Automatic Implementation of Neural Networks Through Reaction Networks—Part I: Circuit Design and Convergence Analysis

Yuzhen Fan, Xiaoyu Zhang, Chuanhou Gao, *Senior Member, IEEE*, and Denis Dochain, *Fellow, IEEE*

***Abstract*—Rapid advancements of artificial neural networks for computer sciences, inspired by biological neuron interaction mechanisms, may be leveraged in reverse to synthetic biology by providing advanced molecular programming paradigms capable of autonomously learning and executing complex tasks. A challenging exploration is to implement neural network functionalities through biochemical reaction networks (BCRNs), a language that is inherently compatible with in vivo, with difficulties especially in constructing an appropriate BCRN that respects computation and information processing steps involved in neural networks. These two-part articles finish programming a nonlinear fully connected neural network (FCNN) with a three-layer structure using BCRNs endowed with mass action kinetics. In part I, we divide the FCNN into assignment, feedforward propagation, judgment, learning, and clear-out modules. Each module is programmed using BCRNs by setting species concentrations to carry information related to all kinds of computations and operations, with the equilibria of some species to represent the computation results and information processing results. The newly formed biochemical neural network exhibits the potential to work independently in vivo. In addition, our construction addresses a critical design gap by integrating two essential modules: the biochemical assignment module that enables iterative input of training samples, and the biochemical judgment module responsible for determining the termination of the training process. We further confirm theoretically that the BCRN system achieves FCNN functionality with exponential convergence to all target computational results. Finally, the performance of this construction is evaluated numerically on binary classification problems.**

Received 5 December 2024; revised 12 March 2025; accepted 19 March 2025. Date of publication 24 March 2025; date of current version 29 September 2025. This work was supported in part by the National Nature Science Foundation of China under Grant 12320101001 and Grant 62303409, and in part by the Jiangsu Provincial Scientific Research Center of Applied Mathematics under Grant BK20233002. Recommended by Associate Editor J. A. Moreno. *(Corresponding author: Chuanhou Gao.)*

Yuzhen Fan and Chuanhou Gao are with the School of Mathematical Sciences, Zhejiang University, Hangzhou 310058, China (e-mail: yuzhen_f@zju.edu.cn; gaochou@zju.edu.cn).

Xiaoyu Zhang is with the School of Mathematics, Southeast University, Nanjing 210096, China (e-mail: Xiaoyu_Z@seu.edu.cn).

Denis Dochain is with ICTEAM, UCLouvain, Bâtiment Euler, Avenue Georges Lemaître 4-6, 1348 Louvain-La-Neuve, Belgium (e-mail: denis.dochain@uclouvain.be).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TAC.2025.

Digital Object Identifier 10.1109/TAC.2025.3554428

***Index Terms*—Biochemical reaction network, computational modules, exponential convergence, mass action kinetics, neural network.**

NOMENCLATURE

Notations	Abbreviations
$\mathbb{R}^n$	n-dimensional real space.
$\mathbb{R}^n_{\geq 0}$	n-dimensional nonnegative real space.
$\mathbb{R}^n_{>0}$	n-dimensional positive real space.
$\mathbb{Z}^n_{\geq 0}$	n-dimensional nonnegative integer space.
$\mathscr{C}(\times;*)$	Set of continuous differentiable functions from $\times$ to $*$.
$\|\cdot\|$	∞-norm in Euclidean space.
$f(A)$	$(f(a_{ij})) \in \mathbb{R}^{n\times n}$ denotes $f(\cdot)$ acts on any element of A with the real value function $f(\cdot)$ and the matrix $A=(a_{ij})_{n\times n}\in\mathbb{R}^{n\times n}$.
$a_{\cdot j}, a_{i\cdot}$	jth column, ith row of the matrix A, respectively.
$\mathbb{1}, \mathbb{0}$	Dimension-suited column vector with all elements to be 1 and 0, respectively.
ξ_l	lth standard basis vector in Euclidean space.
aBCRN	Assignment biochemical reaction network.
BCRN	Biochemical reaction network.
BFCNN	Biochemical fully connected neural network.
CRN	Chemical reaction network.
FCNN	Fully connected neural network.
GAS	Globally asymptotically stable.
jBCRN	Judgment biochemical reaction network.
LAS	Locally asymptotically stable.
lwsBCRN	Linear weighted sum biochemical reaction network.
MAS	Mass action system.
ngBCRN	Negative gradient biochemical reaction network.
pBCRN	Precalculation biochemical reaction network.
SFP	Split factor—multiplication method.
sigBCRN	Sigmoid activation biochemical reaction network.
uBCRN	Update biochemical reaction network.
TP, TN	True positive, true negative.
FP, FN	False positive, false negative.

I. INTRODUCTION

LIVING cells in natural organisms exhibit remarkable adaptability to external stimuli, responding dynamically

to environmental changes by sensing and processing complicated biological signals. These capabilities stem from intricate biomolecular interactions, such as gene regulatory networks and signal transduction pathways, which govern dynamical behaviors within cells. Synthetic biology applies engineering principles to design biological components (e.g., oscillators [1] and toggle switches [2]) and assemble them into more complex biological circuits, imitating natural life behaviors or programming cells with novel functionalities for applications in intelligent medical care [3], energy, and environmental science [4]. Recent advances highlight a "system-level" design, enabling the construction of increasingly complex biochemical reaction circuits to perform sophisticated real-world tasks. Beyond a classical objective in synthetic biology of enhancing robustness typically achieved via engineered molecular control modules, such as regulating protein production under unintended interactions [5] and mitigating disturbances and noise in cellular environments [6], [7], and so on, another crucial aspect is biochemical information processing. It enables biological circuits to recognize molecular patterns [8] and make in vivo decisions based on biological inputs. Natural examples include bacterial chemotaxis [9] and the brain distinguishing complex odor information [10]. Synthetic molecular computation systems are engineered to perform such tasks, namely, embedding computation into a context compatible with the cellular environment where traditional computers are unable to operate, which requires developing the computational capabilities of biochemical kinetics.

DNA molecular computation has emerged as a powerful framework for encoding and processing information at the molecular scale due to the information storage capacity and programmability of synthetic DNA. It primarily refers to utilizing DNA strands as input and output signals and processing signals through specific reaction mechanisms. CRN represent a widely acknowledged mathematical model for describing biomolecular interactions. It has been demonstrated that any CRN equipped with mass action kinetics can be realized as DNA strand displacement reactions (a powerful mechanism for constructing large-scale DNA-based computational systems) [11]. Therefore, CRNs serve as a theoretical bridge to design biological circuits at the system level, especially utilized as a "programming language" [12] in molecular computation to develop advanced computational paradigms. Another notable advantage of CRNs is their ability to store continuous-time information, similar to analog computers. Unlike digital computers, which require discrete media representations (image, audio, video, etc.) and discretization for tasks like solving differential equations, CRN systems can properly encode equations and leverage their convergence to obtain solutions. Currently, traditional DNA computation devices, departing from biological models of computation, imitate simple Boolean paradigms from electronics, which inherently limits their computing power. Biological computing systems performing basic arithmetic operations and differentiation has been well-studied [13], while compiling a molecular computing system to perform complex information processing tasks, including classification and decision-making, remains a challenging issue. In addition, most synthetic circuits are designed for predefined goals. Hence, constructing more advanced biological computing systems exhibiting intelligent learning ability is of great interest, especially applied in smart therapeutics [14], [15], [16], [17].

Artificial neural networks have established a mature computational framework to solve problems in silicon, with parameters trainable based on datasets using mathematically well-defined optimization algorithms, which is akin to the adaptation of biological systems to external environments. Therefore, artificial neural networks are deemed a promising programming paradigm for molecular computation systems capable of learning and performing neuromorphic information-processing functionalities. Previous research [8], [18], [19] has demonstrated a concrete implementation of the feedforward structure of neural networks using DNA strand displacement reactions in wet labs. Moreover, theoretical advancements in programming neural networks through CRN have closely followed and are experiencing rapid development. The perceptron model with rectified linear unit (ReLU) or sigmoid activation functions were realized by molecular sequestration reaction systems and phosphorylation-dephosphorylation cycles, respectively [20], [21], both embedded with negative weights through competitive mechanisms. A class of rate-independent reaction networks was identified to implement neurons with ReLU nonlinearity solely through stoichiometric changes [22]. A general mathematical framework for implementing feedforward computation with a smooth ReLU function was proposed in [23]. However, some of the designed feedforward structures are relatively simple, lacking scalability for more practical applications, and are limited to fixed weight parameters. To devise molecular learning circuits, recent developments have included qualitative results [24] and quantitative implementations [14], [25] of the weight update component.

Admittedly, excellent progress has been made in programming neural networks using CRN systems. However, limitations persist in terms of the autonomy of biological neural networks, efficiency of the realization (here, the realization means the process of implementing the neural networks using CRNs), and theoretical support. Specifically, autonomy refers to the ability of a biological neural network can proceed with training automatically once initialized until completion, as an artificial neural network does on a computer. The crucial part of guaranteeing the automated operation is the existence of an individual biological assignment module, while its absence nowadays causes training samples to be manually fed into chemical neural networks before every iteration by controlling round time. This task performed by the biological assignment module can be regarded as *automatic assignment* in this article. Also, the quantitative design for the learning component is currently limited primarily focusing on programming the gradient descent algorithm by approximating the derivative operator [25], [26] using the difference format. This approach may introduce the corresponding approximation errors and lead to a cumbersome design. Furthermore, the judgment operation for terminating the training process when the preset error precision is satisfied is absent now. In addition, mass action models of chemical systems are polynomial dynamical systems known to exhibit multiple dynamical behaviors [27], [28], [29], [30]. Designing appropriate mass action reaction systems leading to desirable steady states and rapid convergence

speed [23] with given parameters and initial conditions is highly nontrivial. Thus, performing dynamic analyses is essential for guaranteeing the correct construction and these desired properties.

In summary, our main contribution is two-fold. First, we propose a complete biochemical FCNN design that promises to work independently in vivo. This involves a completely novel CRN design for the assignment module utilizing linear transformation, a realization of the sigmoid activation function via modifying an autocatalytic reaction, and the creation of the judgment module controlling the termination of training processes with a bistable reaction system. In addition, we put forward a *split factor—multiplication* method for the multiplication calculation involving any number of factors, enabling the construction of the reaction system, constituted solely by bimolecular reactions (more practical and easier to be realized as DNA strand displacement reactions) with the minimum number of intermediate species, for precise negative gradient calculation, which offers a parallel computation form. In addition to the bimolecular design, this construction handles any combination of factors and is robust to the initial concentrations compared to [14]. Second, we confirm that our designed biochemical systems implement a neural network and have certain desirable properties. The reaction systems in the feedforward module converge to the target output exponentially, and the exponential stable positive equilibrium points of the learning module in the final iteration are the fixed points of the stochastic gradient descent algorithm.

The rest of this article is organized as follows. Section II covers preliminaries about CRN, FCNN, and computation with CRNs. Section III provides a detailed design of every biochemical module, corresponding to specific aspects of FCNN. In Section IV, a comprehensive dynamical analysis of the constructed computational modules is conducted to confirm their performance. The biochemical neural network is applied in three binary classification problems to validate its efficiency in Section V. Finally, Section VI concludes this article.

II. Preliminaries

In this section, we formally introduce the basic concepts of CRNs [31], [32], [33], [34], the FCNN [35], [36] with the specific structure of an input layer, a hidden layer and an output layer, and the computation potential of CRNs as a biochemical programming language.

A. Chemical Reaction Network

Consider a CRN with n species that interact with each other through r reactions

Definition 1 (CRN): A CRN consists of the following three finite sets.

1) *Species set:* $\mathcal{S} = \{X_1, \ldots, X_n\}$ that denotes the subjects participating in the reactions.
2) *Complex set:* $\mathcal{C} = \bigcup_{j=1}^{r}\{C_j, C'_j\}$ whose elements have the form $\sum_{i=1}^{n} a_i X_i$ with $a \in \mathbb{Z}^n_{\geq 0}$ called the complex vector, where a_i is the stoichiometric coefficient of X_i.
3) *Reaction set:* $\mathcal{R} = \bigcup_{j=1}^{r}\{C_j \to C'_j\}$ satisfying that $\forall C_j \to C'_j \in \mathcal{R}$, $C_{\cdot j} \neq C'_{\cdot j}$, and $\forall C_{\cdot j} \in \mathcal{C}$, $\exists C'_{\cdot j} \in \mathcal{C}$ supporting either $C_{\cdot j} \to C'_{\cdot j}$ or $C'_{\cdot j} \to C_{\cdot j}$, where "$\to$" means one complex reacts to another complex.

We often use the triple $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ to represent a CRN.

Based on the above definition, the jth reaction is written as

$$\sum_{i=1}^{n} v_{ij} X_i \longrightarrow \sum_{i=1}^{n} v'_{ij} X_i \tag{1}$$

where $\sum_{i=1}^{n} v_{ij} X_i = v_{1j} X_1 + \cdots + v_{nj} X_n$ is the linear combination of all species and represents the reactant complex of the jth reaction. Further, we define the *reaction vectors* of the jth reaction by $v'_{\cdot j} - v_{\cdot j}$ and the linear subspace spanned by all reaction vectors, called the *stoichiometric subspace*, by

$$\mathscr{S} \triangleq \text{span}\{v'_{\cdot j} - v_{\cdot j}, j = 1, \ldots, r\}. \tag{2}$$

The dynamics of $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ that captures the change of the concentration of each species, labeled by $x \in \mathbb{R}^n_{\geq 0}$, may be reached if a r-dimensional vector-valued function $\mathscr{K} \in \mathscr{C}(\mathbb{R}^n_{\geq 0}; \mathbb{R}^r_{\geq 0})$ is defined to evaluate the reaction rates and the balance law is further utilized, written as

$$\frac{dx(t)}{dt} = \Gamma \mathscr{K}(x), \quad x \in \mathbb{R}^n_{\geq 0} \tag{3}$$

where $\Gamma \in \mathbb{Z}^{n \times r}$ is the *stoichiometric matrix* with the jth column $\Gamma_{\cdot j} = v'_{\cdot j} - v_{\cdot j}$. Under the assumption that the reaction mixture remains spatially homogeneous with constant temperature, total volume, and a sufficiently large molecule count, *mass action kinetics* is usually used to evaluate the reaction rate, which requires the probability of random molecular collisions to be proportional to the product of reactant concentrations and induces the rate of the jth reaction as

$$\mathscr{K}_j(x) = k_j x^{v_{\cdot j}} \triangleq k_j \prod_{i=1}^{n} x_i^{v_{ij}} \tag{4}$$

where $k_j > 0$ represents the reaction rate constant. With k_j alongside the reaction arrow, (1) can be written as

$$\sum_{i=1}^{n} v_{ij} X_i \xrightarrow{k_j} \sum_{i=1}^{n} v'_{ij} X_i.$$

Then, the integral form of the solution to (3) plus (4) is

$$x(t) = x(0) + \sum_{j=1}^{r} k_j \left(\int_0^t \prod_{i=1}^{n} x_i^{v_{ij}}(\tau) d\tau \right) (v'_{\cdot j} - v_{\cdot j}) \tag{5}$$

where $x(0) \in \mathbb{R}^n_{\geq 0}$ represents any given initial point. Clearly, the CRN system endowed with mass action kinetics offers polynomial ordinary differential equations, often termed as MAS and denoted by the quad $(\mathcal{S}, \mathcal{C}, \mathcal{R}, k)$ with $k = (k_1, \ldots, k_r)^\top$. The dynamics of (5) means that any trajectory of a MAS will evolve within a special set, called *stoichiometric compatibility class*.

