A Neural Network Framework Based on Symmetric Differential Equations

Authors: KunJiang, KunJiang

Date: 2024-10-09T00:00:00+00:00

Abstract

Modern mathematical neural networks are derived from biological neural networks, yet the currently popular general large models do not incorporate biological neural networks. The primary reason for this is that the differential equations based on biological neural networks are difficult to manipulate. At present, mathematical neural networks are characterized by their capacity for large-scale deployment, while biological neural networks offer strong biological interpretability. This paper introduces a system of differential equations with perfect symmetry and convenient manipulability, enabling us to manipulate this system as easily as we manipulate numbers in a matrix, thus integrating the advantages of both. As we are introducing a brand-new neural network framework, we first explore the mathematical properties of the differential equations, then define a new signal propagation method, and finally propose a new training approach for the neural network. The training of this new neural network does not rely on the traditional back-propagation algorithm; instead, it depends solely on the propagation of local signals. This implies that we no longer require global information to train the network. Each neuron can adjust based on the signals it receives and its predetermined strategy. As a verification, we mimicked the linking method of a multilayer perceptron (MLP) to create a new neural network and trained it on the MNIST dataset, demonstrating the effectiveness of our methodology.

Full Text

Preamble

A Neural Network Framework Based on Symmetric Differential Equations
Kun Jiang

Abstract

Modern mathematical neural networks are derived from biological neural networks, yet the currently popular general large models do not incorporate biological neural networks. The primary reason for this is that the differential equations based on biological neural networks are difficult to manipulate. At present, mathematical neural networks are characterized by their capacity for large-scale deployment, while biological neural networks offer strong biological interpretability. This paper introduces a system of differential equations with perfect symmetry and convenient manipulability, enabling us to manipulate this system as easily as we manipulate numbers in a matrix, thus integrating the advantages of both. As we are introducing a brand-new neural network framework, we first explore the mathematical properties of the differential equations, then define a new signal propagation method, and finally propose a new training approach for the neural network. The training of this new neural network does not rely on the traditional back-propagation algorithm; instead, it depends solely on the propagation of local signals. This implies that we no longer require global information to train the network. Each neuron can adjust based on the signals it receives and its predetermined strategy. As a verification, we mimicked the linking method of a multilayer perceptron (MLP) to create a new neural network and trained it on the MNIST dataset, demonstrating the effectiveness of our methodology.

Keywords: Symmetric differential equations, Fixed point, Multilayer perceptron, Neural network, Backward propagation

1 Introduction

In 1952, Alan Hodgkin and Andrew Huxley introduced a model based on differential equations (Equation (1.1)) derived from experiments, aimed at elucidating the intricate mechanisms of ion concentration, membrane potential, and conduction current in nerve cells [?]. This pioneering work advanced our understanding of nerve cell function, promoted the proposal of a variety of other models, and earned them the Nobel Prize in 1963.

In Equation (1.1), the first term represents the conduction current, which is influenced by the membrane potential and two ionic currents (including potassium (K) and sodium (Na)), as well as leak current. The subsequent three equations illustrate how membrane potential affects ionic currents and leak current. Notably, the equation governing ionic currents and leak current exhibits good symmetry, while the conduction current equation lacks this property, posing challenges for obtaining desirable mathematical characteristics. Subsequent equations describing nerve cells also have similar problems. This is because they all describe the problem from an experimental perspective, which inevitably leads to some details that cause the equation to lose its symmetry. When the equation lost its symmetry and could not obtain perfect mathematical properties, it became very difficult to deal with such differential equations. Therefore, it was not until the late 20th century that biological neural networks such as cellular neural networks and chaotic neural networks began to develop [?, ?]. During this difficult period, researchers began to simplify the model from a mathematical perspective and established neural networks such as multilayer perceptron (MLP) and Hopfield through various means [?, ?]. These mathematically based models have become the basis of neural networks used on a large scale today. The frameworks such as RNN, CNN and TRANSFORMER that followed have profoundly changed the landscape of neural networks [?]. In this context, a question has become increasingly prominent: humans and other organisms are clearly composed of a series of cells, so why do we need to turn nerve cells into numbers for calculation?

