Received: 3 April 2023 | Revised: 17 May 2023 | Accepted: 25 May 2023 | Published online: 25 May 2023

REVIEW

Journal of Data Science and Intelligent Systems
2023, Vol. 00(00) 1–5
DOI: 10.47852/bonviewJDSIS3202934

BON VIEW PUBLISHING

Interdisciplinary Trends in the Reintegration of Organisms with Perceptron Units

Yuxuan Deng[1], Zhangyue Li[2] and Junpeng Chen[3,*]

[1]*Qiqihar Medical University, China*

[2]*Chongqing University of Posts and Telecommunications, China*

[3]*Westlake University, China*

Abstract: Various artificial algorithms have emerged from imitating the behavior of primitive cells. Despite their independent development, these algorithms still have the potential to integrate with the intelligence of the primitive biological cells and control their behavior. Research on biologically inspired intelligence emerged earlier, with the goal of combining algorithms with the inherent intelligence of living organisms. When microbial behavior meets the requirements of the algorithm, such microorganisms can be used as a carrier for computation. This review analyzes the shortcomings of this approach and points out potential directions for development, including using existing structures inside cells to simulate control systems such as circuits, in order to construct cells that fully satisfy the needs of the algorithm. However, when using classical control systems to achieve the goal, it is necessary to construct a fairly complex internal cell structure, and this review analyzes the shortcomings of this approach. Subsequently, by analyzing recent research, this review points out the latest novel research direction, in which some scholars attempt to construct artificial cell structures to directly achieve the function of the algorithm by building low-implementation-difficulty internal control systems in cells. However, there are still some problems to be solved, and this review summarizes the relevant research and briefly discusses the current theoretical challenges.

Keywords: biological intelligence, convergence, neural network

1. Introduction

With the rise of applications such as Chat-GPT [1], DALL-E [2], and SCUBA [3], various types of neural networks are becoming more and more important. The birth of neural networks originated from early connectionism, which generates a single real-valued output by imitating the network cluster of nerve cells. Neural networks originate from the modeling of cell clusters. Therefore, when designing neural networks, if appropriate bionic constraints are added, various artificial neural network algorithms still have the potential to bind to original biological cells. If the cells themselves can be programmed, it is possible to specifically cure various diseases and control various synthetic reactions. This review categorizes research related to biologically inspired intelligence into the following levels: (1) selecting existing cells that match the computational needs of the algorithm; (2) constructing control systems within the cell using existing organelles to implement the algorithm; and (3) controlling protein synthesis within the cell to directly implement the algorithm by exploiting the physical and chemical properties of molecules and minimizing the impact of complex life processes within the cell. Subsequently, the challenges faced in current research are presented in Section 4, and the breakthroughs in research from a theoretical perspective are elucidated in Sections 5 and 6. Finally, Section 7 summarizes the significance of current research. This review aims to provide an overview of the current state of research in biologically inspired intelligence.

The rise of biological intelligence: The earlier research on biological intelligence aimed to combine algorithms with the innate intelligence of living organisms. Typically, a certain type of microorganism with cellular structure is used, and when there are enough microorganisms, their behavior follows a certain statistical pattern. When this statistical pattern can meet the needs of the algorithm, such microorganisms can be used as a computational carrier. However, this method is usually crude, limited in functionality, and highly unpredictable.

Rough imitation using microbial features: Tero et al. [4] and Mauttone and Urquhart [5] used the swarm intelligence of slime molds to realize a multi-agent optimization algorithm and complete the construction of a railway network. Ajzen and Fishbein [6], Alim et al. [7], and Burchett et al. [8] further set up the Monte Carlo Physarum Machine, using food as the representative of the sample, inputting the slime mold network, and then using the slime mold as a neural unit with its own activation function, and taking the distribution of slime mold on the surface of the object as the network output structure.