Definition 2 (Stoichiometric compatibility class): Given a CRN $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ and $x_0 \in \mathbb{R}^n_{\geq 0}$, the set induced by $\mathcal{P}(x_0) = \{x \in \mathbb{R}^n_{\geq 0} \mid x - x_0 \in \mathscr{S}\}$ is called the *stoichiometric compatibility class* of x_0, and $\mathcal{P}^+(x_0) = \mathcal{P}(x_0) \cap \mathbb{R}^n_{>0}$ is named the *positive stoichiometric compatibility class* of x_0.

We further give the definitions of equilibrium and exponential convergence.

Definition 3 (Equilibrium): For a MAS $(\mathcal{S},\mathcal{C},\mathcal{R},k)$ governed by (3) plus (4), if a constant vector $\bar{x}\in\mathbb{R}^n_{\geq 0}$ satisfies $\Gamma\mathscr{K}(\bar{x})=0$, then $\bar{x}$ is called the *nonnegative equilibrium point*. If for any $x_{\mathcal{A}}\in\mathcal{A}\subset\mathbb{R}^n_{\geq 0}$ there is $\Gamma\mathscr{K}(x_{\mathcal{A}})=0$, then the set $\mathcal{A}$ is called an equilibrium set.

Definition 4 (Exponential convergence): Consider a MAS $(\mathcal{S},\mathcal{C},\mathcal{R},k)$ governed by (3) plus (4) and admitting an equilibrium set $\mathcal{A}$. The solution $x(t)$ of this MAS *converges exponentially* to $\mathcal{A}$ if there are constants $M,\gamma\in\mathbb{R}_{>0}$ supporting $\inf_{y\in\mathcal{A}}\|x(t)-y\|\leq Me^{-\gamma t}$ for all $t\geq 0$. When the system admits an isolated equilibrium, i.e., $\mathcal{A}=\{\bar{x}\}$, *exponential convergence* is ensured if there exist $M,\gamma\in\mathbb{R}_{>0}$ satisfying $\|x(t)-\bar{x}\|\leq Me^{-\gamma t}$.

B. Fully Connected Neural Network

Assume an FCNN to perform a classification task on the data set $\mathbb{D}=\{(x^i,d^i)\}_{i=1}^p$ with input $x\in\mathbb{R}^n$, output $d\in\{0,1\}$, and data size p. The training process utilizes the common stochastic gradient descent algorithm with the mini-batch strategy (at each iteration only part of samples are fed to update weights), and the activation function takes the *Sigmoid function* given by

$$f(z)=\frac{1}{1+e^{-z}}. \tag{6}$$

Note that the activation function will map the input into (0,1). To make it be applicable, it usually needs to normalize the input and output data before training. Mathematically, the training process can be expressed according to feedforward and backward propagation. For simplicity but without the loss of generality, we fix the structure of FCNN to have *two (input layer)-two (hidden layer)-one (output layer)* nodes to exhibit this process, which implies that the data input is 2-D while the output is 1-D.

1) Feedforward Propagation: Denote the *sample matrix* by $\chi\in\mathbb{R}^{3\times p}$ with the ith $(i=1,\ldots,p)$ column to be $\chi_{\cdot i}=(x_1^i,x_2^i,d^i)^\top$ ((x_1^i,x_2^i) represents two-dimensional feature vector, and d^i is the 1-D label); the *input matrix* by $\Xi\in\mathbb{R}^{3\times\tilde{p}}_{\geq 0}$ with $\Xi_{\cdot l}=(x_1^l,x_2^l,1)^\top$ and $\Xi_{3\cdot}=\mathbb{1}^\top$ to handle dumb nodes corresponding to bias in the input layer, and $\tilde{p}$ $(\tilde{p}\leq p)$ to represent the mini-batch size; and the *weight matrices* connecting input-hidden layers and hidden-output layers by $\mathcal{W}_1\in\mathbb{R}^{2\times 3}$ and $\mathcal{W}_2\in\mathbb{R}^{1\times 3}$ with their last columns to represent the bias attributed to the nodes of hidden and output layers, respectively. Then, the feedforward computation process at one iteration follows

$$\begin{aligned}&\mathcal{N}=\mathcal{W}_1\cdot\Xi,\Upsilon=f(\mathcal{N}),\tilde{\Upsilon}=\begin{pmatrix}\Upsilon\\ \mathbb{1}^\top\end{pmatrix}\\ &\tilde{\mathcal{N}}=\mathcal{W}_2\cdot\tilde{\Upsilon},y=f(\tilde{\mathcal{N}})\end{aligned} \tag{7}$$

where $\mathcal{N}\in\mathbb{R}^{2\times\tilde{p}}$, $\Upsilon\in\mathbb{R}^{2\times\tilde{p}}$ is the *hidden matrix*, $\tilde{\Upsilon}\in\mathbb{R}^{3\times\tilde{p}}$ is constructed to add a row of 1 to Υ to handle dumb nodes in the hidden layer, $\tilde{\mathcal{N}}\in\mathbb{R}^{1\times\tilde{p}}$, and $y\in(0,1)^{1\times\tilde{p}}$ is the *output matrix*. For the regression task, the prediction output is assigned with the actual value of $y^l=f^l(\tilde{\mathcal{N}})$ for the training, while for the classification task, the prediction label takes values in $\{0,1\}^{1\times\tilde{p}}$ with 1 if $y^l>\frac{1}{2}$ and 0 otherwise, i.e., y^l is not interpreted as a probability here. Based on these settings, we can compute the mean squared error loss

$$\mathcal{E}(\mathcal{W}_1,\mathcal{W}_2)=\frac{1}{2}(\delta-y)(\delta-y)^\top \tag{8}$$

where $\delta=(d^1,\ldots,d^{\tilde{p}})$ for the fundamental and accurate realization using CRNs. This finishes the feedforward computation of one iteration with $\tilde{p}$ samples.

2) Backward Propagation: This process serves to update weights from the mth iteration to the $(m+1)$th iteration according to the stochastic gradient descent algorithm, which takes

$$\mathcal{W}^{m+1}=\mathcal{W}^m+\Delta\mathcal{W}^m \tag{9}$$

with $\mathcal{W}=\begin{pmatrix}\mathcal{W}_1\\ \mathcal{W}_2\end{pmatrix}$, $\eta\in(0,1]$ to be the learning rate and

$$\begin{aligned}\Delta\mathcal{W}^m_{1_{ij}}&=-\eta\frac{\partial\mathcal{E}^m}{\partial\mathcal{W}^m_{1_{ij}}}=-\eta\sum_{l=1}^{\tilde{p}}e_l^m f'(n_{3l}^m)\mathcal{W}^m_{2_{1i}}f'(n_{il}^m)\Xi_{jl}^m\\ \Delta\mathcal{W}^m_{2_{1j}}&=-\eta\frac{\partial\mathcal{E}^m}{\partial\mathcal{W}^m_{2_{1j}}}=-\eta\sum_{l=1}^{\tilde{p}}e_l^m f'(n_{3l}^m)\tilde{\Upsilon}_{jl}^m\\ &\quad i=1,2;\ j=1,2,3.\end{aligned} \tag{10}$$

Here, $e_l^m=(\delta^m-y^m)_{1l}$, $(n_{1l}^m,n_{2l}^m)^\top=\mathcal{N}_{\cdot l}^m$, $n_{3l}^m=\tilde{\mathcal{N}}_{1l}^m$, and the differentiation is derived by the chain rule. After finishing updating weights, the training for the $m+1$ iteration fires through feeding another group of $\tilde{p}$ samples from $\mathbb{D}$ into the input layer, and performing the feedforward computation repeatedly. The termination condition is based on the precision requirement, i.e., $|e_l^m|$ is less than the preset threshold for all l, otherwise, the training process continues. Note that the termination condition used is not as general as the one for which the validation loss flatlines and the loss is not the commonly used cross-entropy function for classification. While the programmed FCNN may be susceptible to overfitting and suboptimal performance on certain tasks, the primary objective of this study is to develop foundational methods for feasibly translating machine learning paradigms into biochemical systems for potential applications in synthetic biology. As such, the outcomes presented are intended to serve as a proof of concept rather than an attempt to achieve state-of-the-art performance in machine learning algorithmic predictions. A complete flow chart is presented in Fig. 1 to exhibit how the simplified FCNN works.

C. Example of Implementing Addition Calculation With CRNs

There are various calculations, such as addition and subtraction, and information processing, such as assignment, and logical judgment, such as loop, in the process of FCNN. Every operation needs to be implemented by CRNs for our current task, and moreover, each variable (a fixed constant after calculation or processing) involved in FCNN needs to be represented by some counterpart in CRN. Thus, for a designed MAS governed by (3)

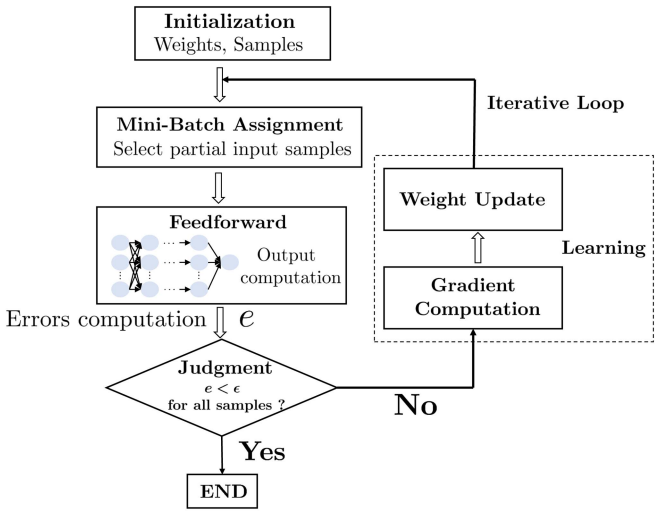

Fig. 1. Flow chart of FCNN.

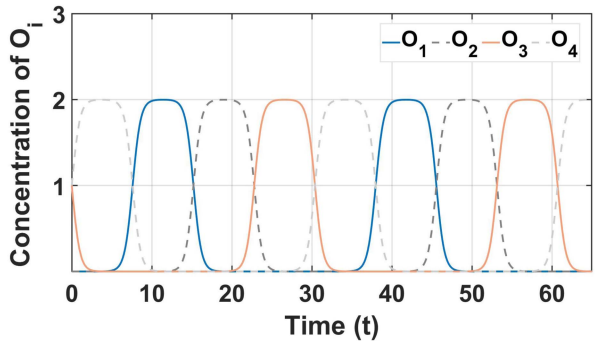

Fig. 2. Response curve of the chemical oscillator with four species and $k_o = 1$. Here, O_1 and O_3 can control two reaction systems.

plus (4) with given $x(0) \in \mathbb{R}^n_{\geq 0}$, this prompts the *limiting steady states* [23] of some species

$$\mathscr{L}[x(t)] = \left(\lim_{t\to\infty} x_i(t)\right)_{X_i \in \mathcal{S}} \tag{11}$$

to represent the corresponding variables in FCNN. Note that (11) includes GAS/LAS equilibrium of partial species if any. In the following, we give an example of programming addition operation using CRN.

Example 1 ([12]): For the addition operation $a + b$ in $\mathbb{R}^n_{\geq 0}$, we can implement it by the MAS of

$$A \xrightarrow{1} A + C, \quad B \xrightarrow{1} B + C, \quad C \xrightarrow{1} \emptyset.$$

From the reactions, A and B clearly act as *catalysts species* (no concentration changes at all times, i.e., $\forall t,\ x_a(t) = x_a(0),\ x_b(t) = x_b(0)$). The dynamics that captures the concentration changes of all species thus only follows that of species C, i.e., $\dot{x}_c(t) = x_a(t) + x_b(t) - x_c(t)$, which determines the limiting steady state of C to be $\overline{x}_c = \lim_{t\to\infty} x_c(t) = x_a(0) + x_b(0)$. Therefore, if the initial concentrations of A and B are set by $x_a(0) = a$ and $x_b(0) = b$, then $\bar{x}_c$ will store the addition result of $a + b$.

Note that all reactions seem not to support the conservation law of mass, which is actually not necessary in the CRN theory. The reaction $C \xrightarrow{1} \emptyset$ is often used to model the matter exchange with the environment. Besides, all variables $a,\ b,\ c$ in addition operation are restricted in $\mathbb{R}^n_{\geq 0}$. Thus, they can be expressed by the concentrations (nonnegative attribute) of species naturally. However, general addition operations may involve negative variables as well. To comply with the nonnegative attribute of species concentration in CRN, we adopt the *dual rail encoding* method to make the right match.

Definition 5 (Dual rail encoding [37]): For any variable $\zeta \in \mathbb{R}$, we take two species $\mathcal{Z}^+, \mathcal{Z}^-$ with nonegative concentrations $\zeta^+(t) \geq 0, \zeta^-(t) \geq 0, \forall t \geq 0$, between which the difference $\zeta^+(t) - \zeta^-(t)$ represents the real value ζ.

Clearly, the number of concentration variables, i.e., the number of species in CRN, need to be increased doubly after using the technique of dual rail encoding, which will increase the complexity of the MASs and further complicate the implementation process. In the subsequent expressions, we use variables in lower case to represent the concentrations of the corresponding species in upper case.

III. CRNs Programming FCNNs: Circuit Design

In this section, we use CRNs to program the *two-two-one* structured FCNN utilizing the modular and layered design principle, i.e., dividing FCNN into the modules of assignment, feedforward propagation, judgment and loop, learning, and clear-out. The accordingly generated biomolecular circuits are called BFCNNs in the context. Moreover, they can be extended to an FCNN with arbitrary depth and width.

