What kind of hardware do we need for pervasive AI?

Alex Yakovlev
Head of microSystems Group
School of Engineering
Newcastle University
https://www.ncl.ac.uk/engineering/research/eee/microsystems/
http://async.org.uk/

CAIR AI Workshop 7 February 2019
Swarm of devices – Future of ICT

Trillions of ubiquitous systems (sensors, probes, monitors, actuators, controllers) are being deployed to operate in myriad of places (organisation, human, body part, household, offices, pets) using harvested energy or micro-batteries.
Granularity of intelligence

• Pervasive Intelligence requires reconsidering many balances:
  – Between software and hardware
  – Between power and compute
  – Between analog and digital
  – Between design and fabrication and maintenance
  – ...

• Granularity of time and energy
Granularity of intelligence

Granulation phenomenon:
- Granularity of power
- Granularity of time
- Granularity of data
- Granularity of function

Questions:
- Can we granulate intelligence to minimum?
- What is the smallest level at which we can make cyber-systems learn – in terms of power, time, data and function?
Grand challenge for pervasive hardware AI:

To enable electronic components with an ability to learn and compute in real-life environments with real-power and in real-time

Research Hypothesis:

We should design systems that are energy-modulated and self-timed, with maximally distributed learning capabilities
Energy-modulated computing
Power-modulated multi-layer system

- Multiple layers of the system design can turn on at different power levels (analogies with living organisms’ nervous systems or underwater life, layers of different cost labour in resilient economies)
- As power goes higher new layers turn on, while the lower layers (“back up”) remain active
- The more active layers the system has the more power resourceful it is
Most of the operations here are done by using conventional binary arithmetic, which is not power-adaptive and uses centrally provided Power and Clocks – hence poor power-proportionality and robustness.
Proposed approach

• Event-driven, robust to power and timing fluctuations
• Decentralised Tsetlin Automata (TAs) for learning on demand
• Mixed digital-analog compute where elements are enabled and controlled by individual TAs
• Natural approximation in its nature, both in learning and compute
• Asynchronous logic for h/w implementation
Why Tsetlin Automata?

Hypothesis: TAs provide a minimalist (energy-wise and robustness-wise) way to adaptation

- TAs can act as generators of control signals, naturally enabling:
  - Compute function shaping (include/exclude parts of compute)
  - Distributed Power and Clock gating

- TAs can be easily implemented in hardware:
  - Directly (in digital or mixed signal way)
  - Via microprogrammable structures, using transition-output tables in memory and simple access microcode in h/w
  - Can be prototyped in FPGAs and on microcontrollers

- TAs can be made:
  - With fixed structure (linear tactics)
  - With variable structure or with tunable memory depths
  - For stationary and varying environments
The idea of Tsetlin Machine:
https://www.dropbox.com/s/usk78fj381k2qrw/Tsetlin_Machine_170119.pdf?dl=0
**Tsetlin Automaton: async design**

4-state TA\(_k\):  

![Diagram of a 4-state Tsetlin Automaton](image)

**Action 1 (Exclude \(x_k\)):**  
\[ X_{11} = x_{12}' \cdot x_{21}' \cdot x_{22}' + p \cdot A_1 \]  
\[ X_{12} = r \cdot A_1 + x_{11}' \cdot x_{12} \]  
\[ A_1 = p \cdot (A_2' \cdot x_{21} + x_{12} + A_1') + r \cdot (x_{12} + x_{11}) \]

**Action 2: Include \(x_k\):**  
\[ X_{21} = x_{11}' \cdot x_{12}' \cdot x_{22}' + p \cdot A_1 \]  
\[ X_{22} = r \cdot A_2 + x_{21}' \cdot x_{22} \]  
\[ A_2 = p \cdot (A_1' \cdot x_{11} + x_{22} + A_2') + r \cdot (x_{22} + x_{21}) \]

**Logic implementation (equations obtained from our tool Workcraft – next slides):**

**Approximate performance:**
- response time can be in the order of 100ps
- energy cost in the order of 100fJ per action
Clause and function computation

- Clause can produce either levels or pulses
- We can use various energy-efficient ways of summing +1’s and -1 as, or accumulating events (e.g. up and down counters)

Feedback computation (after logic minimisation)

**Reward:** \( R_{kj} = C_j \cdot \text{Inc}_k \cdot \text{Type I} \)

**Penalty:** \( P_{kj} = x_k \cdot C_j \cdot \text{Exc}_k \cdot \text{Type I} + x_k' \cdot C_j \cdot \text{Type II} \)

**Inaction:** \( I_{kj} = C_j' + x_k' \cdot \text{Type I} + x_k \cdot \text{Type II} \)

We assume:
- \( \text{Exclude} = \text{Not}(\text{Include}); \text{Exc} = 0/\text{Inc} = 1 \)
- \( \text{Type I} = \text{Not (Type II)}; \text{Type I} = 0/\text{Type II} = 1 \)

Interpretation:
- We reward \( TA_k \) to Include \( x_k \) in \( C_j \) for Type I
- We penalise \( TA_k \) to Exclude \( x_k \) when \( x_k = C_j = 1 \) for Type I or to make \( x_k = 0 \) and \( C_j = 1 \) for Type II
- We maintain state for \( TA_k \) when \( C_j = 0 \) or \( x_k = 0 \) for Type I or \( x_k = 1 \) for Type II
Designing the compute part: mixed analog-digital (DATE’19 paper)

- PMOS and NMOS act as voltage divider.
- Output voltage is inversely proportional to input duty cycle.

The circuit below adds 3 inputs multiplied by 3-bit weights (can be generated by Tsetlin Automata)

Robustness to Frequency and Voltage variations:
Workcraft: Toolkit for designing asynchronous circuits
Example STG for 4-state Tsetlin Automaton
Newcastle people involved in research on real-power computing, asynchronous design, approximate computing and pervasive AI

- Microsystems Group at Engineering:
  Dr Alex Bystrov, Dr Andrey Mokhov, Dr Reza Ramezani, Dr Rishad Shafik, Dr Danil Sokolov, Dr Ahmed Soltan, Dr Fei Xia; PhD students: Sergey Mileiko, Thanasin Bunnam, Adrian Wheeldon

- Computing Science:
  Dr Victor Khomenko, Prof Maciej Koutny

- Academic collaboration:
  Prof Steve Furber’s group in The Uni of Manchester

- Industry collaboration:
  Temporal Computing and Dialog Semiconductor
Thank you!

More information:

https://www.ncl.ac.uk/engineering/research/eee/microsystems/
http://async.org.uk/
http://workcraft.org
https://blogs.ncl.ac.uk/alexyakovlev/