Computation with reaction networks

Badal Joshi
California State University San Marcos

Mathematical Modeling of Chemical Reaction Networks
Monday, July 24, 2023

Autonomous computation in a cell

- **Goal**: Construct synthetic reaction networks that perform pre-determined task.
- Field is called "computation with chemical reaction networks".
- Use sub-cellular substrate (eg. proteins) to devise a "chemical" computer instead of a silicon-based one.
- Tasks can be simple, eg. arithmetic or complex, eg. machine learning.

Elementary operations

- Implementing identity function.
- Suppose we want to assign a pre-determined value a to the species X.
- Think of species A as input species whose concentration a is input value.
- Think of species X as output species whose concentration x will be the output of the computation.
- **Want:** Concentration of input species never changes. Output species converges to the correct value.

Elementary operations: identity

Reference: Buisman, Eikelder, Hilbers, Liekens: Computing algebraic functions with biochemical reaction networks

$$A \xrightarrow{k_1} A + X$$
$$X \xrightarrow{k_2} 0$$

- X degrades autonomously with rate constant k_2.
- A catalyzes production of X with rate constant k_1.
- We do not include the resource (raw material from which X is assembled) and waste (that X breaks down into).
- Set rate constants k_1 and k_2 to 1.

$$\dot{a} = 0$$
$$\dot{x} = a - x$$

Clearly at steady state $x = a = a(0)$.

Elementary operations: identity

$$A \xrightarrow{k_1} A + X$$
$$X \xrightarrow{k_2} 0$$

$$\dot{a} = 0$$
$$\dot{x} = k_1 a - k_2 x$$

Explicit solution:

$$a(t) = a(0)$$
$$x(t) = \frac{1}{k_2}\left(k_1 a_0 - (k_1 a_0 - k_2 x_0)\, e^{-k_2 t}\right)$$

$$x(t) \xrightarrow{t \to \infty} \frac{k_1}{k_2} a_0$$

- **Note:** The convergence happens at an exponential rate.
- We implemented "multiplying by a fixed constant".

Digital vs. Analog

- Usual silicon computers are digital.
- Boolean arithmetic and discretization are used to approximate continuous processes.
- A "chemical computer" is nonlinear and analog.
- When performing arithmetic, we are implementing discrete operations which are approximated with continuous processes.

Elementary operations: multiplication

$$A + B \xrightarrow{k_1} A + B + X$$
$$X \xrightarrow{k_2} 0$$

$$\dot{a} = 0, \quad \dot{b} = 0$$
$$\dot{x} = k_1 ab - k_2 x$$

Explicit solution:

$$a(t) = a(0)$$
$$x(t) = \frac{1}{k_2}\left(k_1 a_0 b_0 - (k_1 a_0 b_0 - k_2 x_0)\, e^{-k_2 t}\right)$$

$$x(t) \xrightarrow{t \to \infty} \frac{k_1}{k_2} a_0 b_0$$

Elementary operations: division

Exercise: How would you implement division?

Elementary operations: division

Exercise: How would you implement division?

$$A \xrightarrow{k_1} A + X$$
$$B + X \xrightarrow{k_2} B$$

$$\dot{x} = k_1 a - k_2 b x$$

Explicit solution:

$$x(t) = \frac{1}{k_2 b_0} \left(k_1 a_0 - (k_1 a_0 - k_2 b_0 x_0)\, e^{-k_2 b_0 t} \right)$$

$$x(t) \xrightarrow{t \to \infty} \frac{k_1 a_0}{k_2 b_0}$$

Elementary operations: addition

Exercise: How would you implement addition?

Elementary operations: addition

Exercise: How would you implement addition?

$$A \xrightarrow{1} A + X$$

$$B \xrightarrow{1} B + X$$

$$X \xrightarrow{1} 0$$

$$\dot{x} = a + b - x$$

Elementary operations: subtraction

Exercise: How would you implement subtraction?

Think first about **why** subtraction is trickier than all previous operations?

Elementary operations: subtraction

Exercise: How would you implement subtraction?
Think first about **why** subtraction is trickier than all previous operations?

- For the previous operations, the input was 1 or 2 positive reals and the output was a positive real. Positive reals can be coded as concentrations of species.
- Applying subtraction to two positive reals can result in a negative real. How to handle this?
- Define proper subtraction (also called monus)

$$a \dot{-} b = \begin{cases} 0 & \text{if } a < b \\ a - b & \text{if } a \geq b. \end{cases}$$

- Many possibilities. Sometimes dual rail construction. One variable tracks positive part while another tracks negative part. $x = x^+ - x^-$ with $x^+, x^- > 0$.

Elementary operations: subtraction

- Non-dual rail construction.
- We need multiple dynamic (non-input) variables.

$$A \xrightarrow{k_1} A + X$$

$$B \xrightarrow{k_1} B + Y$$

$$X \xrightarrow{k_1} 0$$

$$X + Y \xrightarrow{k_2} 0$$

$$\dot{x} = k_1 a - k_1 x - k_2 xy$$

$$\dot{y} = k_1 b - k_2 xy$$

- Explicit solution seems difficult.
- Need nonlinear analysis (note the nonlinear term xy).

Elementary operations: subtraction

Exercise: Find the positive steady states.

Elementary operations: subtraction

- Steady states are

$$x^* = a_0 - b_0,$$
$$y^* = \frac{k_1 b_0}{k_2(a_0 - b_0)}$$

Exercise: Calculate the Jacobian matrix at the steady state and determine its linear stability.

