A New Neural Network for Artificial General Intelligence

Haisong Liang

haisongliang@gmail.com

Abstract

Since artificial intelligence was first introduced several decades ago, neural network has achieved remarkable results as one of the most important research methods, and variety of neural network models have been proposed. Usually, for a specific task, we train the network with large amounts of data to develop a mathematical model, making the model produce the expected outputs according to inputs, which also results in the black box problem. In this case, if we study from the perspective of information meanings and their causal relations with the following measures: denote information by neurons; store their relations with links; give neurons a state indicating the strength of information, which can be updated by a state function or input signal; then we can store different information and their relations and control related information's expression with neurons' state. The neural network will become a dynamic system then. More importantly, we can denote different information and logic by designing the topology of neural network and the attributes of the links, and thus having the ability to design and explain every detail of the network precisely, turning neural network into a general information storage, expression, control and processing system, which is also commonly referred as "Strong AI".

English version is out of date and has some grammar or format errors, please refer to Chinese version if possible.

Revision: 2018-03-26

1 Introduction

In recent years, neural network has achieved remarkable results in many areas such as image recognition [1] and speech recognition. Many neural network models and algorithms have been proposed since then, for example, perceptron [2], DNN [3], BP algorithm [4], and so on. Among them, the majorities are based on the McCulloch – Pitts neuron model [5] and Hebb's learning mechanism [6].

Figure 1 shows the basic structure of the M-P neuron model, where x_i is the input, and w_{ki} is the weight of the input signal; $v_k = \sum_{i=1}^{n} w_{ki}x_i + b_k$, where v_k is the sum function, and b_k is the bias; $\varphi(\mathrm{v})$ is the activation function, and it can be a sigmoid or tanh function.

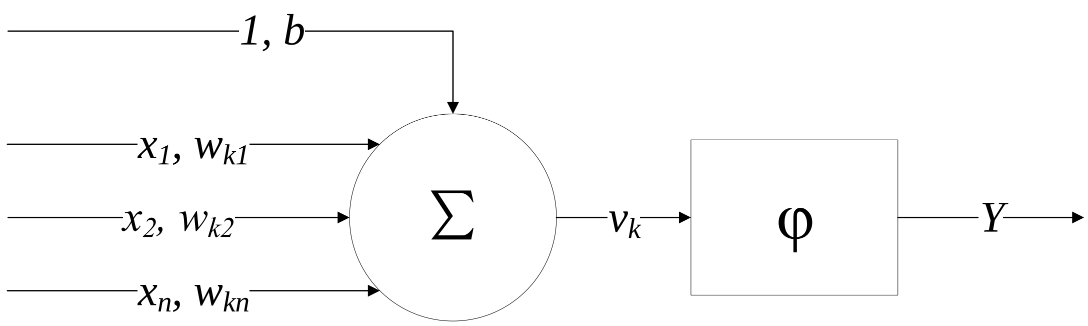

Figure 1 McCulloch – Pitts neuron model

Traditional neural networks treat the neuron as a compute unit and try to modify the weights of synapses by training. We train the network in order to develop a mathematical model, hoping the model produce the expected outputs according to inputs. The neural network is actually a linear or non-linear regression function then. Traditional neural networks are partial to math and statistics, ignoring the information's meaning and their casual relations, which may lead to little progress in artificial general intelligence for decades more or less.

In general, if we add a state representing the strength of the information; store their relations with links; update the state with a state function or input signal; then we can store different information and their relations and control related information expression by changing the neurons' state with state function. For ease of description, information, state, links and state function are named as information node.

If we study from the perspective of information meanings and their causal relations, for example, view neurons as information carrier, view neurons' state as information's expression, and use links to store and transfer the relations of information, then we may be able to get the answers to the principle of intelligent behaviors and the black box problem of the neural networks. Based on information and their relations, this paper proposes a new neural network model.

The paper is organized as follows. Section 2 introduces the new neuron model, and section 3 introduces the neural network based on the new neuron model. In the last, we make a conclusion and give some advice about future work.

2 Neuron Model

Figure 2 shows a simplified neuron model mentioned in this paper. This neuron is composed of an information state, input links, output links, and a state function. The state represents the strength of the information's expression or appearance and is denoted by x_k; Input links are used to send signals to this neuron, and then the state function $\varphi(v)$ update neuron's state. If the neuron's state is over a threshold, then it sends signals to other neurons by output links. Both input and output links are the same links, and w_k is the weight of a link, TTL_k is the duration of a link.

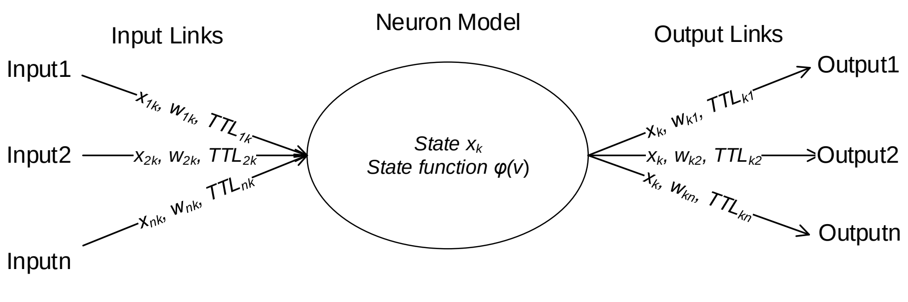

Figure 2 Neuron Model

2.1 Information State

In this paper, we use a state to show the strength of the information's expression or appearance. The state can be denoted by a ranged float value, such as [0, 1], where 0 is the smallest, and 1 is the biggest. The state is a dynamic value, which can be updated by input signal or state function.

2.2 Link

In this paper, links are used to record the influences of information on each other and transfer signals. Commonly, the effects of information can be positive or negative, so we can add an attribute to show this. When a neuron has an output bigger than threshold, positive links send positive signals to connected neurons, and negative links send negative signals to connected neurons.

The link between neurons is directional, and it points from one neuron to another. The links in neurons can be divided into two types, input links and output links. Input links receive signals from other nodes, output links send signals to other nodes on the contrary. For an information node, it may have multiple input links and output links, and there may exist multiple links between two nodes.

2.2.1 Signal Delay

Usually, when the neuron is activated, links will send signals immediately to connected neuron. But in some cases, the link may have signal delay so that it may send signal several clock cycles after the neuron's activation, for example, the action potential in neurons needs time to conduct [7]. So, we can give the link a delay attribute. When the neuron has an output, if the delay is not zero, the link will send signal to connected neuron after the delay.

All the links in this paper will have a default value zero for delay if not specified.

2.2.2 Signal Duration

The effects applied on neurons may have a duration. The duration may different in different links, some long, some short, but within the duration time, the link will always send signals and apply effect on the linked node. The duration is expressed with a TTL (Time to Live). When the neural network runs, we use a runtime TTL to express the remaining time of the signal. If the runtime TTL is not 0, it means that the link is activated and is sending signal.

2.2.3 Link Weights

The influence strength between information is different in most situations, so we give the link a weight attribute to represent the trait. The weight is a float number, and it can be positive or negative. Normally, we set it positive for positive links, and negative for negative links.

The weight of links and the number of links are equivalent and can be converted from one to another.

2.2.4 Persistence

In most cases, the links are persistent between information. But in dynamic systems, the relations between information can be modified dynamically. One of the modifications is that the links may be deleted or invalid after a period of time. Here we add a persistence attribute, or a survival time, to the links. When the link lives longer than its original survival time, then the link is invalid, and it won't transfer signal.

2.2.5 Link's Symbol

Figure 3 shows a symbol of the link. We use an arrow to denote the link, and the arrow points from the starting neuron to the endding neuron. The weight *w* and *TTL* are marked in the middle of the arrow. If the weight is 1 or -1, we mark the result of $w \times TTL$. As shown in Figure 3, -1 means a negative link with TTL 1.

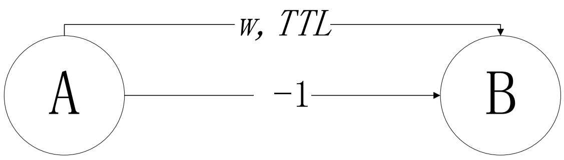

Figure 3 Link's Symbol

2.2.6 Other Attributes

The paragraphs above give some attributes to represent the relations between information, nevertheless the relations are multitudinous and not limited to the ones above. Similarly, we can add new attribute to the link for new relation if necessary.

2.3 State Function

State function is used to assess the state on the basis of links' attributes and input neurons' state. For a neuron, let x_n be the input of the nth link, and w_n the weight of the nth link; Set d the threshold of the neuron. If the weighted sum of links is bigger than d, then the neuron is activated. Let S be the neuron status, and true for active, then we can get the following equation:

Equation 1

$$S = \begin{cases} 1, & \sum_{i=1}^{n} w_i \times x_i > \mathrm{d} \\ 0, & \sum_{i=1}^{n} w_i \times x_i \leq \mathrm{d} \end{cases}$$

Considering the "all-or-none" law in biological nervous system [7], we give the neuron only two states. To express the state more precisely, we can use the activation function $\varphi(\mathrm{v})$ in traditional neural network to make the state in (0, 1), as shown in Equation 2.

Equation 2

$$\varphi(\mathrm{v}) = \varphi(\sum_{i=1}^{n} w_i \times x_i - \mathrm{d})$$

where $\varphi(\mathrm{v})$ can be sigmoid or tanh. More commonly, we can use a linear function to describe the strength of the information, such as ReLu function.

In the above operations, it is needed to pay attention to the x_n. Because of the signal delay and signal duration, the x_n can be different at different time.

2.4 Simplified Model

The neuron model in this paper can be simplified according to the all-or-none law. The state of a neuron can be active or inactive, and the weight can be 1 or -1. For state function, we can use a Heaviside function, and set d = 0. Then we have Equation 3.

Equation 3

$$S = \begin{cases} 1, & \sum_{i=1}^{n} w_i \times x_i > 0 \\ 0, & \sum_{i=1}^{n} w_i \times x_i \le 0 \end{cases}$$

Equation 3 is a step function, and the meaning of the above equation is that, in the input links of a neuron, if the number of positive links is bigger than the negative links', the neuron is activated. The network based on the simplified neurons is called simplified neural network, and the network model accords with the sparse matrix in traditional neural network [8].

2.5 Information Node Types

Classify the nodes by functions, we have three types of information nodes.

1. Input nodes: Input nodes, like input device in computer, receive signals from outside environment. The input links of input nodes are empty.
2. Inner nodes: Inner nodes are the nodes that don't communicate with environment directly. They connect with input nodes, output nodes and inner nodes.
3. Output nodes: Output nodes send signals to environment. When output nodes' status change, output device should react.

The state of information node can be updated by input signals, or by the state function.

3 Neural Network Model

3.1 Introduction

Combine the neurons together with links, we can get a large-scale information storage and processing network, and that is the new neural network model. In the ideal situation, all the information nodes can execute independently with a state function, forming a large-scale parallel system.

