutionary Algorithms?)
- Emotions, Empathy (Artificial Empathy)
- Safety, Ethics (AI Safety)
- Body, Actions (Artificial Life)

Lenia
Use Lenia to...

• Understand evolution by writing programs?
• Teach AI to be curious and creative?
• Teach AI to understand life?
Thank You!

Bert Chan
chakazul.github.io
@BertChakovsky