The cleverness of this experiment is that it avoids the numerical problem of neural networks. The slime mold network is an objective

*Corresponding author: Junpeng Chen, Westlake University, China. Email: chenjunpeng@westlake.edu.cn.

and real spatial position output, rather than the numerical output of the neural network output layer [9–11], but the ultimate goal is the spatial distribution of the railway network, so it does not need to pay attention to the transformation of cell position and numerical output. Therefore, the algorithm is limited to specific path problems, such as the Sloan Digital Sky Survey model [8]: the researchers use the MCOM algorithm to reconstruct the cosmic network image on the SDSS dataset.

The performance of microorganisms can be manually screened to achieve a more precise algorithm control process: In some recent studies, significant progress has been made in optimizing bio-intelligence models. Adamatzky [12] conducted a more in-depth investigation into this field. The results of his research show that different fungi exhibit distinct electrical activity peak characteristics and demonstrate that the distribution of fungal word lengths matches the distribution of human language. This highlights a certain degree of model transferability; when the intelligence of a biological organism aligns with the algorithm's properties, the organism itself can serve as the physical medium for the algorithm, with similar applications found in academia. For instance, in SCUBA, protein sequences are treated as unsupervised time-series text. After dimensionality reduction, proteins are simplified into encodable one-dimensional time-series information, which can then be used to predict protein structures using language models. In recent reports related to Andrew Adamatzky's work, microbial spike signals were used as gate units and mycelial cultures were employed for computation. These computers exhibit a certain level of computational fault tolerance and can be designed with different logic functions.

Whiting and other scholars [13, 14] further confirm the feasibility of microbial programming. They proposed physically programming slime mold computers. This study employed different electrical signals to stimulate microorganisms, causing changes in the network connection weights, which is highly similar to the backpropagation process in neural networks. In some research [15, 16], scholars have already applied convergent models to the construction of microbial networks. However, these studies are limited to graphical computations, meaning they require the use of the target computation's graph properties.

However, the problem is that the reaction inside the microorganism is not always controllable, which requires the organism to be abstracted into a mathematical model and given certain constraints, rather than constructing the corresponding organism according to the conditions. That is to say, myxomycetes can use spatial position as input and spatial position as output of calculation. When the data is not input by spatial position and not output by spatial position, the model is difficult to use. Meanwhile, the biochemical reactions in living organisms are highly uncertain, so this approach has great potential but is difficult to generalize in terms of material conditions. For example, in the case of nanorobots, it is necessary to incorporate existing intelligent algorithms into molecular structures, which is in line with the starting point of the slime mold network. However, it requires controllable molecular reactions rather than being constrained by existing microbial conditions.

2. From the Cellular Level to the Molecular Level

Some researchers have opened up new avenues by attempting to simulate circuits from the interactions of organelles within the cell, utilizing existing cellular structures to construct cells that fully meet the algorithmic needs: The reaction in natural organisms is difficult to control. In order to avoid uncertainty in biological behavior, Hanahan and Weinberg [17] proposed to circuit cells. However, these circuits also have a lot of uncertainties. For example, there is a certain mutual interference between circuit components. Molecules in cells usually show a complex network of interactions. It is difficult to establish a general situation at the same time for both cell modeling and circuit component modeling. The characteristics of electrical components are difficult to be consistent with the biological characteristics of cells. For example, the flow solution in cells is inconsistent with the form of electrical discrete signals, and there is a problem of signal-type conversion. Elements with other electrical variables such as inductance have more than one working state, which is difficult to simulate in cells.

