Continuous-space model of computation

Damien Woods

Thomas J. Naughton, Paul gibson

TASS research group,
Department of Computer Science, NUI Maynooth,
Ireland
Motivations

• Investigate computational power of a novel model of computation
• Relationship between models of computation and scientific theories
CSM definition

- Images are the basic data units in the CSM

- A **complex-valued image** (or simply, an image) is a complex-valued function on the real unit square

\[ f : [0, 1] \times [0, 1] \rightarrow \mathbb{C} \]
CSM definition

A **continuous space machine** is a quintuple $M = (D, L, I, P, O)$, where

- $D = (m, n)$, $D \in \mathbb{N} \times \mathbb{N}$: grid dimensions
- $L = ((s_\xi, s_\eta), (a_\xi, a_\eta), (b_\xi, b_\eta))$: addresses $sta, a, b$
- $I = \{ (\iota_1, \iota_1), \ldots, (\iota_k, \iota_k) \}$: addresses of the $k$ input images
- $P = \{ (\pi_1, p_{1\xi}, p_{1\eta}), \ldots, (\pi_r, p_{r\xi}, p_{r\eta}) \}$, $\pi_j \in \{ h, v, \ast, \cdot, +, \rho, \text{st, ld, br, hlt} \} \cup \mathcal{N}$: the $r$ programming symbols and their addresses
- $O = \{ (o_{1\xi}, o_{1\eta}), \ldots, (o_{l\xi}, o_{l\eta}) \}$: addresses of the $l$ output images.

Also, $(s_\xi, s_\eta), (a_\xi, a_\eta), (b_\xi, b_\eta), (\iota_k, \iota_k), (p_{r\xi}, p_{r\eta}), (o_{l}, o_{l}) \in \{0, \ldots, m - 1\} \times \{0, \ldots, n - 1\}$ for all $k, l \in \{1, \ldots, k\}, r, l' \in \{1, \ldots, r\}, l', l'' \in \{1, \ldots, l\}$.

A **CSM configuration** is a pair $\langle c, g \rangle$
- $c$ is an address called the control, $g = ((i_0, 0, 0), \ldots, (i_{m-1}, n-1, m-1, n-1))$
CSM operations

\begin{align*}
\mathbf{h} & : \text{horizontal 1-D Fourier transform} \\
\mathbf{v} & : \text{vertical 1-D Fourier transform} \\
\ast & : \text{complex conjugate} \\
\cdot & : \text{multiply two images (point by point multiplication)} \\
+ & : \text{add two images (complex addition)} \\
\rho & : \text{image filter using lower and upper amplitude threshold images } z_l \text{ and } z_u.
\end{align*}
CSM operations

\[
\text{st } p1 \ p2 \ p3 \ p4 : p1, p2, p3, p4 \in \mathbb{N}; \text{ copy the image in } a \text{ into the rectangle of images whose bottom left-hand corner address is } (p1, p3) \text{ and whose top right-hand corner address is } (p2, p4).
\]

\[
\text{ld } p1 \ p2 \ p3 \ p4 : p1, p2, p3, p4 \in \mathbb{N}; \text{ copy into } a \text{ the rectangle of images whose bottom left-hand corner address is } (p1, p3) \text{ and whose top right-hand corner address is } (p2, p4).
\]
CSM operations

\[
\text{br} \quad p_1 \quad p_2 \quad : \quad p_1, p_2 \in \mathbb{N} \text{; unconditionally branch to the image at address } (p_1, p_2).
\]

\[
\text{hlt} \quad : \quad \text{halt.}
\]

: move to the next grid image (ignore images that do not represent a programming symbol).
Complexity measures

Computational complexity measures are used to analyse CSM instances

- **TIME** = number of computation steps
- **SPACE** = number of images in grid
- **RESOLUTION** = max spatial resolution, relative to some unit image
- **RANGE** = number of bits required to represent the values in the set $f'$ where

$$f : [0, 1] \times [0, 1] \mapsto f' \subseteq \mathbb{C}$$
Symbols, words, languages

\[ \{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\} \]

\[ L \subseteq \{0, 1\}^* \]

Given \( L \) and \( w \in \{0, 1\}^* \), is \( w \in L \)?
Representing data as images