Before we perform programming, we introduce the chemical oscillator [12], [38] first, which will play an important role in the design of all modules. A major challenge in devising CRNs to perform a series of cumbersome calculations or information processing is that the latter are executed sequentially while chemical reactions occur in parallel. The chemical oscillator is a feasible solution to address this issue, which, also called molecular clock signal reactions, is a set of chemical reactions admitting a certain oscillatory behavior. The usual oscillatory reactions [39] follow

$$\begin{aligned} &O_1 + O_2 \xrightarrow{k_o} 2O_2, \quad O_2 + O_3 \xrightarrow{k_o} 2O_3 \\ &\cdots\cdots \quad , \quad O_r + O_1 \xrightarrow{k_o} 2O_1 \end{aligned} \tag{12}$$

where $O_i,\ i = 1, \ldots, r$ denotes clock signal species. We display the concentration curves of a simple chemical oscillator in Fig. 2 with $k_o = 1$. Note that the oscillating behavior is not globally stable, but this does not impede its use for the current purpose. With certain initial values, each odd-encoded clock signal species $O_i (i = 2k + 1, k \geq 0)$ has an oscillation phase apart from others in which it has nonzero concentration while others are almost zero. We use these signals as catalysts acting on different reaction systems, which will turn ON a reaction system but turn OFF others according to the concentrations (nonzero or almost zero) of catalysts. This allows to control the occurrence

sequence of different reactions without artificial intervention. We will design suitable chemical oscillators to help program all modules mentioned above.

A. Assignment Module

From Fig. 1, the first step of FCNN is initialization, i.e., assigning values. That is, to implement the order like $x = a$ (assign value a to variable x) through chemical reactions. We design *assignment module* to complete this task.

The mini-batch strategy requests to divide the sample matrix χ into $\frac{p}{\tilde{p}}$ blocks, and feed each block according to

$$\text{"block}_1 \to \text{block}_2 \to \cdots \to \text{block}_{\frac{p}{\tilde{p}}} \to \text{block}_1 \to \cdots\text{"}$$

into the input layer one by one during each iteration. There includes two steps to program this process: one step is to take $\tilde{p}$ samples from original p samples in sequence, and the other step is to circulate to feed "$\text{block}_1 \to \ldots \to \text{block}''_{\frac{p}{\tilde{p}}}$ into the input layer again and again. For simplicity but without loss of generality, we assume the samples to be nonnegative, and thus directly introduce the sample species set $\{X_1^i, X_2^i, D^i\}_{i=1}^{p}$, the input species set $\{S_1^l, S_2^l, S_3^l\}_{l=1}^{\tilde{p}}$, the order species set $\{C_i^l\}$, and the auxiliary order species set $\{\tilde{C}_i^l\}$ to facilitate updating $\{C_i^l\}$ after a round of iteration. The aBCRN operating in three phases is designed as $\forall i \in \{1,\ldots,p\},\ l \in \{1,\ldots,\tilde{p}\},\ q \in \{1,2\}$ and any constant k

$$\begin{aligned}
\mathcal{O}_1:\quad & C_i^l + X_q^i \xrightarrow{1} C_i^l + X_q^i + S_q^l,\quad S_q^l \xrightarrow{1} \varnothing \\
& C_i^l + D^i \xrightarrow{1} C_i^l + D^i + S_3^l,\quad S_3^l \xrightarrow{1} \varnothing \qquad (13)\\
\mathcal{O}_3:\quad & C_i^l \xrightarrow{k} \tilde{C}_i^l \qquad (14)\\
\mathcal{O}_5:\quad & \tilde{C}_{o_l}^l \xrightarrow{k} C_{\tilde{p}+o_l}^l,\ \tilde{C}_{p-\tilde{p}+l}^l \xrightarrow{k} C_l^l,\ \tilde{C}_{j_l}^l \xrightarrow{k} C_{j_l}^l \qquad (15)
\end{aligned}$$

where $o_l \in I_l,\ j_l \notin I_l \bigcup \{p-\tilde{p}+l\}$ with $I_l \triangleq \{l, \tilde{p}+l, \ldots, p-2\tilde{p}+l\}$ for any fixed l. In particular, $j_l \in \varnothing$ if $\tilde{p} = 1$ implying that $\tilde{C}_{j_l}^l \xrightarrow{k} C_{j_l}^l$ in $\mathcal{O}_5$ are absent, and $I_l \triangleq \{l\}$ if $3\tilde{p} > p$, i.e., $\tilde{p} + l > p - 2\tilde{p} + l$.

Remark 1: We use $\mathcal{O}_i$ to identify different phases in aBCRN, which means the oscillatory (clock) signal O_i produced by (12) is added to all reactions in this phase as a catalyst when aBCRN works. Their addition will not change the dynamics of aBCRN, but can control the occurrence order of different phases. The explanation in this remark will apply to all BCRNs designed in the subsequent.

In the following, we use $c(t) = (c^1(t), \ldots, c^{\tilde{p}}(t))^\top \in \mathbb{R}_{\geq 0}^{1\times p\tilde{p}}$, where $c^l(t) = (c_1^l(t), \ldots, c_p^l(t))$ with $c_i^l(t)$ to identify the concentration of C_i^l $(i = 1,\ldots,p)$ to exhibit how aBCRN works under a given initial condition on each of the involved species concentrations.

Proposition 1: The aBCRN of (13)–(15) driven by the corresponding oscillatory signals can perform automatic assignment through equilibrium if the initial point is set by $c(0) = (\xi_1, \ldots, \xi_{\tilde{p}})^\top$, where $\xi_l \in \mathbb{R}^p$, $(x_1^i(0), x_2^i(0), d^i(0))^\top = \chi_{\cdot i}$, $\tilde{c}(0) = \mathbb{0}$ and $s(0) \in \mathbb{R}_{\geq 0}^{3\tilde{p}}$ to be any nonnegative value.

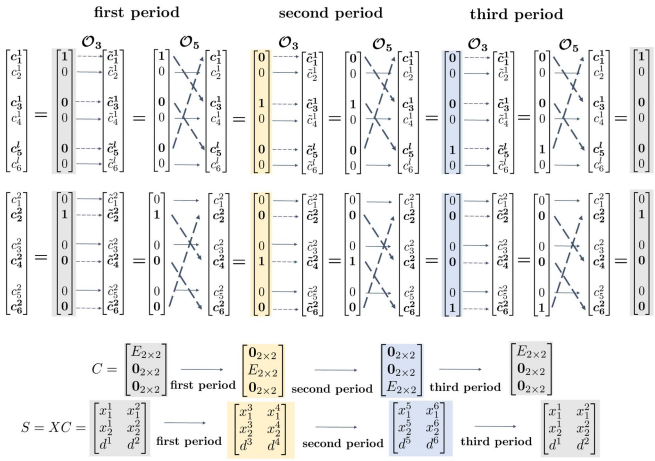

Fig. 3. Concentration change of species C_i^l (denoted as c_i^l), species $\tilde{C}_i^l$ (denoted as $\tilde{c}_i^l$), and input species during several periods governed by the aBCRN with $p = 6$ and $\tilde{p} = 2$. $E_{2\times 2}$ is a 2×2 unit matrix. The concentration matrices are defined as C of order species with column l and row i, X of sample species with column i and row q, and S of input species with column l and row q (the presentation of indicators follows Proposition 1 except that $q \in \{1,2,3\}$ here).

The detailed proofs of propositions, lemmas and theorems are available in the Supplementary Material due to space limits.

Proposition 1 serves to assign the sample values to input according to the mini-batch strategy, which is automatically finished through (13)–(15) driven by the corresponding oscillatory signal (O_1, O_3, and O_5, respectively) and with the given initial condition. During the stage of high O_1 concentration, the concentrations of O_3 and O_5 are nearly zero. This implies that the reactions in (13) are dominant while the reactions in (14) and (15) are almost inactive. Similarly, reactions controlled by O_3 or O_5 will occur at their nonzero stages. This can ensure that the expected operations are completed within one cycle of O_i $(i = 1, 3, 5)$. The equilibria of species S_q^l $(q = 1, 2, 3)$ store the input under a fixed batch size during each iteration (note that the equilibrium of each species S_q^l is unique and GAS relative to its stoichiometry compatibility class). That is, when the whole aBCRN reaches equilibrium in every phase, the equilibria of species S_q^l are the input in that iteration.

To better understand how the aBCRN works, we give an example of the species concentration change during the first three periods with $p = 6, \tilde{p} = 2$. The result is shown in Fig. 3. Based on the first three lines, the concentration matrix C is designed to store the values of the transformation matrix, with its columns governed by the BCRNs (14) and (15) that implement copy (thin dotted and solid arrows) and heterotopic copy (thick dotted arrows), respectively. Then, with the periodic concentration change of order species C_i^l, the linear transformation (matrix multiplication in the last line) is realized by BCRN (13).

B. Feedforward Propagation Module

In this section, we design the *feedforward propagation module*, including wsBCRN and *nonlinear* sigBCRN.

1) Linear Weighted Sum BCRN: This module contributes to programming the computations of the net input to the hidden layer, $\mathcal{N}$, and that to the output layer, $\tilde{\mathcal{N}}$, presented in (7). We use the *dual-rail encoding* technique [37] to handle the negative variables, and omit the iteration identification m of all variables if any in Section II for convenience.

Based on the dual rail encoding technique, we can rewrite $\mathcal{N}$ given in (7) to be

$$\mathcal{N} = (\mathcal{W}_1^+ - \mathcal{W}_1^-) \cdot \Xi = \mathcal{N}^+ - \mathcal{N}^- \tag{16}$$

where $\mathcal{W}_1^+$, $\mathcal{W}_1^-$, $\mathcal{N}^+$, and $\mathcal{N}^-$ all contain nonnegative elements. Thus, we introduce the net input species pair $\{N_{il}^+, N_{il}^-\}$, the weight species pair $\{W_q^+, W_q^-\}$, and the bias species pair $\{B_i^+, B_i^-\}$ with $i = 1, 2$, $l = 1, \ldots, \tilde{p}$ and $q = 1, \ldots, 4$, and construct the MAS

$$\begin{aligned} \mathcal{O}_7^1: \quad & W_i^\pm + S_1^l \xrightarrow{1} W_i^\pm + S_1^l + N_{il}^\pm \\ & W_{i+2}^\pm + S_2^l \xrightarrow{1} W_{i+2}^\pm + S_2^l + N_{il}^\pm \\ & B_i^\pm \xrightarrow{1} B_i^\pm + N_{il}^\pm \\ & N_{il}^\pm \xrightarrow{1} \varnothing. \end{aligned} \tag{17}$$

Note that the phase identification $\mathcal{O}_7^1$ with the superscript "1" represents the first part of the reaction system controlled by the oscillatory signal O_7, and so on. The reaction rate constant is set as 1 here, but it can be adjusted as needed in practice. The dynamics of (17) thus follows:

$$\dot{n}^+(t) = w_1^+(t)s_a(t) - n^+(t) \tag{18}$$

$$\dot{n}^-(t) = w_1^-(t)s_a(t) - n^-(t) \tag{19}$$

where $s_a(t) = (s_{a_1\cdot}(t), s_{a_2\cdot}(t), \mathbb{1}^\top)^\top \in \mathbb{R}_{\geq 0}^{3\times\tilde{p}}$, $n^\pm(t) \in \mathbb{R}_{\geq 0}^{2\times\tilde{p}}$ with $n_{il}^\pm(t)$ to denote the concentration of $N_{il}^\pm$ and $w_1^\pm(t) \in \mathbb{R}_{\geq 0}^{2\times 3}$ is the concentration matrix of weight species. Notably, elements in $w_{1_1\cdot}^\pm(t)$ from left to right denote the concentration of species $W_1^\pm, W_3^\pm, B_1^\pm$, and those in $w_{1_2\cdot}^\pm(t)$ indicate the concentration of species $W_2^\pm, W_4^\pm, B_2^\pm$, respectively. Catalytic reactions (the first three) contribute to increasing the concentration of $N_{il}^\pm$ and the decay reaction (the final one) plays the role of balancing. Obviously, (18) and (19) are both GAS with the equilibrium points $\bar{n}^+ = w_1^+(0)s_a(0), \bar{n}^- = w_1^-(0)s_a(0)$. Since the oscillator separates reaction modules, $s_{a_1\cdot}(0), s_{a_2\cdot}(0)$ refer to the equilibrium concentrations of species $\{S_1^l, S_2^l\}_{l=1}^{\tilde{p}}$ in phase $\mathcal{O}_1$, which stores the input sample values in the current iteration. Also, let $w_{1_{ij}}^+(0) = \mathcal{W}_{1_{ij}}, w_{1_{ij}}^-(0) = 0$ if $\mathcal{W}_{1_{ij}} > 0$, and $w_{1_{ij}}^+(0) = 0, w_{1_{ij}}^-(0) = -\mathcal{W}_{1_{ij}}$ if $\mathcal{W}_{1_{ij}} < 0$. Under these settings, the lwsBCRN system (17) realizes the linear weighted sum computation for the hidden layer through its equilibrium points.

2) Sigmoid Activation Biochemical Reaction Network: Having computed the net inputs in each hidden layer neuron, we need to pass the values through the nonlinear activation function to achieve a nonlinearity in the output. Compared to current methods based on approximation, we draw inspiration from [25] and [40] to calculate the accurate Sigmoid output value by devising the sigBCRN system.

Consider the reversible autocatalytic reaction $P \underset{1}{\overset{1}{\rightleftharpoons}} 2P$ with dynamics $\dot{p}(t) = p(t)(1 - p(t))$, which through the zero deficiency theorem [31] exhibits to have a unique GAS positive equilibrium 1 and a unstable equilibrium 0. Clearly, the solution trajectory $p(t) = \frac{1}{1+\frac{1-p(0)}{p(0)}e^{-t}}$ always falls within $\mathbb{R}_{>0}$ if $p(0) > 0$, and will be $p(t) < 1$ before reaching equilibrium when $p(0) < 1$. In particular, $p(0) = \frac{1}{2}$ will induce the well-known Sigmoid function $p(t) = \frac{1}{1+e^{-t}}$. To make the Sigmoid function be able to activate the concentration of net input species, we design N_{il}^+ as the catalyst to take part in the autocatalytic reaction $P \underset{1}{\overset{1}{\rightleftharpoons}} 2P$, and moreover, it can hydrolyze autonomously. Therefore, the resulting reactions will stop when the concentration of N_{il}^+ becomes zero, and the final states are highly dependent on $n_{il}^+(0)$. This measure ensures that the relationship between the limiting steady states and the prior reaction's output is mechanistically established, thereby changing the independent variable position from t to $n_{il}^+(0)$ in the solution $p(t) = \frac{1}{1+\frac{1-p(0)}{p(0)}e^{-t}}$. In addition, biochemical reaction systems can only handle the function computation of positive independent variables. Thus, we consider calculating $f(z) = 1/(1+e^{-z})$ when $n_{il} = n_{il}^+ - n_{il}^- > 0$, and $\tilde{f}(z) = 1/(1+e^z)$ when $n_{il}^+ - n_{il}^- < 0$. It implies that $n_{il}^+ - n_{il}^-$ should be known by the reaction system in advance.

