GPU Computing for Cognitive Robotics

Martin Peniak, Davide Marocco, Angelo Cangelosi
Acknowledgements

This study was financed by:

- EU Integrating Projects - ITALK and Poeticon++ within the FP7 ICT programme Cognitive Systems and Robotics
- ARIADNA scheme of The European Space Agency

Thanks to my supervisors Prof Angelo Cangelosi, Dr. Davide Marocco and Prof Tony Belpaeme for their support

Thanks to Calisa Cole and Chandra Cheij from NVIDIA for their help
New position at Cortexica
Imperial College London

- Leading provider of visual search and image recognition technology for mobile device

- Creators of a bio-inspired vision system enabling intelligent image recognition using principles derived from the human sight

www.cortexica.com
Overview

- Action and language acquisition in humanoid robots
- Biologically-inspired Active Vision system
- Software development
Action and Language Acquisition in Humanoid Robots
Humans are good at learning complex actions

Constant repetition of movements with certain components segmented as reusable elements

Motor primitives are flexibly combined into novel sequences of actions

Human motor control system known to have motor primitives implemented as low as at the spinal cord and hi-level planning and execution takes place in primary motor cortex
Explicit hierarchical structure vs multiple timescales
slide right
Initial testing of two actions

Experimental Setup

- SOM and MTRNN trained on 2 sequences each repeated 5x with different positions
- Extended version of up to 9 action sequences
- Left and Right hand used individually
- MTRNN input: head, torso and arms (41 DOF)
- Update rate: 50ms
Multiple Time-scales Recurrent Neural Network Experiment on action-language grounding – step 1

Proprioceptive Input → MTRNN → Visual Input

Linguistic Input

<table>
<thead>
<tr>
<th></th>
<th>Action 1</th>
<th>Action 2</th>
<th>Action 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object 1</td>
<td>trained</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object 2</td>
<td></td>
<td>trained</td>
<td></td>
</tr>
<tr>
<td>Object 3</td>
<td></td>
<td></td>
<td>trained</td>
</tr>
</tbody>
</table>
Results

20 trials conducted and each reached the threshold error of 0.000005
Multiple Time-scales Recurrent Neural Network
Scaling up the experiment on action-language grounding

<table>
<thead>
<tr>
<th>Action 1</th>
<th>Action 2</th>
<th>Action 3</th>
<th>Action 4</th>
<th>Action 5</th>
<th>Action 6</th>
<th>Action N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object 1</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
</tr>
<tr>
<td>Object 2</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
</tr>
<tr>
<td>Object 3</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
</tr>
<tr>
<td>Object 4</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
</tr>
<tr>
<td>Object 5</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>untrained</td>
</tr>
<tr>
<td>Object 6</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>untrained</td>
</tr>
<tr>
<td>Object N</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
<td>trained</td>
</tr>
</tbody>
</table>
Multiple Time-scales Recurrent Neural Network

Generalisation testing

Experimental Setup

- For each of the 9 objects, SOM and MTRNN was trained on 9 sequences each repeated 6x with different positions. Total of 478 sequences each with 100 41-wide vectors.

- Left and Right hand used individually

- MTRNN input: head, torso and arms (41 DOF)

- Update rate: 50ms
Self-organising maps
CPU vs GPU Performance
Multiple Time-scales Recurrent Neural Network
CPU vs GPU Performance

<table>
<thead>
<tr>
<th>Neurons</th>
<th>2xGTX470</th>
<th>2xGTX580</th>
<th>2xEC2075</th>
<th>2xK40</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>4.2x</td>
<td>5.6x</td>
<td>4.0x</td>
<td>1.7x</td>
</tr>
<tr>
<td>136</td>
<td>5.9x</td>
<td>7.8x</td>
<td>5.7x</td>
<td>4.0x</td>
</tr>
<tr>
<td>264</td>
<td>4.0x</td>
<td>5.8x</td>
<td>3.9x</td>
<td>4.9x</td>
</tr>
<tr>
<td>520</td>
<td>2.5x</td>
<td>3.5x</td>
<td>2.5x</td>
<td>4.8x</td>
</tr>
<tr>
<td>1,032</td>
<td>7.0x</td>
<td>9.0x</td>
<td>4.9x</td>
<td>3.6x</td>
</tr>
<tr>
<td>2,056</td>
<td>9.4x</td>
<td>14.7x</td>
<td>10.1x</td>
<td>7.8x</td>
</tr>
<tr>
<td>4,104</td>
<td>27.2x</td>
<td>51.4x</td>
<td>26.4x</td>
<td>32.6x</td>
</tr>
<tr>
<td>8,200</td>
<td>21.6x</td>
<td>33.3x</td>
<td>21.0x</td>
<td>37.9x</td>
</tr>
</tbody>
</table>
Biologically-inspired Active Vision system
Specific template or computational representation is required to allow object recognition

