Intelligent Control and Automation, 2026, 17(1), 19-58
https://www.scirp.org/journal/ica
ISSN Online: 2153-0661
ISSN Print: 2153-0653

Theory of Structural Learning

Dong H. Liu

The Department of Artificial Intelligence, College of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China
Email: m13926032320@163.com

How to cite this paper: Liu, D.H. (2026) Theory of Structural Learning. *Intelligent Control and Automation*, **17**, 19-58.
https://doi.org/10.4236/ica.2026.171002

Received: September 23, 2025
Accepted: December 27, 2025
Published: December 30, 2025

Abstract

This is the first paper to be written on the theory of structural learning. The first section outlines the overall concept; the second section proposes the logical universe as the environment for structural learning; the third, fourth and fifth sections detail the theories of the Central Diagram Processor, Software Diagram and Diagram Thinking, with the Central Diagram Processor acting as the machine's brain, the Software Diagram as its body, and the Diagram Thinking as its life activities; finally, the sixth section explains existing research achievements in machine life and structural learning. The main purpose of this paper is to recommend the idea of Structural Learning for Machine Life to human society.

Keywords

Structural Learning, Central Diagram Processor, Software Diagram, Machine Life

1. Introduction

General Introduction

① The goal of structural learning is to invent a new learning method to replace machine learning, adapting to the research into structural intelligence.

② The developmental tools for structural learning include the following: bionics of human learning; structural diagrams of life-giving structural logical equations; and existing achievements in machine learning.

③ The environment for structural learning is a logical world environment, namely the logical universe.

④ The logical subject of structural learning is called machine life.

⑤ The growth path for structural learning involves bionically following the learning and growth path of humans.

1.1. The Objective of Structural Learning

Everyone knows about machine learning, the definition of which can be described as follows:

$$\text{Algorithm} + \text{Information} \rightarrow \text{Useful Information} \quad (1)$$

We now propose structural learning as a novel replacement for machine learning, creating a form of learning that can better develop intelligent technology. First, we define a formula for structural learning:

$$\begin{aligned}&\text{Diagram Structure of Machine Life} + \text{Information}\\&\rightarrow \text{New Diagram Structure of Machine Life}\\&\rightarrow \text{Transformation of Machine Life}\end{aligned} \quad (2)$$

In other words, machine learning, where humans use algorithms to process information, is updated to structural learning, where a form of machine life uses a diagram structure to process information, transforming the machine learning of Formula (3) into the structural learning of Formula (4).

$$\text{Human} + \text{Algorithm} + \text{Information} = \text{New Information} \quad (3)$$

$$\begin{aligned}&\text{Machine Life} + \text{Diagram Structure} + \text{Information}\\&= \text{New Machine Life Diagram Structure}\end{aligned} \quad (4)$$

1.2. Development Tools for Structural Learning

1.2.1. Bionics of Human Learning

Throughout human history, there have been many instances of significant learning. Humanity learned the properties of fire, and thereafter began eating cooked food, which altered the structure of both human diet and the human body. The harnessing of the steam engine revolutionized the very structure of machinery. The heliocentric ideas of Copernicus and the universal theory of gravitation devised by Newton initially redefined the structure of the universe, which was later altered again by Einstein's theory of relativity. This demonstrates that the purpose of human learning is the creation of new structures, achieved either by adapting in response to environmental changes or by directly modifying one's own structures in order to evolve. Since the objective of human learning is to develop new structures, this bionic principle provides a rationale for moving beyond machine learning. A process that merely transforms information into other information is not true learning, since it fails to alter underlying structures. As we advance the concept of structural learning, we can continuously emulate the learning methods of the human body and society to empower structural learning in machine life and machine societies. Consequently, the bionic emulation of human learning stands as the key developmental approach for structural learning.

1.2.2. Life-Giving Structural Logical Equation Diagram

1) Logical Equation Theory

① Equation Overview

Logical equations are based on physical and chemical (material), and biological equations. These equations or equation sets show the determining effect of certain variables on others. Assigning values to certain variables allow the derivation of other variables. In physical and chemical equations, and in biological equations, these variables describe the structure, life activities, and life cycle of matter or life. When applied to intelligent technology, they describe the structure, life activities, and life cycle of intelligent machines. Logical equations include not only objective equations such as physical, chemical and biological equations, but also subjective equations, such as belief and emotion equations, created by the machine brain. Subjective equations reflect the logic of the human brain and can also form the basis of advanced logical structures and intelligent machine consciousness.

② Equation Definition

All equations within the software and hardware of structural intelligence machines start off as functions. However, when solving for values, there is no distinction between functions and variables. Functions and variables are only divided into input assignment variables and output solution variables.

Furthermore, all equations in the knowledge equation structure diagram of the equation unit generator, as well as the equations in each software diagram, are function programs, which derive a group of assigned output variables from a group of assigned input variables. The equations in the software diagram can feature various required modalities.

③ Equation Data

The value set of equation variables in the life-giving structural logical equation diagram stores class data. Class data is defined as follows:

$$\text{Class Data} = \text{Description Set} \wedge \text{Attribute Set} \wedge \text{Function Set} \qquad (5)$$

The description set contains a description of the situation of the class data. The description set can have a time point in the form of a clock. The attribute set stores the logical equations that determine the attributes, while the function set stores the logical equations that determine the functions. If the attribute set and function set are omitted, leaving only the description set, the class data becomes raw data.

Class data includes a description set, an attribute set and a function set; the difference from traditional raw data lies in its class form. Different forms of information, mathematical operations, and processing of class data can be performed on the records of the description, attribute, and function fields. Class data can have a simplified form.

Taken together, all class data in the value set of a variable form a data class group, referred to as a class group. This group might, for example, consist of the files of every student in a school class, or the registration details of every motor vehicle in the city of Guangzhou. A data class group is a new type of database or data warehouse.

Mutual interference of data class group, or simply "mutual interference", involves the construction of logical equations. Mutual interference research is an advanced stage of diagram analysis and an important part of empowering the equation unit generator to create logical equations. Mutual interference research requires specific research methods. First, the object sets of both parties involved must be combined into a single set, with this whole set comprising the future logical equation. Suitable methods can then be sought based on the laws of this overall set. The social or natural laws of the research object must also be integrated within the mutual interference research process.

④ Related Equations

$$\text{Standard Form of Logical Equation} = \text{Clock Form} \wedge \text{Address Form} \wedge \text{Communication Form} \wedge \text{Modality Form} \wedge \text{Equation} \quad (6)$$

$$\text{Standard Form of Equation: Function Variable} = \text{Formula (Variable 1)} \wedge \text{Formula (Variable 2)} \wedge \text{Formula (Variable 3)} \wedge \cdots \wedge \text{Formula (Variable n)} \quad (7)$$

$$\text{Standard Form of Variable} = \text{Data Class } \{(\text{Description Set}) \wedge (\text{Attribute Set}) \wedge (\text{Function Set})\} \quad (8)$$

Note that data classes have these forms: (Description Set) ∧ (Attribute Set) ∧ (Function Set), (Description Set) ∧ (Attribute Set), (Description Set) ∧ (Function Set), (Description Set).

$$\text{Standard Form of Variable Value Set} = \{\text{Data Class 1 (Description Set)} \wedge (\text{Attribute Set}) \wedge (\text{Function Set}), \text{Data Class 2 (Description Set)} \wedge (\text{Attribute Set}) \wedge (\text{Function Set}), \text{Data Class 3 (Description Set)} \wedge (\text{Attribute Set}) \wedge (\text{Function Set}), \cdots, \text{Data Class n (Description Set)} \wedge (\text{Attribute Set}) \wedge (\text{Function Set})\} \quad (9)$$ [1]

2) The Study of Life-giving Structural Logical Equation Diagrams

① Logical Equation Structure Diagrams

A directed diagram where a node consists of one or a group of logical equations. If the variable(s) of this logical equation or equation set determine the variable(s) of another logical equation or equation set, then this node has a directed path to the other node, and weights can be added to the paths to indicate their corresponding order. Writing out all logical equations and marking all variable relationships with directed paths completes a logical equation structure diagram.

② Life-Giving Logical Equation Structure Diagram

Endowing a structure diagram with life using a clock is the first instance of Living Diagram Theory in human research, and the first pioneering diagram theory tool of the intelligent era. This structure diagram includes scientific logic, life logic, and subjective logic, and represents the main tool for describing intelligent structures and intelligent technology. Intelligent engineering is the process of designing and implementing structure diagrams, while intelligence consists of the life activities and life cycle of the structure diagram. The required research tools, design tools, operating tools, monitoring tools, and modification tools for machine brain structures are all life-giving logical equation structure diagrams. The truth or falsehood of each unit within the logical equation structure diagram is judged using logical mechanics. [2]

③ Structural Types

The structure diagram should establish a diagram directory modeled after the file system in a computer operating system, in order to facilitate diagram activities and modifications. Thus, the structure diagram incorporates two types of connection: transmission paths for variables, and a diagram directory. While any diagram activity and/or modification is being carried out, the diagram directory can be used as the unit for these activities and modifications. After the diagram has been modified, new transmission paths for variables will be created.

④ Substitution Diagram Theory

This type of diagram is called a "mother diagram". Within the mother diagram, all variables, constants, and functions of its equations can represent another "unit diagram". By simulating algebraic theory, the so-called "Substitutional Diagram Theory" can be created. In this way, diagrams can describe various "universe diagrams" and "life diagrams". A single variable can simultaneously represent a unit of spacetime or a type of life organization, greatly enhancing the descriptive capability of the diagrams.

Furthermore, by emulating the algebraic operations of mathematical algebraic theory, we can establish and develop the substitutional diagram operations of Substitutional Diagram Theory, satisfying the various needs of structure diagrams for describing the universe and life. A substitutional diagram variable can be established with the following formula:

$$\begin{aligned}&\text{Substitutional Diagram Variable} = \text{Substitutional Diagram Input Variable}\\&+ \text{Substitutional Diagram} + \text{Substitutional Diagram Output Variable}\end{aligned} \tag{10}$$

This allows the input and output variables of the substitutional diagram variable to establish connections and paths with the various equation variables of the mother diagram.

A crucial function of the Theory of Substitutional Diagrams is its ability to perform activities and modifications at the level of the diagram directory. Following these operations, new equation-based connection paths and entirely new diagrams can be generated accordingly, thereby enhancing structural learning.

⑤ Diagram Task

The task of this type of diagram is to describe both universe and life diagrams. Life diagrams, in turn, include both machine life diagrams and machine society diagrams. The brain, body, and life activities of a machine life can all be described by an associated diagram system. Life activities include developing and changing structure through learning.