To construct sigBCRN, we introduce hidden-layer output species pairs $\{P_{il}^+, P_{il}^-\}$ with $i = 1, 2$, $l = 1, \ldots, \tilde{p}$ and species Half maintaining the concentration of $\frac{1}{2}$ (this can be implemented easily by setting Half as a catalyst with the initial concentration $\frac{1}{2}$). The following reactions occupying two phases are for "clarifying" the sign of n_{il} in CRN and presetting initial concentrations of $P_{il}^\pm$

$$\mathcal{O}_9: \quad N_{il}^+ + N_{il}^- \xrightarrow{k} \varnothing \tag{20}$$

$$\begin{aligned} \mathcal{O}_{11}: \quad & N_{il}^\pm + \text{Half} \xrightarrow{1} N_{il}^\pm + \text{Half} + P_{il}^\pm \\ & N_{il}^\pm + P_{il}^\pm \xrightarrow{1} N_{il}^\pm. \end{aligned} \tag{21}$$

Here, the annihilation reaction (20) serves to compute the difference, and intuitively speaking, the fixed point $\bar{n}_{il}^+ = n_{il}$ if $n_{il} > 0$ and $\bar{n}_{il}^- = -n_{il}$ if $n_{il} < 0$ after $\mathcal{O}_9$. Owing to the reaction mechanism designed in $\mathcal{O}_{11}$, only the net input species N_{il}^+ or N_{il}^- with nonzero initial concentrations can trigger the reactions involving the related species Half, $N_{il}^\pm, P_{il}^\pm$ in phase $\mathcal{O}_{11}$. Then, the following reaction systems were built occurring in phase $\mathcal{O}_{13}$ to calculate the output P_{il}^+, P_{il}^-

$$\mathcal{O}_{13}: N_{il}^+ + P_{il}^+ \underset{1}{\overset{1}{\rightleftharpoons}} 2P_{il}^+ + N_{il}^+, \; N_{il}^+ \xrightarrow{1} \varnothing \tag{22}$$

$$\begin{aligned} & N_{il}^- + 2P_{il}^- \xrightarrow{1} 3P_{il}^- + N_{il}^- \\ & N_{il}^- + P_{il}^- \xrightarrow{1} N_{il}^- \xrightarrow{1} \varnothing \end{aligned} \tag{23}$$

for all samples l. The system designed above is composed of two uncoupled subnetworks without dynamical interference. The MAS (22) is designed to realize $\frac{1}{1+e^{-n_{il}^+(0)}}$, and (23) is a

slightly revised version of (22) to compute $\frac{1}{1+e^{n_{il}^-(0)}}$. Recall that, after $\mathcal{O}_{11}$, initial conditions in phase $\mathcal{O}_{13}$ will become $p_{il}^+(0)=\frac{1}{2}, p_{il}^-(0)=0$ if $n_{il}>0$ and $p_{il}^+(0)=0, p_{il}^-(0)=\frac{1}{2}$ if $n_{il}<0$. Under these settings, in phase $\mathcal{O}_{13}$, (22) will occur only when the net input n_{il} take positive values by which we could get $f(n_{il})$ in the equilibrium of P_{il}^+, and meanwhile P_{il}^- is going to maintain zero concentration in this phase. If $n_{il}<0$, (23) is initiated instead of (22) to provide $\tilde{f}(-n_{il})$ in the equilibrium of P_{il}^-. In the end, to use one species to denote the final outputs Υ_{il} in the hidden layer in either case, the new species $\{P_i^l\}$ was introduced and the following reactions are added in phase $\mathcal{O}_{13}$:

$$P_{il}^+ \xrightarrow{1} P_{il}^+ + P_i^l,\quad P_{il}^- \xrightarrow{1} P_{il}^- + P_i^l,\quad P_i^l \xrightarrow{1} \varnothing \tag{24}$$

where, at least, one of the initial concentrations of P_{il}^+ and P_{il}^- must be zero. Therefore, the sigBCRN system (20)–(24) implements the Sigmoid function computation for real independent variables. Our design is confirmed quantitatively by the dynamic behavior analysis in Section I of the Supplementary Material. Then, regarding the hidden-output layer, we further introduce new species pairs $\{N_{3l}^+, N_{3l}^-\}, \{W_q^+, W_q^-\}$, and $\{B_3^+, B_3^-\}$ with $q=5,6, l=1,\ldots,\tilde{p}$ and give the following MAS without illustration:

$$\begin{aligned}\mathcal{O}_{15}^1:\quad & W_5^\pm + P_1^l \xrightarrow{1} W_5^\pm + P_1^l + N_{3l}^\pm\\ & W_6^\pm + P_2^l \xrightarrow{1} W_6^\pm + P_2^l + N_{3l}^\pm\\ & B_3^\pm \xrightarrow{1} B_3^\pm + N_{3l}^\pm\\ & N_{3l}^\pm \xrightarrow{1} \varnothing.\end{aligned} \tag{25}$$

Consequently, applying the sigBCRN design on species $\{N_{3l}^\pm, \text{Half}, Y_l^\pm, Y^l\}_{l=1}^{\tilde{p}}$, the y_l in FCNN is exhibited by the equilibrium concentration $\bar{y}_l$ of the output species Y^l.

C. Judgment Module

One tough problem of using BCRNs to program the learning process based on the gradient descent is the compilation of derivative operators, $\frac{\partial \mathcal{E}}{\partial \mathcal{W}_{1_{ij}}}, \frac{\partial \mathcal{E}}{\partial \mathcal{W}_{2_{1j}}}$ with $i=1,2$, $j=1,2,3$. Our method is to compute the backpropagation (10) derived from the chain rule. Notably, due to $f'(n_{3l})=y_l(1-y_l), f'(n_{il})=\tilde{\Upsilon}_{il}(1-\tilde{\Upsilon}_{il})$, some of the factors in (10) like $\mathcal{W}_{1_{ij}}, \mathcal{W}_{2_{1j}}, \Xi_{il}, y_l, \tilde{\Upsilon}_{il}$ have been already computed and stored in the corresponding species, while there are no species concentrations to carry other signals that need to be subtracted in advance, i.e., $e_l, 1-\tilde{\Upsilon}_{il}, 1-y_l$. Therefore, this section first involves the reaction network design for *Preceding Calculation* BCRN with the attempt to store such signals and prepare for the subsequent judgment and learning part. Actually, this kind of precomputing skill is quite ubiquitous in the computer environment to enhance computation efficiency.

We introduce species set $\{Y_e^l, Y_s^l, \tilde{Y}^l, I_y^l\}$, $\{P_{is}^l, \tilde{P}_i^l, I_{p_i}^l\}$, $\{S_y^l, S_{p_i}^l\}$, $\{E_l^+, E_l^-, E_l\}$ for $l\in\{1,\ldots,\tilde{p}\}$ and device the pBCRN as follows:

$$\mathcal{O}_{23}^1\quad \begin{cases} Y^l \xrightarrow{k} Y_e^l + Y_s^l + \tilde{Y}^l\\ Y_e^l + S_3^l \xrightarrow{k} \varnothing\\ Y_s^l + I_y^l \xrightarrow{k} \varnothing\\ P_i^l \xrightarrow{k} P_{is}^l + \tilde{P}_i^l\\ P_{is}^l + I_{p_i}^l \xrightarrow{k} \varnothing \end{cases} \tag{26}$$

$$\mathcal{O}_{23}^2\quad \begin{cases} S_3^l \xrightarrow{1} E_l^+ + S_3^l,\quad E_l^+ \xrightarrow{1} \varnothing\\ Y_e^l \xrightarrow{1} E_l^- + Y_e^l,\quad E_l^- \xrightarrow{1} \varnothing\\ I_y^l \xrightarrow{1} S_y^l + I_y^l,\quad S_y^l \xrightarrow{1} \varnothing\\ I_{p_i}^l \xrightarrow{1} S_{p_i}^l + I_{p_i}^l,\quad S_{p_i}^l \xrightarrow{1} \varnothing \end{cases} \tag{27}$$

$$\mathcal{O}_{23}^3\quad \begin{cases} E_l^+ \xrightarrow{1} E_l^+ + E_l\\ E_l^- \xrightarrow{1} E_l^- + E_l\\ E_l \xrightarrow{1} \varnothing. \end{cases} \tag{28}$$

Here, the concentration of species $I_y^l, I_{p_i}^l$ should be maintained at one. The first three reactions of (26) are designed to measure the difference between outputs y_l and labels d^l as well as between output and one. Specifically, the first reaction with boundary equilibrium points performs a replication operation to assign the concentration of Y_l to $Y_e^l, Y_s^l, \tilde{Y}^l$. Here, Y_e^l, Y_s^l are used in the subsequent two annihilation reactions for subtraction, and output concentrations could be stored in $\tilde{Y}^l$ for backpropagation. A similar explanation applies to the last two reactions of (26). The MAS (27) comprises four group catalytic–hydrolysis reactions intending to assign the catalyst concentration to that of degraded species, and so does as (28).

The pBCRN system works in the following way. Two possible initial concentration relationships between $y_l(0)$ and $s_{3l}(0)$, i.e., $y_l(0)>s_{3l}(0)$ and $y_l(0)<s_{3l}(0)$, relates to the sign of training errors $e_l=d^l-y_l$ in one iteration. Correspondingly, the positive training error is provided in the equilibrium concentrations of species S_3^l, E_l^+, the negative one is stored in Y_e^l, E_l^-, and species E_l can represent the training error result in either case. In addition, (26) and (27) works in the same way for computing $1-y_l, 1-\tilde{\Upsilon}_{il}$ that are only stored in equilibrium concentrations of species S_y^l, S_{pi}^l since the $y_l(0), p_i^l(0)$ is always smaller than one due to the Sigmoid output in sigBCRN.

Translating the judgment to biochemical reaction network language corresponds to determining whether the biochemical learning process carried out by follow-up reactions will happen. This part utilizes a bistable biochemical reaction system to perform the *Judgment Module*. Upholding the low-complexity design principles and introducing species $\{A_l\}$, we use the minimal 2-dimensional bistable MAS, proposed by Wilhelm [41], with four reactions to assist in our design

$$\begin{aligned}\mathcal{O}_{25}^1:\quad & A_l \xrightarrow{k_1} 2E_l,\quad 2E_l \xrightarrow{k_2} E_l + A_l\\ & E_l + A_l \xrightarrow{k_3} A_l,\quad E_l \xrightarrow{k_4} \varnothing.\end{aligned} \tag{29}$$

Note that E_l is the error species also participating in pBCRN in phase $\mathcal{O}_{23}$, which serves as input species in this module. Hopf bifurcation can be excluded as the trace

of the Jacobian matrix is always negative, and the MAS (29) is dissipative [41], whose dynamical behavior depends on its rate constants, i.e., there is no positive equilibrium when $D = k_1k_2 - 4k_3k_4 < 0$, a saddle-node bifurcation will occur at $D = 0$, and the system exhibits bistability if $D > 0$. Given specific parameters satisfying $D > 0$, the system has three equilibrium points, $(\bar{e}_{l_1}, \bar{a}_{l_1}) = (0, 0)$, $(\bar{e}_{l_2}, \bar{a}_{l_2}) = (\frac{k_1k_2 - \sqrt{k_1k_2(k_1k_2-4k_3k_4)}}{2k_2k_3}, \frac{k_2}{k_1}(\bar{e}_{l_2})^2)$, $(\bar{e}_{l_3}, \bar{a}_{l_3}) = (\frac{k_1k_2 + \sqrt{k_1k_2(k_1k_2-4k_3k_4)}}{2k_2k_3}, \frac{k_2}{k_1}(\bar{e}_{l_3})^2)$, which satisfy $\bar{e}_{l_1} < \bar{e}_{l_2} < \bar{e}_{l_3}$. Lyapunov's first method can be used to prove that the first and the third are locally asymptotic stable, and the second is unstable. The following reactions are added so that species C_a inherit the concentration sum of $\bar{e}_l$ for all input samples and act as the catalysts of subsequent CRNs

$$\mathcal{O}_{25}^2: \quad E_l \xrightarrow{k} E_l + C_a,\ C_a \xrightarrow{k} \varnothing \quad \forall l \in \{1, \ldots, \tilde{p}\}. \tag{30}$$

We call (29) and (30) the jBCRN system. It works in the premise that the $\bar{e}_{l_2}$ is set to the preset training threshold by adjusting rate constants and regarding $e_l(0)$ as the absolute value of the training error. In this way, if the initial conditions of species are perturbed upward at $(\bar{e}_{l_2}, \bar{a}_{l_2})$, the system travels to the third value; if they are perturbed downwards, it settles to (0,0). Consequently, only if $|e_l|$ of all input samples in the current round meet the requirement, the concentration of C_a will reach the zero equilibrium, and the following update reactions will not happen. Thus, the jBCRN system implements the judgment module under these parameter selections.

We construct the biochemical *Loop module* through chemical oscillators for our current task, aiming to replicate iterative loops in Fig. 1 while running an algorithm if the judgment condition is not achieved. Due to the periodical variations in concentrations of catalyst species, the chemical oscillator serves a dual purpose: not only do they regulate all modules to ensure sequential execution, but they also compel modules to iterate multiple times, facilitating the training process. Clearly, it is more convenient and practical for fully automated implementation compared to the cell-like compartment used in our previous work [42].

D. Learning Module

Here, are two main steps for updating weights using CRNs, i.e., first programming the backpropagation formula for each weight and then getting the new weight species concentrations for the next iteration. This section provides the design of the *learning module* composed of ngBCRN and uBCRN, which plays the main role in equipping the biochemical neural networks with learning abilities.

The derivative computation is divided into multiplication and the sum of multiplication concerning samples according to (10). Common factors existing in the multiplication part for different weights contribute to our realization. In our case, updating nine weights $\mathcal{W}$ produces four types of multiplication formulas. Different types are composed of different numbers of factors, and deeper weight formulas have fewer terms. Specifically, according to the chain rule, $e_l y_l(1 - y_l)$ for $\mathcal{W}_{2_{13}}$ (3-item) is a global common factor that appears in every formula, and $\tilde{\Upsilon}_{1l}, \tilde{\Upsilon}_{2l}$ is

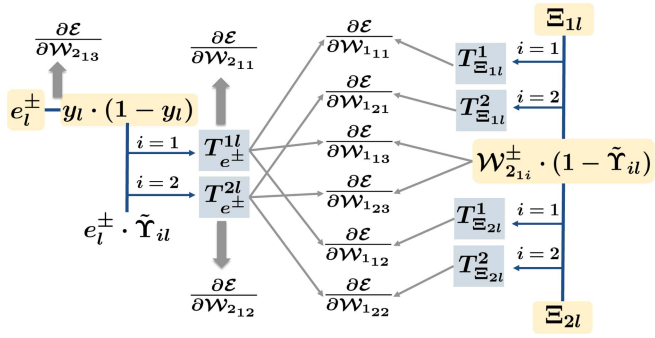

Fig. 4. Computation flow diagram and intermediate variables produced by the SFP.

multiplied for $\mathcal{W}_{2_{11}}, \mathcal{W}_{2_{12}}$ (4-item). When error is propagated to the input-hidden connection, $(1 - \tilde{\Upsilon}_{1l})\mathcal{W}_{2_{11}}, (1 - \tilde{\Upsilon}_{2l})\mathcal{W}_{2_{12}}$ will be multiplied again for $\mathcal{W}_{1_{13}}$ and $\mathcal{W}_{1_{23}}$ (6-item), respectively, and Ξ_{1l}, Ξ_{2l} are finally considered for $\mathcal{W}_{1_{11}}, \mathcal{W}_{1_{21}}$ and $\mathcal{W}_{1_{12}}, \mathcal{W}_{1_{22}}$ (7-item), respectively. Furthermore, the *dual-rail encoding* for real variables $e_l, \mathcal{W}_2$ will complicate the process. Here, we propose the design method, called SFP, for computing the multiplication consisting of any number of factors using a MAS consisting of reactions with no more than two reactants. The main computation strategy in SFP refers to combining two factors at a time and giving a combination that produces the fewest intermediate variables based on the aforementioned common-factor rule. To be specific, we first carry out product operation for common two-factors $e_l^{\pm}\tilde{\Upsilon}_{il}, y_l(1 - y_l), \mathcal{W}_{2_{1i}}^{\pm}(1 - \tilde{\Upsilon}_{il})$ (shaded yellow in Fig. 4), then combine single-factor $\Xi_{il}, e_l^{\pm}$ to perform second multiplication to obtain three-factor and four-factor shared monomials (shaded blue). After this step, we have the multiplication formula for $\mathcal{W}_2$ in the hidden-output connection. Finally, the formula for $\mathcal{W}_1$ is computed by the third multiplication.