\( \psi \in \{0, 1\} \) is represented by the binary symbol image \( f_\psi \),

\[
f_\psi(x, y) = \begin{cases} 
1, & \text{if } x, y = 0.5, \psi = 1 \\
0, & \text{otherwise}
\end{cases}
\]

\( w = w_1w_2 \cdots w_k \in \Sigma^+ \) is represented by the binary list image \( f_w \),

\[
f_w(x, y) = \begin{cases} 
1, & \text{if } x = \frac{2i-1}{2k}, y = 0.5, w_i = 1 \\
0, & \text{otherwise}
\end{cases}
\]

\( w = w_1w_2 \cdots w_k \in \{0, 1\}^+ \) is represented by the binary stack image \( f_w \),

\[
f_w(x, y) = \begin{cases} 
1, & \text{if } x = 1 - \frac{3}{2^{k-i+2}}, y = 0.5, w_i = 1 \\
0, & \text{otherwise}
\end{cases}
\]

List/stack image \( f_w \) is said to have length \( k \in \mathbb{N} \). \((f_w, k)\) uniquely represents \( w \).
Representing data as images

$r \in \mathbb{R}$ is represented by the real number image $f_r$

$$f_r(x, y) = \begin{cases} r, & \text{if } x, y = 0.5 \\ 0, & \text{otherwise} \end{cases}$$

$R \times C$ matrix $A$, with real-valued components $a_{ij}$, is represented by the $R \times C$ matrix image $f_A$

$$f_A(x, y) = \begin{cases} a_{ij}, & \text{if } x = 1 - \frac{1+2k}{2j+k}, y = \frac{1+2l}{2i+l} \\ 0, & \text{otherwise} \end{cases}$$
Language deciding by CSM

CSM $M_L$ decides $L \subseteq \Sigma^*$ if $M_L$ has initial configuration $\langle c_s, g_s \rangle$ and final configuration $\langle c_h, g_h \rangle$, and the following hold:

- sequence $g_s$ contains the two input elements $(f_w, \iota_1\xi, \iota_1\eta)$ and $(f_{1|w|}, \iota_2\xi, \iota_2\eta)$
- $g_h$ contains the output element $(f_1, o_1\xi, o_1\eta)$ if $w \in L$
- $g_h$ contains the output element $(f_0, o_1\xi, o_1\eta)$ if $w \notin L$
- $\langle c_s, g_s \rangle \vdash^*_M \langle c_h, g_h \rangle$, for all $w \in \Sigma^+$.

Where $f_w$ is the binary stack image representation of $w \in \Sigma^+$, $f_{1|w|}$ is the unary stack image representation of the unary word $1|w|$. Images $f_0$ and $f_1$ are the binary symbol image representations of the symbols 0 and 1, respectively.
Analog recurrent neural networks

- Finite size feedback first-order neural networks with real weights

- Model of analog computation, by Siegelmann and Sontag, TCS, 1994

\[
x_i(t + 1) = \sigma \left( \sum_{j=1}^{N} a_{ij} x_j(t) + \sum_{j=1}^{M} b_{ij} u_j(t) + c_i \right), \quad i = 1, \ldots, N
\]

\[
\sigma(x) = \begin{cases} 
0, & \text{if } x < 0 \\
x, & \text{if } 0 \leq x \leq 1 \\
1, & \text{if } x > 1 .
\end{cases}
\]
## CSM simulation of ARNN