When we go back to the starting point, the answer is that differential equations cannot be manipulated as conveniently as numbers, and we lack a differential system with perfect mathematical properties to describe biological nerve cells. With this insight, we started from symmetry logic and established a completely symmetrical set of differential equations. In terms of logic, we chose the Five Elements (Wuxing) Theory, which has been circulated for more than 2,000 years. In terms of differential equations, we chose the widely recognized predator equation. While the Wuxing Theory embodies complete symmetry, it lacks corresponding mathematical equations; conversely, although the predator equation is widely applied, it does not exhibit symmetry [?]. By integrating these two concepts, we formulated a new set of differential equations.

With the establishment of the new differential equations, we proceeded to investigate its fixed point. In chaos theory, the fixed point is a fundamental mathematical property of the system. However, in many systems, this property is difficult to grasp. Sometimes we don't even know whether the system has a fixed point, let alone control such a property. The differential equation established in this paper can easily specify the fixed point of the system due to its good symmetry. Fixed point theory is also used in early mathematical neural networks, such as Hopfield neural networks. People get the fixed point of the system through iterative calculations for information storage. That method is very easy to implement in this paper, and we can easily specify any fixed point through calculation.

In this paper, using fixed points to store information is not a particularly big breakthrough. The major breakthrough of this paper is that a new signal propagation method is established, which is related to perturbation theory. On this basis, a new training method is established, which has been shown to be effective. Since the system has complete symmetry, the propagation of signals is completely reversible. Consequently, the new training method developed does not rely on traditional back propagation, but only on local signal propagation.

Among the many training methods of neural networks, back propagation plays the most critical role and has achieved the most and greatest success. However, its biological rationality has always been doubted by many researchers. Although people have made a lot of efforts to explain the rationality of back propagation, this research has not yet reached a consensus [?]. The new training method proposed in this paper is a training method based on local signal propagation. Unlike the traditional back propagation algorithm, it does not require global information, but only needs to focus on its own signal propagation, and then adjust according to the predetermined strategy to achieve the training goal. This method provides a new set of research ideas for the reasonable explanation of the learning process of biological neural systems.

As a verification, we constructed a multilayer perceptron (MLP) with 784 inputs and 10 outputs based on the MNIST dataset, and the results show that the method is effective. Although we only describe some positive results (not the best) in this paper, it is mainly because the adjustment parameters are limited, so the system generalization ability is insufficient. A few months ago, we had achieved 60% accuracy on the same model, which is about the level of neural networks in the early 21st century.

We did not try specific structures such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs); rather, we opted to simulate a multilayer perceptron (MLP) in its simplest form to demonstrate its potential. As a new neural network framework, our work has surpassed the development of the past few decades, and we think this should be a significant development. The high scalability and biological interpretability of the current system further support our conviction that this new framework holds great promise. Considering the huge computing power improvement brought by CUDA to neural networks, we also implemented parallel computing, which is obvious because biological neural networks are inherently parallel.

Figure 1 [Figure 1: see original paper] shows the basic framework of this article. Figure 1.a is a very simple MLP, on which we will replace the digital neurons with a system of differential equations. Figure 1.b shows the traditional Wuxing logic, which is a closed system. Such logical relationships have been circulated in China for more than 2,000 years and have profoundly changed Chinese culture and philosophy. In this logical system, there are five different elements, which generate or inhibit each other, forming a closed logical system. This system has perfect symmetry, so many people have been fascinated by it since ancient times and used it to explain the laws of the world. In this article, this makes some sense, but their mistake is that they did not form Wuxing neurons into a larger system to simulate the world.

Figure 1.c. We adapted the predator-prey model to formalize Wuxing logic and introduced a self-attenuation term changing the system into an open neuron. Accordingly, each element is considered as a neural interface that functions as both input and output, with this process being reversible but disallowing simultaneous input and output.

Figure 1.d. The connectivity of neural elements. We employ the hierarchical structure of a multilayer perceptron, which may involve random connections. However, since each neural element has only five interfaces, not all interfaces

are guaranteed to connect with others, potentially leaving some interfaces unconnected.