Thus, given index sets $I_s = \{1, 2, 3, 4, 7, 8\}$, $I_v = \{5, 6, 9\}$, the monomial species $\{Q_{s\pm\pm}^l\}_{s\in I_s}$, $\{Q_{v\pm}^l\}_{v\in I_v}$, the intermediate species $\{M_l^{a,\pm}, M_y^l, M_{b,\pm}^l\}$, $\{T_l^{a,\pm}, T_{b,\pm}^l\}$ with $a, b = \{1, 2\}$ for $\forall l \in \{1, 2, \ldots, \tilde{p}\}$ and gradient species pairs $\{\mathrm{Par}_i^+, \mathrm{Par}_i^-\}_{i=1}^9$ are introduced here, and then we exhibit the ngBCRN based on SFP by computing $-\frac{\partial \mathcal{E}}{\partial \mathcal{W}_{1_{11}}}$. Defining $\mathrm{par}_1^{\pm} \in \mathbb{R}^{2\times 3}$, $\mathrm{par}_2^{\pm} \in \mathbb{R}^{1\times 3}$ and using the *dual-rail encoding*, we write

$$-\frac{\partial \mathcal{E}}{\partial \mathcal{W}_{1_{11}}} = \mathrm{par}_{1_{11}}^+ - \mathrm{par}_{1_{11}}^-$$

where

$$\mathrm{par}_{1_{11}}^+ = \sum_{l=1}^{\tilde{p}} (e_l^+ \mathcal{W}_{2_{11}}^+ + e_l^- \mathcal{W}_{2_{11}}^-) y_l(1 - y_l)\tilde{\Upsilon}_{1l}(1 - \tilde{\Upsilon}_{1l})\Xi_{1l}$$

$$\mathrm{par}_{1_{11}}^- = \sum_{l=1}^{\tilde{p}} (e_l^+ \mathcal{W}_{2_{11}}^- + e_l^- \mathcal{W}_{2_{11}}^+) y_l(1 - y_l)\tilde{\Upsilon}_{1l}(1 - \tilde{\Upsilon}_{1l})\Xi_{1l}.$$

It involves computing $4\tilde{p}$ seven-factor monomials and requires species $\{Q_{1\pm\pm}^l\}$ and intermediate species of $a = 1, b = 1$. The multiplication computation when combining e_l^+ and $\mathcal{W}_{2_{11}}^+$ is

implemented by designing the following MAS:

$$\begin{aligned}
\mathcal{O}_{27}^1: \quad & E_l^+ + P_1^l \xrightarrow{1} E_l^+ + P_1^l + M_l^{1,+}, \;\; M_l^{1,+} \xrightarrow{1} \varnothing \\
& \tilde{Y}^l + S_y^l \xrightarrow{1} \tilde{Y}^l + S_y^l + M_y^l, \;\; M_y^l \xrightarrow{1} \varnothing \\
& S_{p_1}^l + W_5^+ \xrightarrow{1} S_{p_1}^l + W_5^+ + M_{1,+}^l, \;\; M_{1,+}^l \xrightarrow{1} \varnothing \\
& M_l^{1,+} + M_y^l \xrightarrow{1} M_l^{1,+} + M_y^l + T_l^{1,+}, \;\; T_l^{1,+} \xrightarrow{1} \varnothing \\
& M_{1,+}^l + S_1^l \xrightarrow{1} M_{1,+}^l + S_1^l + T_{1,+}^l, \;\; T_{1,+}^l \xrightarrow{1} \varnothing \\
& T_l^{1,+} + T_{1,+}^l \xrightarrow{1} T_l^{1,+} + T_{1,+}^l + Q_{1++}^l, \;\; Q_{1++}^l \xrightarrow{1} \varnothing
\end{aligned}$$

and the result is stored in Q_{1++}^l. A similar CRN design structure is applied in generating species $Q_{1+-}^l, Q_{1-+}^l, Q_{1--}^l$ with replacing the error species, weight species, and the intermediate species positions by $\{E^+, M_l^{1,+}, T_l^{1,+}, W_5^-, M_{1,-}^l, T_{1,-}^l\}$, $\{E^-, M_l^{1,-}, T_l^{1,-}, W_5^+, M_{1,+}^l, T_{1,+}^l\}$ and $\{E^-, M_l^{1,-}, T_l^{1,-}, W_5^-, M_{1,-}^l, T_{1,-}^l\}$, respectively. The CRNs in $\mathcal{O}_{27}^1$ for calculating negative gradient values concerning $\mathcal{W}_{1_{12}}$, $\mathcal{W}_{1_{21}}$, $\mathcal{W}_{1_{22}}$ have the same design as $\mathcal{W}_{1_{11}}$ but need the index $\{a, b\}$ of intermediate species set being $\{1, 2\}$, $\{2, 1\}$ and $\{2, 2\}$, to finally get $\{Q_{3\pm\pm}^l\}$, $\{Q_{2\pm\pm}^l\}$, $\{Q_{4\pm\pm}^l\}$, respectively. Regarding $\mathcal{W}_{1_{13}}, \mathcal{W}_{1_{23}}$, we devise the new combination between $\{T_l^{1,\pm}\}$ and $\{M_{1,\pm}^l\}$, $\{T_l^{2,\pm}\}$ and $\{M_{2,\pm}^l\}$ to consequently produce $\{Q_{7\pm\pm}^l\}$ and $\{Q_{8\pm\pm}^l\}$, respectively. As for $\mathcal{W}_2$, it requires us to add only one new combination of species $E_l^\pm$ and $\{M_y^l\}$ to have new species $\{Q_{9\pm}^l\}$ that relates to $\mathcal{W}_{2_{13}}$ because the already existing $\{T_l^{1,\pm}\}$, $\{T_l^{2,\pm}\}$ can be regarded as $\{Q_{5\pm}^l\}$, $\{Q_{6\pm}^l\}$ for $\mathcal{W}_{2_{11}}$ and $\mathcal{W}_{2_{12}}$, separately. In the sum part, we devise reactions $\mathcal{O}_{27}^2$ below to calculate $\text{par}_1^\pm, \text{par}_2^\pm$

$$\begin{array}{l|l}
Q_{s++}^l \xrightarrow{1} Q_{s++}^l + \text{Par}_s^+, & Q_{s+-}^l \xrightarrow{1} Q_{s+-}^l + \text{Par}_s^- \\
Q_{s--}^l \xrightarrow{1} Q_{s--}^l + \text{Par}_s^+, & Q_{s-+}^l \xrightarrow{1} Q_{s-+}^l + \text{Par}_s^- \\
Q_{v+}^l \xrightarrow{1} Q_{v+}^l + \text{Par}_v^+, & Q_{v-}^l \xrightarrow{1} Q_{v-}^l + \text{Par}_v^- \\
\text{Par}_s^+ \xrightarrow{1} \varnothing, \text{Par}_v^+ \xrightarrow{1} \varnothing, & \text{Par}_s^- \xrightarrow{1} \varnothing, \text{Par}_v^- \xrightarrow{1} \varnothing
\end{array}$$

with $s \in I_s, v \in I_v$. Following our design, all shared intermediate variables are calculated once, stored, and ready for arbitrary use, so the SFP-based design could be extended to the FCNN of any depth at a lower computational cost.

Having known the negative gradient of $\mathcal{E}$ with respect to every weight, we consequently intend to get the change quantity $\Delta\mathcal{W}^m$ and the new weight $\mathcal{W}^{m+1}$ for the next iteration. The iterative formula (9) is rewritten as

$$\begin{aligned}
\mathcal{W}^{m+1} &= (\mathcal{W}_+^m + \eta \cdot \text{par}^+) - (\mathcal{W}_-^m + \eta \cdot \text{par}^-) \\
&= \mathcal{W}_+^{m+1} - \mathcal{W}_-^{m+1}
\end{aligned} \tag{31}$$

where $\text{par}^\pm = ((\text{par}_1^\pm)^\top, (\text{par}_2^\pm)^\top)^\top$. To program (31), we add incremental weight species pairs $\{\Delta W_i^+, \Delta W_i^-\}_{i=1}^6$, incremental bias species pairs $\{\Delta B_j^+, \Delta B_j^-\}_{j=1}^3$, old weight-bias species $\{\Gamma_i^+, \Gamma_i^-\}_{i=1}^9$, and learning rate species $\{L\}$. Then, we design the uBCRN below by adopting the same pattern in the previous work [42]

$$\begin{aligned}
\mathcal{O}_{29}: \quad & \text{Par}_i^\pm + L \xrightarrow{1} \text{Par}_i^\pm + L + \Delta W_i^\pm, \;\; \Delta W_i^\pm \xrightarrow{1} W_i^\pm, \\
& \Gamma_i^\pm \xrightarrow{1} \Gamma_i^\pm + W_i^\pm, \;\; W_i^\pm \xrightarrow{1} \varnothing \\
& \text{Par}_{j+6}^\pm + L \xrightarrow{1} \text{Par}_{j+6}^\pm + L + \Delta B_j^\pm, \;\; \Delta B_j^\pm \xrightarrow{1} B_j^\pm, \\
& \Gamma_{j+6}^\pm \xrightarrow{1} \Gamma_{j+6}^\pm + B_j^\pm, \;\; B_j^\pm \xrightarrow{1} \varnothing.
\end{aligned}$$

with $i \in \{1, \ldots, 6\}$, and $j = \{1, 2, 3\}$.

Remark 2: The natural isolation device, the oscillator, helps us to make $\{\Gamma_i^\pm\}_{i=1}^9$ carry the weights and bias of the present iteration in a much simpler way than before [42] by the following reactions added in phase $\mathcal{O}_7$

$$\mathcal{O}_7^2: \quad \begin{array}{c|c} W_i^\pm \xrightarrow{1} \Gamma_i^\pm + W_i^\pm, & B_j^\pm \xrightarrow{1} \Gamma_{j+6}^\pm + B_j^\pm \\ \Gamma_i^\pm \xrightarrow{1} \varnothing, & \Gamma_{j+6}^\pm \xrightarrow{1} \varnothing \end{array}$$

which transfers the initial concentrations of $\{W_i^\pm\}_{i=1}^6$ and $\{B_{j=1}^\pm\}_{j=1}^3$ of this period to $\{\Gamma_i^\pm\}_{i=1}^9$. Apart from that, we set $l(0) = \eta$ and give any positive initial concentration of incremental species, the uBCRN realizes the weight update process for mth training round.

E. Clear-Out Module

Until now, we have presented all the core module designs of the biochemical neural network. However, they do not completely meet the needs of an automated implementation of BFCNN since some given initial concentration conditions are required in each iteration. We categorize them into two categories: 1) the initial concentration should be completely emptied; 2) the initial concentration should be fixed to a certain positive value. Therefore, in our design, some species concentrations in the sigBCRN and pBCRN need to be adjusted in the final module.

1) Nonlinear CRN: The initial concentration of output species pair $\{P_{il}^+, P_{il}^-\}, \{Y_l^+, Y_l^-\}$ should be zero before every iteration to make sure that only one of each pair should be present with concentration $\frac{1}{2}$ before $\mathcal{O}_{13}$. However, in practice, the final equilibrium concentration of these species after each iteration, depending on the computational results in FCNN, may be unpredictable for us, i.e., some of them might not be zero as intended. Therefore, we give the *clear-out* BCRN in the final phase to consume all their concentrations

$$\mathcal{O}_{31}^1: \; P_{il}^+ \xrightarrow{1} \varnothing, \; P_{il}^- \xrightarrow{1} \varnothing, \; Y_l^+ \xrightarrow{1} \varnothing, \; Y_l^- \xrightarrow{1} \varnothing.$$

2) Precalculationg CRN: All the species comprising the product complex $Y_e^l + Y_s^l + \tilde{Y}^l, P_{is}^l + \tilde{P}_{is}^l$ should be none in the reactor for target equilibrium concentration, which depends on initial concentrations closely in our construction. Species $I_y^l, I_{p_i}^l$ should be constant one before every iteration for subtraction. But the practical situation is that $I_y^l, I_{p_i}^l$ must be consumed every iteration and the others may have residual concentrations. Thus, we give both the *clear-out* BCRN and the *compensatory* BCRN (far right)

$$\mathcal{O}_{31}^2: \begin{array}{l|l|l} Y_s^l \xrightarrow{1} \varnothing, & P_{is}^l \xrightarrow{1} \varnothing, & I \xrightarrow{1} I + I_y^l \\ \tilde{Y}^l \xrightarrow{1} \varnothing, & \tilde{P}_i^l \xrightarrow{1} \varnothing, & I \xrightarrow{1} I + I_{p_i}^l \\ Y_e^l \xrightarrow{1} \varnothing, & & I_y^l \xrightarrow{1} \varnothing, \; I_{p_i}^l \xrightarrow{1} \end{array}$$

where I is a unit species with the initial concentration one.

IV. Convergence Analysis of the Computation Reaction Modules

Utilizing such a realization strategy, the asymptotic stability and convergence speed of the designed biochemical reaction systems are significant factors in determining the performance of biochemical neural networks. In this section, a rigorous stability analysis of the computational modules built above will be carried out to demonstrate that all computation modules have exponential convergence properties, which ensures that the BFCNN system can approach the final results at a very fast rate and has a convergent training process.

A. Feedforward Propagation Module

The asymptotic stability of the assignment module is shown in Proposition 1 and the lwsBCRN system has been shown to be a GAS system for given parameters. Here, we still study reactions of the sigBCRN system occurring in different phases separately. The following lemma is first given to show the exponential convergence of the reaction system in $\mathcal{O}_9$.