The nodes use a state function to update its states from input links all the time, and when nodes are activated, output links send signals to connected nodes. The time these operations cost is called a period T. For the neural network, if all the nodes do the same operation at the same time, then it is a synchronous network, otherwise it is an asynchronous network.

If the nodes' number or type, or links' number or attributes, can be modified during runtime, it is a dynamic network.

3.2 Execution Process

The execution flow of the network is simple, which mainly contains operations of different nodes:

1. Initialize the network if necessary;
2. For any of the input nodes, if it receives signals form environment, sends signals to connected nodes through output links;
3. For inner nodes, update their status according to state function, and send signals if activated;
4. For output nodes, update their status according to state function, and ask the output device to response.
5. Update links' attribute;
6. Repeat the above operations.

3.3 Network Simulation

In an ideal network, all the information nodes have executive capacity, which means that the nodes can update their status and send or receive signals on their own. But currently, there is no dedicated hardware designed for the network described in this paper, so it is necessary to simulate the network based on traditional computer architecture. For a complicated network, it may contain nodes and links over one ten trillion [9], demanding great storage and concurrency capabilities.

The following is an algorithm suitable for simulation. In practice, it is a great idea to adopt OPENCL or CUDA to speed up.

1. Save all the nodes in a collection, save the temporary weighted sum in node; Save all the links in a collection, save the runtime status of links, such as TTL and delay;
2. Traverse all the links, update the temporary weighted sum of the active links' connected nodes;
3. Update input nodes according input; Update inner and output nodes according to state function;

4. For links, update their runtime status; For activated input nodes and inner nodes, reset their runtime status to default; For output nodes, let the output device response;
5. Reset all the nodes' status.
6. Repeat step 2-5;

The algorithm processes nodes status first, processing links first can also work.

3.4 Persistent Node

Sometimes when we design network structures, we need some node keeping active to send continuous signals. Such nodes can be got by adding a link to itself, as shown in Figure 4.

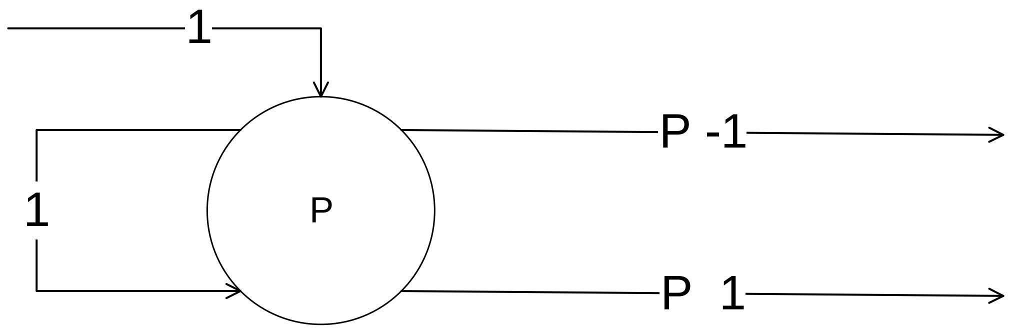

Figure 4 Persistent Information Node

With a link with TTL 1 pointing to itself, the node can keep active all the time, therefore we can get positive or negative signals at any time.

To tell the persistent links apart, here we add a "P" in the middle of the arrow. There is an input link with TTL = 1 in the node. This link is initialized active after the network begins, so the node can work the next period. Another solution is setting node status instead of link status.

3.5 Structures and Functions

By changing the nodes' topology and links' attributes, we can control the states of the nodes at different time, and thus control the information's expression. With different nodes and links, we can denote different information and logic.

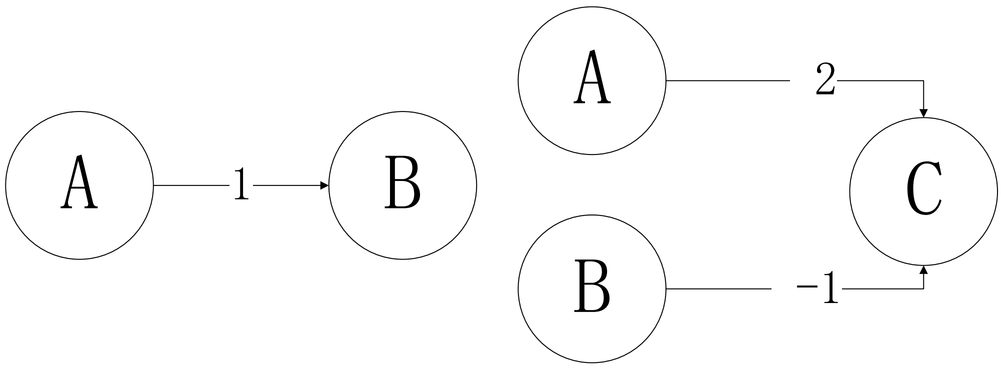

Figure 5

On the left part of the above network in Figure 5, if A is active, then B will receive activation signal from A one period later, and then B will be active; On the right part, the link A-C has a two-period positive signal, link B-C has a one-period negative signal, if at some period T A is active, then if we want C to be active, then B should not be active in T_1 and T_1+1.

The way nodes and links combined together is not unique, and there are multiple network structures for one function.

3.6 Examples of Logical Function

The neurons in the network have states, which can be updated by input signals or state function. The states can be Boolean, hence we can realize various logic based on the states for various information. This section we analyze the logical functions and their corresponding network structures based on the synchronous simplified neural network mentioned above.

3.6.1 AND

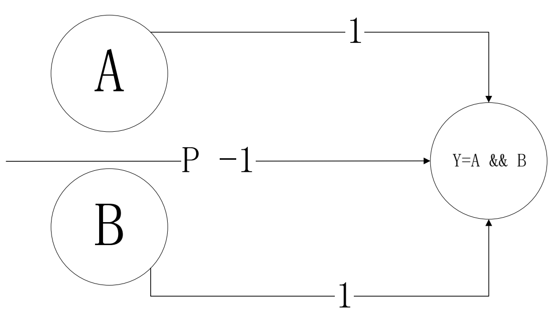

Figure 6 AND

Figure 6 shows the structure for AND. Node A denotes feature A, node B denotes feature B, and they both have a link pointing to node Y with TTL 1. In addition, Y has a negative link pointing to itself with a permanent TTL 1. Let 1 for active, and 0 for inactive. According to the state function, we can have the following truth table.

Table 1 AND

Input (T)		Output (T+1)
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

In Table 1, the states of A, B are their states at time T, the state of Y is its state at time T+1. From the table, we can draw a conclusion that Y = A AND B. Therefore, Y means that A and B active at the same time. A and B can be replaced with any other information, for example, pixels and sound strength.

3.6.2 OR

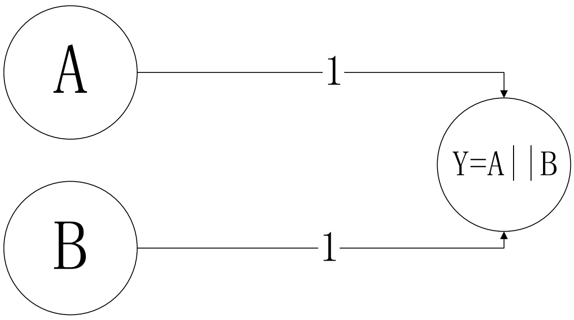

Figure 7 OR

Figure 7 shows the structure for OR. Node A and B both have a positive link pointing to node Y with TTL 1. According to state function, we can have Table 2.

Table 2 OR

Input (T)		Output (T+1)
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

In Table 2, the states of A, B are their states at time T, and the state of Y is its state at time T+1. From the table, we can draw a conclusion that Y = A OR B.

3.6.3 NOT

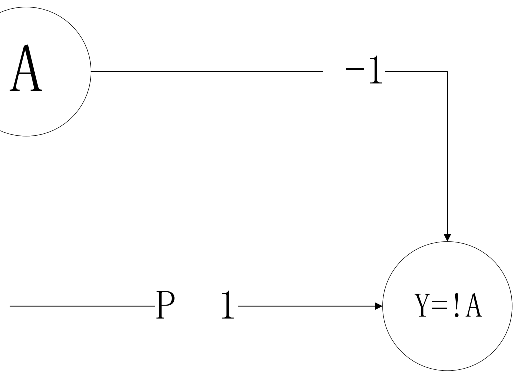

Figure 8　NOT

Figure 8 shows the structure for NOT. Node A has a negative link pointing to node Y with TTL 1. In addition, Y has a positive link pointing to itself with permanent TTL 1. According to state function, we can have the following truth table.

Table 3　NOT

Input（T）	Output（T+1）
A	Y
0	1
1	0

In the table, the state of A is its state at time T, the state of Y is its state at time T+1. From the table, we can draw a conclusion that Y = NOT A.

3.6.4 NAND

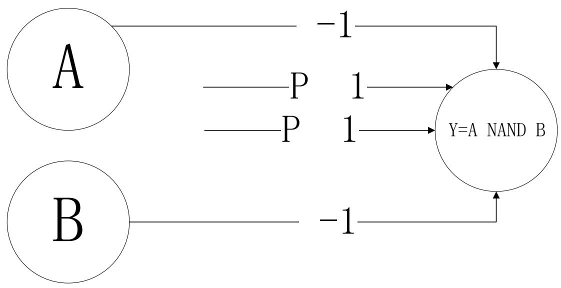

Figure 9　NAND

Figure 9 shows the structure of NAND. Because of the simplified neural network, we use two persistent positive links with TTL 1. The truth table is shown in Table 4

Table 4 NAND

Input（T）		Output（T+1）
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

3.6.5 NOR

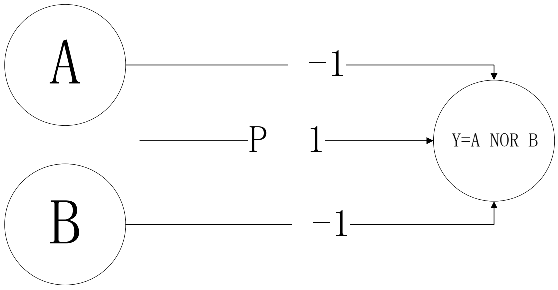

Figure 10 NOR

Figure 10 shows the structure of NOR, and the truth table is shown in Table 5.