Since the traditional Wuxing system is closed, modifications are necessary. We refer to the predator equation to create an open and natural system, as illustrated in Figure 1.c. In this system composed of differential equations, the propagation of signals is completely reversible. Each element can function as an input node or an output node, but it cannot perform both roles simultaneously, which will cause signal interference.

Figure 1.d depicts the interconnection of multiple Wuxing systems. We refer to the structure of MLP in Figure 1.a. The difference is that the connection between neuron nodes is one-to-one. Since each Wuxing neuron has only 5 nodes, some nodes may be unconnected.

The subsequent chapters of this paper are organized as follows: In Chapter 2, we will introduce the logic of the Wuxing Theory and establish the corresponding cause-effect relationship. We will also present the predator-prey model, integrating these concepts to formulate a new set of differential equations. In Chapter 3, we will first generalize this set of differential equations to reveal the universality of the system. Also, we will explore the fixed points of the system and prove that fixed points can be used to store information. Chapter 4 will focus on the establishment of a new signal propagation method, which is related to perturbation theory. Under this theory, the initial value of the elements was set to the fixed point, and all signals are the displacement of elements near the fixed point. In view of this, we established a unified signal propagation path. In Chapter 5, we studied the training method of neural networks and proposed the instinctive design method to completely localize the training of neural networks. For the training of a single neuron, it is no longer necessary to obtain global information, but only to care about its own signal propagation process. Finally, the paper will conclude with a summary and some prospects for future development.

2 Wuxing Logical Relationship and Corresponding Differential Equations

Since ancient times, humans have sought to understand and replicate nature, leading to the development of various theories and sciences. One such theory is the Wuxing Theory. Ancient Chinese thought held that the world was composed of five distinct elements (represented as J, M, S, H, T in Figure 1.b). Through the observation of natural phenomena, each element was associated with specific relationships. For example, Water (S) can extinguish Fire (H), indicating that Water (S) restrains Fire (H). Conversely, Water (S) supports the growth of plants, thus Water (S) generates Wood (M). Based on these observations, numerous entities were classified into one of the five elements, and their interactions could be inferred from these classifications.

However, the Wuxing logical relationship is merely a formal representation. To convert this relationship into a mathematical equation, we draw on the predator-prey model, which describes the dynamic interactions between predators and prey.

$$ax - bxy \tag{1}$$

Equation (2.1) represents the predator-prey model used in this paper [?]. In this model, x denotes prey, such as rabbits, while y represents predators, such as wolves. The prey population is influenced by two main factors: its own reproduction rate and the predation by wolves. The predator population is affected by two factors: the availability of prey (rabbits) and natural mortality. However, this model is not complete or symmetrical, and it appears somewhat unnatural. For example, rabbits do not die of natural causes, and wolves do not increase in number through reproduction.

In contrast, the Wuxing Logic (Figure 1.b) maintains complete symmetry among its elements. Therefore, to align with the Wuxing logical structure, the system must be modified to be more natural. For instance, the rabbit population should be influenced by three factors: 1) reproduction limited by natural resource availability, 2) natural mortality of rabbits, and 3) the impact of external factors (such as predation by wolves). The first factor is constrained by external conditions, the second by intrinsic factors, and the third by interactions with other elements.

Incorporating the logical relationships from Figure 1.b, the mathematical model of the Wuxing theory can be expressed as Equation (2.2), and its logical structure is illustrated in Figure 1.c.

$$\langle MATH_2.2 \rangle \tag{2}$$

In biological neural networks, most models establish a set of differential equations to describe the propagation of electrical signals in neurons, with the Hodgkin–Huxley model being one of the earliest examples [?]. These equations stem from experimental data, resulting in numerous fitting processes and a lack of symmetry. In contrast, the differential equations proposed in this paper are grounded in logical structures, exhibiting complete symmetry and possessing excellent mathematical properties, including specified fixed points and reversible propagation. These key features provide the system with robust expansion and training capabilities.

3 Mathematical Structure and Properties of Differential Equations

This section primarily explores the mathematical characteristics of the system to facilitate the subsequent construction of neural networks.