Lemma 1: Suppose that $n_{il}(t) = (n_{il}^+(t), n_{il}^-(t))^\top, c = n_{il}^-(0) - n_{il}^+(0), a = \frac{n_{il}^+(0)}{n_{il}^-(0)}$. There exists $M_1 > \frac{c}{1-a}, \alpha_1 \in (0, -c)$ such that the MAS (20) converges exponentially to the unique boundary equilibrium $(\bar{n}_{il})^1 = (n_{il}^+(0) - n_{il}^-(0), 0)$ in any compatibility class $\{(x, y) \in \mathbb{R}^2_{\geq 0} | y = x + b\}$ for $\forall b \in \mathbb{R}_{\geq 0}$, and also $\exists\, M_2 > \frac{ac}{1-a}, \alpha_2 \in (0, c)$ such that it converges exponentially to the unique equilibrium $(\bar{n}_{il})^2 = (0, n_{il}^-(0) - n_{il}^+(0))$ in the compatibility class $\{(x, y) \in \mathbb{R}^2_{\geq 0} | y = x + b\}$ for $\forall b \in \mathbb{R}_{<0}$.

Lemma 1 shows that (20) realize computing the net input at a fast speed by regarding N_{il}^+ as output signal carriers when $n_{il} > 0$, and using species N_{il}^- as the output if $n_{il} < 0$. Using $p_{il}^\pm(t)$ to denote the concentration of species $P_{il}^\pm$, the dynamic equations of the MAS (21)

$$\mathcal{O}_{11}: \quad \dot{p}_{il}^\pm(t) = n_{il}^\pm(t)\left(\frac{1}{2} - p_{il}^\pm(t)\right) \tag{32}$$

with $n_{il}^\pm(t) = n_{il}^\pm(0) > 0$ has the unique globally exponentially stable (GES) equilibrium point $\frac{1}{2}$ whose convergence rate depends on $\bar{n}_{il}^\pm$ in phase $\mathcal{O}_9$, that is, $|p_{il}^\pm(t) - \bar{p}_{il}^\pm| = |p_{il}^\pm(t) - \frac{1}{2}| = |p_{il}^\pm(0) - \frac{1}{2}|e^{-n_{il}^\pm(0)t}$. If one of $n_{il}^\pm(0) = 0$, we can have $p_{il}^\pm(t) = p_{il}^\pm(0)$ for $\forall t \geq 0$. Then, the proposition below implies that MAS (22), (23) of sigBCRN will globally converge exponentially to an equilibrium set, in which at least one species concentration maintains at zero, and provides the appropriate initial concentrations range for $P_{il}^\pm$ to guarantee that their equilibrium concentration always falls within the range of Sigmoid function.

Proposition 2: For all $l = \{1, 2, \ldots, \tilde{p}\}$, $i = \{1, 2\}$, given nonnegative initial condition $(p_{il}^+(0), n_{il}^+(0)) \in \mathbb{R}^2_{\geq 0}$ and $(p_{il}^-(0), n_{il}^-(0)) \in \mathbb{R}^2_{\geq 0}$, the reaction system (22) and (23) will globally asymptotically converge to a unique boundary equilibrium set $\mathcal{A}_p = \{(p_{il}^+, n_{il}^+) | p_{il}^+ \geq 0, n_{il}^+ = 0\}$ and $\mathcal{A}_n = \{(p_{il}^-, n_{il}^-) | p_{il}^- \geq 0, n_{il}^- = 0\}$, respectively. Moreover, given initial concentrations $p_{il}^+(0), p_{il}^-(0) \in (0, 1)$, there exist $M_p^{il} = n_{il}^+(0), M_n^{il} = n_{il}^-(0), \gamma_p = \gamma_n = 1$ such that (22) and (23) can converge exponentially to $A_p^e = \{(p_{il}^+, n_{il}^+) | p_{il}^+ \in (0, 1), n_{il}^+ = 0\}$ and $A_n^e = \{(p_{il}^-, n_{il}^-) | p_{il}^- \in (0, 1), n_{il}^- = 0\}$, respectively.

Proposition 2 provides theoretical support that as long as the initial concentration of the corresponding output species is fixed at $\frac{1}{2}$, the MAS (22) and (23) have the limiting steady states of output species to represent the accurate Sigmoid activation output for arbitrary net input. Although they are both degenerate systems only admitting nonisolated boundary equilibrium points in the compatibility class, the attractive regions of every equilibrium can be determined from $\{(p_{il0}^+, n_{il0}^+) | p_{il0}^+ = \frac{1}{1+\frac{1-b}{b}e^{n_{il0}^+}}, b = \frac{1}{1+e^{-a}}, \forall a \in \mathbb{R}\}$ and $\{(p_{il0}^-, n_{il0}^-) | p_{il0}^- = \frac{1}{1+\frac{1-b}{b}e^{-n_{il0}^-}}, b = \frac{1}{1+e^{a}}, \forall a \in \mathbb{R}\}$.

In addition, to save the running time of BFCNN by decreasing the number of phases, the MAS (24), serving to represent target outputs by one species concentration, is embedded as a subnetwork into the MAS in phase $\mathcal{O}_{13}$. This strategy is also adopted in the subsequent design. We can demonstrate through the following lemma 2 that the asymptotic stability and the exponential convergence rate, if any in the unembedded system, could be inherited in the parent reaction network.

Lemma 2: Consider the kinetic equation of MAS $\mathscr{M}$

$$\dot{s}(t) = g(s(t)), \quad s_0 = s(0) \tag{33}$$

where $s(t) \in \mathbb{R}^n_{\geq 0}$, and the set $I \subset \{1, 2, \ldots, n\}$. Assume that a certain species subset $\{S_i\}_{i \in I}$ of $\mathscr{M}$ has the limiting steady states and admits the exponential convergence, i.e., $\exists\, q_i, \beta_i > 0$ such that $|s_i(t) - \bar{s}_i| < q_i e^{-\beta_i t}$ for $\forall t \geq 0$. If the MAS $\mathscr{M}'$ (called type-I CRN)

$$\begin{aligned} S_i &\xrightarrow{1} S_i + H \quad \forall i \in I \\ H &\xrightarrow{1} \varnothing \end{aligned} \tag{34}$$

is embedded as a subsystem into $\mathscr{M}$, then species H in the parent network $\mathscr{M}_1^*$ has the limiting steady state $\sum \bar{s}_i$ and will inherit the exponential convergence from $\mathscr{M}$. Moreover, if the equilibrium points of S_i are asymptotically stable with respect to any finite initial concentrations (not necessarily exponentially) in $\mathscr{M}$, so as the H in $\mathscr{M}_1^*$.

Lemma 2 helps to show that the MAS in phase $\mathcal{O}_{13}$ exponentially converge to the output result in hidden-layer neurons, which is stored in species P_i^l, even though there are no isolated LAS/GAS points for the MAS (22) and (23). Then, we conclude with the following.

Theorem 3: The biochemical feedforward module composed of the MAS (17), (20)–(24) implements the basic feedforward computation with an exponential convergence to the target results. Such design can be extended to the feedforward FCNN of any depth and width.

B. Preceding Calculation System

The pBCRN is composed of (26)–(28). For showing the asymptotic stability, we first consider the uppermost part (26), which can be decomposed into three independent subnetworks [32] with disjoint species set $\mathscr{M}^1 = \{Y^l, Y_e^l, Y_s^l, \tilde{Y}^l, S_3^l, I_y^l\}$, $\mathscr{M}_{(i)}^2 = \{P_i^l, P_{is}^l, \tilde{P}_i^l, I_{p_i}^l\}$ with $i =$

1, 2. For convenience, we represent $\mathscr{M}^1, \mathscr{M}^2_{(i)}$ using the following 6-D network and 4-D CRNs

$$\underbrace{\begin{array}{l} X_1 \xrightarrow{k} X_2 + X_3 + X_4 \\ X_2 + \tilde{X}_2 \xrightarrow{k} \varnothing \\ X_3 + \tilde{X}_3 \xrightarrow{k} \varnothing \end{array}}_{\mathscr{M}^1} \quad \underbrace{\begin{array}{l} Y_1 \xrightarrow{k} Y_2 + Y_3 \\ Y_2 + \tilde{Y}_2 \xrightarrow{k} \varnothing \end{array}}_{\mathscr{M}^2_{(i)}}. \tag{35}$$

Note that $\mathscr{M}^1$ and $\mathscr{M}^2_{(i)}$ have the same structure with one replication reaction and several annihilation reactions. The following propositions aim to prove that both have boundary equilibrium points that are GAS relative to the compatibility class.

Proposition 4: Given that $y = (y_1, y_2, y_3, \tilde{y}_2)^\top, y(0) = (y_1(0), y_2(0), y_3(0), \tilde{y}_2(0))^\top \in \mathbb{R}^4_{\geq 0}, d_1 = y_1(0) + y_3(0), d_2 = y_1(0) + y_2(0) - \tilde{y}_2(0)$. For any positive constant $k \in \mathbb{R}_{>0}$ the MAS $\mathscr{M}^2_{(i)}$ with the dynamical equations

$$\dot{y}_1(t) = -k y_1(t),\ \dot{y}_2(t) = k y_1(t) - k y_2(t)\tilde{y}_2(t)$$
$$\dot{y}_3(t) = k y_1(t),\ \dot{\tilde{y}}_2 = -k y_2(t)\tilde{y}_2(t)$$

is GAS in a stoichiometric compatibility class.

The proof in the Supplementary Material is completed by utilizing the candidate polyhedral Lyapunov function $V = \|y - \bar{y}\|_1$ for stability analysis. In particular, it is impossible to find a uniform positive constant K for all $t \geq 0$ to describe the global exponential stability of $\mathscr{M}^2_{(i)}$ because multiple boundary equilibrium reactions in our design force the reaction rate to be slow when the trajectory is about to reach equilibrium. Therefore, we provide the following definitions.

Definition 6 (Out-epsilon region): For any point $\bar{x}$ in $\mathbb{R}^n_{\geq 0}$ and $\forall \epsilon > 0$, we call the region $\mathcal{O} = \{x \in \mathbb{R}^n_{\geq 0} | \|x - \bar{x}\|_1 \geq \epsilon\}$ *out-epsilon region of* $\bar{x}$.

Definition 7 (Finite-time exponential stability): Consider a MAS admitting an asymptotic stable equilibrium point $\bar{x}$. It is *finite-time exponential stable* if for any finite time T, there exists an out-epsilon region $\mathcal{O}_T$ of $\bar{x}$ and positive constants $M, \gamma > 0$ such that $\|x(t) - \bar{x}\| \leq M e^{-\gamma t}$ for $x(t) \in \mathcal{O}_T$.

It implies that the system will converge exponentially to any ϵ-neighborhood of the equilibrium. Then, we give the convergence description of the MAS $\mathscr{M}^2_{(i)}$.

Proposition 5: The MAS $\mathscr{M}^2_{(i)}$ allows the finite-time exponential stability for the boundary equilibrium point in any stoichiometric compatibility class.

It is worth saying that the exponential order will decrease following the decrease of ϵ, which is consistent with our intuition. The following proposition shows the stability of $\mathscr{M}^1$.

Proposition 6: Let $x(t)$ be the concentration vector of $\mathscr{M}^1$ and $x(0) = (x_1(0), x_2(0), x_3(0), x_4(0), \tilde{x}_2(0), \tilde{x}_3(0))^\top \in \mathbb{R}^6_{\geq 0}$. Assume that $d_1 = x_1(0) + x_4(0), d_2 = x_1(0) - \tilde{x}_3(0) + x_3(0), d_3 = x_1(0) - \tilde{x}_2(0) + x_2(0)$. For any positive constant k, the MAS $\mathscr{M}^1$ with equations

$$\dot{x}_1(t) = -k x_1(t),\ \dot{x}_4(t) = k x_1(t)$$
$$\dot{x}_2(t) = k x_1(t) - k x_2(t)\tilde{x}_2(t),\ \dot{\tilde{x}}_2 = -k x_2(t)\tilde{x}_2(t)$$
$$\dot{x}_3(t) = k x_1(t) - k x_3(t)\tilde{x}_3(t),\ \dot{\tilde{x}}_3 = -k x_3(t)\tilde{x}_3(t)$$

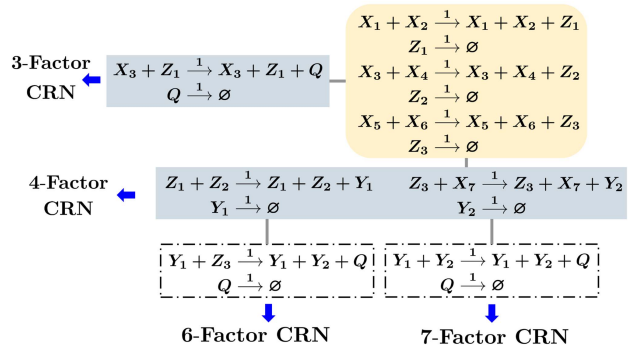

Fig. 5. Schematic diagrams of reaction network structures appeared in $\mathcal{O}_{27}$. The *root block* (shaded yellow) for the first product is shared by all factor CRNs. Different *stem blocks* (shaded blue) for the second product are shared by different factor CRNs. Only six/seven-factor CRNs need the *leaf blocks* (dashed frame) for the third multiplication.

is GAS in a stoichiometric compatibility class and allows finite-time exponential stability.

Since $\mathscr{M}^1$ and $\mathscr{M}^2_{(i)}$ are independent subnetworks without noncoupling components, the parent network (26) also admits a unique global asymptotically stable boundary equilibrium and finite-time exponential stability following Propositions 4–6. Further, let $x^l_e = (e^+_l, e^-_l, s^l_y)^\top, y^l_p = (s^l_{p1}, s^l_{p2})^\top$. The exponential convergence of the pBCRN system in the out-epsilon region is consequently confirmed by considering x^l_e, y^l_p, e_l via Lemma 2. Regarding the jBCRN system, Lemma 2 also promotes to induce the asymptotic stability of species C_a following the bistability of (29).

C. Backpropagation Module

The MAS uBCRN has been demonstrated as GES in the past work [42]. In this section, we work on the ngBCRN system and show that it has a GES equilibrium point, which indicates that all reactions occurring in parallel can execute sequential calculations to get all negative gradient values. Furthermore, the final fixed point of the learning module after phase $\mathcal{O}_{29}$ precisely aligns with the fixed point of the stochastic gradient descent iterative algorithm.

In phase $\mathcal{O}_{27}$, four structural paradigms designed for computing multiplication are shown in Fig. 5. Every paradigm is constituted by several blocks that consist of reactions calculating two-factor multiplications. In particular, the concentration of reactant complexes of the root block is fixed, whereas the reactant complexes of stem and leaf blocks come from the product complexes in the upper blocks whose concentration might vary over time. Thus, we propose a variant of Lemma 2 to handle the composition of multiplication blocks.