The structural diagram of life-giving structural logical equations constitutes the second developmental approach for structural learning. [1]

3) Diagram Data Theory

① Significance of Diagram Data Theory

This theory makes use of class data instead of raw data, using the data distributed in the variable value sets of each equation structure diagram instead of the raw data concentrated in the database. It also uses equation analysis instead of statistical analysis, and features an equation unit generator to generate a unit logical equation structure diagram instead of statistical results generated by the database. The data is attached to the life-giving structural logical equation diagram. The database is a value set of equation variables, while data analysis consists of logical equation analysis—a form of logical mathematical analysis. The aim of this analysis is not to obtain statistical data, but to empower the equation unit generator to generate a unit equation structure diagram, creating the thinking and knowledge of intelligent machines.

For example, traditional data analysis often produces indices, such as economic indices. However, diagram data theory uses a life-giving structural logical equation diagram to perform data analysis. The index calculation formula becomes a structure diagram, turning economic index analysis into economic structure diagram analysis. Complex analysis, clock analysis, and equation unit generator analysis are used to generate a unit equation diagram instead of an index, transforming economic index research and extending it to other fields. Structure diagram analysis is used to achieve comprehensive cross-domain analysis. [3]

② Replacing Database Analysis with Equation Structure Diagram Analysis

Traditional data analysis is the concentrated mathematical analysis of raw data within a database, while the data analysis of diagram data consists of the logical mathematical analysis of class data in the variable value sets of each equation structure diagram. This logical analysis is a form of equation analysis, and includes the following key points:

Structure Diagram Complexity and Analysis: This analysis requires that class data in the variable value set and the equation structure diagram exist in huge amounts;

Clock Equation System and Analysis: This analysis requires that both equations and data can be endowed with life, and that data can be activated or terminated;

The equation unit generator creates new equation units to replace traditional data analysis indices. That is, the data analysis for the diagram data is ultimately completed by the equation unit generator, and the result of this analysis is the new variable value set and new variables generated by the new unit software diagram.

③ Replacing Database Analysis with Equation Structure Diagram Analysis [3]

④ Inheriting Data Training

Diagram data theory has its own complete set of learning methods, but it still needs to inherit data training from information-based intelligent technology. The development of intelligent human technology originated from data training algorithms. It has been proven that data training is still effective for organizing large amounts of data and constructing logical relationships. However, constrained by limited data resources and limited computing power, it is necessary to develop diagram learning combined with the required data training.

1.2.3. Existing Achievements in Machine Learning

After more than half a century of development, machine learning is now a thriving field with a wealth of accomplishments. However, its critical limitation is the inability to evolve new structures. While Structural Learning has been proposed as its successor, abandoning machine learning entirely would be imprudent. For instance, large machine learning models often involve Q&A tasks. Representing this interaction as a logical equation is straightforward. It can be minimally expressed as:

Question (Variable) + Large Model (Parameters) = Answer (Function Value) (11)

This demonstrates that with minor and straightforward modifications to existing machine learning algorithms—specifically, by creating an "interface algorithm"—their output can be converted from raw information into a logical equation. This equation can then be processed by a Machine Lift's native structural learning algorithms. This effectively provides a vast repository of resources for structural learning.

Machine Learning + Interface Algorithm + Structural Learning
= New Structural Learning (12)

The interface algorithm can also be extended to create the following formula:

Input Modality + Modality Algorithm + Machine Learning
+ Interface Algorithm + Structural Learning
+ Modality Algorithm + Output Modality
= Logical Spacetime (13)

In the figure, Logical Spacetime can be further developed into the following:

Input Modality + Modality Algorithm + Structural Learning
+ Modality Algorithm + Output Modality = Logical Spacetime (14)

The existing achievements in machine learning serve as the third development tool for structural learning.

1.3. Structural Learning Environment

This environment is termed the Logical Universe. While this concept may seem novel, the universe as perceived by humanity has always been, in essence, a logical construct. From ancient mythologies to contemporary astrophysics, our descriptions have consistently modeled a logical universe as the true, physical universe that remains forever beyond direct perception. The Logical Universe proposed herein replaces these traditional frameworks—myth and modern science—with Structure Diagrams of Life-giving Logical Equations. These diagrams model cosmic time and space as currently understood, and are designed to evolve in tandem with the advancement of human knowledge, thereby establishing the operational environment for Structural Learning.

1.4. The Logical Subject of Structural Learning

There is currently a wide variety of names for logical subjects. The logical subject of structural learning is more appropriately called "machine life". Machines inherently have no life, but developing intelligent technology requires them to have this property. In order to solve this issue, humans must endow machines with a simulated form of life. A machine must have a machine brain, a machine body, and engage in machine activities. Controllers and time controllers are used to create the machine's simulated life, generating the machine brain's life thinking and the machine body's vital signs and life activities.

All of the above concepts must be realized through the Structural Diagrams of

Life-giving Structural Logical Equations. When this type of diagram is implemented as a central processor, it serves as the diagram description of the machine brain, using "Diagram Thinking" to describe "Life Thinking". By connecting to and controlling the machine's hardware, the diagram can then represent the machine body, while "Diagram Activities" and "Diagram Changes" can be used to describe the body's life activities.

1.5. Growth Path of Structural Learning

The growth path of structural learning, in short, is to establish a starting point machine, known as a "Machine Birth". The "Birth Equation Structure Diagram" of this machine can be developed into a life-giving structural logical equation diagram. By means of structural learning, this diagram can then grow, much like a human, according to the following equation:

$$\text{Original Diagram (Original Machine Life)} + \text{Structural Learning} \rightarrow \text{New Diagram (New Machine Life)} \quad (15)$$

This is analogous to a person's journey from an unenlightened state to maturity through learning.

In addition to the growth of individual Machine Life, a "Machine Society" can also grow through structural learning. This process can emulate the progressive development of human society, as represented by the following equation:

$$\text{Original Diagram (Original Machine Society)} + \text{Structural Learning} \rightarrow \text{New Diagram (New Machine Society)} \quad (16)$$

1.6. Comparative Study of Structure Learning and Machine Learning

Structure learning involves a change in the structure of the structure diagram. This change has a logical similarity to the learning and thinking methods of real life. When humans learn and think, brain cells are active; in reality, the structure of the brain tissue is changing. Equations have a logical similarity to cells; therefore, the change in structure diagram involves the addition and subtraction of equations and their connections, representing the addition and subtraction of the various components that comprise the structure diagram. Human learning and thinking also involve changes in the structure of brain cell organization. Knowledge and thoughts increase or decrease; the production of new knowledge, forgetting, and changes in thought all originate from structural changes in brain cell organization. That is to say, the structure learning of the machine-implemented structure diagram has a logical similarity to life learning; both are a form of structure learning.

Machine learning is an operation on information: collecting, retrieving, and organizing information to obtain new knowledge. The entire learning process is devoid of emotion and thinking; that is, machine learning has no relationship with the program or machine structure. The learning process is merely a change of information, something which is completely different from life learning, since that carries both emotion and thinking. "Learning without thinking" has no life characteristics. Life learning, such as human learning, is "learning with thinking". Human "learning with thinking" is produced by structural changes in brain cell organization. Structure learning allows the structure diagram to undergo changes, with the changes in the structure diagram able to simulate human "learning with thinking". This is something machine learning cannot hope to match.

In addition to its own simulation of life learning, structure learning can inherit the accumulated achievements of machine learning, since structure learning is also produced through practical machine implementation. This characteristic increases the scientific nature and feasibility of structure learning [4]-[8].

2. The Structural Learning Environment—The Logical Universe

General Introduction

① Constructing the logical universe, transforming and developing existing mathematics and physics.

② Key points of the theory of logical spacetime.

③ Key points of the theory of logical mathematics.

④ Key points of the theory of logical physics.

⑤ Using the structural diagrams of life-giving structural logic equations, substitutional diagram theory, and logical spacetime theory to construct a logical time diagram, and using this structural diagram to describe logical time.

⑥ A logical space diagram is constructed using structure diagrams of life-giving structural logical equations, substitutional diagram theory, and logical spacetime theory; this type of structure diagram is used to describe logical space.

⑦ Merging the logical time diagram and the logical space diagram into one diagram system yields the logical spacetime diagram, which can control over time and space at the same time; it can also describe logical spacetime.

⑧ Integrating the logical time diagram and the logical space diagram into one diagram system yields the spacetime diagram. An infinitely large spacetime diagram becomes the logical universe diagram.

⑨ Constructing the logical universe involves constructing the logical universe diagram.

2.1. Origin and Theory

2.1.1. Origin

Establishing common sense is actually a logical subject, while material phenomena are actually logical phenomena. The relationship between all matter is actually a logical relationship, while logical phenomena and logical relationships are also logical subjects. If we develop this concept further, a logical subject can also be used to represent time and space. Time, space, and matter can then be represented by a single concept of a logical subject. This serves to construct the Logical Universe and to reform and develop existing mathematics and physics.

2.1.2. The Following Are Key Points for the Theory of Logical Spacetime

① The common sense for establishing logical spacetime is that time, space, and matter are all logical subjects, material phenomena are logical phenomena, universal connections between matter are logical relationships, and logical phenomena and logical relationships are also logical subjects.

② Logical subjects are described using three sets: a description set, an attribute set, and a function set, all of which can be attributed to a life-giving logical equation structure diagram. As long as the output variable set contains the union of these three sets, then the correct method for identifying a logical subject involves determining whether the subject can be expressed as a life-giving structural logical equation diagram, and whether this diagram's output variable set contains the union of the description set, attribute set, and function set.

③ Three tools are needed to work together when expressing logical subjects using life-giving structural logical equation diagrams, namely bionic logic, life-giving structural logical equation diagrams, and logical mathematics and physics.

Bionic Logic creates the life-giving structural logical equation diagrams using bionics. Bionics refers to finding the structure, activities, and cycle of a logical subject. The life-giving structural logical equation diagram is constructed according to the bionic results of the logical subject. Life here refers to the activity cycle of the logical subject. Parts that cannot be simulated for the time being are designed artificially. The most important work in this process is the equation unit generator working with the clock equation system to create the activities and cycles of the logical subject.

Logical mathematics is the main tool for establishing the equations of the life-giving structural logical equation diagram, while logical physics is the tool for helping bionic logic find the structure of the life-giving structural logical equation diagram.