3.1 General Mathematical Expressions of Differential Equations

We will start with a general description of Equation (2.2). We use $\langle MATH_3.1_a \rangle$ to denote the five different elements and $\langle MATH_3.1_b \rangle$ to represent various parameters. Equation (2.2) can be rewritten as Equation (3.1):

$$\langle MATH_3.1 \rangle \tag{3}$$

Among which, $\langle MATH_3.1_c \rangle$ at different positions are not identical. In Equation (3.1), the elements represented by $\langle MATH_3.1_d \rangle$ are different, and the initial value of $\langle MATH_3.1_e \rangle$ is uncertain, making identification of another fixed point quite challenging. Therefore, we define the order of elements in $\langle MATH_3.1_f \rangle$ as {J, S, M, H, T}, with different offset numbers used to indicate different elements (see Equation 3.2). Consequently, Equation (2.2) can be conveniently described in terms of Equation (3.2) by specifying the number of elements in the equation beforehand.

$$\langle MATH_3.2 \rangle \tag{4}$$

Equation (3.2) is a general form of equation that describes a class of equations with two rings, one of which is a generating ring and the other is an inhibiting ring. The parameter of the generating ring is $\langle MATH_3.2_a \rangle$ and the parameter of the inhibiting ring is $\langle MATH_3.2_b \rangle$. In this paper, the number of elements is 5, and as the research progresses, the number of elements can also be other numbers. Specifically, when the number of elements is one, and if $\langle MATH_3.2_c \rangle$ is appropriately chosen, we obtain the Logistics Equation (3.3). In the same way, if we choose appropriate parameters, we can also get an equation similar to the one in Equation (1.1) that describes the particle concentration. These equations are widely used. For example, the Logistics Equation can describe the natural growth of organisms under certain conditions.

$$\langle MATH_3.3 \rangle \tag{5}$$

3.2 Fixed Points of Differential Equations

Obviously, zero is a fixed point of the equation, but this point is not stable. If the system is slightly perturbed, the system will move to another fixed point. However, if the parameters $\langle MATH_3.4_a \rangle$ are different and the initial value of $\langle MATH_3.4_b \rangle$ is uncertain, identifying another fixed point of the equation becomes quite challenging. We assume $\langle MATH_3.4_c \rangle$ is greater than 0 but not excessively large to avoid system instability. Under this assumption, we further consider that all parameters within $\langle MATH_3.4_d \rangle$ are respectively equal. At this point, we can analytically calculate the fixed point $\langle MATH_3.4_e \rangle$ (as shown in equation (3.4)) for equation (3.2).

$$\langle MATH_3.4\rangle \quad (6)$$

Although this formula is derived under the ideal condition where each set of parameters is equal, it remains the most significant formula in this article. It illustrates how different parameters affect the fixed point, even when they are not equal. In many cases, calculating the fixed point for varying parameters is complex, but we can still adjust the parameters in $\langle MATH_3.4_f\rangle$ according to Equation (3.4) to achieve the target fixed point.

The fixed point is a crucial parameter in this system because signal propagation relies entirely on fixed points in Chapter 4. If the fixed point of the system cannot be determined, the initial value of the system will remain indeterminate. Consequently, the signals generated by the system in response to external stimuli also cannot be defined, leading to a complete failure in the overall signal propagation of the system. Properly setting and utilizing the fixed point is a key skill for adjusting the system.

For instance, the fixed point can be used to store specific values by comparing the target with the fixed point to obtain the error signal. Consider a model with parameters $\langle MATH_3.5_a\rangle$ = {1, 1, 1, 1, 1}, $\langle MATH_3.5_b\rangle$ = {0.5, 0.5, 0.5, 0.5, 0.5}, and an initial value of $\langle MATH_3.5_c\rangle$ set to 1. The following is a simple example of adjusting the fixed point: According to Equation (3.4), the system initial value stays at its fixed point. To move the target fixed point to {0.8, 0.6, 1.5, 1.0, 0.9}, we can use the following method to obtain error information and adjust $\langle MATH_3.5_d\rangle$ = {0.5, 0.5, 0.5, 0.5, 0.5} and $\langle MATH_3.5_e\rangle$ accordingly.