Lemma 3: Consider the MAS $\mathscr{M}$ with the kinetic (33), and assume that the certain species subset $\{S_1, S_2\}$ has the limiting steady states, and $\exists\ q_i, \beta_i > 0$ such that $|s_i(t) - \bar{s}_i| < q_i e^{-\beta_i t}$ for $\forall t \geq 0$. Then if we embed the MAS $\mathscr{M}$ (called type-II CRN)

$$S_1 + S_2 \xrightarrow{1} S_1 + S_2 + H$$
$$H \xrightarrow{1} \varnothing \tag{36}$$

TABLE I
ALL BIOCHEMICAL REACTION MODULES OF THE BFCNN AND THE RELATED STABILITY TYPES

Modules		Reactions	Stability
Assignment Module	$\mathcal{O}_1$:	$C_i^l + X_q^i \xrightarrow{1} C_i^l + X_q^i + S_q^l,\ S_q^l \xrightarrow{1} \varnothing,\ q \in \{1,2\},\ l \in \{1,...,\tilde{p}\}$ $C_i^l + D^i \xrightarrow{1} C_i^l + D^i + S_3^l,\ S_3^l \xrightarrow{1} \varnothing,\ i \in \{1,...,p\}$	GES
	$\mathcal{O}_3$:	$C_i^l \xrightarrow{k} \tilde{C}_i^l$	GES_rs [1]
	$\mathcal{O}_5$:	$\tilde{C}_{o_l}^l \xrightarrow{k} C_{\tilde{p}+o_l}^l,\ \tilde{C}_{p-\tilde{p}+l}^l \xrightarrow{k} C_l^l,\ \tilde{C}_{j_l}^l \xrightarrow{k} C_{j_l}^l$	GES_rs
Feedforward Module *lwsBCRN*	$\mathcal{O}_7/\mathcal{O}_{15}$: [2]	$W_i^\pm + S_1^l \xrightarrow{1} W_i^\pm + S_1^l + N_{il}^\pm,\ B_i^\pm \xrightarrow{1} B_i^\pm + N_{il}^\pm,$ $W_{i+2}^\pm + S_2^l \xrightarrow{1} W_{i+2}^\pm + S_2^l + N_{il}^\pm,\ N_{il}^\pm \xrightarrow{1} \varnothing,\ i \in \{1,2\}$	GES
sigBCRN	$\mathcal{O}_9/\mathcal{O}_{17}$:	$N_{il}^+ + N_{il}^- \xrightarrow{k} \varnothing,$	GES_rs
	$\mathcal{O}_{11}/\mathcal{O}_{19}$:	$N_{il}^\pm + Half \xrightarrow{1} N_{il}^\pm + Half + P_{il}^\pm,$ $N_{il}^\pm + P_{il}^\pm \xrightarrow{1} N_{il}^\pm$	GES
	$\mathcal{O}_{13}/\mathcal{O}_{21}$:	$p_{il}^+(0) > 0$: $N_{il}^+ + P_{il}^+ \underset{1}{\overset{1}{\rightleftharpoons}} 2P_{il}^+ + N_{il}^+,\quad N_{il}^+ \xrightarrow{1} \varnothing$ $p_{il}^-(0) > 0$: $N_{il}^- + 2P_{il}^- \xrightarrow{1} 3P_{il}^- + N_{il}^-,$ $N_{il}^- + P_{il}^- \xrightarrow{1} N_{il}^- \xrightarrow{1} \varnothing$ $P_{il}^+ \xrightarrow{1} P_{il}^+ + P_i^l,\quad P_{il}^- \xrightarrow{1} P_{il}^- + P_i^l,\quad P_i^l \xrightarrow{1} \varnothing$	GAS_es
Judgment Module *pBCRN* (e.g., 6-dimensional network $\mathscr{M}^1$ + type-I CRN)	$\mathcal{O}_{23}$:	$Y^l \xrightarrow{k} Y_e^l + Y_s^l + \tilde{Y}^l,\ Y_e^l + S_3^l \xrightarrow{k} \varnothing,\ Y_s^l + I_y^l \xrightarrow{k} \varnothing$ $S_3^l \xrightarrow{1} E_l^+ + S_3^l,\quad E_l^+ \xrightarrow{1} \varnothing$ $Y_e^l \xrightarrow{1} E_l^- + Y_e^l,\quad E_l^- \xrightarrow{1} \varnothing$ $I_y^l \xrightarrow{1} S_y^l + I_y^l,\quad S_y^l \xrightarrow{1} \varnothing$ $E_l^+ \xrightarrow{1} E_l^+ + E_l,\ E_l^- \xrightarrow{1} E_l^- + E_l,\ E_l \xrightarrow{1} \varnothing$	Finite-time exponential stability
jBCRN	$\mathcal{O}_{25}$:	$A_l \xrightarrow{k_1} 2E_l,\quad 2E_l \xrightarrow{k_2} E_l + A_l,$ $E_l + A_l \xrightarrow{k_3} A_l,\quad E_l \xrightarrow{k_4} \varnothing$ $E_l \xrightarrow{k} E_l + C_a,\ C_a \xrightarrow{k} \varnothing,\ \forall l \in \{1,...,\tilde{p}\}$	Bistability
Learning Module *ngBCRN* (e.g., 6-factor CRN for multiplication + the negative part of gradient for addition)	$\mathcal{O}_{27}$:	***Multiplication*** | ***Addition*** $X_1 + X_2 \xrightarrow{1} X_1 + X_2 + Z_1,\ Z_1 \xrightarrow{1} \varnothing$ | $Q_{s+-}^l \xrightarrow{1} Q_{s+-}^l + Par_s^-$ $X_3 + X_4 \xrightarrow{1} X_3 + X_4 + Z_2,\ Z_2 \xrightarrow{1} \varnothing$ | $Q_{s-+}^l \xrightarrow{1} Q_{s-+}^l + Par_s^-$ $X_5 + X_6 \xrightarrow{1} X_5 + X_6 + Z_3,\ Z_3 \xrightarrow{1} \varnothing$ | $Q_{v-}^l \xrightarrow{1} Q_{v-}^l + Par_v^-$ $Z_1 + Z_2 \xrightarrow{1} Z_1 + Z_2 + Y_1,\quad Y_1 \xrightarrow{1} \varnothing$ | $Par_s^- \xrightarrow{1} \varnothing$ [3] $Z_3 + Y_1 \xrightarrow{1} Z_3 + Y_1 + Q,\quad Q \xrightarrow{1} \varnothing$ | $Par_v^- \xrightarrow{1} \varnothing$ [3]	GES
uBCRN	$\mathcal{O}_{29}$:	$Par_i^\pm + L \xrightarrow{1} Par_i^\pm + L + \Delta W_i^\pm,\ \Delta W_i^\pm \xrightarrow{1} W_i^\pm,$ $\Gamma_i^\pm \xrightarrow{1} \Gamma_i^\pm + W_i^\pm,\ W_i^\pm \xrightarrow{1} \varnothing,\ i \in \{1,...,6\}$ $Par_{j+6}^\pm + L \xrightarrow{1} Par_{j+6}^\pm + L + \Delta B_j^\pm,\ \Delta B_j^\pm \xrightarrow{1} B_j^\pm,$ $\Gamma_{j+6}^\pm \xrightarrow{1} \Gamma_{j+6}^\pm + B_j^\pm,\ B_j^\pm \xrightarrow{1} \varnothing,\ j \in \{1,2,3\}$	GES
Clear-out Module	$\mathcal{O}_{31}$:	***clear − out BCRN*** | ***compensatory BCRN*** $X \xrightarrow{1} \varnothing$ | $I \xrightarrow{1} I + X$ | $X \xrightarrow{1} \varnothing$	GES

[1] For the type of stability, the abbreviation *GES_rs* means GES relative to its stoichiometry compatibility class and *GAS_es* means GAS convergence to the equilibirums set.
[2] In the Feedforward Module, the two-phase mark, i.e., $\mathcal{O}_i/\mathcal{O}_j$, implies that the type of BCRN listed in the considering row implements the linear/nonlinear feedforward computation in the input-hidden layer (operate in the smaller-subscript phase) and hidden-output layer (operate in the bigger-subscript phase), respectively.
[3] $s \in \{1,2,3,4,7,8\}$, $v \in \{5,6,9\}$

into $\mathscr{M}$, the species H will inherit the exponential convergence of equilibrium $\bar{s}_1\bar{s}_2$ in the parent network $\mathscr{M}_2^*$.

Since the reaction system in the root block in Fig. 5 is GAS, all types of factor CRNs systems are easily shown to be GAS in a similar way as the proof of Lemma 2. Then, combining Lemma 3, we have the next theorem.

Theorem 7: Given any nonnegative catalyst concentration, ngBCRN allows global exponential stability and implements a negative gradient calculation. Moreover, the fixed point of uBCRN equals the new weight matrix in every iteration.

Therefore, based on our termination mechanism, the final fixed point of the uBCRN system is exactly the fixed point of the stochastic gradient descent algorithm. In addition, compared to [14], the computation results of our backpropagation module are robust to the initial concentrations of gradient species and weights species due to GES.

V. APPLICATION TO BINARY CLASSIFICATION PROBLEMS

In this section, we validate the performance of the proposed BFCNN in Table I on two logic classification problems having linear ("OR" gate) and nonlinear ("XOR" gate) separation boundaries, and one real-world dataset of COVID-19 Immune Altas: peripheral blood mononuclear cell (PBMC) data via computer simulations. By solving the dynamical equations composed of all modules in the BFCNN, it can finish the workflow of the biochemical neural network. Naturally, the initial concentrations of some species should be given properly to solve ODEs. All

TABLE II
RUNTIME OF THE BFCNN TRAINING PROCESS

Dataset (size)	$\tilde{p}$	TSPAN	Runtime
"OR" Gate (4)	2	[0 2000]	40.701095s
"XOR" Gate (4)	2	[0 3000]	86.102931s
PBMC (40)	2	[0 9000]	230.731983s
PBMC (60)	4	[0 10000]	303.653511s
PBMC (80)	4	[0 10000]	363.823153s

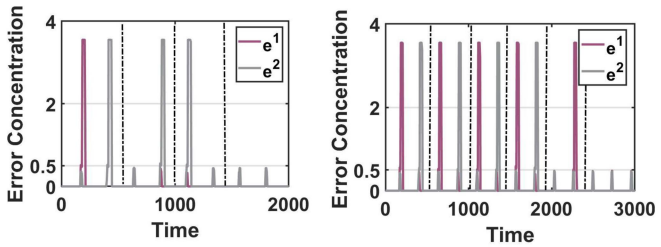

Fig. 6. Concentration curves of error species for training the binary classification functions: "X_1 OR X_2" (left) and "X_1 XOR X_2" (right), where black dotted lines divide different epochs.

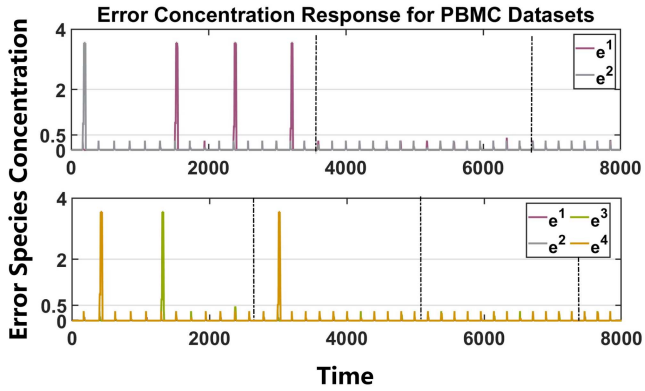

Fig. 7. Concentration-response curves of error species for training the PBMC (40) dataset (top) and PBMC (60) dataset (bottom), where black dotted lines divide different epochs.

experiments are made in the MATLAB (2021b) environment, and the solver forordinary differential equations (ODEs) is the *ode45*.

From Table I, there are 32 oscillatory species used in (12) for the BFCNN. We set the oscillator rate constant $k_o = 1$, the initial concentrations of O_{31}, O_{32} to be 1 while thoseof others oscillatory species to be 10^{-6} [25]. The error threshold is set as 0.5, which results from setting the rate constants of (29) to be $k_1 = 8, k_2 = 1, k_3 = 2, k_4 = 0.4375$.

A. Logic Classification Dataset

1) "OR" Gate: This dataset is for a 2-D classification problem comprising four points $\chi_{\cdot 1}^1 = (1,0,1)^\top, \chi_{\cdot 2}^1 = (0,0,0)^\top, \chi_{\cdot 3}^1 = (1,1,1)^\top, \chi_{\cdot 4}^1 = (0,1,1)^\top$, i.e., $p = 4$, where the third element of every point represents label. By setting the batch size $\tilde{p} = 2$ and the initial condition (see details in Section II of the Supplementary Material), we apply the BFCNN to perform the classification task on this dataset. The experiment stops at the sixth iteration at which the concentration of error species E_l in the jBCRN falls below 0.5, and the final output is $y = [0.6654, 0.3950, 0.6358, 0.5724]$, corresponding to the label output $[1, 0, 1, 1]$. Clearly, the BFCNN is well-behaved in this dataset, since those four samples are classified right. Tables II further report the runtime that the BFCNN needs, where the "TSPAN" column indicates the time span for solving the corresponding ODEs. Note that the time magnitude (more than 40 s) looks rather large compared with directly using the neural network. This phenomenon is normal since solving ODEs is very time consuming in the computer environment. If the BFCNN works in vivo, the time will be greatly reduced since biochemical reactions occur rapidly. We also record the runtime that each module needs at one iteration, being 0.004902 s, 0.007000 s, 0.066171 s, and 0.103077 s for assignment, feedforward, judgment, and backpropagation module, respectively.

To observe the error of the BFCNN during the iteration process, we present Fig. 6 to exhibit the concentrations of error species at different iteration times. It is clear that in the fourth epoch, the errors, e_1 and e_2, are smaller than 0.5. Further, the heatmap is provided to visualize the linear ("OR" gate) and nonlinear (for the following "XOR" gate) classification surface in Section III of the Supplementary Material.

2) "XOR" Gate: This dataset is also for a 2-D classification problem with four points $\chi_{\cdot 1}^2 = (1,0,1)^\top, \chi_{\cdot 2}^2 = (0,0,0)^\top, \chi_{\cdot 3}^2 = (1,1,0)^\top, \chi_{\cdot 4}^2 = (0,1,1)^\top$. Following the same logic as that for "OR" gate, we conduct experiments on this dataset with $\tilde{p} = 2$ and the initial condition given in Section II of the Supplementary Material. At the 11th iteration, the training is terminated to produce the output $y =$ [0.5129,0.4188.0.4503,0.5082], i.e., label output [1,0,0,1], which completely coincides with the samples. Likewise, the training time and the error for this dataset are presented in Table II and Fig. 6.