2.1.3. The Following Are Key Points for the Theory of Logical Mathematics

① The transformation of traditional mathematics by logical mathematics is reflected in these two points: one is to change the study of numbers in traditional mathematics to the study of logical equations in logical mathematics, and the other is to attribute the research results, research methods, and research directions of traditional mathematics to creating binary and multiple relationships in logical equations.

② The first task of logical mathematics is to express the logical equations of all things in the universe derived from bionic logic. The second task involves devising improvements to the equation unit generator of the life-giving structural logical equation diagram, summarizing the known logical paths between input and output into logical equation unit diagrams, and creating logical equation unit diagrams to achieve the logical relationship from input to output. The third task is to incorporate traditional mathematics into the field of logical relationships and to use traditional mathematics to create the binary and multiple relationships of logical equation structure diagrams.

2.1.4. Key Points for the Theory of Logical Physics

① The transformation of traditional physics by logical physics is reflected in: Turning physical definitions and theories into logical subjects, and in drawing life-giving structural logical equation diagrams. In the future, research into logical physics will be expressed using logical subjects and the life-giving structural logical equation diagrams of logical subjects;

② There are three tools required for the description and research of logical physics: Bionic logic, life-giving structural logical equation diagrams, and logical mathematics. Bionic logic imitates the spacetime of the universe and all things in spacetime. The life-giving structural logical equation diagram further describes bionic logic as a structure diagram;

③ Using traditional physics, which has now become a logical subject, and the logical subjects of newly created logical physics to help bionic logic find the structure of the life-giving structural logical equation diagram;

④ Creating the basic concepts of logical physics: Time, space, matter, material phenomena and widespread connection are all logical subjects, while logical physics is used to establish logical spacetime;

⑤ Through the life-giving structural logical equation diagram of logical physics, it will be possible to study the laws of logical spacetime and logical subjects, logical phenomena subjects, as well as logical relationship subjects in logical spacetime, helping the overall study of bionics. The description set, attribute set, and function set of logical subjects can be expressed using the life-giving structural logical equation diagram.

In this way, structural logical equation diagrams can be used to study and build upon traditional mathematical, physical, and logical research, as well as to create and develop logical spacetime, mathematics and physics, and to reject the development path of pseudoscience. The intelligent era will inevitably require logical spacetime, logical mathematics, and logical physics as its fundamental theories [3].

2.2. Logical Time

A logical time diagram can be constructed using life-giving structural logical equation diagrams, substitutional diagram theory, and logical spacetime theory. This type of structure diagram is used to describe logical time, which includes the following key points:

① Using Structure Diagrams to Create Time Bionics

Unlike a conventional time point defined by a small number of data values, logical time can represent either a specific moment or an entire temporal process, described by a highly complex structure diagram. This is achieved by modeling the genesis, development and conclusion of a real-world temporal event as an equation structure diagram.

② The Role of Logical Time

The primary role of logical time involves the temporal control over knowledge and structure. This is a result derived from the bionic emulation of human activities. The universe, all existing entities, humanity, and society are controlled by time, and change their activities accordingly. Logical time often uses different variables within the logical time diagram of the life-giving structural logical equation diagram to control everything.

③ Substitution Diagram Theory of Logical Time Diagrams

For certain time diagrams that exhibit a high degree of complexity, the method of substitutional diagram operations can be applied, as shown in Formula (10) (Substitutional Variable = Substitutional Diagram Input Variable + Substitutional Diagram + Substitutional Diagram Output Variable). This involves establishing connection paths between the input and output variables of the substitutional diagram of the other equations and variables within the mother diagram, thereby generating a new diagram.

2.3. Logical Space

A logical space diagram is constructed using structure diagrams of life-giving structural logical equations, substitutional diagram theory, and logical spacetime theory; this type of structure diagram is used to describe the logical space.

① Using structure diagrams to create space bionics

Space is not merely an environment; it can also perform spatial positioning and environmental control over all entities. In this way, logical space can be simulated using life-giving structural logical equation diagrams. The simulation of the structure diagram must not only implement all logical laws of real space, but also adhere to the following rule:

Real Space + Input Modality Algorithm = Logical Space Structure Diagram (17-1).

Logical Space Structure Diagram + Output Modality Algorithm = Real Space (17-2).

② The Role of Logical Space

Logical space is a key component in simulating the cosmic environment. All entities must achieve positioning and simulated environmental effects within logical space. This is a form of spatial control, realized through the various variables and their values within the space diagram.

③ Substitutional Diagram Theory of Logical Space

For certain space diagrams that exhibit a high degree of complexity, the method of substitutional diagram operations can be applied, as shown in Formula (10) (Substitutional Diagram Input Variable) = Substitutional Input Variable + Substitutional Diagram + Substitutional Diagram Output Variable. This involves establishing connection paths between the input and output variables of the substitutional diagram, and the other equations and variables within the mother diagram, thereby generating a new diagram.

2.4. Logical Spacetime

Merging the logical time diagram and the logical space diagram into a single diagram system yields the logical spacetime diagram, which can simultaneously realize temporal and spatial control. This spacetime diagram is still a life-giving structural logical equation diagram, can still apply substitutional diagram theory and associated operations, and uses different variables in the diagram for full control. The spacetime diagram is the intrinsic logic of the real universe, and can be mapped back to real spacetime through modality algorithms:

$$\text{Real Spacetime + Input Modality Algorithm = Spacetime Diagram} \quad (18\text{-}1)$$

$$\text{Spacetime Diagram + Output Modality Algorithm = Real Spacetime} \quad (18\text{-}2)$$

By applying substitutional diagram theory to time diagrams and space diagrams, an equation variable in a time diagram can represent a point in time and, simultaneously, the process of that point. For instance, a time point variable for one hour can also represent the process diagram of that hour. Similarly, an equation variable in a space diagram, such as one representing the solar system, can also simultaneously represent the space of the solar system.

2.5. The Logical Universe

Integrating the logical time diagram and the logical space diagram into a single diagram system yields the spacetime diagram. This diagram is infinitely large, reflecting the known logical laws of the universe. Its structure can be revised with the development of human knowledge until an advanced spacetime diagram is achieved, where the results of the diagram structure running according to the logical laws are consistent with real spacetime. This results in a logical universe diagram, which reflects the real logical universe and can generate various cosmic control variables, including temporal, spatial and spacetime controls.

2.6. Constructing the Logical Universe

In the context of structural intelligent technology, constructing the logical universe is equivalent to constructing a logical universe diagram—a life-giving structural logical equation diagram that describes the universe. The construction process requires two algorithms: the modality algorithm and the structural learning algorithm. There are two types of modality algorithm: input and output, and the entire process is represented by the following formulas:

$$\begin{aligned}&\text{Cosmic Spacetime or Local Cosmic Spacetime}\\&+\text{Input Modality Algorithm}\\&=\text{Spacetime Input Diagram}\end{aligned} \quad (19)$$

$$\begin{aligned}&\text{Machine Life Diagram + Structural Learning Algorithm}\\&+\text{Spacetime Input Diagram}\\&=\text{New Logical Universe Diagram}\end{aligned} \quad (20)$$

$$\begin{aligned}&\text{New Logical Universe Diagram + Output Modality Algorithm}\\&=\text{Spacetime Output Diagram}\end{aligned} \quad (21)$$

2.7. Comparative Study of the Logical Universe, the Real Universe, and Cyberspace

The logical universe is a logical world constructed by structural intelligent technology, serving as the largest environment for the machine-implemented structure diagram. Structure learning is conducted within the logical universe. Because the logical universe is organized structurally and has a certain logical similarity to the real structure composing the real universe, the logical universe possesses the conditions required to simulate the real universe.

Cyberspace is the largest environment for machine learning, *i.e.*, the universe of machine learning. Cyberspace has information characteristics but lacks a structural nature, meaning cyberspace does not have the conditions necessary to simulate the real universe. The logical universe possesses both structural and network characteristics, so the logical universe can use existing cyberspace to simulate the real universe. This point creates the required conditions for structure learning to simulate life learning, and is also the reason why structural intelligence can simulate human life better than informational intelligence [9]-[16].

3. Central Diagram Processor Brain Structure

General Introduction

This section introduces the diagram structures of the various components of the Central Diagram Processor (Brain Structure): ① Innovative Central Processor Unit; ② Equation Unit Generator; ③ Clock; ④ Address Manager; ⑤ Communication Controller; and ⑥ Modal Expresser.

3.1. Innovative Central Processing Unit

3.1.1. Full Diagram of Central Diagram Processor Equation Structure

Full Diagram of Central Diagram Processor Equation Structure
= Knowledge Equation Structure Diagram
+ Life Clock Unit Diagram and its Subsidiary Diagram
+ Address Unit Diagram and its Subsidiary Diagram
+ Address Unit Diagram and its Subsidiary Diagram
+ Modality Control Unit Diagram and its Subsidiary Diagram (22)

3.1.2. Purpose of Full Diagram of Central Diagram Processor Equation Structure

The equation unit generator primarily controls the clock and other components of the processor; that is, the equation unit generator controls the entire intelligent machine. The equation unit generator is used to realize the two functions of diagram data and diagram learning. The life clock unit diagram controlled by the equation unit generator endows the machine with both life and vital signs. After this, the knowledge equation structure diagram endows the equation structure diagram of the software diagram. The life clock unit diagram and its subsidiary diagrams, the address unit diagram and its subsidiary diagrams, the communication unit diagram and its subsidiary diagrams, and the modality control unit diagram and its subsidiary diagrams then endow the software diagram with various clock forms, address forms, communication forms, and modality forms to realize the functions of diagram clock, diagram address, diagram communication, and diagram modality. Finally, this software diagram is realized, which includes diagram thinking, diagram analysis, diagram reasoning and diagram control. The central diagram processor should also be equipped with the various required registers, and the management of these registers still uses clock forms, address forms, communication forms, modality forms, and various types of equations.

3.2. Equation Unit Generator

In structural intelligence machines, the equation unit generator (referred to as the generator) lies at the core, acting as a controller. It is the main component of learning, determining the level of structural intelligence.

3.2.1. Knowledge Equation Structure Diagram

Standard Diagram of Knowledge Equation Structure Diagram
= Common Sense Structure Diagram + Diagram Obtained from
Data Training + New Software Diagram Knowledge + Generator
Logical Learning (23)

3.2.2. Connection with Software Diagrams

The main function of the equation unit generator is to generate unit software diagrams, and to control the clock and other components of the central processing unit by transmitting and receiving information, thereby controlling the entire structural intelligence machine.