1. Set the value of $\langle MATH_3.5_f\rangle$ to the fixed point to be stored $\langle MATH_3.5_g\rangle$ = {0.5, 1.5, 1.3, 0.6, $\langle MATH_3.5_h\rangle$}. Iterate a small time step according to the equation to get the change $\langle MATH_3.5_i\rangle$ (Equation (3.5)).

$$\langle MATH_3.5\rangle \quad (7)$$

2. According to $\langle MATH_3.6_a\rangle$ combined with (3.4), the corresponding $\langle MATH_3.6_b\rangle$ can be adjusted (equation (3.6), among which $\langle MATH_3.6_c\rangle$ is small positive numbers).

$$\langle MATH_3.6\rangle \quad (8)$$

3. Repeat steps 1-3 until $\langle MATH_3.7_a\rangle$ is small enough, at which point it can be regarded that the target fixed point has been reached.

Figure 2 [Figure 2: see original paper] illustrates the training process for the fixed point. In Figure 2.a, the initial value of the element is simply set to the target value, and no further action is taken. This results in the system returning to its original fixed point, an inherent characteristic of the system. Figures 2.b and 2.c display the training curves achieved by adjusting $\langle MATH_3.7_b\rangle$ and $\langle MATH_3.7_c\rangle$ respectively. The initial value of the element is reset to the target value every one second. Using Equation (3.5), we calculate the error signal, followed by parameter adjustments according to Equation (3.6). After approximately 20 adjustments, the system gradually stabilizes at the new fixed point. A similar outcome occurs when adjusting $\langle MATH_3.7_d \rangle$, though the training speeds differ slightly. Figure 2.d verifies that the trained system successfully reaches the new fixed point; when set to the original fixed point, the system stabilizes at the target fixed point.

This is a very simple fixed point adjustment method. However, sometimes we don't know what the fixed point should be set to. In this case, we need some other strategies to adjust the parameters, which we will discuss in the following chapters.

3.3 Some Additional Explanations

The main goal of this paper is to implement a neural network similar to a multilayer perceptron (MLP). While MLPs have significant potential and can be used to develop other powerful neural network structures such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), we will not explore these extensions here to maintain a clear focus. Numerous studies have been conducted on neural network architectures, and many problem-solving methods are available for reference. We will not delve into these unless specifically required. Additionally, there are some minor adjustments in our approach. For instance, we use the improved Euler method instead of the Runge-Kutta method for calculations. This choice is due to the improved Euler method's lower computational cost and acceptable error margin. In summary, this work aims to establish a new methodological framework, with refinements to be addressed in future work. The following measures were implemented, along with the rationale for each:

1. All parameters are constrained to positive numbers within a specific range (to ensure stability).
2. All instantaneous element values are restricted in a specific range (to ensure stability).
3. The improved Euler method is employed in differential equation calculations due to its lower computational demand and acceptable error (for simplification).
4. Connections between neurons are assumed to be uniformly random (based on experience).
5. All neurons are set at their fixed points before receiving signals (to ensure stability).
6. Parallel programming and asynchronous updates are used to accelerate calculations, though results may vary slightly with different parallel parameters (to improve efficiency).

4 Signal Propagation and Network Structure

The previous chapters mainly discussed some characteristics of the system itself and did not address how the system interacts with external signals. In this section, we will define the related issues of signal propagation and network connection.

4.1 Signal Definition

First, we define an input signal $\langle MATH_4.1_a \rangle$, which corresponds to the number of elements. In this article, we consider the case where the number of elements is 5, meaning the dimension of the input signal for a single neuron is five. The previous Equation (3.2) becomes:

$$\langle MATH_4.1 \rangle \tag{9}$$

At this point, the system will deviate from its original fixed point. Assume that the initial values of all elements in the system are at their fixed points; we define the perturbation between $\langle MATH_4.2_a \rangle$ generated by the system and the fixed point as the signal $\langle MATH_4.2_b \rangle$.

$$\langle MATH_4.2 \rangle \tag{10}$$

Here, $\langle MATH_4.2_c \rangle$ is still controlled by Equation (3.4). This signal does not match Equation (3.4) because the parameters are not unique, but the value of $\langle MATH_4.2_d \rangle$ can be further propagated to other neurons as input (see Figure 3 [Figure 3: see original paper].a). Thus, we observe a pattern of signal propagation: when one neuron initially at its fixed point receives an input signal, $\langle MATH_4.2_e \rangle$ will deviate from the original fixed point, generating a new signal. This signal is then propagated through the network of neuron connections and eventually reaches the output.