In recent research, bio-inspired algorithms, mainly based on neural networks, have made significant progress. To reproduce the functionality of these algorithms in cells, some scholars propose that the behavior of cells can be fully controlled by controlling the intracellular macromolecules, thus reproducing the specific algorithms based on neural networks. Chen et al. [18] considered the need to program the inside of the cell and use the molecular structure to solve these problems. Chen et al. [18] have used the protein synthesis logic gate structure and implemented the Winner-take-all neural network, which is based on the synthesis of artificial proteins with specific functions. The artificial protein synthesis technology led by AlphaFold2 has made a major breakthrough [19, 20]. The synthesis of proteins similar to perceptron units based on artificial neural network algorithms is becoming increasingly achievable. This can fundamentally solve the problem of low controllability of cell responses and use proteins to form a perceptron of the neural network to realize the functional operation of the neural network inside the cell and classify molecular functions [21]. This may solve the problem of molecular controllability of nanorobots and identify signals generated by early cell differentiation in genomics.

3. The Difference Between Cell Molecular Level and Network Unit Level

In the relevant research, the focus is on replicating neural network algorithms within cells. However, there are still many challenges to reproducing algorithms such as neural networks. This is because algorithms such as neural networks, which do not accurately simulate the characteristics of microorganisms during their design, have significant differences: The weighted value of the input protein concentration is used to form the basic perceptron unit (hereinafter referred to as the protein element), but there are differences in detail with the perceptron unit in the computer (hereinafter referred to as the algorithm element) [22]. The operation of the algorithmic meta-network is implemented in stages, that is, input-feature accumulation-output. Each step is performed in order, and the result is not affected by the time unit. The feature accumulation process of the algorithm element itself is continuous, that is, the input is continuously stored on a single perceptron and then output. However, the calculation of the protein network is a continuous and time-consuming process. The activation of the protein itself is impulsive, which is caused by the accumulation of compound concentration, and the protein itself does not participate in the process of feature accumulation. Their difference lies in the nature of information processing. Algorithmic elements need to focus on real value, while protein

elements need to focus on time value, which requires differentiated neural network model design, or even two different development directions [23].

However, this direction still holds potential because biological carriers can achieve more complex functions. Based on the above characteristics, the protein element can make full use of time, frequency domain, and spatial information, while the traditional algorithm element cannot. Traditional algorithm elements need to do manifold learning modeling alone to mine time-frequency information hidden in the data dimension. The protein element may be able to directly use these hidden information to implement dynamic adjustment of neurons [24]. Meanwhile, the calculation of algorithmic elements is frame-driven, while the protein elements are mixed input with sparse time dimension, which can realize deep perception in the form of information flow and exceed hundreds of algorithmic elements in the hidden layer on time-series problems. However, there are some problems in the above operations. In the algorithm design and the use of protein elements, special dynamic datasets are needed as processing.

4. Unresolved Issues

Current research mainly focuses on the following aspects (Table 1):

regular signals with a large number of irregular disturbances, which cause various random changes in the signal. When using protein elements in practical applications, the protein element receives a chaotic signal, which can attempt to be orthogonalized using mathematical algorithms, but the protein element may also be able to orthogonally process the chaotic signal at the protein structure level and have innate advantages in processing chaotic information. By combining the dynamic characteristics of protein elements and performing phase space transformations on the features of input objects on material particles, it is possible to cluster the features of input objects, which may be feasible.

The greater the number of layers in a neural network, the more weight is placed on artificial design. Therefore, some scholars have attempted to simulate shallow networks to weaken the influence of artificial design. In addition, Chen et al. [21] attempted to fit nonchaotic inputs using unsupervised learning. In shallow networks, there is a lack of Gaussian processes between neurons, which can lead to potential overfitting problems. In current algorithms, a single neuron is often used to represent a distribution space instead of an entire layer of neurons, which improves the control over a single neuron but reduces the available feature distribution space and limits the ability to apply "similar but not identical" scenarios. In complex intracellular applications, it is difficult to achieve fitting, and the stability of