Elementary (unary) operation: nth root

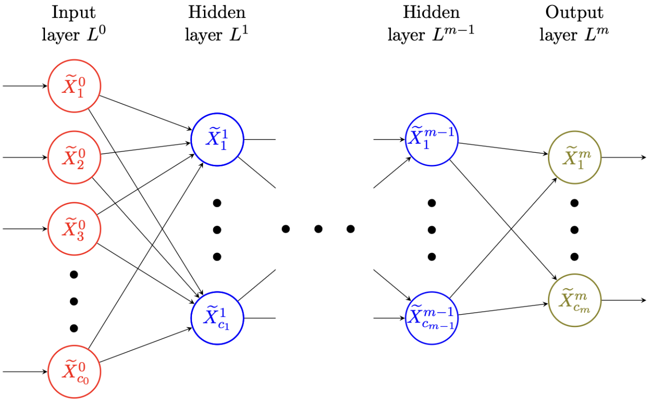
$$A \xrightarrow{k_1} A + X$$
$$nX \xrightarrow{k_2} 0$$

$$\dot{x} = k_1 a - k_2 x^n$$

Exercises

1. Show that the steady state for subtraction is exponentially stable. Does this prove that for all initial values we will have convergence to the steady state?
2. Some of the operations (eg. multiplication, nth root) required higher-than-bimolecular complexes. Can you modify the construction such that only bimolecular complexes are involved?

Structure of a Neural Network



Source: David F. Anderson, Badal Joshi, and Abhishek Deshpande. On reaction network implementations of neural networks, *Journal of Royal Society Interface*, Vol. 18, Issue 177, (April 2021), arXiv.

Hardwired neural network

A pair of consecutive layers $L^{\ell-1}$ and L^{ℓ} along with all edges between the two layers, encode a function $\psi^{\ell} : \mathbb{R}^{c_{\ell-1}} \to \mathbb{R}^{c_{\ell}}$ which is defined via

$$\psi^{\ell}(y) = \varphi(W^{\ell} y + \beta^{\ell}).$$

Taking compositions, a hardwired neural network is then simply a visual representation for the function $\Psi_{(G,\mathcal{P},\varphi)} : \mathbb{R}^{c_0} \to \mathbb{R}^{c_m}_{\geq 0}$ defined via

$$\Psi_{(G,\mathcal{P},\varphi)} = \psi^{m} \circ \psi^{m-1} \circ \cdots \circ \psi^{1}.$$

The function φ is called activation function.

Hardwired neural network

Commonly used activation functions:

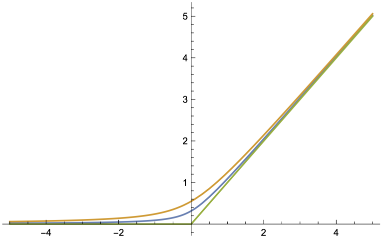
1. ReLU (Rectified Linear Units)

$$\varphi_0(y) = \max(0, y)$$

2. For $h \geq 0$,

$$\varphi_h(y) = \frac{1}{2}\left(y + \sqrt{y^2 + 4h}\right).$$

Plugging $h = 0$ in φ_h gives ReLU. φ_h is smoothed version of ReLU while being strictly monotone.



Hardwired neural network

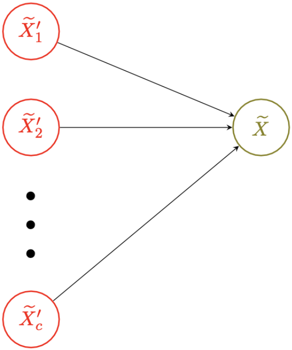
Which aspect of neural network is implemented chemically?	Chemical implementation of a single directed edge $(\widetilde{X}', \widetilde{X})$ of the neural network	Which term results in the ODE for the species X?
Closeness to ReLU	$H \longrightarrow H + X$	h
Input $\widetilde{X}'$ and weight of the edge $(\widetilde{X}', \widetilde{X})$	$X' + W^+ + X \longrightarrow X' + W^+ + 2X$ $X' + W^- + X \longrightarrow X' + W^-$	$(w^+ - w^-)x'x$, where $w := w^+ - w^-$ implements the edge weight
Additive node bias of $\widetilde{X}$	$B^+ + X \longrightarrow B^+ + 2X$ $B^- + X \longrightarrow B^-$	$(b^+ - b^-)x$, where $b := b^+ - b^-$ implements the node bias
q-polynomial decay Stability/convergence from ∞	$qX \longrightarrow X$ $(q > 1)$	$(q-1)x^q$

Chemical implementation of a single directed edge $(\widetilde{X}', \widetilde{X})$ along with nodes $\widetilde{X}'$ and $\widetilde{X}$ of the neural network.

$$\frac{d}{dt}x(t) = h + \left(\left(b^+ - b^-\right) + (w^+ - w^-)x'\right) x(t) - (q-1)x(t)^q.$$

Hardwired neural network

$$\frac{d}{dt}x(t) = h + \left((b^+ - b^-) + \sum_{i=1}^{c} (w^+_{x'_i,x} - w^-_{x'_i,x}) x'_i \right) x(t) - x(t)^2$$
$$= h + \rho_x x(t) - x(t)^2$$



Step two: node $\widetilde{X}$ along with all its c inputs.

Thank you!