Identifying biological circuits capable of specific functionalities using ML-based ODE solutions

Karthik Raman (IIT Madras) & Santi Adavani (RocketML)

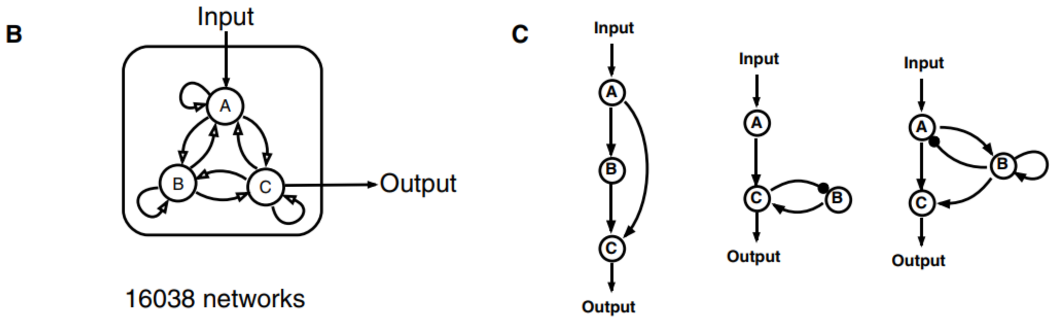
Background

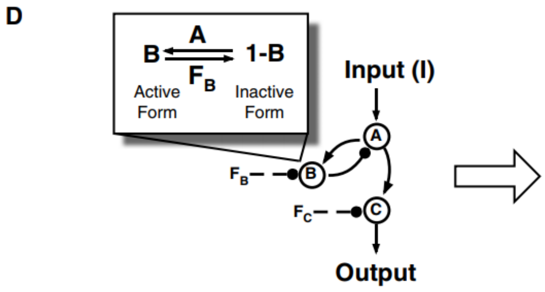
Living cells consist of complex networks of metabolism, signalling and regulation. With the advent of synthetic biology, which involves the *de novo* design of biological networks, a common question that arises is as to how one designs a biological network that performs a specific function. For example, how does one design a circadian oscillator? Or, how does one design a (tuneable) switch? Also, are there any clear design principles underlying the construction of these systems? Current approaches to designing such systems are based on trial and error, or brute force explorations of the design space[1]. We have also developed systems-theoretic approaches to design such networks[2].

The classic approach by Ma *et al.* (2009)[1] involved the simulation of over 16,000 networks comprising three proteins each, each with 10,000 different parameter realisations. Given this massive complexity, for even a small three-node network, it becomes imperative to develop a more scalable approach, perhaps leveraging ML approaches to solve the ODEs, or even identify topology–parameter combinations.

Methodology

Generating ODEs





$$\frac{dA}{dt} = k_{IA} I \frac{(1-A)}{(1-A)+K_{IA}} - k'_{BA} B \frac{A}{A+K'_{BA}}$$

$$\frac{dB}{dt} = k_{AB} A \frac{(1-B)}{(1-B)+K_{AB}} - k'_{F_B B} F_B \frac{B}{B+K'_{F_B B}}$$

$$\frac{dC}{dt} = k_{AC} A \frac{(1-C)}{(1-C)+K_{AC}} - k'_{F_C C} F_C \frac{C}{C+K'_{F_C C}}$$

This figure from Ma *et al.*[1] shows (B) possible links between three proteins, resulting in 16,038 combinations, i.e. networks, (C) some example realisations from these 16K networks, and (D) an example of translating a network to ODEs, which are then solved and evaluated for their "behaviour", the behaviour in this case being "perfect adaptation", characterised by specific responses to perturbations, etc.

ML-based solution of ODEs

In recent years, several data-driven and data-free approaches for solving partial/ordinary differential equations (PDEs/ODEs) have been proposed. The backbone of these approaches is the use of (deep) neural networks, which have proven to be capable of learning complex non-linear relationships between the inputs and the outputs. Current ML approaches are primarily ***data-driven*** where the training is performed based on existing ODE simulation data. While a large fraction of data-free neural solvers are designed for ***pointwise predictions***[3, 4], i.e., the networks in these cases take as input *t* in temporal domain (0, T], and produces an output vector u, by calculating the value of state variable at each t. They exploit the ideas of automatic differentiation to solve the ODE by minimizing the residual over a set of sampled points t. Due to this implicit representation, these methods do not require entire time discretization and rely on collocating points from the domain randomly. Apart from minimizing the residual, these approaches also satisfy the prescribed initial conditions. Another data-free approach is based on ***full-field predictions***[5,6], where a neural network is trained for a parameter set of a family of ODEs i.e., for a given input control parameter vector the network predicts all state variables for all the time steps in a single shot. In this project, we will explore both data-driven and pointwise and full-field data-free deep learning methods.

Deliverables

- A scalable approach to solve 10K+ ODE systems (all inter-related, with mild variations), with many many parameter realisations

References

1. Ma, W., Trusina, A., El-Samad, H., Lim, W. A. & Tang, C. Defining network topologies that can achieve biochemical adaptation. *Cell* **138**, 760–773 (2009).
2. Bhattacharya, P., Raman, K. & Tangirala, A. K. Discovering adaptation-capable biological network structures using control-theoretic approaches. *PLoS Comput. Biol.* **18**, e1009769 (2022).
3. Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational physics 378 (2019): 686-707.
4. Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E., 2021. DeepXDE: A deep learning library for solving differential equations. SIAM Review, 63(1), pp.208-228.
5. Shah, V., Joshi, A., Ghosal, S., Pokuri, B., Sarkar, S., Ganapathysubramanian, B. and Hegde, C., 2019. Encoding invariances in deep generative models. arXiv preprint arXiv:1906.01626.

6. Botelho, S., Joshi, A., Khara, B., Rao, V., Sarkar, S., Hegde, C., Adavani, S. and Ganapathysubramanian, B., 2020, November. Deep generative models that solve pdes: Distributed computing for training large data-free models. In 2020 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC) and Workshop on Artificial Intelligence and Machine Learning for Scientific Applications (AI4S) (pp. 50-63). IEEE.