Table 1
The findings of current research

Authors	Findings
Hanahan and Weinberg [17]	Circuit cells
Pershin et al. [13], Whiting et al. [14]	Physically programmed the slime mold computer.
Tero et al. [4]	Using the swarm intelligence of slime mold to realize the multi-agent optimization algorithm and complete the construction of the railway network
Kay et al. [15], Zhang et al. [16]	Applying convergent models to the construction of microbial networks.
Chen et al. [18]	Synthesized logic gate structure using protein and implemented Winner-take-all neural network
Burchett et al. [8]	A Monte Carlo velvet vesicle machine was established, taking food as a sample representative, inputting the slime mold network, and then using the slime mold as a neural unit with its own activation function, and taking the distribution of the slime mold on the surface of the object as the network output structure.
Adamatzky [12]	Different fungi exhibit different electrical activity peak signatures, and the distribution of fungal word lengths matches that of human speech
Chen et al. [21]	The perceptron of the neural network composed of protein realizes the functional operation and molecular function classification of the neural network in the cell
Cramer [19] and Chowdhury et al. [20]	Major breakthroughs in artificial protein synthesis technology led by AlphaFold2

Through the analysis above, many issues have gradually been revealed about the differences: the stochastic pulse characteristics of protein elements are nonquantifiable, which makes it possible to implement a series of nonlinear activation functions, dropout and other network layers for protein elements. Once these are implemented, the concept of clustering (or networkization of perceptron units) can be introduced.

The difficulty of controlling chemical reactions inside cells is high because cells are essentially a multi-base, mixed-input, and strongly stochastic large-scale computing body. Therefore, the academic community mainly uses shallow neural networks as simulation objects. Chaotic signals can be simply understood as network training also decreases because the test set samples are difficult to produce stable feature representative values. The fitting results of a single neuron will oscillate within a certain space. Although this "oscillation" can be specially fitted in the chaotic analytical layer, no scholars have worked on this aspect yet.

5. Current Demand for Principles

In the current research, the design process of algorithms is often independent of research on biological intelligence, which means that some potentially useful algorithms may not be considered [25, 26]. For example, redundancy is something that needs to be avoided when

designing neural networks, but it may be beneficial for the implementation of biological intelligence. The network with a large number of neurons should also be considered. When identifying ECG signals, only the linear layer is used as the Gaussian process input, and then the output is directly activated, and the accuracy can exceed 90 %. This is a typical artificial neural network design theory. When the number of perceptron units in the network is much larger than the object required for operation, the learning ability of the neural network will produce redundancy. This redundancy is harmful in the algorithm element, but it may be beneficial in the protein element. Only a single network layer is required, and this network layer only needs to implement a simple Gaussian process to achieve the traditional artificial neural network function. Once the protein element forms a cluster and objectively follows the Gaussian process in substance, it has the opportunity to achieve similar effects.

When considering the design of neural networks, it is also necessary to consider the characteristics of the input and output objects. In the actual construction of large-scale computing neural networks, the protein elements are clustered, rather than individual protein elements. When simulating the neural network, the stimulation of the neural network is the stimulation of the protein cluster. Therefore, the neural network solidified into the protein cluster should have several characteristics: (1) it is better to be multi-input nodes so that it is not necessary to pay attention to the direction of material flow. The input object of the protein cluster is difficult to control its position into the protein cluster, and the protein itself flows in the liquid. (2) Protein flows in the liquid, but to ensure the stability of the whole network, which makes me think of the "myxomycete model." (3) However, if the amount of protein is too much, the concentration of the identified object is hierarchical, which involves the problem that the local protein cluster is shielded.

This requires that any input position is given, and a statistical, protein cluster overall process is used to obtain the output of a cluster overall decision, rather than the traditional algorithm element calculated and output in order. Complex network [27] has similar research with graph neural network and quantum neural network, that is, the position of the input perceptron is changeable.

6. Prospects for Algorithm Development Direction

End-to-end learning refers to the joint learning of all parameters or parameters that previously needed to be determined in several steps in a deep learning model, rather than learning step by step. The current framework of algorithms cannot be directly applied to biological intelligence because the current frameworks mainly use end-to-end frameworks.