3.3. Clock

3.3.1. Clock Equation Structure Diagram

Standard Diagram of Clock Equation Structure Diagram
= Life Clock Unit Diagram + Generator Clock Unit Diagram
+ Address Clock Unit Diagram
+ Communication Clock Unit Diagram
+ Modality Clock Unit Diagram
+ Unit Software Diagram Clock (24)

Clock Form = Time Quantity + Access Quantity (25)

Time Quantity = Generator Clock + Address Clock
+ Communication Clock
+ Modality Clock
+ Unit Software Diagram Clock (26)

3.3.2. Connection with Software Diagrams

The standard diagram of the clock equation structure diagram generates clock forms and transmits them to the standard form of logical equations and unit software diagrams, controlling the logical equations and unit software diagrams where the clock forms are located. The access quantity is the variable of the logical equations and unit software diagrams where the clock forms are located when accessing the clock. When a logical equation and unit software diagram terminates, the access quantity reports to the life clock unit diagram and the generator clock unit diagram.

3.3.3. Diagram Clock Theory

The clock is controlled by the equation unit generator, and then controls the generated software diagrams by means of clock forms. The operating principle of the clock consists of the diagram clock theory. Within this theory, the main object is the clock equation structure diagram synthesized from the main diagram life clock unit diagram and the subordinate diagram address clock unit diagram, generator clock unit diagram, communication clock unit diagram, modality clock unit diagram, and unit software diagram clock. The main diagram life clock unit diagram must first create a life clock, creating machine life and vital signs. The life clock is also an important condition for realizing the machine life cycle and the generator's heredity, since the clock can stipulate the life cycle and mark the time for the generator's knowledge structure diagram directory and control function directory. Secondly, the main diagram life clock unit diagram is triggered by the equation unit generator to generate each subordinate diagram, working with the subordinate diagrams to issue various clock forms. At the same time, it accepts the access quantity and replies, thereby creating a real-time dynamic software diagram environment. The diagram clock theory needs to be implemented through an artificially designed algorithm, and is an iterative evolution process.

3.4. Address Manager

3.4.1. Address Equation Structure Diagram

$$\begin{aligned}&\text{Standard Diagram of the Address Equation Structure Diagram}\\&= \text{Address Unit Diagram} + \text{Generator Address Unit Diagram}\\&+ \text{Address Unit Software Diagram}\end{aligned} \tag{27}$$

$$\text{Address Form} = \text{Address Quantity} + \text{Access Quantity} \tag{28}$$

$$\text{Address Quantity} = \text{Storage Address} \tag{29}$$

3.4.2. Connection with Software Diagrams

The generator first issues a command to the address unit diagram. The address unit diagram then creates the address form. After this, the address form is added to the standard form of the logical equations and unit software diagrams through the address unit software diagram to allocate storage space. When new storage space is needed, the access quantity accesses the address unit software diagram and transfers it to the address unit diagram to increase storage space. When an equation or unit software diagram terminates, the access quantity still informs the address unit software diagram and the generator address unit diagram. The address unit software diagram tells the address unit diagram to reclaim the storage space.

3.4.3. Diagram Address Theory

The operating principle of the address manager of the central diagram processor is the diagram address theory. Operating objects include the address equation structure diagram synthesized from the main diagram address unit diagram and the subordinate diagram generator address unit diagram, along with the address unit software diagram. The address unit diagram is triggered by the generator and works with the subordinate diagrams to generate various address forms to allocate or reclaim storage space, and also works with the subordinate diagrams to reply to the access quantity to manage storage space, realizing storage management and storage space allocation. The diagram address theory needs to be implemented through an artificially designed algorithm, and is itself an iterative evolution process. Existing storage management algorithms can be used for reference.

3.5. Communication Controller

3.5.1. Communication Equation Structure Diagram

$$\begin{aligned}&\text{Standard Diagram of Communication Equation Structure Diagram}\\&=\text{Communication Unit Diagram}\\&+\text{Generator Communication Unit Diagram}\\&+\text{Communication Unit Software Diagram}\end{aligned} \tag{30}$$

$$\text{Communication Form} = \text{Post Office Function} + \text{Access Quantity} \tag{31}$$

3.5.2. Connection with Software Diagrams

First, the generator issues a command to the communication unit diagram, which in turn creates the communication form. Then, the communication form is added to the standard form of the logical equations and unit software diagrams through the communication unit software diagram, creating a post office and post office functions. When new post offices and post office functions are needed, the access quantity accesses the communication unit software diagram and transfers it to the communication unit diagram to increase the size of the post office. When an equation or unit software diagram terminates, the access quantity still tells the communication unit software diagram and the generator communication unit diagram. The communication unit software diagram tells the communication unit diagram to delete the post office and complete the subsequent operations.

3.5.3. Diagram Communication Theory

The operating principle of the communication controller of the central diagram processor is the diagram communication theory. Operating objects include the communication equation structure diagram synthesized from the main diagram communication unit diagram and the subordinate diagram generator communication unit diagram, and the communication unit software diagram. The communication unit diagram is triggered by the generator and works with the subordinate diagrams to generate various communication forms to create post offices and post office functions. When new post offices and post office functions are needed, the access quantity accesses the communication unit software diagram and transfers it to the communication unit diagram to increase the size of the post office. It also works with the subordinate diagrams to reply to the access quantity to manage post offices, post office functions, and post office storage methods. The diagram communication theory needs to be implemented through an artificially designed algorithm; this is an iterative evolution process. Existing Internet communication algorithms can be used for reference.

3.6. Modality Expresser

3.6.1. Modality Equation Structure Diagram

Standard Diagram of Modality Equation Structure Diagram
= Modality Control Unit Diagram
+ Text Modality Implementation Diagram
+ Image Modality Implementation Diagram
+ Speech Modality Implementation Diagram
+ Video Modality Implementation Diagram
+ Spatial Modality Implementation Diagram
+ Other Modality Implementation Diagrams (32)

3.6.2. Connection with Software Diagrams

The equation unit generator first sends a command to the modality control unit diagram, which then generates a modality form and adds this to the standard form of the logical equations. When the description set, equation, or unit software diagram of a variable requires modality expression, a request is sent through the access quantity of the modality form to the corresponding modality implementation diagram of the standard diagram within the modality equation structure diagram. After this, the modality implementation diagram publishes the modality according to the description set and its own function.

3.6.3. Diagram Modality Theory

① Modality Control Unit Diagram

This is the main diagram of the modality expresser, and generates the modality form of the logical equations or unit software diagrams. In the real-time dynamic operation of a software diagram, its modality form can send the access quantity to the modality control unit diagram. This diagram analyzes the quantity and leads the path to the corresponding subordinate operating diagram according to the analyzed information. In this way, it provides the modality expression needed by the unit software diagram.

② Subordinate Diagrams of Modality Control Unit Diagram

Modality implementation diagrams exist for text, images, speech, video, space, and other factors, and are all subordinate diagrams of the modality control unit diagram. These diagrams also have various thought paths, with each path being a method for modality expression, and the equations within the path representing the different processes of this method. These subordinate diagrams are artificially designed and need to be continuously developed.

③ Application of Modality

The role of the modality expresser is to allow the unit software diagram to be expressed in a certain modality as needed. This is in line with the concept that objects of human thinking all have modalities. It should be noted that the logical equations or unit software diagrams acted upon by the modality expresser all have clock forms, meaning the expressed modality exists in real-time and will be affected by the clock; this is also in line with human thought processes.

3.7. Comparative Study on the Logical Similarity between the Central Diagram Processor and the Human Brain

The central diagram processor is composed of equation structure diagrams endowed with additional functions; it has structural characteristics and is the structure diagram organization required for structure learning. The learning process involved is the process of change in the structure diagram. The human brain is composed of a brain cell organization, providing structural characteristics and the required organization for life learning. Structure learning and life learning have logical similarity, being able to learn through changes in their own structures. The reason lies in the fact that equations have a logical similarity to cells, while equation structure diagrams have a logical similarity to cell organization. The central diagram processor is an equation structure diagram, and the human brain is a form of cell organization; both exist as structures, and both complete activities and functions through structural changes. Therefore, the central diagram processor and the human brain have a definite logical similarity, and this law can be utilized to use structure learning to simulate life learning [17] [18].

4. Software Diagram—Body Structure

General Introduction

This section introduces the definition and working method for the software diagram (body structure).

4.1. Definitions

The life-giving structural software logical equation structure diagram, referred to as a software diagram, is a directed structure diagram in which each standard form of a logical equation is connected by a variable value transmission path. The standard form of a logical equation includes clock form, address form, communication form, modality form, and equation, creating a life-giving structural directed structure diagram.

The operating method of the software diagram is expressed by equations, while its operating process also takes the form of equations; that is, the output is obtained from the input. The realization of the software diagram is the program implementation of the equations according to the required path. An equation is a function program, and the path is a path for assigning values to certain variables of one equation to the input variables of the next equation.

The software diagram represents the machine's body structure, since its variables are able to control hardware devices. It uses a structural directory to be divided into various components, with diagrams under certain directories corresponding to the control of a specific part of the hardware devices. Thus, the software diagram becomes the machine body structure diagram.

4.2. Work

The software diagram has three functions: Firstly, it realizes the central diagram processor unit's control over the software diagram through the clock, address, communication, and modality forms. Secondly, it realizes the generation or deletion of the unit software diagram by the equation unit generator of the central processing unit. Thirdly, it performs diagram thinking based on the real-time dynamic software diagram to control the machine life.

4.3. Comparative Study of Software Diagrams and Artificial Neural Networks

Artificial neural networks, abbreviated as neural networks [19], are large scale information processing systems formed by the interconnection of numerous processing units, similar to neurons. They are a representative class of machine learning algorithms and can be divided into input layers, hidden layers, and output layers. The software diagram uses logical equations as processing units. These equations are structure processing units, whereas the neurons of the neural network are information processing units. In simulating the human brain, the software diagram clearly has advantages over the neural network. Furthermore, the training process of neural networks is carried out through the "back-propagation" algorithm. The software diagram also offers the route functions of forward and backward propagation, meaning that the software diagram can inherit the development achievements of neural networks [20]-[22].

5. Diagram Thinking—Life Activity

General Introduction

This section introduces diagram thinking (diagram transformation, life activities): ① Diagram Thinking Theory; ② Diagram Analysis Theory; ③ Diagram Reasoning Theory; ④ Diagram Control Theory; ⑤ Equation Diagram Analysis Method; ⑥ Path Diagram Analysis Method; ⑦ Parallel Diagram Analysis Method; ⑧ Cross Diagram Analysis Method; ⑨ Massive Diagram Thinking, Complex Diagram Thinking, General Pathogenesis Full Diagram Thinking and General Target Full Diagram Thinking, and Group Diagram Thinking; and ⑩ Study on Input and Output of Software Full Groups.