Table 5 NOR

Input（T）		Output（T+1）
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

3.6.6 XOR

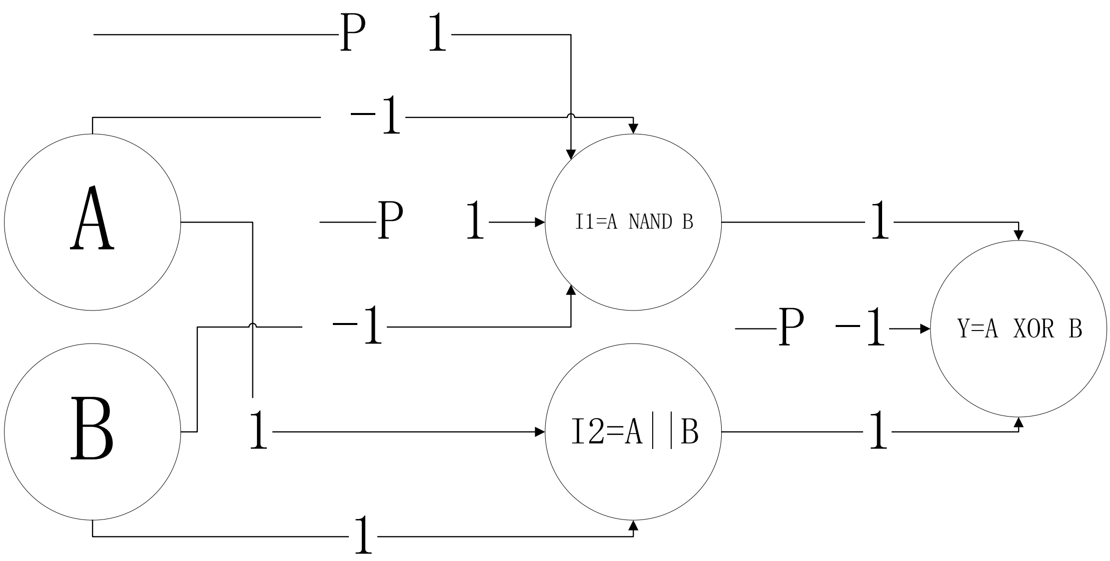

Figure 11 XOR

Figure 11 shows the structure of XOR. XOR is different because it is based on OR, AND and NAND, and it has intermediate nodes. The truth table is shown in Table 6.

Table 6 XOR

Input (T)		I1 (T+1)	I2 (T+1)	Output (T+2)
A	B	I1	I2	Y
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0

3.6.7 Other Logic

Some basic logic operations are listed above, and other logic operations can be got easily according to the theory mentioned in the paper.

3.7 Examples of Flow Control

Given a series of nodes, if the nodes denote some operations, then we can control the flow by controlling the nodes' activation order.

3.7.1 Sequential Structure

Given a sequence of operations A, B, C, the sequential structure is shown in Figure 12.

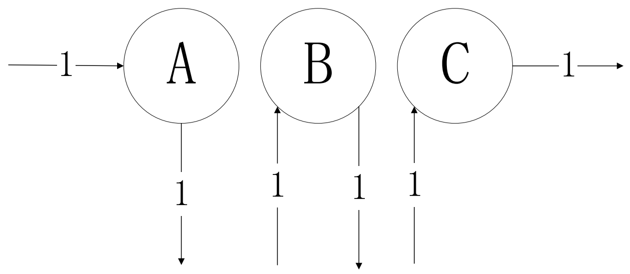

Figure 12 Sequential Structure

The start signal transfers to A, and A becomes active, so the operation A begins; After A is finished, it sends activation signal to B, then B begins executing; After B is finished, C begin executing, so we get the sequential structure.

3.7.2 Selective Structure

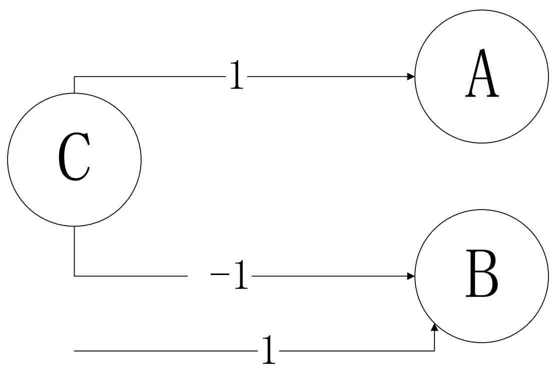

Figure 13 Selective Structure

Figure 13 shows a selective structure, where C is a control node. If C is active, A executes; If C is inactive, B executes.

3.7.3 Cycle Structure

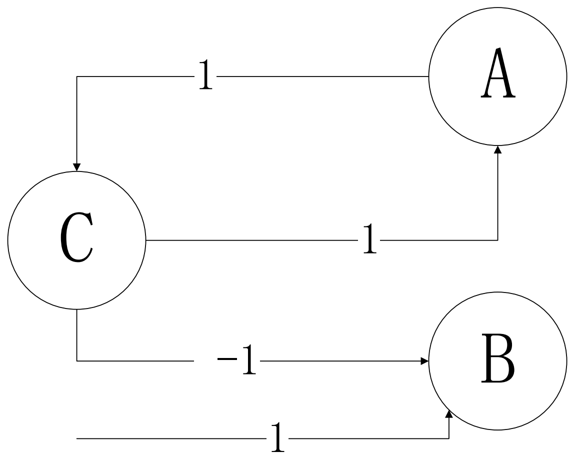

Figure 14 Cycle Structure

Figure 14 shows a cycle structure. C is a control node, and A is a set of operations. If C is active, A executes continually; If C is inactive, B executes.

3.8 Comparisons with Traditional Neural Network

The neural network discussed in this paper is mainly focused on information's meaning, their relations and their expression. The network can express information with neurons' states and can also control other information's expression with a state function. The network and its behaviors can be designed and explained precisely by the topology of the network and the links' attributes. Compared to the traditional neural networks based on M-P neuron model, we can find the following similarities:

1. Mathematical Model

   Both networks have the form $\varphi(v_{k)} = \varphi(\sum_{i=1}^{n} w_{ki}x_i + b_k)$, where $\sum_{i=1}^{n} w_{ki}x_i$ is the weighted sum, and b_k is a bias. The function $\varphi(v_{k)}$ can be sigmoid, tanh or ReLu.

2. Network Model

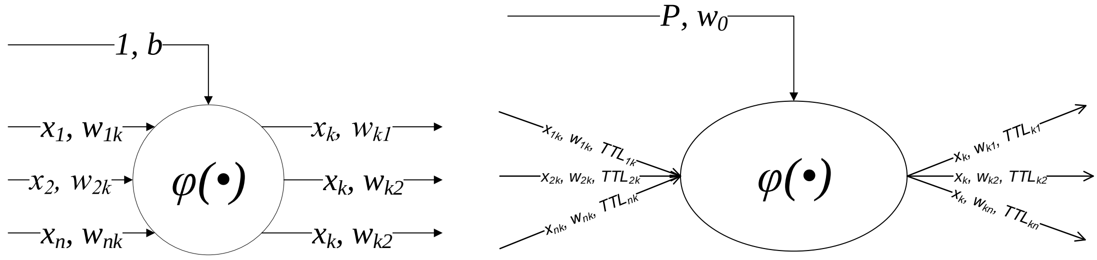

Figure 15 Two Types of Neural Network

Figure 15 shows the structure of the neurons. The left is the M-P neuron, and the bias is treated as an input with $x_n = 1$; the right is the neuron talked in this paper, whose threshold d can be replaced by an input link. Both networks have similar activation or state function, and the neurons can receive inputs or send outputs to other neurons.

Therefore, although they start from different way, they have the same mathematical and network model. They are similar in some sense, and it is possible to convert them between each other.

Traditional models train the network to modify the weights, and usually the networks have many layers; Ours network is mainly designed ahead of time by humans, to give the network specific functions. Also, our neurons can link to any neurons without limitations. Besides, the links have signal delay and duration to represent the relations between information. The most important thing is that we can design and explain our neural network, and we can answer the question of intelligence with information's expression and their relations, so the new neural network is much better than the originals.

According to the contents above, it is obvious that the traditional neural network is a special case of the new neural network.

For years I had always thought that it was hopeless to study intelligence from mathematics and statistics, but until recently I found the similarities between math method and information expression method. The similarities mean that the theory in this paper can explain the black box problem of traditional networks, in turn the successful results of traditional networks can be used to prove the correctness of the theory.

3.9 Information Storage and Coding

In traditional computer systems, information is stored as bits in memory after sampling and coding, and then the CPU process it. In our network, information is denoted and saved by a neuron. In addition, we have links to record and transfer the relations between information. The information nodes are not isolated in the network, on the contrary they are linked together by links, forming a complicated network, as is shown in Figure 16.

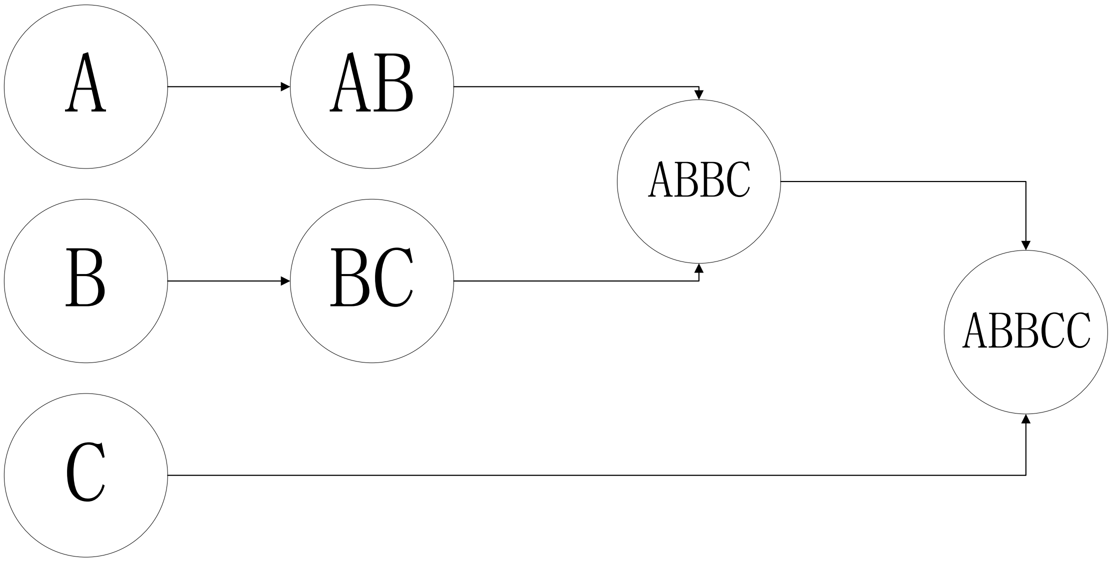

Figure 16 Information Storage and Coding in Neural Network

Because the links can ensure the relations between information, and the network is mainly used to express the information, so it is not necessary to save the information in the neurons. Instead, we can simply save information states, and point the links to the states.

3.10Network Complexity

For some functions, there exists multiple network that can work. To measure the efficiency of a network structure, hear we introduce the concept of network complexity. The complexity is calculated using the formula below:

Equation 4

$$\mathrm{C} = \mathrm{a} \times \mathrm{T} \times \mathrm{N} + \mathrm{b} \times \sum_{i=1}^{n} TTL_i$$

Where C is the network complexity, a is the factor of node, b is the factor of link, n is the number of nodes, T is the max duration of all the links, TTL_n is the nth link's TTL.