In breaking free from framework limitations, we can draw inspiration from the success of the Chat-GPT-4 model [1]. The model first uses high-dimensional information with numerous weak constraints as input and then extracts single-dimensional information as output. Chat-GPT combines semantic logic and convolutional recognition in parallel, abstracting the graphical information obtained through convolution into constraints between elements, which enables understanding the content of images and responding through language models. For example, when asked, "What is in the blue vase?", the Chat-GPT model first abstracts the vase and roses into parameters in a spatial matrix. Then, based on the semantic information conveyed, directional constraints are added to the spatial matrix of the vase and roses, that is, weights are added to recognize the position, and then the answer "roses" is given based on the selected parameters in the position. In this process, both semantic and graphical information are abstracted into spatial parameter matrices as knowledge graphs, which ultimately break free from end-to-end framework limitations. This means that constraints exist between information in different dimensions, such as types of expressions, expression levels, and relevance. The constraints' effects are related to all dimensions, but the constraint relationships may not necessarily be extracted in any given dimension, and information within the same dimension should be adjustable according to constraint conditions.

As with language generation models, the core is to obtain sparse representations of encodable information and then serialize the sparse information over time. By parsing the time-serialized trajectories in space, the activation states of various activation gates (forget gate, memory gate, and output gate) are determined. The forget gate removes low-information components, the memory gate accumulates time-series information, and the output gate extracts language logic constraints, forming a typical high-dimensional data network. This shows that feature information can be effectively accumulated in time series, such as the arrangement of amino acids in proteins, which also uses sequence information accumulated over time, leading to the development of protein generation models.

When cell experiment data are obtained on different time axes, new models are input with data from different batches. Existing models mostly use frame-by-frame time trajectories to construct feature models and can describe the types and expression levels of gene expression within cells as a continuous, time-series trajectory. The problem with this approach is that when sampling, it is difficult to effectively define the position of the target cell on the time axis using time-series trajectories, making it difficult to accurately describe the feature trajectories and to define the feature boundaries of each critical state of the cell with feature values, which involves the issue of describing open boundaries.

7. Instruments

In previous studies, some researchers have used the swarm intelligence shown by myxobacteria to calculate the optimal railway network structure in North America [4]. These are all about the study of random input positions. They all have a common feature: through the continuous trajectory of time, a graphic feature is formed. When using the algorithm element to simulate the random pulse characteristics, it is good at processing the graphic features, and it is known that the protein element is output with random pulse characteristics. Whether there is an interesting relationship between them is an unknown problem. Computer operation is a binary logic operation, and protein operation can be nonbinary: the underlying logic of the computer has only two states of 0 and 1, and the protein can express more than one state, which is very potential. With reference to quantum neural networks, if designed from the theoretical level of neural networks, it is likely that novel subdisciplines will be born.

Funding Support

This article is funded by the "Science and Technology Innovation Plan," Chinese Academy of Sciences (20194001882).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[2] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ..., & Sutskever, I. (2021). Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, 8821–8831.

[3] Huang, B., Xu, Y., Hu, X., Liu, Y., Liao, S., Zhang, J., ..., & Liu, H. (2022). A backbone-centred energy function of neural networks for protein design. *Nature*, *602*(7897), 523–528.

[4] Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., ..., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, *327*(5964), 439–442.

[5] Mauttone, A., & Urquhart, M. E. (2009). A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, *36*(8), 2440–2449.

[6] Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. USA: Prentice-Hall.

[7] Alim, K., Andrew, N., Pringle, A., & Brenner, M. P. (2017). Mechanism of signal propagation in Physarum polycephalum. In *Proceedings of the National Academy of Sciences, 114*(20), 5136.

[8] Burchett, J. N., Elek, O., Tejos, N., Prochaska, J. X., Tripp, T. M., Bordoloi, R., & Forbes, A. G. (2020). Revealing the dark threads of the cosmic web. *The Astrophysical Journal Letters*, *891*(2), L35.