B. COVID-19 Immune Altas: PBMC Dataset

The PBMC dataset is a subset of a gene expression dataset (CellXGene[1]) from various types of cells with different origins (i.e., whether each cell is from a COVID-19 patient). Previous studies have shown that in the monocyte-dendritic cell system of COVID-19 patients, the expression level of *SIGLEC1* is significantly elevated, while *HLA-DRB5* expression is downregulated by syndrome coronavirus 2. Based on this, we selected $40,\ 60,\ 80$ cell samples from this system to categorize individuals by analyzing the expression of *SIGLEC1* and *HLA-DRB5* in COVID-19 patients (labeled as "0") versus healthy individuals (labeled as "1"), with the expression of these two genes serving as a two-dimensional input. The datasets are randomly divided into training and test sets with a split of $40(32 + 8)$,

[1]The data is available on https://cellxgene.cziscience.com/collections/b9fc3d70-5a72-4479-a046-c2cc1ab19efc [43]

$60(48+12)$, and $80(64+16)$, the batch sizes are selected as $2, 4$, and 4, respectively, and the initial conditions are given in the Section II of the Supplementary Material. Three training groups met the error requirements after the 32th, 36th, and 32th iterations, respectively, and the classification performance of the updated BFCNN is evaluated by using its feedforward component on the test sets. The classifier ultimately generates the following confusion matrice (refer to Fig. S1 in the Supplementray Material): TP $= 7$, FN $= 0$, TN $= 1$, FP $= 0$ for the 40 set, TP $= 10$, FN $= 0$, TN $= 2$, FP $= 0$ for the 60 set, and TP $= 12$, FN $= 0$, TN $= 4$, FP $= 0$ for the 80 set with accuracy $= 1$ and F_1 score $= 1$ for all cases. Here, we only display the concentration change of error species to visualize training the 40 and 60 datasets in Fig. 7 due to space limit.

VI. Conclusion and Points of Future Study

In Section I of this article, we have devised a biochemical reaction system with learning capability that first implements a complete FCNN with the gradient descent backpropagation process. The assignment module introduces an innovative mechanism for automatically inputting training samples akin to computers, making it applicable to the biological realization of any iterative algorithm. The Sigmoid function, concerning real inputs, is precisely computed by a revision of the autocatalytic reactions. In addition, to endow our system with learning capacity, we present a precise and robust realization of the learning module, with its final equilibrium concentration being exactly the fixed point of the gradient descent algorithm. Subsequently, we offer theoretical support for our design through rigorous dynamic analysis, ensuring asymptotic stability and exponential convergence. This might facilitate the building of a computational module library with explicit properties, benefiting future applications.

Despite all computational modules allowing exponential convergence, realization errors are inevitable due to each step of computation, each times of information processing, the finite phase length of oscillators, and the intricate construction. Moreover, as the neural network becomes complex, the compounding error will increase further. In Section II, we propose a strict error analysis method and provide an error upper bound of our BFCNN. However, while CRNs can theoretically implement FCNN functionality unbiasedly, direct implementation in biological environments is currently highly challenging due to multiple factors, including the architecture of BFCNN (constructed by CRNs that may be out of the "fixed/standard set"), noise, uncertainties, and unintended interactions in biological environments, which are actually giant challenge for synthetic biology. Thus, our work is not intended as an immediate experimental blueprint but a necessary theoretical exploration step to explore the computational potential of CRN model, i.e., to identify the fundamental principles and constraints governing molecular computational circuits.

It should be emphasized again that the main purpose of this series of papers is to break the barrier between computer language and "language understood by living beings" not to construct a novel architecture that competes with other artificial neural networks in terms of training speed and accuracy in the computer environment. Naturally, this is a foundational work and the resulting network cannot handle large-scale data well. However, there are no technical difficulties in programming more complicated neural networks using BCRN. Specifically, the information processing module (biological assignment, judgment, and loop modules) can be directly applied to any neural network framework and even to other machine learning algorithms implemented with CRNs. The core design modules, performing linear (lwsBCRN) and nonlinear (sigBCRN) feedforward computation, gradient computation (ngBCRN) and the update process (uBCRN) in a biological way, can also be extended as building blocks for realizing other architectures. The modular designs may potentially serve as "module libraries" for some CRN simulation software to facilitate cost-effective redesigns and accelerate the development of automated "neural network to BCRN" pipelines. In addition, all designs hinge on precise rate constant settings and specific initial concentration choices. As carriers of all kinds of information or values (not too large normally in biological environments), the species concentrations are efficient to represent them, since in theory any initial concentration value may be designed no matter how large or how small it is. We put the implementation of other techniques in deep learning and developing new types of neural network algorithms based on dynamical systems theory as points of our future study.

References

[1] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, 2000.

[2] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, pp. 339–342, 2000.

[3] C. Zhang et al., "Cancer diagnosis with DNA molecular computation," *Nature Nanotechnol.*, vol. 15, no. 8, pp. 709–715, 2020.

[4] Y. Qian, C. McBride, and D. Del Vecchio, "Programming cells to work for us," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 411–440, 2018.

[5] Y. Qian and D. Del Vecchio, "Robustness of networked systems to unintended interactions with application to engineered genetic circuits," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 4, pp. 1705–1716, Dec. 2021.

[6] S. K. Aoki, G. Lillacci, A. Gupta, A. Baumschlager, D. Schweingruber, and M. Khammash, "A universal biomolecular integral feedback controller for robust perfect adaptation," *Nature*, vol. 570, no. 7762, pp. 533–537, 2019.

[7] F. Xiao and J. C. Doyle, "Robust perfect adaptation in biomolecular reaction networks," in *Proc. 2018 IEEE Conf. Decis. Control*, 2018, pp. 4345–4352.

[8] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with dna-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, 2018.

[9] G. H. Wadhams and J. P. Armitage, "Making sense of it all: Bacterial chemotaxis," *Nature Rev. Mol. Cell Biol.*, vol. 5, no. 12, pp. 1024–1037, 2004.

[10] K. Mori, H. Nagao, and Y. Yoshihara, "The olfactory bulb: Coding and processing of odor molecule information," *Science*, vol. 286, no. 5440, pp. 711–715, 1999.

[11] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *Proc. Nat. Acad. Sci.*, vol. 107, no. 12, pp. 5393–5398, 2010.

[12] M. Vasić, D. Soloveichik, and S. Khurshid, "CRN++: Molecular programming language," *Natural Comput.*, vol. 19, no. 2, pp. 391–407, 2020.

[13] H. Buisman, H. M. M. ten Eikelder, P. A. Hilbers, and A. M. Liekens, "Computing algebraic functions with biochemical reaction networks," *Artif. Life*, vol. 15, no. 1, pp. 5–19, 2009.

[14] M. R. Lakin, "Design and simulation of a multilayer chemical neural network that learns via backpropagation," *Artif. Life*, vol. 29, no. 3, pp. 308–335, 2023.

[15] N. Zhao et al., "Synthetic biology-inspired cell engineering in diagnosis, treatment, and drug development," *Signal Transduct. Target. Ther.*, vol. 8, no. 1, 2023, Art. no. 112.

[16] X. Yan, X. Liu, C. Zhao, and G.-Q. Chen, "Applications of synthetic biology in medical and pharmaceutical fields," *Signal Transduct. Target. Ther.*, vol. 8, no. 1, 2023, Art. no. 199.
[17] S. Ausländer, M. Wieland, and M. Fussenegger, "Smart medication through combination of synthetic biology and cell microencapsulation," *Metabolic Eng.*, vol. 14, no. 3, pp. 252–260, 2012.
[18] S. Okumura et al., "Nonlinear decision-making with enzymatic neural networks," *Nature*, vol. 610, no. 7932, pp. 496–501, 2022.
[19] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with dna strand displacement cascades," *Nature*, vol. 475, no. 7356, pp. 368–372, 2011.
[20] C. C. Samaniego, A. Moorman, G. Giordano, and E. Franco, "Signaling-based neural networks for cellular computation," in *Proc. IEEE 2021 Amer. Control Conf.*, 2021, pp. 1883–1890.
[21] A. Moorman, C. C. Samaniego, C. Maley, and R. Weiss, "A dynamical biomolecular neural network," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 1797–1802.
[22] M. Vasić, C. Chalk, A. Luchsinger, S. Khurshid, and D. Soloveichik, "Programming and training rate-independent chemical reaction networks," *Proc. Nat. Acad. Sci.*, vol. 119, no. 24, 2022, Art. no. e2111552119.
[23] D. F. Anderson, B. Joshi, and A. Deshpande, "On reaction network implementations of neural networks," *J. Roy. Soc. Interface*, vol. 18, no. 177, 2021, Art. no. 20210031.
[24] D. Blount, P. Banda, C. Teuscher, and D. Stefanovic, "Feedforward chemical neural network: An in silico chemical system that learns xor," *Artif. Life*, vol. 23, no. 3, pp. 295–317, 2017.
[25] D. Arredondo and M. R. Lakin, "Supervised learning in a multilayer, nonlinear chemical neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7734–7745, Oct. 2023.
[26] C. C. Samaniego, J. Kim, and E. Franco, "Sequestration and delays enable the synthesis of a molecular derivative operator," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 5106–5112.
[27] X. Zhang and C. Gao, "Persistence of delayed complex balanced chemical reaction networks," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1658–1669, Apr. 2021.
[28] X. Zhang, C. Gao, and D. Dochain, "On persistence of some $\{\mathcal{W}_I\}$ - endotactic chemical reaction networks," *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 7420–7435, Dec. 2023.
[29] D. Angeli, M. A. Al-Radhawi, and E. D. Sontag, "A robust Lyapunov criterion for nonoscillatory behaviors in biological interaction networks," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3305–3320, Jul. 2022.
[30] D. B. Forger, *Biological Clocks, Rhythms, and Oscillations: The Theory of Biological Timekeeping*. Cambridge, MA, USA: MIT Press, 2017.
[31] M. Feinberg, "Complex balancing in general kinetic systems," *Arch. Rational Mechan. Anal.*, vol. 49, no. 3, pp. 187–194, 1972.
[32] M. Feinberg, "Lectures on chemical reaction networks," Notes of lectures given at the Mathematics Research Center, Univ. Wisconsin, 1979.
[33] F. Horn and R. Jackson, "General mass action kinetics," *Arch. Rational Mechan. Anal.*, vol. 47, no. 2, pp. 81–116, 1972.
[34] M. Feinberg, *Foundations of Chemical Reaction Network Theory*. Cham, Switzerland: Springer, 2019.
[35] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham, Switzerland: Springer, 2018.
[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT press, 2016.
[37] M. Vasic, C. Chalk, S. Khurshid, and D. Soloveichik, "Deep molecular programming: A natural implementation of binary-weight relu neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9701–9711.
[38] X. Shi, C. Gao, and D. Dochain, "Accurate control to run and stop chemical reactions via relaxation oscillators," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10441–10446, 2023.
[39] M. Lachmann and G. Sella, "The computationally complete ant colony: Global coordination in a system with no hierarchy," in *Proc. Eur. Conf. Artif. Life*, 1995, pp. 784–800.
[40] F. Fages, G. Le Guludec, O. Bournez, and A. Pouly, "Strong turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs," in *Proc. 15th Int. Conf. Comput. Methods Syst. Biol.*, 2017, 15, pp. 108–127.
[41] T. Wilhelm, "The smallest chemical reaction system with bistability," *BMC Syst. Biol.*, vol. 3, pp. 1–9, 2009.
[42] Y. Fan, X. Zhang, and C. Gao, "Towards programming adaptive linear neural networks through chemical reaction networks," *IFAC-PapersOnLine*, vol. 55, no. 18, pp. 7–13, 2022.
[43] P. S. Arunachalam et al., "Systems biological assessment of immunity to mild versus severe COVID-19 infection in humans," *Science*, vol. 369, no. 6508, pp. 1210–1220, 2020.

Yuzhen Fan received the B.Sc. degree in information and computational science from the College of Science, Northeastern University, Shenyang, China, in 2020. She is currently working toward the Ph.D. degree in operation research and cybernetics with the School of Mathematical Sciences, Zhejiang University, Hangzhou, China.

Her research interests include areas of control theory, synthetic biology, and chemical reaction networks theory.

Xiaoyu Zhang received the B.Sc. degree in mathematics from Anhui University, Hefei, China, in 2017, and the Ph.D. degree in operation research and cybernetics from Zhejiang University, Hangzhou, China, in 2022.

She is currently an Associate Researcher with the School of Mathematics, Southeast University, Nanjing, China. Her research interests are in the areas of control and chemical reaction networks theory.

Chuanhou Gao (Senior Member, IEEE) received the B.Sc. degree in chemical engineering from the Zhejiang University of Technology, Hangzhou, China, in 1998, and the Ph.D. degree in operational research and cybernetics from Zhejiang University, Hangzhou, China, in 2004.

From 2004 to 2006, he was a Postdoctoral with the Department of Control Science and Engineering, Zhejiang University. Since June 2006, he has joined the Department of Mathematics, Zhejiang University, where he is currently a Professor. His research interests include data-driven modeling, control and optimization, chemical reaction network theory, and thermodynamic process control.

Dr. Gao is an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL and *International Journal of Adaptive Control and Signal Processing*.

Denis Dochain (Fellow, IEEE) received the master degree in electrical engineering and the dual joint Ph.D. degree in applied science with a "thèse d'agrégation de l'enseignement supérieur" from Université Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, in 1982, and 1986 and 1994, respectively.

He has been CNRS Associate Researcher with the LAAS, Toulouse, France, in 1989, and Professor with the Ecole Polytechnique de Montréal Montreal, QC, Canada in 1987–1988 and 1990–1992. Since 1990, he has been with the FNRS, Belgium since 1990. Since September 1999, he has been a Professor with the ICTEAM (Institute), Université Catholique de Louvain, and Honorary Research Director of the FNRS. He has been invited Professor with Queen's University, Kingston, ON, Canada between 2002 and 2004. Since 2005, he has been a Full Professor with the UCL since 2005. He has (coauthored) authored five books, more than 160 papers in refereed journals, and 260 international conference papers. His main research interestsinclude nonlinear systems, thermodynamics-based control, parameter and state estimation, adaptive extremum seeking control and distributed parameter systems, with application to microbial ecology, environmental, biological, and chemical systems, and electrical and mechanical systems.

Dr. Dochain is the Editor-in-Chief of the *Journal of Process Control*, Senior Editor for IEEE TRANSACTIONS OF AUTOMATIC CONTROL, and Associate Editor of *Automatica*. He is active in IFAC since 1999 (Council member, Technical Board member, Publication Committee member and Chair, TC and CC chair). He was the recipient of the IFAC outstanding service award in 2008 and the title of Doctor Honoris Causa from the INP Grenoble on December 13, 2020, and is an IFAC Fellow since 2010. He received the title of Doctor Honoris Causa from the INP Grenoble on December 13, 2020.