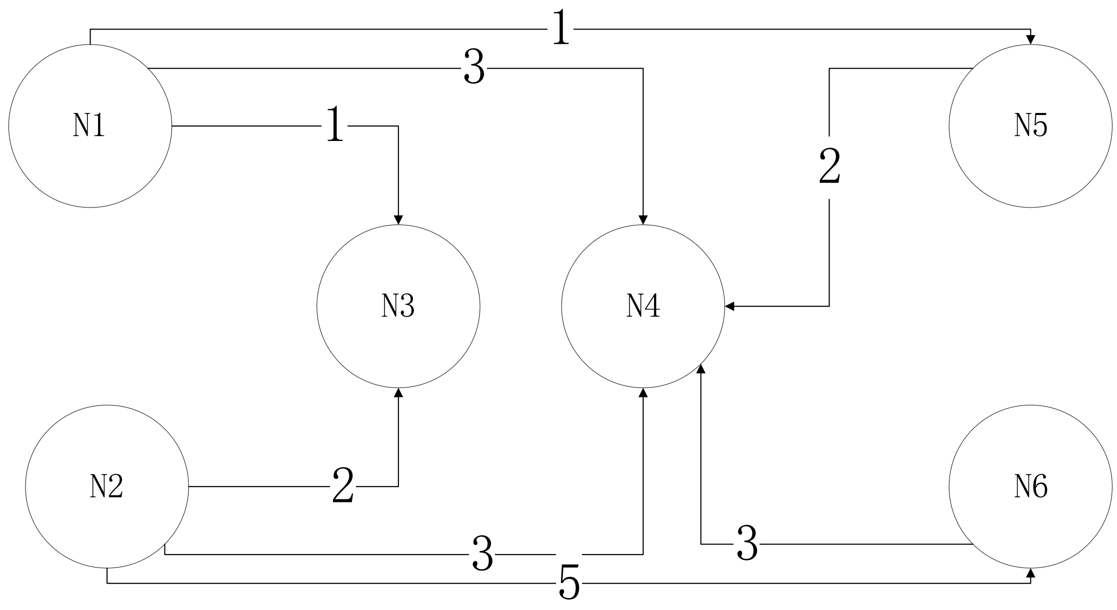

Figure 17 Network Complexity

Let a = b = 1, for the network shown in Figure 17, n = 6, T = 5, so C = 5×6 + 20 = 50.

4 Conclusion and Future Work

4.1 Conclusion

Up to today, though great progress has been made on humans' brain, there is no explicit answer for the crucial consciousness. Traditional AI research methods put emphasis on math and statistics, but here we put forward a new approach for information storage and expression based on their meanings and relations, hoping that it starts a new research filed for general artificial intelligence.

Though the new and the old neural network have similarities, they are different in other aspects. Traditional networks study from the perspective of math and statistics and try to develop a model by training for classification and prediction; The new network studies from information relations and expression, and is designed to realize specific functions. It is noteworthy that the latter can be used to develop a general information storage, expression, control and processing system, especially with dynamic neural networks.

The neural network in this paper can be designed by the topology and the links' attributes, giving us the ability to control everything in the network, especially the ability to control information expression. The new neural network is raised not as an alternative method for traditional methods, but as a general information storage and expression system targeting at artificial general intelligence.

4.2 Future Work

Because the topology structures and links for the neural network are too numerous to mention, this paper only covers a little, and there are many undiscovered realms waiting for us. In summary, some important research fields for artificial general intelligence based on the new neural network are listed below:

1. Information Representation: This filed studies how information are stored and organized in the network, for example, 1D, 2D and 3D information; In addition, to support perception and imagination, this network need connections with them.
2. Feature Extraction: Usually, the feature of information can be extracted with neural network, for example, timing characteristic, strength characteristic, spatial characteristic.
3. Logic Representation: For representing logic relations between information.
4. Flow Control: For how to control the operation flow.
5. Perception and Retrieve: For a neural network, perception means that the corresponding neuron is active. In addition, human can focus attention on specific information when facing a lot of information, so there must be a perception layer for the network to map the active neurons to the layer selectively; At the same time, there should be a retrieval network, to retrieve other information's state based on the layer's output; Together with the language, perception, inner or outer state of the agent, we can get a consciousness system.
6. Task Scheduling: For how to execute, monitor and schedule tasks.
7. Emotions: We can represent a 1D information with a set of neurons, and the active one in them can be changed by other neurons' state, and the change can send signals to other neurons. This can be a basic for emotions.
8. Language: Neurons can stay active for some time, and it can be got by other neurons and then output to the outside; Similarly, input signal can keep the inner neurons keep active for a duration. Therefore, we can design a network to describe the inner network state and transfer to the output or reappear the activation states in neural network by the inputs. This structure

can be used to support language.

9. Dynamic Neural Network: Dynamic network supports dynamic modifications of nodes and links, and is the basic of advanced intelligent behaviors, such as learning and memory.
10. Learning: The network structure can be designed ahead of time before running, or can be modified at runtime, and the latter is extremely important for strong AI. The learning here means the network modify itself with outputs to the interface of dynamic network, with congenital or acquired rules.
11. Memory: For dynamic network, combine language and perception system to establish a network for Long Short-Term Memory.
12. Imagination an d Association: By particular designed network, we can give connectivity to different neurons with a switch, and the switch can be controlled by the outputs from perception layer. Combined the learning ability, the neural network can create new nodes and links, so we can give the neural network creativity

They are the research directions for strong AI. Of course, they are not isolated, usually they have connections with each other. Only we integrate them organically into a neural network system can we develop a true artificial intelligent system.

5 References

1 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012: 1097-1105.

2 Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.

3 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.

4 Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533.

5 McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.

6 Hebb D O. The organization of behavior: A neuropsychological theory[J]. 2005.

7 Nicholls J G, Martin A R, Wallace B G, et al. From neuron to brain[M]. Sunderland, MA: Sinauer Associates, 2001.

[8] Olshausen B A, Field D J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images[J]. Nature, 1996, 381(6583): 607.

[9] Haykin S S. Neural networks and learning machines[M]. Upper Saddle River, NJ, USA:: Pearson, 2009.

一种适用于通用人工智能的神经网络

梁海松

haisongliang@gmail.com

摘要

自从人工智能提出以来，神经网络作为一种重要的研究方法，取得了巨大的进展，并诞生了多种多样的神经网络模型。一般地，针对某项特定的任务，我们通过大量的训练来修改神经元之间的参数，建立数学模型，使网络产生预期的输出，这也导致神经网络内部成为一个黑盒。如果我们从信息的意义、信息之间的因果关系和信息的表达的角度入手，用神经元表示某种信息，使用连接来记录和表示信息之间的作用关系，使用状态标记来表示信息的状态，状态由输入信号或者状态函数者根据连接及对应信息的状态更新，那么，神经网络就可以表示各种信息及他们之间的关系，并且可以根据信息的状态来控制不同信息的表达，成为一个动态的系统。因此，通过设计神经网络的拓扑结构及连接属性，就可以表示不同的信息和逻辑，从而实现对神经网络的精确设计和解释，并使得神经网络成为一种通用的信息存储、表达、控制和处理系统。

Revision: 2018-03-31

1 引言

神经网络作为研究人工智能的一种重要方法，近年来在图像识别【1】、语音识别等领域取得了显著的成果。自从神经网络诞生以来，诞生了许多神经网络结构，如感知器【2】，深度神经网络【3】等模型，并提出了许多算法，如 BP 算法【4】等等。其中，大多数网络模型和算法的基本原理都是基于 Warren McCulloch 和 Walter Pitts 提出的 McCulloch - Pitts 神经元模型【5】和 Donald Hebb 提出的学习规则【6】。

图 1 是 McCulloch - Pitts 神经元模型的基本结构，主要由输入集合，加法器，激活函数组成。其中，x_i表示输入信号，w_{ki}表示输入信号的连接权重；v_k是求和函数，$v_k = \sum_{i=1}^{n} w_{ki}x_i + b_k$，$b_k$为神经元的偏置；φ(v)为激活函数，一般可取 sigmoid 函数、tanh 函数等等。

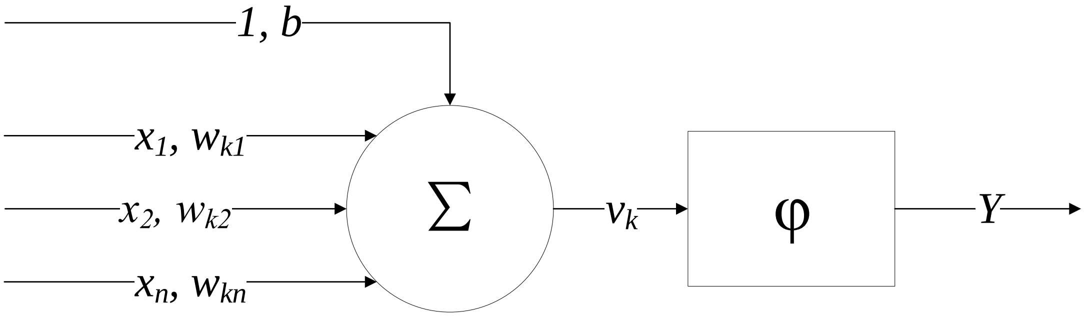

图 1　　McCulloch - Pitts 神经元模型

传统的神经网络，是将神经元视为一个计算单元，通过训练来修改连接的权值，建立一个数学模型，实现对输入信息的分类，其本质是一个线性或者非线性拟合函数。这些神经网络偏向于数学模型，没有重视信息的意义和信息之间的因果作用关系，或多或少地导致了几十年来通用人工智能的研究没有进展。

一般地，如果我们在处理信息的时候，给信息增加一个状态，用以表示信息出现或者表达的强弱；信息之间的各种因果作用关系，使用连接来表示并存储；信息的状态，使用状态函数根据连接的属性和相关的信息的状态来更新。从而，通过控制信息的状态标志，就可以控制其他相关信息的状态，从而控制相关信息的表达。为了便于描述，将信息、信息状态、连接、状态函数等统称为一个信息节点。

以烽火台为例，烽火用来传递敌情。如果烽火出现了，那么说明有敌情。在这个例子中，烽火和敌情是两个信息，烽火的出现与否是它的状态，敌情是要传递的信息，烽火出现则表

示敌情出现。

如果我们从信息的意义及信息之间的关系的角度出发，将神经元视为信息的载体，神经元的状态视为信息出现或者表达，突触用来记录和传递信息之间的各种作用关系，那么，我们或许能够解答智能行为的原理和传统神经网络的黑盒问题。基于上述原理，本文从信息的意义、信息的关系、信息的表达的角度出发，提出了一种新的神经网络模型。