[9] Nakagaki, T., Yamada, H. & Hara, M. (2004). Smart network solutions in an amoeboid organism. *Biophysical Chemistry*, *107*(1), 1–5.

[10] Shirakawa, T., Gunji, Y. P., Sato, H., & Tsubakino, H. (2020). Diversity in the chemotactic behavior of physarum plasmodium induced by Bi-modal Stimuli. *International Journal of Unconventional Computing*, *15*(4), 275–285.

[11] Takamatsu, A., Takaba, E. & Takizawa, G. (2009). Environment-dependent morphology in plasmodium of true slime mold Physarum polycephalum and a network growth model. *Journal of Theoretical Biology*, *256*(1), 29–44.

[12] Adamatzky, A. (2022). Language of fungi derived from their electrical spiking activity. *Royal Society Open Science*, *9*(4), 211926.

[13] Pershin, Y. V., La Fontaine, S., & Di Ventra, M. (2009). Memristive model of amoeba learning. *Physical Review E*, *80*(2), 021926.

[14] Whiting, J. G., Jones, J., Bull, L., Levin, M., & Adamatzky, A. (2016). Towards a Physarum learning chip. *Scientific Reports*, *6*(1), 1–14.

[15] Kay, R., Mattacchione, A., Katrycz, C., & Hatton, B. D. (2022). Stepwise slime mould growth as a template for urban design. *Scientific Reports*, *12*(1), 1322.

[16] Zhang, X., Adamatzky, A., Chan, F. T., Deng, Y., Yang, H., Yang, X. S., ..., & Mahadevan, S. (2015). A biologically inspired network design model. *Scientific Reports*, *5*(1), 1–14.

[17] Hanahan, D., & Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, *100*(1), 57–70.

[18] Chen, Z., Kibler, R. D., Hunt, A., Busch, F., Pearl, J., Jia, M., ..., & Baker, D. (2020). De novo design of protein logic gates. *Science*, *368*(6486), 78–84.

[19] Cramer, P. (2021). AlphaFold2 and the future of structural biology. *Nature Structural & Molecular Biology*, *28*(9), 704–705.

[20] Chowdhury, R., Bouatta, N., Biswas, S., Floristean, C., Kharkar, A., Roy, K., ..., & AlQuraishi, M. (2022). Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, *40*(11), 1617–1623.

[21] Chen, Z., Linton, J. M., Zhu, R., & Elowitz, M. (2022). A synthetic protein-level neural network in mammalian cells. *bioRxiv Preprint: 2022.07.10.499405*,

[22] Lin, X. H., Wang, X. W., Zhang, N., & Ma, H. F. (2015). Supervised learning algorithms for spiking neural networks: A review. *Acta Electonica Sinica*, *43*(3), 577.

[23] Cao, Y., Chen, Y., & Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, *113*(1), 54–66.

[24] Chen, J., Chen, Z., Li, C., Yang, K., Li, X., Jiang, J., ..., & Li, Y. (2023). Preprocessing and pattern recognition for Single-Lead cardiac dynamic model. *Biomedical Signal Processing and Control*, *82*, 104544.

[25] Boujnouni, I. E., Zili, H., Tali, A., Tali, T., & Laaziz, Y. (2022). A wavelet-based capsule neural network for ECG biometric identification. *Biomedical Signal Processing and Control*, *76*, 103692.

[26] Mazzia, V., Salvetti, F., & Chiaberge, M. (2021). Efficient-capsnet: Capsule network with self-attention routing. *Scientific Reports*, *11*(1), 14634.

[27] Baggio, G., Bassett, D. S., & Pasqualetti, F. (2021). Data-driven control of complex networks. *Nature Communications*, *12*(1), 1429.

How to Cite: Deng, Y., Li, Z., & Chen, J. (2023). Interdisciplinary Trends in the Reintegration of Organisms with Perceptron Units. *Journal of Data Science and Intelligent Systems* https://doi.org/10.47852/bonviewJDSIS3202934