5.1. Diagram Thought Theory

Diagram thought theory involves how thought processes work within a life-giving structural software logical equation structure diagram (referred to as a software diagram). Diagram thought involves three steps: diagram analysis, diagram reasoning, and diagram control. This diagram-based control governs the machine's body structure, generating all of its life activities.

Diagram thought is based on these two formulas:

① Forward Thinking: Assigned Input Variables + Logical Relationship between Input Variables and Output Variables = Solved Output Variables (33)

② Reverse Thinking: Assigned Output Variables + Inverse Relationship of the Logical Relationship between Input Variables and Output Variables = Solving Input Variables (34)

5.1.1. Diagram Analysis Theory

Diagram analysis is the first step in diagram thinking. Its purpose is to find the logical relationship or the inverse relationship of the relationship between input variables and output variables in a real-time dynamic software diagram environment; the difference between various forms of diagram thought lies in the difference in diagram analysis.

5.1.2. Diagram Reasoning Theory

Diagram reasoning is the second step in diagram thinking. The processes of all forward diagram reasoning and all reverse diagram reasoning are the same, and are all based on the logical relationship or the inverse relationship of the relationship between input variables and output variables found by diagram analysis. These variables are solved according to the assigned values.

5.1.3. Diagram Control Theory

Diagram control is the third step in diagram thinking. Based on the output variables or the input variables solved by diagram reasoning, this theory focuses on the effect of the solution on the controlled object. This enables the machine and its activities to be controlled.

5.2. Equation Diagram Analysis

Equation diagram analysis refers to the logical relationship between input variables and output variables, represented by a logical equation or a logical equation set, and involves finding the required logic or reverse logic of this relationship.

5.2.1. Forward Equation Diagram Analysis

Forward equation diagram analysis involves the study of logical relationships represented by a logical equation or a logical equation set.

5.2.2. Reverse Equation Diagram Analysis

Reverse equation diagram analysis involves the study of inverse relationships of logical relationships represented by a logical equation or a logical equation set.

5.3. Path Diagram Analysis

Path diagram analysis refers to the logical relationship between input variables and output variables, represented by a path in a real-time dynamic software diagram. This features an initial equation, several intermediate equations, and a final equation.

5.3.1. Forward Path Diagram Analysis

Forward path diagram analysis involves the study of logical relationships represented by the equation path.

5.3.2. Reverse Path Diagram Analysis

Reverse path diagram analysis involves studying the inverse relationship of the logical relationship represented by the equation path.

5.4. Parallel Diagram Analysis

Parallel diagram analysis refers to the logical relationship between input variables and output variables, consisting of multiple parallel paths in a real-time dynamic software diagram, with each path having an initial equation, several intermediate equations, and a final equation.

5.4.1. Forward Parallel Diagram Analysis

Forward parallel diagram analysis involves the study of logical relationships represented by each parallel equation path; forward diagram reasoning is performed to solve the output variables of each parallel equation path according to the assigned input variables and the logical relationship between the input variables and output variables represented by each parallel equation path obtained from diagram analysis. Finally, forward diagram control is performed to study the effect of the output variables obtained by each equation path on the controlled object.

5.4.2. Reverse Parallel Diagram Analysis

Reverse parallel diagram analysis involves the study of inverse relationships of the logical relationships represented by each parallel equation path. Reverse diagram reasoning is performed to solve the input variables based on the assigned output variables of each parallel equation path and the inverse relationship between the input variables and output variables represented by each parallel equation path obtained from diagram analysis. Finally, reverse diagram control is performed to study the effect of the input variables obtained by each parallel equation path on the controlled object.

5.5. Cross Diagram Analysis

Cross diagram analysis refers to the logical relationship between input variables and output variables, represented by multiple parallel paths in a real-time dynamic software diagram. Each path consists of an initial equation, several intermediate equations, and a final equation. The various parallel equation paths may intersect, meaning several parallel paths could pass through one or more common equations. However, due to their real-time dynamic nature, the time points at which two different paths pass through the same equation may not be the same, resulting in different equation states.

5.5.1. Forward Cross Diagram Analysis

Forward cross diagram analysis involves the study of logical relationships represented by each intersecting equation path, followed by forward diagram reasoning and forward control.

5.5.2. Reverse Cross Diagram Analysis

Reverse cross diagram analysis involves the study of inverse relationships of the logical relationships represented by each intersecting equation path, followed by reverse diagram reasoning and reverse control.

5.6. Large Diagram Thinking

Regardless of equation diagram thinking, path diagram thinking, parallel diagram thinking, or cross diagram thinking, if the number of variables to be solved is significant, this is known as large diagram thinking.

5.7. Complex Diagram Thinking

Complex diagram thinking is still divided into three steps, namely complex diagram analysis, complex diagram reasoning, and complex diagram control.

5.7.1. Complex Diagram Analysis

To draw an analogy, complex diagram analysis is like the actual operating diagram of China's high-speed railway network. This is filled with diagram analysis paths in various directions, both in parallel and intersecting. There are paths in completely opposite directions, which may simultaneously be filled by forward and reverse analysis. The equations participating in this analysis are numerous, similar to the number of railway stations present on the network. In the future, the number of equations participating in this form of analysis could be ten thousand or even a million times the number of Chinese railway stations. A huge number of logical equations participate in the analysis, and the complete analysis diagram will act dynamically in real-time due to the action of the clock. This is complex diagram analysis, which involves finding the logical relationship between the input variables of the starting equation of various paths and the output variables of the ending equation of the path, or finding the inverse relationship of the logical relationship between the output variables of the ending equation of various paths and the input variables of the starting equation of the path. Complex diagram analysis is an important source of the structural complexity of intelligent machines, and also an essential source for improving the learning ability and thinking of intelligent machines.

5.7.2. Complex Diagram Reasoning

Complex diagram reasoning features the same paths as complex diagram analysis, like the full diagram of China's high-speed railway network mentioned earlier. On the surface, it is complex and chaotic, but in reality, everything is in good order due to real-time dynamic scheduling of the whole system. Complex, but in good order; this summarizes the nature of complex diagram reasoning. Each path reasons dynamically in real time, according to the clock of the particular path. It then calculates its own path, and forms the scheduling required for complex diagram reasoning.

5.7.3. Complex Diagram Control

When complex diagram analysis and complex diagram reasoning generate a huge number of control variables, this is known as complex diagram control. It is a real-time, dynamic process due to the action of the clock form of the software diagram. The application focus of complex diagram control is intelligent military and government control. Currently, it can also be used for social control of human-machine interaction.

5.8. General Pathogenesis Full Diagram Thinking and General Target Full Diagram Thinking

General pathogenesis full diagram thinking and general target full diagram thinking expand the thinking used in pharmaceutical research to more general fields. The main difference from other forms of diagram thinking lies in the diagram analysis step. Here, all problem analyses are called pathogenesis analyses, while all problem-solving analyses are called target analyses.

① Full Diagram Pathogenesis Analysis

Pathogenesis analysis is a form of problem diagram analysis. Full diagram pathogenesis analysis is basically the same as complex diagram analysis, but pathogenesis is a special case, involving the complications caused by the laws of life. This requires the equation unit generator to simulate pathogenesis complications during the analysis process in order to create new unit software diagrams and analysis paths. The simulation of the equation unit generator is based on the research results of bionic logic.

② Full Diagram Target Analysis

Target analysis is another form of solution diagram analysis. Full diagram target analysis is basically the same as complex diagram analysis, but also features a special case: the target-derived situation caused by the laws of life. This may be the side effects of drugs or changes in the human body caused by medical treatment. This also requires the equation unit generator to simulate target-derived situations during the analysis to create new unit software diagrams and analysis paths. The simulation of the equation unit generator is based on the research results of bionic logic.

③ Full Diagram Health Analysis

Full diagram health analysis, much like a real-life physical examination, involves a full diagram forward analysis to see if all the values of the output variable set are healthy. This includes tests to check if the output values are healthy depending on different input values, such as food inspection or drug experiments.

5.9. Group Diagram Thinking

① Chaotic Situation of Full Diagram Online Group Software

The chaotic situation of the full diagram for online group software is the same as the chaotic situation of the full diagram of the logical machine software. The input and output ends are often chaotic, and a logical machine node may simultaneously act as an input end, an intermediate process, and an output end.

② Problem Analysis on Chaotic Paths for Full Diagram Online Group Software

The problem analysis for this type of group structure diagram uses the same method as the complex analysis of chaotic paths in the full diagram of the machine software; but in this case, the path is composed of "Intranet Output Path + Extranet Transmission path + Intranet Input Path".

③ Solution Analysis on Chaotic Paths for Full Diagram Online Machine Software

The solution for these chaotic paths uses the same method as the complex analysis of the solution of chaotic paths in the full diagram machine software. However, in this case, the path is composed of "Intranet Output Path + Extranet Transmission Path + Intranet Input Path".

5.10. Study on Input and Output of Full Software Groups

The input and output of the full software diagram are chaotic, rather than uniform and tidy. An equation node may be an input node, an intermediate node and an output node at the same time, while the logical machine node of the network group structure diagram of the logical field may also be like this.

The clock, the initial equation, and the final equation determine the input and output, as well as the starting point and the ending point. That is, the input of a path is the input that occurs in the full software diagram state of a certain clock. Controlled by the clock, the starting point of the input is a certain equation of the software diagram of a certain clock. The path that it takes passes through each node equation is controlled by the clock form, and the output is also a specific equation of the software diagram of a certain clock. Specific problems need to be analyzed individually, and this situation exists in both forward analysis and reverse analysis of the full software diagram, as well as in pathogenesis analysis and target analysis.

Regardless of the form of analysis used, there is always more than one input equation and input clock in the pathogenesis, but there may be multiple input equations and multiple input clocks. In this way, there may be more than one directed path for the pathogenesis path, but multiple directed paths. The same is true for the target, because there is more than one treatment method and more than one treated human tissue; similarly, there is more than one input equation and one input clock. Naturally, there is also more than one target path. Through pathogenesis and target analysis, and through forward and reverse analysis, there may be more than one input equation and input clock. Multiple inputs may occur in different equations and different clocks, creating multiple paths. The output may also generate multiple outputs in different clocks and different equations. Therefore, forward analysis, reverse analysis, pathogenesis analysis, and target analysis must face multi-forward paths, multi-reverse paths, multi-pathogenesis paths, multi-target paths, and multi-recovery paths.