使用信息的意义和关系来研究通用人工智能的人并不只有我一个，Michael Miller 也基于类似的原理提出了一个 PAM-P2 系统【7】。

本文后续的内容安排如下，第二章介绍了新的神经元模型，第三章介绍了基于新的神经元模型的神经网络，第四章对本文的内容做出总结并对未来的研究方向做出展望。

2 神经元模型

图 2 是本文所述神经元的一种简化的模型。这种神经元由状态标志、输入连接、输出连接、状态函数组成。状态标志表示信息出现或者表达的强弱，用x_k表示；输入连接向神经元传递输入信号，即其他信息的状态，用于状态函数$\varphi(v)$判定信息节点的状态，并通过输出连接向其他信息节点传送。输入连接和输出连接统称为连接，w_k表示连接的权值，TTL_k为连接持续的时间。

需要说明的是，图 2 所示的神经元模型没有标注连接的延时、持久性等属性。

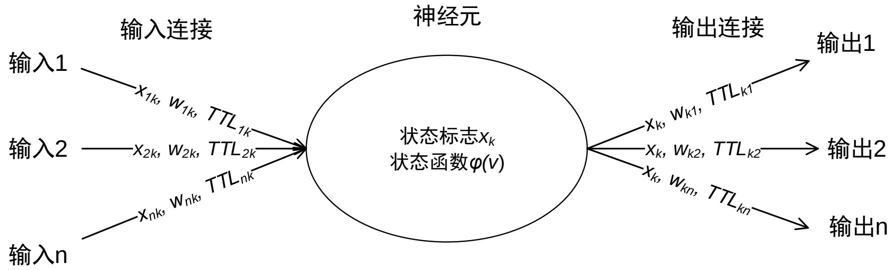

图 2 神经元模型

2.1 信息状态

在本文中，使用信息状态来表示信息出现或者表达的强弱程度。一般地，可以使用一个区间范围来表示，如$[0,1]$。其中，0 表示信息强度最小，1 表示信息强度最大。信息状态是

一个可以动态变化的值，可以由输入信号更新，也可以由状态函数更新。

2.2 信息关系

信息之间的各种关系，用连接来表示。通常，一个信息对另外一个信息的作用，可以是刺激性的，也可以是抑制性的。因此，可以给连接增加一个信号属性，用来表示刺激作用或者抑制作用。在神经元有输出时，刺激性连接向相连的神经元发送刺激性信号，抑制性连接向相连的神经元发送抑制性信号。这种规则符合神经递质对神经细胞的超极化或者去极化现象【8】。

连接具有方向性，由一个神经元指向任意一个神经元，可分为输入连接和输出连接。输入连接将其他信息节点的信号传递到本信息节点，输出连接将本信息节点的信号传递到其他信息节点。对于某一个神经元，可以具有多个输入连接和输出连接，并且在两个神经元之间可以存在多条连接。

2.2.1 信号的延时

一般地，在神经元激活时，可以通过连接立即将对应的信号传递出去。但是在实际情况中，信息之间的作用可能存在延时，并且在神经系统中动作电位的传递也存在时间【8】。因此，可以给连接一个延时属性，在神经元产生输出时，如果连接的延时不为 0，那么在经过相应的时间后，连接激活，向相连的神经元传递信号。

在本文中，若未做特殊说明，所有连接的延时均为 0，即没有延时。

2.2.2 信号的时长

一个信息对其他信息的作用效果，可以具有一定的持续时间，这个时间可以通过给连接增加信号持续时间来表示。在神经元产生输出后，连接激活，向相连的神经元传送信号，在信号持续时间之内，信号都会作用于相连的神经元。信号的持续时长用 TTL（Time To Live）表示，在网络运行时，除了原始的 TTL 外，还需要有一个运行时的 TTL，用以表示信号剩余的作用时间。若剩余 TTL 不为 0，则表示连接已经处于激活状态，正在通过连接向其他节点发送信号。

当然，信息作用效果的持续时间也可以通过给神经元增加激活持续时间来表示，在神经

元产生输出后的一定持续时间内，神经元都通过输出连接向相连的神经元发送信号。但是，这种方法有个弊端，即神经元无法区分对其他不同神经元的作用时长，也不易处理后续的神经元输出，因此，不如使用信号持续时长简单、灵活和强大。

2.2.3 连接的权值

一个信息受到多个输入信息的共同作用时，每个输入信息对最终的信息的状态的影响能力可能不同。在神经网络中，这种影响能力可以通过连接的权值来体现。通常，将信号的属性和连接的权值结合起来表示，将刺激性连接的权值设为正数，抑制性连接的权值设为负数。

连接的权值和连接的数量之间具有等效性，可以根据权值来调整两个神经元之间的连接数量，从而表示信息之间的影响能力。

2.2.4 持久和非持久性

通常，信息间的关系是持久的。但是在动态的系统中，信息之间的联系可以动态地改变，这种改变体现在连接可以新增或者减少，或者具有一定的持续时间。在持续时间之内，连接可以传递信号并作用于相连的节点；若超出持续时间，则连接失效，等同于连接不存在。根据连接是否是持久的，将连接分为持久连接和非持久连接。

2.2.5 连接的符号

图 3 是连接的一种图示符号。使用一个箭头表示连接，箭头的起点始于起始神经元，箭头的终点指向目标神经元，权值 w 和持续时长 TTL 标注在箭头中间。如果权值为 1 或者 -1，则直接标注$w \times TTL$的结果。如图 3，-1 表示一条权值为-1，持续时间为 1 的抑制性连接。

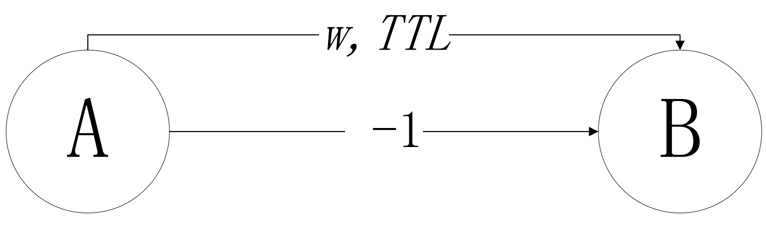

图 3　　连接的符号

2.2.6 其他特性

上面列出了使用连接表示信息之间的部分关系的方法，然而，信息之间的关系是多种多样的，连接的属性并不局限于上面几种。如果需要表示其他的关系，可以通过给连接增加对应的属性来实现。

2.3 信息表达与状态函数

连接及其传递的信号对神经元状态的影响，通过状态函数来确定。对于一个神经元，记x_n为第 n 个输入连接的输入，记w_n为第 n 个输入连接的权值，记 d 为神经元产生输出的阈值，如果刺激性信号和抑制性信号的加权和超过阈值 d，那么神经元激活，产生输出。用 S 表示神经元状态，用 1 表示激活，0 表示非激活，那么状态函数可以表示为公式 1。

公式 1

$$\mathrm{S} = \begin{cases} 1, & \sum_{i=1}^{n} w_i \times x_i > \mathrm{d} \\ 0, & \sum_{i=1}^{n} w_i \times x_i \le \mathrm{d} \end{cases}$$

基于生物神经系统的全或无定律【8】，上述神经元只采用了两种状态。为了更精确地表示信息的强弱程度，可以采用在传统神经网络中使用的激活函数φ(v)来对神经元的输出进行区间化，如公式 2 所示。

公式 2

$$\varphi(\mathrm{v}) = \varphi(\sum_{i=1}^{n} w_i \times x_i - \mathrm{d})$$

其中，激活函数可以采用传统的 sigmoid 或者 tanh 等函数。

更为一般地，在加权和超过阈值后，可以采用线性函数来描述神经元的激活的强弱程度，如 ReLu 函数。使用 sigmoid，tanh，ReLu 函数及其等效形式，不符合神经元的激活特性，但是在数学形式和信息强度的表达上面是可以解释的。

在上述操作中，需要特别注意的是，输入x_n是当前正在生效的连接所表示的信息状态。由于连接具有延时型、持续性等特征，因此，在不同的时刻，连接的输入x_n可能会不同。

2.4 简化模型

在生物神经系统中，存在全或无定律，即刺激达到神经元的阈限时，它便以最大的脉冲振幅加以反应；刺激达不到阈限时，神经元便不发生反应。

根据神经系统的全或无定律，可对本文所述的神经元做出简化。对于神经元状态，可以直接采用激活和非激活状态，用以表示信息的全或无特征；对于连接，可以将刺激性连接的权值设为 1，抑制性连接的权值设为-1，连接的权重通过修改连接的数量来实现；对于状态函数，可以省去激活函数，直接使用公式 1 的函数。更进一步地，设阈值 d＝0，公式 1 可以简化为公式 3。

公式 3

$$
S = \begin{cases} 1, & \sum_{i=1}^{n} w_i \times x_i > 0 \\ 0, & \sum_{i=1}^{n} w_i \times x_i \le 0 \end{cases}
$$

上述公式是一个阶跃函数，其意义是，在神经元处于激活状态的输入连接中，如果刺激性连接的数量大于抑制性连接的数量，那么神经元激活，信息被表达出来。

可以看出，上述神经元是本文所述的神经元的一种简化模型，将这种神经元组成的网络称为简化神经网络。基于这种神经元构建的神经网络，符合人脑同一时刻只有部分神经元活动和传统神经网络中的稀疏矩阵【9】现象。

2.5 节点类型

神经元是一种信息节点。信息节点按照功能进行分类，可以分为以下三种：

1. 输入节点：输入节点感知外部环境的输入，若外界有信息输入，那么信息节点激活。输入节点的输入连接为空，输出连接和内部节点或者输出节点连接；
2. 内部节点：内部节点是在系统内部表示信息的节点。内部节点可以接受输入节点或者内部节点的连接，也可以向输出节点或者内部节点发出连接。内部节点也可以用来表示内部信息、输入信息、输出信息节点之间的关系；
3. 输出节点：输出节点向外部产生输出信号。输出节点的输出连接为空，输入连接可以接受来自输入节点或者内部节点的输入，当节点激活时，外部设备根据节点状态做出响应。

信息节点的状态，可以受外部信息的刺激改变，也可以通过状态函数在内部信息节点和输入信息节点的作用下改变。

3 神经网络模型

3.1 网络组成

将上述神经元通过连接组合起来，可以形成一个大规模的信息表达和处理网络，这种网络就是本文所述的神经网络模型。理想情况下，神经网络由大量的神经元通过连接组合起来，每个节点都能够独立地运行，是一个大规模并行系统。

在神经元工作时，不断地根据输入连接，使用状态函数计算并更新神经元的状态，然后处理输出连接，将神经元处理一次上述操作的时间称为一个周期。在神经网络中，不同节点处理上述工作的起始时间可能相同，也可能不同。将所有神经元同步工作的网络称为同步网络，否则称为异步网络。

另外，如果网络的神经元数量、类型或者连接的数量、属性等等可以动态地改变，那么称这种网络为动态网络，否则称为静态网络。

基于本文所述的神经元构成的神经网络，其原理是基于信息关系和信息表达，并因此而形成各种逻辑功能，在形式上面不再符合冯诺依曼提出的计算机架构【10】，并且没有数据和代码的概念。当然，这并不表示本文所述的神经网络可以完全脱离传统的计算机体系结构，用于支撑神经网络运行的运算器、控制器和存储器仍然可以使用传统的软硬件模型。