<table>
<thead>
<tr>
<th></th>
<th>sta</th>
<th>( \bar{u} )</th>
<th>( \Sigma AX )</th>
<th>( \Sigma BU )</th>
<th>( t_1 )</th>
<th>a</th>
<th>b</th>
<th>( t_2 )</th>
<th>( O )</th>
<th>( I )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>br</td>
<td>0</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ii)</td>
<td>ld</td>
<td>( I )</td>
<td>st ( t_{1a} )</td>
<td>st ( I )</td>
<td>ld ( t_1 )</td>
<td>st ( \bar{u} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(iii)</td>
<td>ld</td>
<td>( \bar{x} )</td>
<td>whl ( N-1 )</td>
<td>st ( t_3 )</td>
<td>ld ( \bar{x} )</td>
<td>ld at ( t_3 )</td>
<td>end</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(iv)</td>
<td>st</td>
<td>b</td>
<td>ld ( A )</td>
<td>( \cdot )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(v)</td>
<td>st</td>
<td>( t_2 )</td>
<td>ld ( 0 )</td>
<td>whl ( N-1 )</td>
<td>st ( t_1 )</td>
<td>ld ( t_2 )</td>
<td>st ( bt_2 )</td>
<td>ld ( t_1 )</td>
<td>+</td>
<td>end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(vi)</td>
<td>st</td>
<td>b</td>
<td>ld ( B )</td>
<td>( \cdot )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(vii)</td>
<td>st</td>
<td>( t_2 )</td>
<td>ld ( 0 )</td>
<td>whl ( M-1 )</td>
<td>st ( t_1 )</td>
<td>ld ( t_2 )</td>
<td>st ( bt_2 )</td>
<td>ld ( t_1 )</td>
<td>+</td>
<td>end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(viii)</td>
<td>ld</td>
<td>( \Sigma AX )</td>
<td>st</td>
<td>b</td>
<td>ld</td>
<td>( \Sigma BU )</td>
<td>+</td>
<td>st</td>
<td>( c )</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ix)</td>
<td>( \rho )</td>
<td>0</td>
<td>1</td>
<td>st ( t_3 )</td>
<td>ld</td>
<td>0</td>
<td>st ( t_1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x)</td>
<td>whl</td>
<td>( N-1 )</td>
<td>ld ( t_3 )</td>
<td>st ( at_3 )</td>
<td>ld ( t_{1a} )</td>
<td>st ( t_1 )</td>
<td>end</td>
<td>ld ( t_3 )</td>
<td>st</td>
<td>ab</td>
<td>ld ( t_{1a} )</td>
<td>st ( t_1 )</td>
</tr>
<tr>
<td>(xi)</td>
<td>whl</td>
<td>( N-1 )</td>
<td>ld ( t_1 )</td>
<td>st ( t_{1a} )</td>
<td>ld</td>
<td>ab</td>
<td>st</td>
<td>b</td>
<td>end</td>
<td>ld ( t_1 )</td>
<td>st ( t_{1a} )</td>
<td>ld</td>
</tr>
<tr>
<td>(xii)</td>
<td>st</td>
<td>b</td>
<td>ld ( P )</td>
<td>( \cdot )</td>
<td>st</td>
<td>( t_1 )</td>
<td>ld</td>
<td>( O )</td>
<td>ld ( t_{1a} )</td>
<td>st</td>
<td>( O )</td>
<td></td>
</tr>
</tbody>
</table>

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | \( \ldots \) | \( \bar{x} \) | \( N-1 \) | \( M-1 \) | A | B | \( c \) | P |

Note: address \( t_3 \) is located at grid coordinates \((10, 14)\)
CSM simulation of ARNN

(i) \( \overline{u} := I.\text{pop()} \)
(ii) \( \overline{X} := \text{push } \overline{x} \text{ onto itself vertically } N - 1 \text{ times} \)
(iii) \( \overline{AX} := A \cdot \overline{X} \)
(iv) \( \Sigma \overline{AX} := \Sigma_{i=1}^{N} (\overline{AX}.\text{pop}_i()) \)
(v) \( \overline{U} := \text{push } \overline{u} \text{ onto itself vertically } N - 1 \text{ times} \)
(vi) \( \overline{BU} := B \cdot \overline{U} \)
(vii) \( \Sigma \overline{BU} := \Sigma_{i=1}^{M} (\overline{BU}.\text{pop}_i()) \)
(viii) affine-comb := \( \Sigma \overline{AX} + \Sigma \overline{BU} + \overline{c} \)
(ix) \( \overline{x}' := \rho(\text{affine-comb, } 0, 1) \)
(x) \( \overline{x} := (\overline{x}')^T \)
(xi) \( O.\text{push } (\overline{P} \cdot \overline{x}) \)
(xii) goto step (i)
CSM decides any \( L \subseteq \{0, 1\}^+ \)

- Formal nets; a class of ARNNs that decide languages

- For each \( L \subseteq \{0, 1\}^+ \) there exists formal net \( \mathcal{F}_L \) that decides \( L \)