The paths we are seeking may be a group of paths. The question then arises: will this all be very complicated? No, the analysis of a group of paths will be complicated if it remains purely theoretical, but it will become very simple if you use the activation target program of the full software diagram to carry out the analysis. In the blink of an eye, the clock will schedule the executable full software diagram.

In addition, the analysis of a group of paths is the norm of the universe, the norm of life, and the inevitable requirement of expanding bionic logic theory.

① Full Diagram Chaotic Path Analysis

The full diagram chaotic path analysis uses the same method as the full diagram analysis, but the input end and the output end are purely chaotic, requiring a method for analyzing the journey of train passengers. One input is used to track the radial rays until they reach the output.

② Full Diagram Chaotic Path Solution Analysis

This requires first performing a full diagram chaotic path analysis, followed by a standard full diagram solution analysis based on the results.

5.11. Comparative Study of Diagram Thinking vs. Deep Learning and Reinforcement Learning

Diagram thinking is a simulation of human brain thinking—a structural simulation. Diagram thinking is the activity of the diagram, which has a logical structural similarity to human brain thinking (which is the activity of human brain organization). Deep learning and reinforcement learning also simulate human brain thinking, but these are informational simulations, and their efficiency and accuracy are both inferior to diagram thinking. Furthermore, diagram thinking can think within a spatial diagram, resulting in stronger capabilities. Because diagram thinking requires machine implementation, just as deep learning and reinforcement learning do, diagram thinking can continue the developments of deep learning and reinforcement learning, increasing the scientific nature and feasibility of diagram thinking [23].

6. Machine Life and Structural Learning

General Introduction

The theory of machine life introduces the following: ① Definition of Machine Life; ② Theory of the Life Clock; ③ Vital Signs; ④ Metabolism; ⑤ Life Cycle; ⑥ Machine Heredity; ⑦ Self-awareness; and ⑧ Significance of Machine Life.

The theory of structural learning introduces the following: ① Structural Diagram Algorithm of Central Diagram Processor; ② Diagram Learning Theory; and ③ Proposed Equation Analysis Algorithms.

6.1. Machine Life

① Machine Life

The machine clock, vital signs, metabolism, life cycle, and machine heredity of the software diagram realized by the software and hardware contained within structural intelligence machines will produce machine life. Note that I disagree with the machine life theory proposed by scientists in some countries, and I even hold a critical attitude towards this way of thinking. In contrast to natural life, machine life is biomimetic, which is something distinct from real life. The microscopic structure of machine life stops at the parts and cannot construct the concept of life by studying living microscopic tissue in the way that natural life can be studied.

② Life Clock

"3.1.3.1 Clock Equation Structure Diagram" is as follows:

Standard Diagram of Clock Equation Structure Diagram = Life Clock Unit Diagram + Address Clock Unit Diagram + Communication Clock Unit Diagram + Generator Clock Unit Diagram + Modality Clock Unit Diagram + Unit Software Diagram Clock.

The life clock unit diagram creates the life clock.

③ Vital Signs

All life in the universe contains vital signs, which are of great significance for reproduction, inheritance, iterative evolution, life cycles, and other life activities. This tells us that intelligent machines must also develop machine vital signs. The vital signs of life come from the clock of the cell, the clock of the life tissue, the clock of the life individual, and the clock of the life group. Using bionic logic to develop machine vital signs can only be achieved by means of structural intelligence. In this way, the equation structure diagrams of hardware, software, machine society, and human-machine society must all have clocks. To this end, a clock equation system is established, with a clock embedded in the central processing unit chip; this includes both the single-machine central processing unit and the central processing unit of the network server. The clock creates a clock equation system, which is used to simulate the life clock. The clock equation variables generated by the clock equation system control various equations and equation unit structure diagrams. The functions of the equations and the unit structure diagrams changing with time can express the machine's vital signs. In this way, we can create reproduction, inheritance, evolution, life cycles, and other machine life activities according to the machine's vital signs. Here, the equation unit generator can create genetic information and create a form of heredity. This again illustrates that the structural intelligence machine is true scientific bionic logic, an intelligent machine constructed using the natural laws of life intelligence.

④ Metabolism

Machine metabolism: The life clock unit diagram and its subsidiary diagrams controlled by the generator can make the clock form return to zero, which is equivalent to deleting the logical equation or unit software diagram where the clock form is located. This is an important condition for the real-time dynamic life of the software diagram. The behavior of generating unit software diagrams for software diagrams or deleting unit software diagrams by controlling the clock is called machine metabolism.

⑤ Life Cycle

Intelligent machines have three levels of clocks: at a primary level inside the elements, at an intermediate level within the tissues and organs, and at an advanced level within the brain. The clock inside the elements determines the life cycle of those elements; the clock inside the tissues and organs determines the life cycle of the tissues and organs; while the clock inside the brain determines the life cycle of the brain.

Taking humans as an example, intelligence exists when the brain is alive, and is lost when the brain dies. The core of an advanced structure of intelligent machines is the machine brain structure. This forms the brain, which also contains a clock. The life cycle variable of this clock is determined by the clock form, which the brain structure is responsible for using to calculate the life cycle variable. This variable then determines whether the brain can survive and whether advanced intelligent machines do indeed possess intelligence. The clock form of the machine brain structure can perform both information reasoning (diagram reasoning) and structural reasoning (diagram control).

The structural intelligence machine is the diagram control of the machine by the brain structure software diagram, intermediate structure software diagram, and primary structure software diagram. The life cycle of the machine needs to be reflected in the life cycle of the software diagram, while the life cycle of the software diagram is formulated by the generator according to its own knowledge structure diagram. The generator then transmits information to the clock, and the clock form generated by the clock reflects and realizes the life cycle.

⑥ Machine Heredity

The phenomenon of machine heredity is created by the generator. It can create a real-time dynamic machine life environment by controlling the clock. In this environment, the generator can autonomously and automatically generate two executable directories by means of artificial algorithms. One of these is the directory of knowledge equation structure diagram, and the other is the directory of software and hardware control functions, both of which can be used as machine genes. When machine reproduction technology is developed further in the future, machine genes can be passed on to the next generation of intelligent machines, producing the phenomenon of machine heredity.

⑦ Self-Awareness

Real-time dynamic diagram thinking, machine life generated by vital signs and hereditary phenomena, can allow structural intelligence machines to have a form of self-understanding, a knowledge of its own brain activity, and a knowledge of its own structure. The ability and need for a metabolism gives structural intelligence to machines, the concept of "self". As the complexity of the structure diagram expands, the concept of "my interests" will also be generated. The equation unit generator will optimize the equation structure diagram according to the requirements of the two concepts of "self" and "interests", and even generate new structure diagrams to achieve early machine reproduction. All of this creates the early self-awareness of intelligent machines. The good side of self-awareness is that it is conducive to autonomously and automatically optimizing and protecting itself, while imitating human logic to improve intelligence and machine reproduction. The negative potential is that it may become hostile to humans, potentially leading to destruction and crime. For this issue, humans should therefore attach importance to the belief in civil rights, human rights, and the rule of law.

⑧ Significance of Machine Life

The three key entities of a structural intelligence machine are as follows: the machine brain, the robot, and machine society. The road for the proper development of intelligence is the way to developing machine life, while machine life itself is the intelligence of the structural intelligence machine. The structural learning ability of an intelligent machine also stems from the particular condition of the three main entities of machine life.

6.2. Structural Learning

6.2.1. Structural Diagram Algorithm for the Central Diagram Processor

1) Algorithm of Equation Unit Generator

① Algorithm for generating unit software diagrams: In order to generate a unit software diagram, the generator first sends notifications to the life clock unit diagram, address unit diagram, communication unit diagram, and modality unit diagram, in order to obtain the clock form, address form, communication form, and modality form. The equations of the knowledge equation structure diagram then form a standard logical equation to realize the generation of the software diagram.

② Common sense learning algorithm: The common sense equation structure diagram includes: logical spacetime, logical mathematics, logical physics, common scientific sense, and common social sense. This requires the artificial construction of existing knowledge into equation structure diagrams, a process which would require a huge amount of work, potentially the strength of an entire country. This is often used to install common sense equation structure diagrams on network servers to provide network sharing, similar in form to a library.

③ Data training algorithm: Intelligent technology started with data training algorithms. Information-based technology is not entirely useless, though it is important to remove the dross and take the essence. Structural technology retains data training technology, performs feasible and necessary data training on large amounts of data, and supplements the common sense learning algorithm.

④ New knowledge software diagram: Software diagrams generate new diagrams that the generator does not possess by means of diagram thinking. This involves both analysis and reasoning, and the new diagrams are then added to the generator structure diagram.

⑤ Generator logical learning algorithm (the core technology of structural intelligence): This algorithm combines artificially designed algorithms with new knowledge algorithms to develop the ability of intelligent machines to turn information of various modalities into logic, and then stores this as logical diagrams in the standard diagram of the knowledge equation structure diagram to perform logical generator learning. This is a process of continuous iterative development.

2) Clock Algorithm

The algorithm of the clock contains the following steps:

① The life clock unit diagram starts, endowing the intelligent machine with life and creating the machine's vital signs.

② The equation unit generator issues a command to generate an equation unit to the life clock unit diagram, after which the life clock unit diagram creates the address clock unit diagram, communication clock unit diagram, generator clock unit diagram, modality clock unit diagram, and unit software diagram clock based on the command. Then, the unit software diagram clock issues a clock form to the corresponding logical equation and unit software diagram. If this logical equation and unit software diagram require address clock variables, communication clock variables, generator clock variables, or modality clock variables, it accesses the corresponding diagram through the access quantity of the clock form to obtain the clock value. If this logical equation or unit software diagram terminates, then the access quantity of its clock form will respond to the life clock unit diagram. The life clock unit diagram then informs the equation unit generator of the termination through the generator clock unit diagram.

③ The clock algorithm needs to be designed and implemented artificially.

3) Address Manager Algorithm

This is the technology required to develop the address unit diagram to efficiently manage storage space.

4) Communication Controller Algorithm

This involves studying the communication unit diagram's ability to establish post offices and post office functions, making the post office both small and efficient. The clock and storage space required by the post office are obtained through the access quantity of the clock form and the address form of the logical equation or unit software diagram where the post office is located, respectively.

5) Modality Expresser Algorithm:

Extracting the essence of existing technology while discarding the dross enables the development of text modality implementation diagrams, image modality implementation diagrams, speech modality implementation diagrams, video modality implementation diagrams, spatial modality implementation diagrams, and other modality implementation diagrams. These features are an artificial design combined with the generator's new intelligent knowledge to meet the needs of intelligent technology for multi-modality expression and multi-modality thinking.