3.2 运行流程

神经网络的运行，主要是各种信息节点的运行，主要流程如下：

1. 初始化网络；
2. 对于输入节点，若接收到外部输入，则通过连接向所有连接的节点发送信号；
3. 对于内部节点，根据状态函数判定节点的状态，通过连接向所有连接的节点发送信号；
4. 对于输出节点，根据状态函数判定节点的状态，输出设备对节点的状态进行响应；
5. 更新连接的相关属性；
6. 重复步骤 2 - 5。

3.3 网络仿真

在理想的神经网络中，神经元以独立的角色存在，并通过连接和其他神经元通信。但是，鉴于目前并没有专用的神经网络硬件，因此需要使用传统的计算机来进行仿真。对于一个复杂的神经网络，如人类大脑，其中的节点和连接的数量可能达到十万亿级别以上【11】，这对于系统的存储能力和并发能力要求非常高。

基于现有的计算机体系结构，下面是一种可行的处理算法。在算法的实施过程中，可以采用 OPENCL 或者 CUDA 等并发技术来提高并发能力。

1. 将所有节点存储为一个集合，节点中存储有临时的状态函数的加权和；将所有连接存储为一个集合，连接中存储连接运行时的状态，如 TTL 和延时等；
2. 遍历所有连接，对于活动连接，更新连接的目标节点的状态函数的加权和；
3. 根据外部输入更新输入节点的状态；根据状态函数更新内部节点、输出节点的状态；
4. 对于连接，更新连接的运行时的状态；对于激活的输入节点和内部节点，重置输出连接的状态；对于外部节点，外部设备根据节点状态执行对应操作；
5. 重置所有节点的状态；
6. 重复 2-5 步骤；

上面的算法根据连接的状态更新节点的状态，也可以根据节点的状态更新连接的状态。

3.4 持久节点

在设计某些神经元和连接的时候，有时候需要持久刺激性或者抑制性连接，这种连接可以采用如下方式实现。

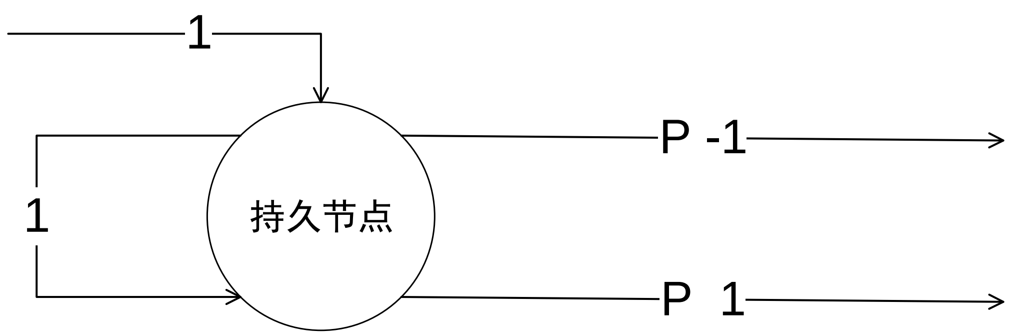

图 4　　持久节点

这种节点通过一个指向自身，且 TTL＝1 的连接，可以保证自身每个周期都处于激活状

态，因此可以提供持久性的刺激性或者抑制性信号，将这种节点称为持久节点。

为了将持久连接和普通连接区分开来，在连接前面加上 P，如图 4。在图 4 中，还有一个 TTL＝1 的输入，这个输入在神经网络初始化之后被设置成激活状态，从而确保在后续的周期之内节点能够激活；除此之外，也可以不用这个输入连接，采用设置节点的初始激活状态来进行处理。

3.5 结构和功能

本文所述的神经网络，通过改变神经元的拓扑结构和连接的属性，可以表示不同的信息和逻辑；通过控制神经元的激活状态，可以控制其他相关神经元的状态，从而控制信息的表达。

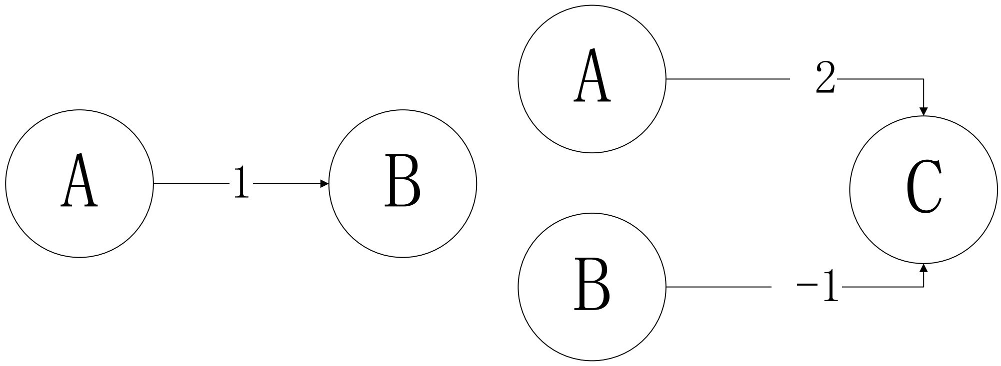

图 5

在图 5 所示的网络中，对于左边的网络，如果 A 激活，那么 B 在一个周期后收到 A 的激活信号，从而 B 激活；在右边的网络中，连接 AC 是持续两个周期的刺激性信号，连接 BC 是持续一个周期的抑制性信号，如果在某个时刻 T_1，A 激活，那么如果要使 C 激活，B 就不能在 T_1 和 T_1+1 的时刻处于激活状态。

需要特别说明的是，对于神经网络来说，神经元和和连接的组合方式千变万化，实现某种功能的神经网络结构并不唯一。

3.6 逻辑功能示例

本文提出的神经元，可以通过状态函数，根据输入连接的状态来更新自身的状态，从而表示不同的逻辑关系。基于上节提出的简化的同步神经网络，本小节对常见的逻辑操作对应的网络结构进行了分析。

3.6.1 与

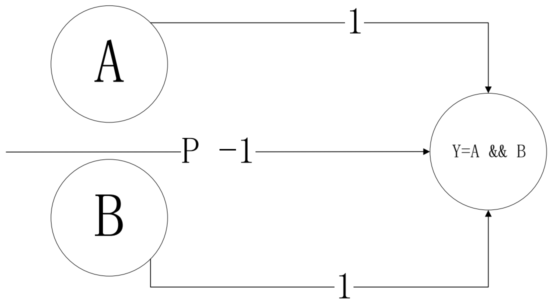

图 6　　AND 结构图

图 6 是一种 AND 逻辑结构示意图。其中，用节点 A 表示特征 A，用节点 B 表示特征 B，A、B 各有一条指向 Y 的 TTL＝1 的刺激性连接，另外，Y 还有一条指向自己的 TTL 恒为 1 的抑制性连接。用 1 表示节点的激活状态，0 表示节点非激活状态，根据状态函数，其真值表如表格 1 所示：

表格 1　AND 真值表

输入（T）		输出（T+1）
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

其中，A、B 的状态为在 T 时刻的状态，Y 的状态为在 T+1 时刻的状态。由真值表可见，Y 表示 A 和 B 的与逻辑。其意义是，当且仅当 A 和 B 均激活时，Y 激活。因此，Y 可以表示 AB 同时出现的情况。在神经网络中，特征 A、B 可以代表其他任意的信息，如像素、波形等等。

3.6.2 或

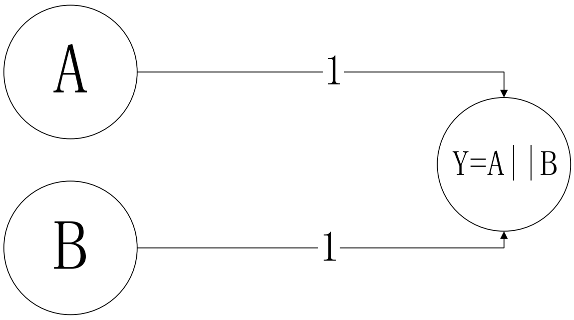

图 7　　OR 结构图

图 7 是一种 OR 逻辑结构示意图。其中，用节点 A 表示特征 A，用节点 B 表示特征 B，A、B 各有一条指向 Y 的 TTL=1 的刺激性连接，根据状态函数，其真值表如表格 2 表格 1 所示：

表格 2　OR 真值表

输入（T）		输出（T+1）
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

其中，A、B 的状态为在 T 时刻的状态，Y 的状态为在 T+1 时刻的状态。由真值表可见，Y 表示 A 和 B 的或逻辑。

3.6.3 非

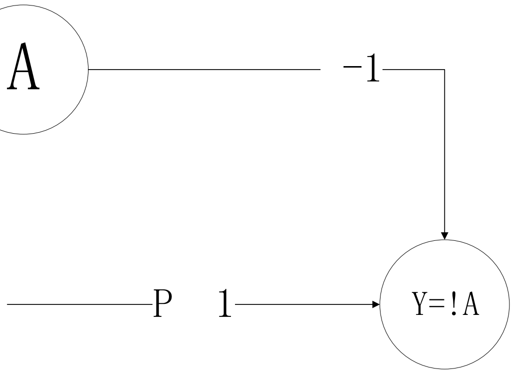

图 8　　NOT 结构图

图 8 是一种 NOT 逻辑结构示意图。其中，用节点 A 表示特征 A， A 有一条指向 Y 的

TTL=1 的抑制性连接，另外，Y 还有一条指向自己的 TTL 恒为 1 的刺激性连接。用 1 表示节点的激活状态，0 表示节点非激活状态，根据状态函数，其真值表如表格 3 所示：

表格 3 NOT 真值表

输入（T）	输出（T+1）
A	Y
0	1
1	0

其中，A 的状态为在 T 时刻的状态，Y 的状态为在 T+1 时刻的状态。由真值表可见，Y 表示 A 的非逻辑。

3.6.4 与非

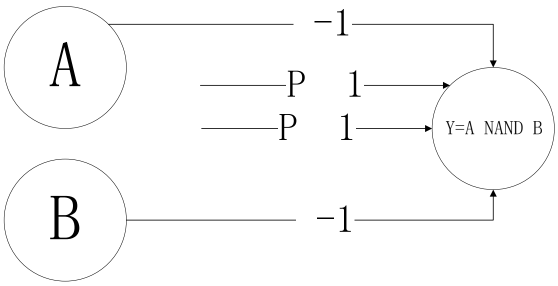

图 9 NAND 结构图

图 9 是一种 NAND 逻辑结构示意图。由于本示例基于的神经网络采用的是简化版本的同步神经网络，因此，最终的结果节点 Y 有两个持久性的刺激连接。若采用的是非简化版本的神经网络，则可以调节 Y 的输入连接的权重。根据状态函数，其真值表如表格 4 所示：