4.2 Network Structure

Here, we will imitate the multi-layer network structure of an MLP to build a similar network. Figure 3.a illustrates a network with three signal inputs and two signal outputs, organized into four layers with uniformly random connections between each layer.

Figure 3 [Figure 3: see original paper] shows the forward and backward propagation of Wuxing neural networks. Figure 3.a depicts the forward propagation path of the Wuxing neural network and corresponding equations. This network adopts a randomly connected multi-layer neural network structure. Unlike the traditional MLP, each neuron has only 5 interfaces, and the connections between the interfaces are one-to-one. Figure 3.b shows the backward propagation path of the Wuxing neural network and corresponding equations. Compared with the forward network structure, the connection method has not changed; the only change is the direction of signal propagation. The reverse propagation equation is changed according to the connection method.

Additionally, based on the reversibility of propagation, we can derive the corresponding backward propagation network (as shown in Figure 3.b). In Figure 3.b, not only does the direction of signal propagation between neurons change, but the direction of signal propagation within neurons also reverses. According to equation (4.1), we obtain a new propagation equation (4.3).

$$\langle MATH_4.3\rangle \tag{11}$$

In Equation (4.3), both the order of $\langle MATH_4.3_a\rangle$ and the order of $\langle MATH_4.3_b\rangle$ are altered. This is because $\langle MATH_4.3_c\rangle$ describes the relationship between different elements. When the direction of signal propagation changes, $\langle MATH_4.3_d\rangle$ must continue to represent the relationship between the original two elements, necessitating a change in the order of $\langle MATH_4.3_e\rangle$ in Equation (4.3).

Biological neural networks have long had an advantage: they can be easily deployed in circuit systems. In practical applications, when trying to deploy the neural network structure, and if someone doubts whether the system is reversible, we can consider the following solution. In Figure 3, there are two systems. If we build two systems at the same time that are mirror images of each other, then we only need to synchronize their parameters to achieve the function of a reversible system. At the same time, the establishment of two systems also allows forward propagation and reverse propagation to be carried out at the same time, which improves the efficiency of the system.

5 Wuxing Neural Network Training

In this section, we will discuss the training methods of neural networks. Based on the characteristics of biological neural networks, we propose an instinctive design method so that the training of neural networks no longer depends on global information, but instead relies on its own signal propagation in a decentralized manner.

5.1 Training Theory

Neural network training has long been a highly complex problem, prompting the development of numerous training methods. Among these, the backpropagation algorithm is the most efficient and widely used [**?**]. This algorithm continuously optimizes connection parameters by computing the partial derivatives between inputs and outputs. However, the biological plausibility of backpropagation has been questioned, with some researchers doubting that organisms could perform such intricate continuous derivative calculations.

In neural networks, causality and nonlinearity are fundamental characteristics. Neural networks function as mappings from inputs to outputs, with causality ensuring the determinism of this mapping and nonlinearity providing its diversity. Thus, to develop an effective neural network, it is crucial to satisfy both of these characteristics. Additionally, to enhance the operability and versatility of neural networks, they should be designed to be as elegant and symmetrical as possible. In mathematically-based neural networks, neurons are represented by numbers, which allows for convenient and rapid manipulation, though at the cost of extensive calculations. Conversely, biological neural networks do not possess advantageous mathematical properties, but they can be deployed in many circuits with high biological interpretability.

As previously mentioned, the essence of neural networks lies in nonlinearity and causality. Therefore, in theory, a neural network can be formed as long as these two principles are maintained. However, this requirement does not specify how to train the neural network. To address this, we propose adding a reversibility concept to the training process. In simple terms, the input signal at the front end passes through a nonlinear system to produce an output signal at the back end. This output signal is then compared with the target to generate a feedback signal (also known as an error signal). Because the system is causally reversible (as illustrated in Figure 3), the feedback signal is used as a new input to the back end and eventually returns to the front end. The propagation mode of these two signals is essentially the same, with the only difference being their directions of propagation.