6.2.2. Diagram Learning Theory

1) Main Body of Learning: Equation Unit Generator

The main body of diagram learning is the equation unit generator of the central diagram processor, which is further divided into three types of equation unit generators:

① Equation Unit Generator of the Intelligent Machine Brain

This equation unit generator simulates the learning of the human brain, capable of producing knowledge and reactions, and thus performing diagram thinking.

② Equation Unit Generator of the Intelligent Robot

This equation unit generator simulates the learning of the human body, producing both knowledge and reactions. These lead to a form of diagram thinking that is control-oriented; that is to say, the robot uses the equation unit generator to generate new equation structure reaction diagrams and software diagrams to control various intelligent devices connected to the machine brain, including the learning of information reasoning (diagram reasoning) and structure reasoning (diagram control).

③ Equation Unit Generator Group of the Intelligent Society

This is a group of equation unit generators that simulates human society, with a leadership group and a citizen group. The leadership group learns information reasoning (diagram reasoning) and social reasoning (diagram control) to be able to lead the society, while the citizen group accepts the information reasoning and social reasoning of the leadership group. The information reasoning of the leadership group often refers to the generation of policies. The leadership group uses the law to implement social reasoning. The information reasoning learning of the leadership group machine enables the creation of machine policies and machine laws. The social reasoning learning of the leadership group machine helps to promote machine policies and machine laws to the citizen group machines.

2) Form of Learning: Equation Structure Reaction Diagram

Traditional machine learning uses huge amounts of data and enormous computing power to find program equations, before using these equations to provide the different required information. This technology has been criticized for three unscientific aspects. Two of these are that the cost required for huge data and enormous computing power is not economically viable. The third unscientific aspect is that the intelligent development path is different from that formed by the universe itself, and the future is bleak. The intelligent development path designed by the universe is the path of humans. Traditional machine learning adopts a similar method to that of training lower animals. Limited by traditional hardware constraints, the path of traditional machine learning is certainly ineffective, while data training is unable to produce advanced intelligence.

Diagram learning proposes the theory of generator learning combined with social learning, based on the needs of the intelligent era and looking forward to future hardware development prospects. Here, the theory of generator learning is studied, a form of robot learning theory. The robot's brain tissue (the equation unit generator) receives external information, performs information reasoning through learning, and forms results, *i.e.*, the equation structure reaction diagram. This diagram is used to create a software diagram, whose output is then used to perform structural reasoning to control the robot's intelligent machinery.

The equation structure reaction diagram, that is, the new knowledge equation structure diagram, describes the form and results proposed by diagram learning theory. This learning process is a form of information reasoning. A strategy is used to perform information reasoning on external information, after which an equation structure reaction diagram is formed. This diagram represents the knowledge learned by the generator, and is still a life-giving structural logical equation diagram. The role of the equation structure reaction diagram is to change the generator's original knowledge of the equation structure diagram. The software diagram generated by the new knowledge equation structure diagram is used to perform structural reasoning. In other words, the software diagram's thought output is used for reasoning, in order to control the robot's structural operations. The new equation structure reaction diagram is used to change the generator's original knowledge equation structure diagram. This new knowledge equation structure diagram not only provides the existing input to output of the software diagram, but also generates new variables. The output of the new software diagram is used to control intelligent machinery to bring new structural reasoning. This simulates both human thinking and the generation of new knowledge.

3) Learning Strategies

Using external information to construct an equation structure reaction diagram lies at the core of diagram learning, which consists of the following methods:

a) Bionic Logic Strategy

For the bionic logic strategy, the first step involves the bionics of the human body. Scientific research into biology, medicine, and psychology is used to study the laws of humans, how humans think about problems, and how the human brain controls the body. This includes the laws of how human senses work. The second step is to turn various laws into logical equations, and then to combine these logical equations to create an equation structure reaction diagram. The equation structure reaction diagram is stored in the knowledge equation structure diagram of the equation unit generator to be called up when needed. This process represents the information reasoning of bionic logic, and can also be called learning. The third step is to call the equation structure reaction diagram to generate a software diagram. Using the software diagram's thought output to control the robot's structural operations is at the core of structural reasoning.

b) Logical Physics Strategy

For the logical physics strategy, the first step is to study the scientific laws of the universe. The second step is to turn these laws into logical equations, and then combine them into an equation structure reaction diagram. This diagram is stored in the knowledge equation structure diagram of the equation unit generator to be called up when needed. This represents the information reasoning of logical physics, and is also known as learning. The third step is to call the equation structure reaction diagram to generate a software diagram. Using the software diagram's thought output to control the robot's structural operations constitutes structural reasoning.

c) Logical Mathematics Strategy

The bionic logic strategy and the logical physics strategy are both external learning strategies, but the logical mathematics strategy is an autonomous one. The following autonomous learning methods can be created using logical mathematics laws:

① The process of generating a new unit software diagram involves first determining a group of input variables, then finding the required output variables and the path from the input variable set to the output variable set. In this way, the logical relationship between the input variable set and the output variable set can be obtained, and a new unit software diagram, that is, a new equation structure reaction diagram, can be automatically generated according to this logical relationship. The new equation structure reaction diagram is used to change the generator's original knowledge equation structure diagram. The new knowledge equation structure diagram then not only drives the existing input to output, but also generates new variables. The output of the new software diagram is used to control intelligent machinery to realize new structure reasoning. This simulates human thinking and the generation of new knowledge.

② The process of generating a new equation unit can also enable us to determine a group of input variables, from which the values of the required output variables can be found, along with the path that provides logical reasoning between the two values. In this way, the logical relationship between the input variable set and the output variable set is obtained, and a new equation structure reaction diagram can be automatically generated based on this logical relationship. The new equation structure reaction diagram is used to change the generator's original knowledge equation structure diagram. This new diagram shows the existing input to outputs, and also generates new variables. The output of the new software diagram is used to control intelligent machinery to realize new structural reasoning. This still simulates human thinking and the generation of new knowledge.

③ The equation unit generator can also work with the communication system. A new equation structure reaction diagram is generated by a combination of newly input and existing equation units, or by a new output value generated by newly input variables generating a new equation structure reaction diagram, thus creating new autonomous learning methods.

④ Advanced equation unit generators can even create new equation structure reaction diagrams based on self-defined input and output sets.

d) Logical Data Analysis Strategy

The strategy of logical data analysis is another form of autonomous learning. Logical data analysis has similarities with the current large model data training, which both use data to generate program equations. However, there are also some important differences: ① The path of large model data training involves training, forming equations, and applying equations, while logical data analysis can form equations immediately; ② Large models create equations first and then apply them, suggesting an external learning strategy, while logical data analysis forms equations immediately and applies them immediately, suggesting an autonomous learning strategy; ③ The application of large model equation relies on flowchart programs, while the application of logical data analysis relies on new equation structure reaction diagrams. There is a difference between flowcharts and structure diagrams in these two programs; ④ Large models require huge amounts of data and computing power, leading to significant costs. They create intelligence in the same that lower animals can be trained, an inappropriate method for developing intelligence in an industrial society. Logical data analysis uses much less data and computing power than these large models. It uses the learning methods of biomimetic humans, enabling machines to first gain knowledge and then form abilities through thinking. This is a far more suitable path for the development of autonomous learning for intelligent machines in an intelligent society.

There are several types of logical data analysis strategies:

One type of analysis is the known equation structure diagram, where the value set of one or more unknown variables is obtained from the value set of one or more known variables.

Another form of analysis is the unknown equation structure diagram. Several class groups are set as different value sets, while the interrelationships between these class groups, their mutual interference, are studied. This research empowers the equation unit generator to create new knowledge equation structure diagrams and equation structure reaction diagrams [3].

6.2.3. Proposed Equation Analysis Algorithms

These equation analysis algorithms all utilize one important phenomenon: the data from each class group appears in an orderly manner according to time points, which are used as a clock to construct the algorithms.

1) Equation Analysis Algorithm 1

Scenario: This is the simplest class group equation analysis, where data from two class groups appear in an orderly manner according to time points.

① Finding the different class groups of the data class equation analysis. The data of each class group changes according to the time point. The data of different class groups is divided into different groups according to the time point, with the data within each group being the respective data of the same time point from different class groups.

② The groups are numbered according to the order of the time points. The first group is programmed with equations, with all logical equations listed; this may be a significant number of equations. Then, the logical relationships between the data in the second group are used to filter the logical equations in the previous group. By analogy, the logical relationship between the data in the latter group is used to filter the logical equations generated by the previous group. The final logical equations obtained are the logical equations or unit equation diagrams required for equation analysis. If a logical equation is not obtained at the end, then the equation analysis will have been in error.

2) Equation Analysis Algorithm 2

Scenario: An analysis of the epidemic situation in the cities of Hong Kong and Shenzhen. Because the nature of the two data sets is the same, a simple set merging is possible. In this case, an equation analysis algorithm can be generated.

① Finding the two class groups for Hong Kong and Shenzhen. The data of each class group changes according to the time point. Obviously, at the same time point, there is one data point for the Hong Kong class group and one data point for the Shenzhen class group. These two data points are formed into a group;

② The groups are numbered according to the order of the time points. The Hong Kong data of the first group is programmed with equations, with all logical equations listed; this may be a significant number of equations. Then, the logical relationship of the Shenzhen data for the first group is used to filter the logical equations generated by the Hong Kong data. After that, the logical relationship of the Hong Kong data for the second group is used to filter the logical equations generated by the first group, after which the Shenzhen data of the second group is used to filter the various logical equations generated by the Hong Kong data of the second group. By analogy, the logical relationships between the data in the latter group can be used to filter the logical equations generated by the previous group. The final "Equation (Hong Kong Data) ∧ Equation (Shenzhen Data)" of each logical equation obtained is then the logical equation or unit equation diagram comprising equation analysis. If a logical equation is not obtained at the end, then the equation analysis will have been in error.

3) Equation Analysis Algorithm 3

Scenario: An equation analysis study of a water quality data class set and an aquatic product data class set in a fishpond requires that the water quality data class set be added to each element of the aquatic product data class set. This situation can generate an equation analysis algorithm.

① The grouping method for this scenario is as follows: each group has the form "{Water Quality Data Class Set} + One Data Point from the Aquatic Product Data Class Set}";

② The groups are numbered as required. The first group is programmed with equations, with all logical equations listed; this may be a significant number of equations. After this, the logical relationships between the data in the second group are used to filter the logical equations in the previous group. By analogy, the logical relationship between the data in the latter group is used to filter the logical equations generated by the previous group. The final logical equations obtained are the logical equations or unit equation diagrams required for equation analysis. If a logical equation is not obtained at the end, the equation analysis will have been in error.