表格 4 NAND 真值表

输入（T）		输出（T+1）
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

3.6.5 或非

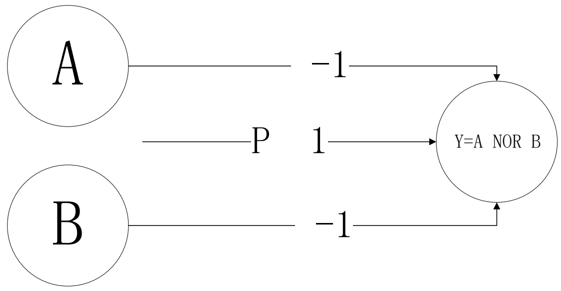

图 10　　　　NOR 结构图

图 10 是一种 NOR 逻辑结构示意图。根据状态函数，其真值表如表格 5 所示：

表格 5　NOR 真值表

输入（T）		输出（T+1）
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

3.6.6 异或

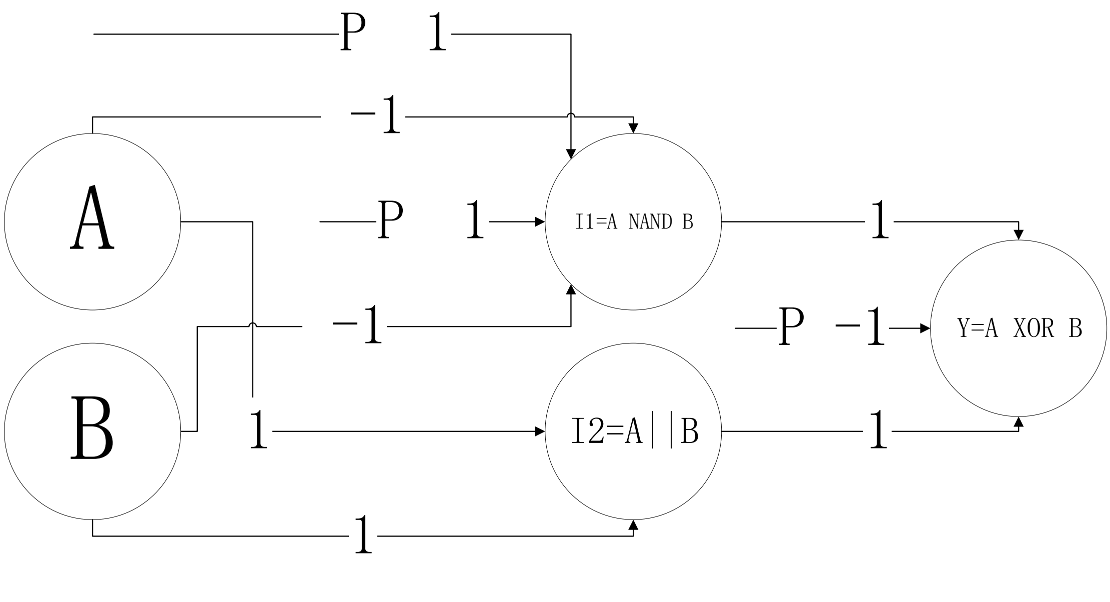

图 11　　XOR 结构图

图 11 是一种 XOR 逻辑结构示意图。不同于其他的逻辑，异或操作是基于 OR、AND 和 NAND 三种逻辑实现的，并引入了中间节点。输入节点在 T 时刻的状态，要在 T+2 时刻

才在结果节点上面显示出来。根据状态函数，其真值表如表格 6 所示：

表格 6 XOR 真值表

输入（T）		中间节点（T+1）	中间节点（T+1）	输出（T+2）
A	B	I1	I2	Y
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0

3.6.7 其他逻辑

上面列出了常见的逻辑操作对应的网络结构。其他各种信息的逻辑，参考上述方式，可以很容易地实现，此处不再赘述。

3.7 流程控制

给定一系列节点，如果这些节点代表某种操作集合，那么，控制这些节点的激活顺序，形成不同的激活序列，则可以控制操作的流程。

3.7.1 顺序

给定一个任务序列 A，B，C，顺序的序列结构如图 12 所示。

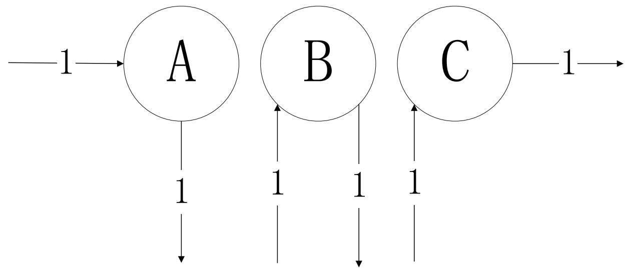

图 12 顺序结构

任务起始信号传递到 A，A 激活，向外发出信号，启动任务；当任务完成后，向 B 送激活信号，执行 B；在 B 完成后，执行 C，从而实现顺序执行的功能。

3.7.2 选择

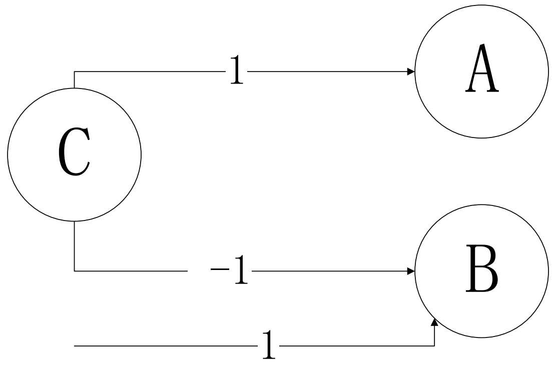

图 13　选择结构

图 13 所示是一个分支结构，C 为控制节点，若 C 激活，则执行 A 操作；若 C 失活，则执行 B 操作。

3.7.3 循环

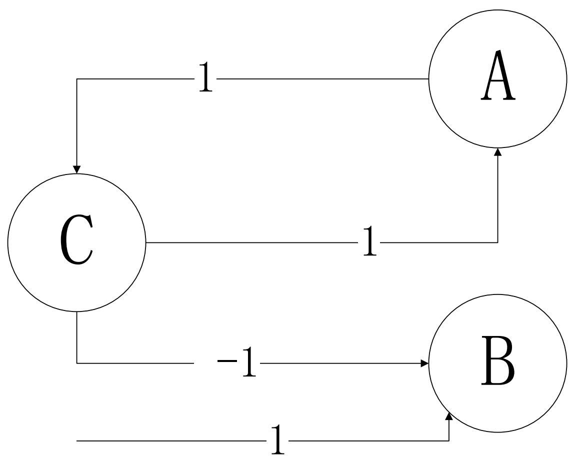

图 14　循环结构

图 14 所示是一个循环结构。C 为控制节点，A 为一个操作集合。在 C 激活时，不断地进行 A 操作；在 C 失活时，执行 B 操作。

3.8 与传统神经网络的比较

本文提出的神经网络，从信息状态及信息联系的角度入手，通过更改网络拓扑结构和连接的属性，可以对神经网络的行为进行精确设计和解释，从而避免了传统神经网络的黑盒问

题。和基于 McCulloch – Pitts 神经元的传统神经网络相比，本文所述的神经网络在如下方面具有相似性：

1. 数学模型

两种神经网络的神经元输出均可以写成$\varphi(v_{k}) = \varphi(\sum_{i=1}^{n} w_{ki}x_{i} + b_{k})$的形式。其中，$\sum_{i=1}^{n} w_{ki}x_{i}$为输入信号的加权和，$b_k$为偏置变量，$\varphi(v_{k})$为激活函数，可取 sigmoid、tanh 函数等。

2. 网络模型

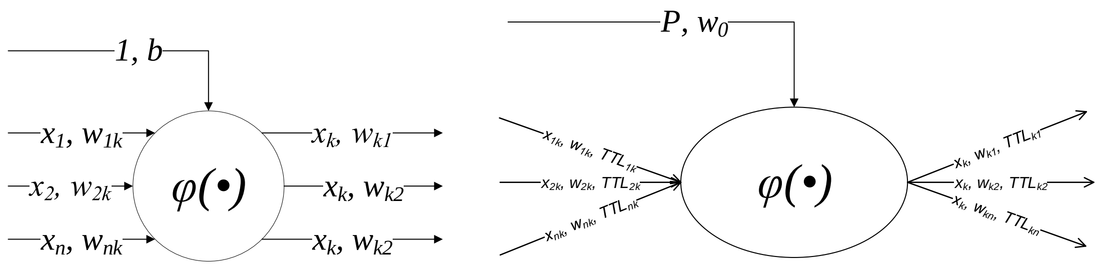

图 15 两种神经网络模型

图 15 是两种神经网络的图示。左边是基于 McCulloch – Pitts 神经元的神经网络，其偏置 b 被作为单独的输入信号为 1 的连接提取出来；右边是本文所述的神经元，其阈值 d 也可以提取出来，用一个等效的输入连接代替。两者可以采用同样的求和函数和激活函数，并且，两者均可以接受任意的神经元输入，向任意神经元输出。

因此，尽管两类神经网络的研究切入角度不同，但是在数学形式和网络结构上面，两类神经网络具有等效性，其最终应用的网络结构具有相互转化的可行性。

传统的网络模型通过训练来修改网络行为，并且网络通常分为很多层；本文所述的神经网络，主要依赖于对网络的主动设计，使之具有特定的功能，且任意的神经元之间都可以不限制地连接。除此之外，本文所述的神经网络，其中的连接还具有延时、时长、惰性等特性，用以表示信息之间的作用关系，所以，本文所述的神经网络比传统的神经网络具有更大的潜力。

基于上述分析，不难发现，传统的神经网络是本文所述的神经网络的一种特例。

多年以来，我一直认为从数学和统计的角度出发去研究人的智能是不对的，数学公式不可能代表人的意识和思想。但直到最近，我才发现从信息表达的角度和数学的角度所得到的网络模型具有惊人的相似性。这种相似性意味着，本文所述的理论可以对传统神经网络的黑盒问题提供解释，相应地传统神经网络至今所取得的成果可以为本文所述理论的正确性提供

支持。

3.9 信息存储与编码

在传统的计算机系统中，外部信息经采样和编码，按照某种格式存储在计算机中，然后被计算机进行处理。在本文的神经网络中，由于神经元可以表示某种信息，神经元本身就是存储单元；另外，神经网络还存储着连接，表示信息之间的关系；信息的状态，在神经网络运行时改变，表示信息表达的强弱。在神经网络中，各个信息节点并不是孤立的，信息节点通过连接和其他节点建立因果关系，接受其输入，形成一个静态的、多个信息节点级联的网络结构，如图 16 所示。信息之间通过连接建立联系，这就是神经网络对信息的编码方式。