- We carry this result over to the CSM by giving a CSM \( \mathcal{D} \) that
  - is consistent with the definition of language deciding by CSM
  - decides \( L \) by simulating \( \mathcal{F}_L \)
### CSM $\mathcal{D}$

| sta | $\bar{u}$ | $\Sigma AX \Sigma BU$ | $t_1$ | $a$ | $b$ | $t_2$ | $f_{\psi w}$ | $fw$ | $f_{1|w|}$ | 0 | 1 |
|-----|----------|---------------------|------|----|----|------|-------------|-----|-----------|----|---|
| br  | 0        | 18                  |      |    |    |      |             |     |           |    |   |
| ld  | $f_w$    | st b ld 0 st t_2    |      |    |    |      |             |     |           |    |   |
| whl | $f_{1|w|}$ | ld b st $t_1 a$ st | b   | ld | $t_2$ | ld | $t_1 a$ st | $t_2$ | end st | $f_w$ |
| ld  | $f_w$    | st $t_1 a$ st $f_w$ | ld  | $t_1$ | st | 13  | 16        | 14  | 14        | 17 | 14 |
| st  | b        | ld $f_{1|w|}$ st $f_1$ | $t_1 a$ | st | $f_{1|w|}$ | ld | $t_1$ | st  | 13 | 16  | 14  | 14 |
| ld  | 12       | 15 14 14 + st $\bar{u}$ | br  | 0 | 13 |     |           |     |           |    |   |
| st  | b        | ld $P$ st $t_3$ st | $t_1$ | whl | $O_v$ | st | $t_1 a$ end | ld | $t_1$ | br  | 0 | $\hat{a}$ |
| ld  | $t_3$    | whl $O_d$ st $t_1 a$ end | ld | $t_1$ | st | $f_{\psi w}$ | hlt |     |     |     |     |     |
| br  | 0        | 16                  |      |    |    |      |             |     |           |    |   |

Note: address $t_3$ is located at grid coordinates $(10, 18)$
CSM $\mathcal{D}$

CSM $\mathcal{D}$, in the worst case, requires

- linear TIME

$$T(N, M, T(|w|), |w|, d, v) = 12|w| + 7d + (49N + 7v + 67)T(|w|) + 22$$

- exponential RESOLUTION

$$R(N, M, T(|w|), |w|, d, v) = \max(2^{|w|}, 2^{(2N-2)})$$

- constant SPACE

- infinite RANGE $\omega$
Needle in haystack problem

• Given \( w \in 0^*10^* \), what is the index of the ‘1’ in \( w \)?

• Conventional (serial) computer requires \( \Theta(n) \) steps, worst case

• Grover’s quantum computer algorithm requires \( O(\sqrt{n}) \) comparisons, average case

• CSM algorithm requires \( \Theta(\log_2 n) \) steps, worst case
Needle in haystack problem

<table>
<thead>
<tr>
<th>i1</th>
<th>i2</th>
<th>f0</th>
<th>f1</th>
<th>sta</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td></td>
<td></td>
<td></td>
<td>br</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>ld</td>
<td>e</td>
<td>st</td>
<td>de</td>
<td>br</td>
<td>0</td>
<td>*d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ld</td>
<td>i2</td>
<td>st</td>
<td>e</td>
<td>ld</td>
<td>f0</td>
<td>st</td>
<td>c</td>
<td>br</td>
</tr>
<tr>
<td>2</td>
<td>hv</td>
<td>st</td>
<td>b</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ld</td>
<td>il</td>
<td>st</td>
<td>ab</td>
<td>br</td>
<td>0</td>
<td>2</td>
<td></td>
<td>ld</td>
</tr>
<tr>
<td>0</td>
<td>ld</td>
<td>c</td>
<td>hlt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ld</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Needle in haystack problem

procedure search(i1, i2)
e := i2
c := f₀
while (e.pop() = f₁)
  rescale i₁ over both image a and image b
  FT, square, and FT image a
  if (a = f₁)
    i₁ := LHS of i₁
    c.push(f₀)
  else /* a = f₀ */
    i₁ := RHS of i₁
    c.push(f₁)
  end if
end while
a := c
end procedure
Needle in haystack problem

On input word of length $n$, CSM needle in haystack algorithm, in the worst case, requires

- log time, $T(n) = 23 \log_2 n + 11$
- linear resolution, $R(n) = 2n$
- constant space
- constant range
Future work

• Prove further computability and complexity results

• Investigate (computationally less powerful) variants of the CSM
Summary

• Presented the continuous space machine
• Analog recurrent neural network simulation
• A log time solution to the needle in haystack problem
• Acknowledgements: TASS, IRCSET

This work is in submission to TCS, available as NUIM-CS-TR-04-2003.