By comparing the signals traveling in both directions through the same neuron, we can determine the corresponding adjustment methods. This localized method for training is also called instinctive design. Through the instinctive design method, all neurons no longer need to obtain global information, but only need to pay attention to the information flow in two directions flowing through themselves, and then they can adjust according to the predetermined strategy. We can insert any number and type of neurons at any position in the neural network, so that when neurons are expanded on a large scale, there is no need to notify all other neurons.

We do not believe that individual will can precisely control each neuron. Instead, a neuron's ability to learn or forget information is more likely due to inherent system characteristics or preset strategies. Personal will is better suited to influencing broader aspects, such as the overall learning rate.

5.2 Training Methodology

We have provided an overall description above. Now, we will delineate specific issues and train this network using the MNIST dataset. MNIST is a well-known dataset for digit recognition; thus, a network model was constructed with 784 input nodes and 10 output nodes, consistent with the dataset's structure. At the output layer, we integrated the output signal over time with 10 dimensions according to Equation (5.1). The largest component of $\langle MATH_5.1_a \rangle$ was selected as the final output result. If this output corresponds to the training label, the training is deemed successful; otherwise, it is classified as a failure.

$$\langle MATH_5.1 \rangle \tag{12}$$

In cases of successful training, parameter adjustments are unnecessary. However, if the results do not align with the expected outcome, an error signal must be generated. Assuming that the $\langle MATH_5.2_a \rangle$ component should be the largest, the error signal for this component can be calculated using Equation (5.2). The error signals for the other output components are derived from Equation (5.3). In these equations, $\langle MATH_5.3_a \rangle$ and $\langle MATH_5.3_b \rangle$ represent two predefined target values, where $\langle MATH_5.3_c \rangle$ is the larger value and $\langle MATH_5.3_d \rangle$ is the smaller one. This method will make the value of the $\langle MATH_5.3_e \rangle$ component larger after training, while the others will be smaller, allowing the system to achieve a higher accuracy rate.

For the $\langle MATH_5.4_a \rangle$ component, the adjustment error is:

$$\langle MATH_5.2 \rangle \tag{13}$$

For other components, the adjustment error is:

$$\langle MATH_5.3 \rangle \tag{14}$$

Similarly, the error signal propagates in the reverse network and also generates a new propagation signal $\langle MATH_5.5_a \rangle$, which is determined by Equation (5.4).

$$\langle MATH_5.4 \rangle \tag{15}$$

where $\langle MATH_5.5_b \rangle$ is the element value in the backpropagation and $\langle MATH_5.5_c \rangle$ is the fixed point determined by the backward propagation Equation (4.3). By comparing the different signals received by the same element during forward and backward propagation, we can derive the appropriate adjustment method.

In Equation 4.1, there are three sets of parameters: $\langle MATH_5.6_a \rangle$. This article will focus on correcting the model by adjusting $\langle MATH_5.6_b \rangle$. We first define a comparison function, where its magnitude and sign reflect the correlation between the forward signal and the backward signal.

$$\langle MATH_5.5 \rangle \tag{16}$$

Of course, the size of $\langle MATH_5.7_a \rangle$ may exceed a certain limit, so we use the inverse tangent function to process the result as follows:

$$\langle MATH_5.6\rangle \quad (17)$$

In (5.6), $\langle MATH_5.8_a\rangle$ is the adjustment parameter, $\langle MATH_5.8_b\rangle$ is the control value after adjustment, $\langle MATH_5.8_c\rangle$ can be adjusted according to $\langle MATH_5.8_d\rangle$.

$$\langle MATH_5.7\rangle \quad (18)$$

Figure 4 [Figure 4: see original paper] shows the accuracy curve on the training set. Only $\langle MATH_5.9_a\rangle$ was adjusted, and the accuracy curve on the training set after 10 training sessions is displayed. After 10 training sessions, we achieved an accuracy rate of 25.7% on the training set and 25.9% on the test set. The following Figure 4 shows the curve of the accuracy rate during the 10 training sessions. This result is not particularly ideal, but it shows that our training method is effective. There are two main reasons for this result. First, we only adjusted one parameter $\langle MATH_5.9_b\rangle$, which resulted in insufficient generalization ability of the system. Second, in order to ensure stability, we adopted a more conservative parameter range, which resulted in limited adjustment ability of the system.