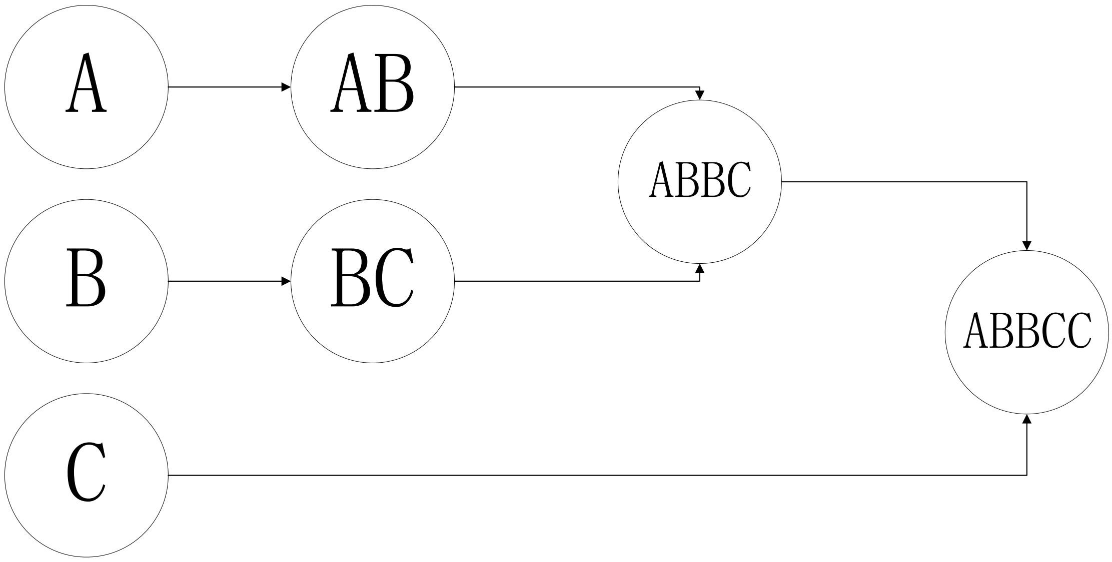

图 16　神经网络的信息编码

由于连接直接指向信息节点，从而确定信息之间的关系，并且神经网络的主要功能是用来控制信息的表达，所以，信息节点只要能够和连接对应起来即可。针对本文所述的神经网络，可直接在存储单元上面存储信息的状态，利用存储单元地址来区分不同的信息，然后将连接的目标指向这些存储单元。

3.10网络复杂度

对于一个神经网络，为了优化网络的结构，提高执行效率，特引入网络复杂度概念。网络复杂度是对网络的结构和运行的时间的一种评估方法，其计算方法如公式 4：

公式 4

$$C = a \times T \times N + b \times \sum_{i=1}^{n} TTL_i$$

其中，C 表示网络复杂度，a 表示节点影响系数，N 表示节点数量，T 表示所有连接中的信号的最大持续时间；b 表示连接影响系数，TTL_i表示第 i 个连接的 TTL。

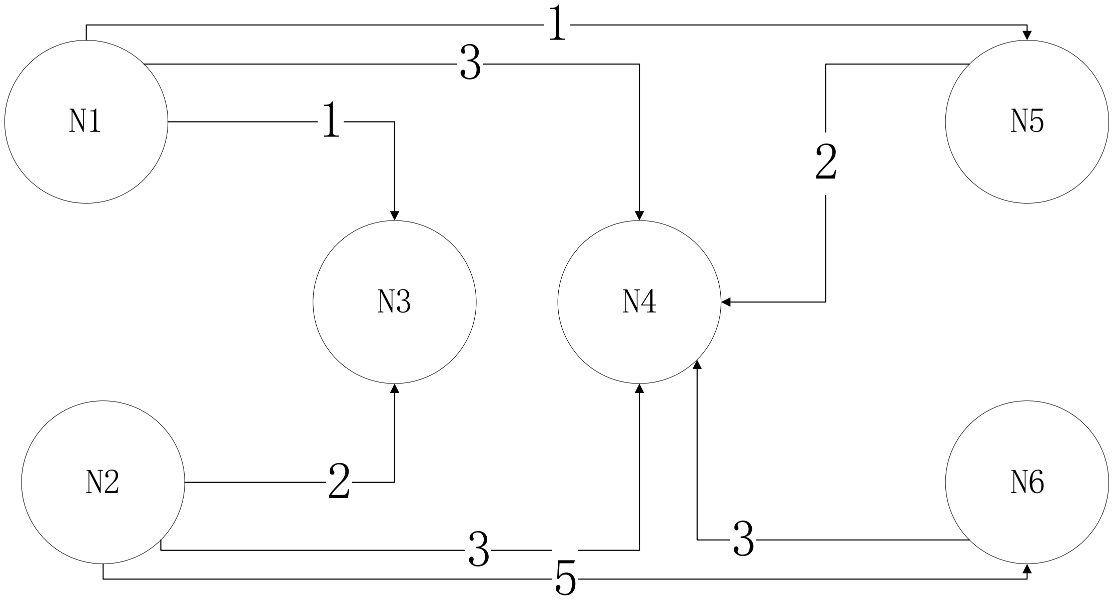

图 17　网络复杂度

取 a = b = 1，对于图 17 所示的网络，节点有 6 个，最大 TTL 为 5，复杂度 C = 5 * 6 + 20 = 50。

4 结论与展望

4.1 结论

时至今日，虽然人类在大脑的结构和功能的研究上面取得了不错的进展，但是在至关重要的意识及思维上面，却还没有得到明确的答案。传统的人工智能研究方法偏向于数学和逻辑推理，而本文则从神经元所代表的意义出发，基于信息及其之间的关系，提出了一种新的神经网络模型，希望能够为解决人工智能的相关问题提供新的思路。

尽管两类神经网络在一些功能上面具有等效性，并且具有一些通用的方法，但是，两者的设计方式和主要用途具有明显区别。传统的神经网络侧重于构建数学模型，通过学习来训

练网络参数，然后进行分类和预测；而本文所述的神经网络则是根据信息之间的因果关系提前设计好网络结构，用来实现通用的逻辑控制与信息表达功能。

本文提出的神经网络模型，从信息及信息之间的因果作用关系入手，不仅在功能上面和传统的神经网络具有等效性，而且通过设计网络的拓扑结构和连接属性，能够精确地控制各种信息的表达，使得神经网络的行为能够被设计和预测。本文提出的神经网络，其目的不是作为一种传统神经网络的替代方法，而是作为一种通用的信息处理系统，用于解决强人工智能的相关问题。

4.2 展望

本文从信息的意义及信息之间的关系的角度出发，对神经网络的部分结构和功能做了简单的介绍。但是，由于神经元之间的拓扑结构和连接属性千变万化，所能表示的信息及实现的功能无法一一枚举，本文所述的内容仅仅是冰山一角，仍然有很多的未知领域待探索。对于本文所述的神经网络，主要的工作是设计网络的拓扑结构，用于实现特定的功能，这点类似于传统的计算机科学中的数据结构和算法。本文所述的神经网络的待研究内容主要如下：

1. 信息表示：神经元可以表示各种信息及他们之间的关系，为此，需要研究各种信息在神经网络中的拓扑结构及编码方式，如一维信息、二维信息、三维信息等等；另外，为了支持感知系统和联想系统，这些信息在结构上面还要与其相连，这些相连的交互式结构还需要研究；
2. 特征提取：一般地，信息具有某些特征，如时序变化特征，空间特征，强度特征等等，结合神经元之间的空间关系及连接的属性，这些特征可以使用神经网络提取并使用神经元表示出来；
3. 逻辑表示：如上述逻辑功能示例中所示，研究如何使用神经网络表示各种逻辑关系；
4. 流程控制：给定一系列操作，每个操作由神经元的状态控制，那么，通过更改神经元的激活顺序，就可以实现不同的控制结构；
5. 感知和检索：对于一个神经网络来说，感知到某种信息意味着对应的信息节点处于激活状态。另外，对于人来说，可以将注意力集中于某些信息，而不是同时接受到的全部信息，因此，应该存在一个感知层，可以从所有激活的信息节点中在对应的感知层内选择性地激活某些信息节点；同时，也应该存在一个检索网络，用于根据感知层的输出来控

制其他信息节点在感知层上面的激活状态；结合语言系统，感知网络，系统外部状态及内部相关网络状态的输出，可以实现一个意识系统；

6. 任务调度：给定一个任务序列，对这些任务进行执行、监测和调度；
7. 情感：可以使用一组神经元表示一个一维的信息，这个一维信息中激活的神经元可以受到其他神经元输入而改变，而这种神经元的改变情况又可以向其他神经元发送信号，从而可以作为一种情感的实现基础（针对简化神经网络模型，非简化神经网络的信息状态是一个区间，可直接使用一个神经元）；
8. 语言和文字：神经元的激活状态可以具有持续时间，而这些状态可以被其他的神经元捕获到，进而通过输出节点向外部环境输出；另外，外部的信息也可以通过输入节点输入，引起内部神经元维持某种状态。因此，可以设计一种网络结构，用于负责将神经网络的内部状态向外部输出，或者根据外部信息来重构其在神经网络中的表示状态，这种网络结构可以用作语言和文字的实现基础；
9. 动态神经网络：所谓动态神经网络，即支持节点或连接动态修改的网络，动态神经网络是学习、记忆等高级活动的基础；如何修改，在什么时候修改，修改哪些目标，是在先天规则还是后天意识下修改等等，这些都是需要研究的内容；
10. 学习：本文所述的神经网络，可以通过提前设计来确定行为，也可以通过动态神经网络的接口在运行时修改行为，其中后者对于强人工智能至关重要。这儿所说的学习，是指在网络运行的过程中，通过动态神经网络的接口，在先天被动功能或者后天主动意识的控制下，按照某种规则对神经网络进行动态修改的过程；
11. 记忆：针对动态神经网络，结合语言、感知等网络结构，研究如何在连接的有效性的属性的支持下，建立一个支持长短期记忆的网络结构；
12. 联想和想象：通过特别设定的网络结构，可以使得不同的信息节点之间存在连通性，这种连通性可以根据感知层的状态，在索引网络的控制下输出到感知层，然后结合神经网络的学习能力，就可以建立新的节点或者新的连接，从而为神经网络提供创造力。

上面列举了本文所述的神经网络的主要待研究内容并提供了一些思路。当然，各个研究方向并不是孤立的，每个方向都会或多或少地和其他方向有关联。只有将这些功能有机地整合到神经网络中，才能够实现一个真正的人工智能系统。

5 参考文献

1 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012: 1097-1105.
2 Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.
3 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.
4 Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533.
5 McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.
6 Hebb D O. The organization of behavior: A neuropsychological theory[J]. 2005.
7 Miller, Michael SP. "The Neural Proposition: Structures for Cognitive Systems." AAAI Spring Symposium: Creativity and (Early) Cognitive Development. 2013.
8 Nicholls J G, Martin A R, Wallace B G, et al. From neuron to brain[M]. Sunderland, MA: Sinauer Associates, 2001.
9 Olshausen B A, Field D J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images[J]. Nature, 1996, 381(6583): 607.
10 Von Neumann, J. (1993). First Draft of a Report on the EDVAC. IEEE Annals of the History of Computing, 15(4), 27-75.
11 Haykin S S. Neural networks and learning machines[M]. Upper Saddle River, NJ, USA:: Pearson, 2009.