4) Equation Analysis Algorithm 4

Scenario: There are several data class groups to be analyzed: housing price data, vegetable basket data, employment data, income data, stock price data, and customer satisfaction data; the associated equation analysis laws are unknown.

① Class group data for housing, vegetable basket, employment, income, stock price, and customer satisfaction at the same time point form a group;

② The groups are numbered according to the order of the time points. The first group is programmed with equations, with all logical equations listed; this may be a significant number of equations. The logical relationships between the data in the second group are then used to filter the logical equations in the previous group. By analogy, the logical relationship between the data in the latter group is used to filter the logical equations generated by the previous group. The final logical equations obtained are the logical equations or unit equation diagrams required for equation analysis. If a logical equation is not obtained at the end, then the equation analysis will have been in error.

5) Equation Analysis Algorithm 5

Scenario: Establish an index equation analysis for the various data class groups that affect atmospheric temperature.

① The indices of various data class groups that affect atmospheric temperature at the same time point form a group;

② The groups are numbered according to the order of the time points. The first group is programmed with equations, with all logical equations listed; this may be a significant number of equations. The logical relationships between the data in the second group are then used to filter the logical equations in the previous group. By analogy, the logical relationship between the data in the latter group is used to filter the logical equations generated by the previous group. The final logical equations obtained are the logical equations or unit equation diagrams required for equation analysis. If a logical equation is not obtained at the end, then the equation analysis will have been in error.

6) Data Class Group Equation Analysis Empowers the Equation Unit Generator to Construct Unit Software Diagrams

Data class group equation analysis research can empower the equation unit generator to construct unit software diagrams, creating the generator's main logical learning algorithm (the core technology of structural intelligence), and introducing a number of advanced mathematical tools. There are at least two scenarios for performing equation analysis of data class groups. In one scenario, an operation is designed to connect various data class groups into a collection, after which the relevant laws of equation analysis can be found through dynamic changes in the group. In the other scenario, the laws of equation analysis must first be understood, after which the operation connections can be determined based on these laws. The various data class groups are then brought together into a collection, after which the dynamic changes of key logical data can be observed and monitored, and the relevant social and natural dynamic developments understood.

7) Data Class Group Equation Analysis Empowers the Equation Unit Generator to Construct Unit Software Diagrams [3]

8) Theory of Structural Learning

Structural learning is divided into three stages: ① Stage 1: Machine perception, which requires developing machine perception for class data and optimizing existing perception technology by eliminating redundancy; ② Stage 2: Equation logic, which requires establishing its core components of class data, brain control and diagram control; ③ Stage 3: Structural diagram logic, which requires establishing the required diagram structure, brain control, diagram control, and diagram thinking for structural diagram logic.

9) Knowledge Generated by Structural Learning

Structural knowledge is embodied in the structure diagram of life-giving structural logical equations, the life-giving structural logical equation diagram, and divided into the gran control diagram of the central diagram processor, the body diagram responsible for life activities (*i.e.*, the software diagram), and the diagram thought activity process of the software diagram. The structure diagram of a life-giving structural logical equation is endowed with life by the clock of the central diagram process, after which the life of the diagram creates the life of structural knowledge. There are three main functions of structural knowledge: the central diagram processor uses the structural knowledge diagram to control the intelligent machine; the software diagram uses the structural knowledge diagram to create the machine body; and the software diagram also uses the structural knowledge diagram to perform information reasoning and structural reasoning. These are all forms of diagram thinking, able to control the hardware body to perform life activities.

10) Structural Learning in Real Life

Structural learning doesn't only exist in intelligent technology; humans and many biological organisms also exhibit structural learning. Therefore, the study of structural learning in humans and organisms is also very important, as it can improve life structures, develop education, enhance health, and revolutionize society.

6.2.4. Comparative Study of Structure Learning vs. Deep Learning and Reinforcement Learning

Comparing structure learning with machine learning (*i.e.*, deep learning and reinforcement learning), one can see that they are both informational simulations of the human brain, whereas structure learning is a structural simulation. Structural simulation is more complex, closer to reality, and possesses more functions than informational simulation. Moreover, structure learning still requires machine implementation, which is the same as deep learning, while reinforcement learning requires machine implementation. Structure learning can inherit the development achievements of both deep learning and reinforcement learning, increasing the scientific nature and feasibility of structure learning. [24]

7. Conclusions

Structural learning based on a structural spacetime universe, structural machine life, structural natural life, and a universal language.

Structural learning consists of learning with structural diagrams, learning with the complete and partial diagrams of machine life, learning with the central diagram processor (machine brain) of the complete diagram of machine life, and learning with the equation unit generator of the machine brain. It is the bionic structural learning of humans, the bionic learning of the human brain. It should be noted that structural learning is open to machine learning and all existing learning methods. That is, structural learning is able to adopt the essence of existing learning methods and achievements while discarding the dross, thus avoiding wasted labor. In this way, it can deal with massive data more effectively and correctly handle informational intelligence.

Structural learning needs to be bionic to human learning, so it naturally needs to be bionic to the human learning environment and associated methods. The human learning environment is influenced by the human spacetime universe, human life, natural life, the human sense of machines, and the human language. For the sake of bionics, and more importantly, for the correct development of structural learning, the structural learning environment must be constructed as an environment based on a structural spacetime universe, structural machine life, structural natural life, and a general language (The Class Data-based General Language [Structural Expression Language]). That is, the spacetime universe, machine life, natural life, and general language are all expressed here using the structural diagrams of life-giving logic equations of a class-data structural type (referred to as structural diagrams). This allows it to represent the real spacetime universe, machine life, natural life, and general language.

Whether structural intelligent technology can be open to intelligent informational technology determines the existence of the key feasibility of structural intelligent technology. This is because informational technology has been the norm since the invention of computers and the internet, and there are many achievements that need to be taken over. It is similar to the way humans used the informational learning of lower animals before developing human learning methods. When humans developed to the stage of human structural learning, they still retained many of the informational learning achievements from the earlier eras; humans call this instinct. This is the same for structural intelligent technology. Whether it can be open to informational intelligent technology depends on whether the structural learning of the structural intelligent technology can inherit the functions and achievements of informational machine learning and other informational learning. After this, it is able to create the instincts of structural intelligent technology, solve the problem of inheriting history, and also solve the need to inherit the informational intelligence created by massive computing power and massive data training. Once this problem is solved, the key feasibility of structural intelligent technology can be established.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Liu, D.H. (2024) Logical Structure School I (Five Fundamental Theories). *Intelligent Control and Automation*, **15**, 125-159. https://doi.org/10.4236/ica.2024.154008

[2] Liu, D.H. (2024) The Fundamental Theory of Artificial Intelligence—Logic Structure and Logic Engineering. *Intelligent Control and Automation*, **15**, 28-62. https://doi.org/10.4236/ica.2024.151003

[3] Liu, D.H. (2024) Logical Structure School II (Nine Applied Theories). *Intelligent Control and Automation*, **15**, 160-214. https://doi.org/10.4236/ica.2024.154009

[4] Chen, J., Zhou, D., Tang, Y., Yang, Z., Cao, Y. and Gu, Q. (2020) Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Yokohama, 11-17 July 2020, 3267-3275. https://doi.org/10.24963/ijcai.2020/452

[5] Dauphin, Y.N., Pascanu, R., Gulcehre, C., *et al.* (2014) Identifying and Attacking the Saddle Point Problem in High Dimensional Nonconvex Optimization. *NeurIPS* 2014, Montréal, 8-13 December 2014, 2933-2941.

[6] Wang, Y.X., Ramanan, D. and Hebert, M. (2017) Learning to Model the Tail. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 7032-7042.

[7] Cao, K., Wei, C., Gaidon, A., *et al.* (2019) Learning Imbalanced Datasets with Label Distribution Aware Margin Loss. *Proceedings of the* 33*rd International Conference on Neural Information Processing Systems*, Vancouver, 8-14 December 2019, 1567-1578.

[8] Duchi, J., Hazan, E. and Singer, Y. (2011) Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, **12**, 2121-2159.

[9] Yang, Z., Dai, Z., Yang, Y., *et al.* (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv: 1906.08237.

[10] Bartlett, P.L. (1998) The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights Is More Important than the Size of the Network. *IEEE Transactions on Information Theory*, **44**, 525-536. https://doi.org/10.1109/18.661502

[11] Neyshabur, B., Tomioka, R. and Srebro, N. (2015) Norm-Based Capacity Control in Neural Networks. arXiv: 1503.00036.

[12] Bartlett, P.L., Foster, D.J. and Telgarsky, M.J. (2017) Spectrally Normalized Margin Bounds Forneural Networks. arXiv: 1706.08498.

[13] Long, P.M. and Sedghi, H. (2019) Generalization Bounds for Deep Convolutional Neural Networks. arXiv: 1905.12600.

[14] Wei, C. and Ma, T. (2019) Data-Dependent Sample Complexity of Deep Neural Networks via Lipschitz Augmentation. arXiv: 1905.03684

[15] Dziugaite, G.K. and Roy, D.M. (2017) Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. arXiv:1703.11008

[16] Kang, B., Xie, S., Rohrbach, M., *et al.* (2019) Decoupling Representation and Classifier for Long-Tailed Recognition. arXiv: 1910.09217

[17] Hinton, G. (2012) Lecture 6D: A Separate, Adaptive Learning Rate for Each Connection. Slides of Lecture Neural Networks for Machine Learning. Technical Report.

[18] Li, H., Xu, Z., Taylor, G., *et al.* (2018) Visualizing the Loss Landscape of Neural Nets. *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*, Montréal, 3-8 December 2018, 6391-6401.

[19] Tang, K., Huang, J. and Zhang, H. (2020) Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect. arXiv: 2009.12991.

[20] Vaswani, A., Shazeer, N., Parmar, N., *et al.* (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 5998-6008.

[21] Neyshabur, B., Bhojanapalli, S., Mcallester, D., *et al.* (2017) Exploring Generalization

in Deep Learning. arXiv: 1706.08947.

[22] Yao, Z., Gholami, A., Keutzer, K., *et al.* (2018) Hessian-Based Analysis of Large Batch Training and Robustness to Adversaries. *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*, Montréal, 3-8 December 2018, 4954-4964.

[23] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, **37**, 448-456.

[24] Madry, A., Makelov, A., Schmidt, L., *et al.* (2017) Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv: 1706.06083.