In Equation (5.5), $\langle MATH_5.10_a\rangle$ and $\langle MATH_5.10_b\rangle$ are local signals, not global ones, which means that a single neuron does not need to know the global information, but can adjust its own parameters according to the predetermined plan based on the methods in Equation (5.5) and later.

People have been looking for a training method with good biological interpretability for a long time. Although the traditional back-propagation algorithm has achieved the most and greatest achievements, its biological interpretability has always been questioned [**?**]. The method proposed in this paper only relies on the propagation of local signals, and only signals are propagated in neurons. The only difference is that the propagation directions of forward signals and backward signals are different, which is natural. In addition, this paper does not make special settings for the signals, so we believe that the method proposed in this paper has good biological interpretability.

Conclusion

1. This paper presents a novel neural network designed to bridge the significant gap between mathematical and biological neural networks. The model incorporates the advantages of both: large-scale parallel applications and strong logical interpretability. The main contributions of this paper are as follows: (1) Designing a neuron structure based on symmetric differential equations to replace traditional MLP neurons; (2) Defining the propagation and connection of signals; (3) Developing a new training method instead of using traditional back-propagation algorithm. Ulti-

mately, we trained the model on the MNIST dataset and achieved promising results. The three breakthroughs mentioned above will bring many new research perspectives to the development of neural networks.

2. The neural network proposed in this paper features a completely new structure, with training based on a novel set of theories. To clarify this topic, we will systematically address how to interpret and train such a neural network in subsequent articles. In this study, we focused solely on adjusting the parameters. To ensure system stability, we employed a relatively conservative range for these parameters, which is why the reported accuracy is not particularly high. Although we achieved an accuracy of 60% a few months ago and parameter adjustments can enhance accuracy, this is not the primary focus of this article.

3. This paper introduces a new training method primarily based on the concept of instinctive design, aiming to minimize special intervention measures. However, in some practical applications, specific interventions and structure might prove to be more effective. At the same time, we also show many potential uses, including expanding the range of the number of elements, using more complex connection mechanisms, and forming more complex systems. We did not try these methods in this paper, mainly to show the real core value of this work, rather than a complex system made up of various techniques. Our work is a tracing back to the development of neural networks in the past hundred years, and also a Chinese romantic encounter across 2000 years.

Acknowledgements

Thanks to China Scholarship Council (CSC) for their support during the pandemic, which allowed me to get through those difficult days and gave me the opportunity to put my past ideas into practice, ultimately resulting in the article I am sharing with you today.

References

Hodgkin, A.L. and A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology, 1952. 117(4): p. 500.

Chua, L.O. and L. Yang, Cellular neural networks: Theory. IEEE Transactions on circuits and systems, 1988. 35(10): p. 1257-1272.

Aihara, K., T. Takabe, and M. Toyoda, Chaotic neural networks. Physics letters A, 1990. 144(6-7): p. 333-340.

Hopfield, J.J., Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 1982. 79(8): p. 2554-2558.

Hornik, K., M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators. Neural networks, 1989. 2(5): p. 359-366.

He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Vaswani, A., Attention is all you need. Advances in Neural Information Processing Systems, 2017.

Radford, A., Improving language understanding by generative pre-training. 2018.

Volterra, V., Variations and fluctuations of the number of individuals in animal species living together. ICES Journal of Marine Science, 1928. 3(1): p. 3-51.

Hinton, G., The forward-forward algorithm: Some preliminary investigations. arXiv preprint arXiv:2212.13345, 2022.

Rumelhart, D.E., et al., Backpropagation: The basic theory, in Backpropagation. 2013, Psychology Press. p. 1-34.

Note: Figure translations are in progress. See original paper for figures.