Must be flexible enough to account with all kinds of variations
Biological Vision

“Researchers have been interested for years in trying to copy biological vision systems, simply because they are so good” ~ David Hogg - computer vision expert at Leeds University, UK

- Highly optimized over millions of years of evolution, developing complex neural structures to represent and process stimuli

- Superiority of biological vision systems is only partially understood

- Hardware architecture and the style of computation in nervous systems are fundamentally different
Imagine that the U.S. is preparing for the outbreak of an unusual Asian disease, which is expected to kill 600 people. Two alternative programs to combat the disease have been proposed. Assume that the exact scientific estimate of the consequences of this program are as follows:

If Program A is adopted 400 people will die.

If Program B is adopted there is 1/3 probability that nobody will die and 2/3 probability that 500 people will die.

Which of the two programs would you favor?

Program A

Program B
Seeing is a way of acting
Active Vision

Inspired by the vision systems of natural organisms that have been evolving for millions of years

In contrast to standard computer vision systems, biological organisms actively interact with the world in order to make sense of it

Humans and also other animals do not look at a scene in fixed steadiness. Instead, they actively explore interesting parts of the scene by rapid saccadic movements
Creating Active Vision Systems

Evolutionary Robotics Approach
New technique for the automatic creation of autonomous robots

Inspired by the Darwinian principle of selective reproduction of the fittest

Views robots as autonomous artificial organisms that develop their own skills in close interaction with the environment and without human intervention

Drawing heavily on biology and ethology, it uses the tools of neural networks, genetic algorithms, dynamic systems, and biomorphic engineering
Artificial neural networks (ANNs) are very powerful brain-inspired computational models, which have been used in many different areas such as engineering, medicine, finance, and many others.

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution.
Related Research
Mars Rover obstacle avoidance (Peniak et al.)
Method

Evolution of the active vision system for real-world object recognition
- training the system in a parallel manner on multiple objects viewed from many different angles and under different lighting conditions

Amsterdam Library of Object Images (ALOI)
- provides a color image collection of one-thousand small objects
- recorded for scientific purposes
- systematically varied viewing angle, illumination angle, and illumination color

Active Vision Training
- trained on a set of objects from the ALOI library
- each genotype is evaluated during multiple trials with different randomly rotated objects and under varying lighting conditions
- evolutionary pressure provided by a fitness function that evaluates overall success or failure of the object classification
- trained on increasingly larger number of objects

Active Vision Testing
- robustness and resiliency of recognition of the dataset
- generalization to previously unseen instances of the learned objects
Experimental Setup

Recurrent Neural Network
- Inputs: 8x8 neurons for retina, 2 neurons for proprioception (x,y pos)
- No hidden neurons
- Outputs: 5 object recognition neurons, 2 neurons to move retina (16px max)

Genetic Algorithm
- Generations: 10000
- Number of individuals: 100
- Number of trials: 36+16 (object rotations + varying lighting conditions)
- Mutation probability: 10%
- Reproduction: best 20% of individuals create new population
- Elitism used (best individual is preserved)
Experimental Setup

- Each individual (neural network) could freely move the retina and read the input from the source image (128x128) for 20 steps.

- At each step, neural network controlled the behavior of the system (retina position) and provide recognition output.

- The recognition output neuron with the highest activation was considered the network’s guess about what the object was.
  - Fitness function = number of correct answers / number of total steps.
GPUs were used to accelerate:

- Evolutionary process – parallel execution of trials
- Neural Network – parallel calculation of neural activities
Results

- Fitness can not reach 1.0 since it takes few time-steps to recognize an object
- All objects are correctly classified at the end of the each test
Evolved Behavior
Software development
Heterogeneous computing

Device

GPU

Use GPU to Parallelise

Compute-Intensive Functions

Application Code

Rest of Sequential CPU Code

Host

CPU
What is Aquila?

- Heterogeneous software architecture for the development of modules loosely coupled to their graphical user interfaces

- Provides simple and user friendly GUI client
  - Distribute, control and visualise existing modules
  - Generate new modules
  - Monitor connected server
  - Tools

- Modules
  - Run heterogeneous CPU-GPU code doing the actual work
What is Aquila?

- Developed in C++ and CUDA

- Cross-platform
  - Linux
  - OSX
  - Windows

- Dependencies
  - Qt
  - YARP
  - CUDA
Existing Aquila Ecosystem

MTRNN
Multiple Time-scales Recurrent Neural Network

SOM
Self-organising Map

ERA
Epigenetic Robotics Architecture

Tracker
Object tracking

ESN
Echo State Networks

MTRNN Benchmark Example
2xGTX580(P2P) vs 8 core Intel Xeon
"Imagination is the highest form of research"
Albert Einstein

Questions?