Feedforward Neural Network Reconstructed from High-order Quantum Systems

1st Junwei Zhang
College of Intelligence and Computing,
Tianjin University,
Tianjin, China.
Email: junwei@tju.edu.cn

2nd Zhao Li (✉)
Zhejiang University, Zhejiang, China.
Hangzhou Yugu Technology Co., Ltd., China.
Link2Do Technology Ltd., China.
Email: zhao_li@zju.edu.cn

3rd Hao Peng
Beihang University,
Beihang, China.
Email: penghao@buaa.edu.cn

4th Ming Li
The Key Laboratory of Intelligent Education Technology and
Application of Zhejiang Province, Zhejiang Normal University,
Zhejiang, China.
Email: mingli@zjnu.edu.cn

5th Xiaofen Wang
Shijiazhuang Railway University,
Shijiazhuang, China.
Email: wangxiaofen@std.edu.cn

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892720

***Abstract*—Neural Networks (NNs) are widely used because of their superior feature extraction capabilities, among which Feedforward Neural Network (FNN) is used as the basic model for theoretical research. Recently, Quantum Neural Networks (QNNs) based on quantum mechanics have received extensive attention due to their ability to mine quantum correlations and parallel computing. Since two classical bits are required to simulate one qubit (i.e., quantum bit) on a classical computer, it brings challenges for simulating complex quantum operations or building large-scale QNNs on a classical computer. Hardy et al. extended the classical and quantum probability theories to the Generalized Probability Theory (GPT), so it is possible to construct high-order quantum systems. This paper regards the entire feature extraction and integration process of FNN as the evolution process of the high-order quantum system, and then leverages quantum coherence to describe the complex relationship between the features extracted by each layer of the network model. Intuitively, we reconstruct FNN to change the general vector processed by each layer into the state vector of the high-order quantum system. The experimental results on four mainstream datasets show that FNN reconstructed from the high-order quantum system is significantly better than the classical counterpart.**

***Index Terms*—Feedforward Neural Network, Quantum Neural Networks, Generalized Probability Theory**

I. Introduction

Neural Networks (NNs) are widely studied and applied because of their efficient feature extraction and integration capabilities, and the Feedforward Neural Network (FNN) is used as the basic model for theoretical research.

Neural network models based on quantum mechanics theory [1] are called **Quantum Neural Networks** (QNNs) [2], and they have attracted extensive attention from researchers because of their ability to mine strong correlations between features [3], [4], such as quantum entanglement or quantum coherence. In addition, because quantum mechanics allows the existence of superposition states, the computational model based on quantum mechanics has the ability to perform parallel operations, which makes the quantum model significantly better than the classical model in terms of computational efficiency [5].

For a long time, QNNs have been widely studied. In 2000, Ventura and Martinez [6] proposed a quantum implementation of the associative memory model. In 2003, a qubit (i.e., quantum bit) neural network was introduced by Kouda et al. [7]. Subsequently, a number of quantum neural network models, such as quantum convolutional neural network [8], quantum recurrent neural network [9], quantum Hopfield neural network [10], and quantum graph neural network [11], were proposed, indicating that the neural network models inspired by quantum theory are widely accepted and recognized by researchers.

However, since two classical bits are required to simulate one qubit on a classical computer, it is difficult to construct large-scale quantum systems on a classical computer. And the compound operation of quantum systems on a classical computer is a tensor operation, that is, 2^N classical bits are needed to construct a compound system of N qubits [5], so it is not suitable for building large-scale quantum neural network models like classical deep learning models. In addition, quantum coherence (such as the widely mentioned quantum entanglement) is an important quantum resource [12], [13], and large-scale quantum systems will help to reproduce and study the strong correlations revealed by quantum coherence. **Therefore, how to effectively utilize the limited computing resources on classical computers to simulate large-scale quantum systems is an fundamental work to advance the research of QNNs.**

Hardy et al. [14], [15] extended the classical probability theory (namely, Kolmogorov probability) and quantum probability theory (namely, mathematical principles of quantum mechanics) to the **Generalized Probability Theory** (GPT), so it is possible to construct **high-order quantum systems**. That is to say, based on the GPT, we can construct high-order general bits, similar to qubits, which conform to all the

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

properties of quantum mechanics. In this paper, we regard the entire feature extraction and integration process of FNN as the evolution process of the high-order quantum system, and then leverage quantum coherence to describe the complex relationship between the features extracted by each layer of the network model. Intuitively, we reconstruct the FNN through the GPT to change the general vector processed by each layer of the network model into the state vector of the high-order quantum system. The experimental results on four mainstream datasets show that the FNN reconstructed from the high-order elementary quantum system is significantly better than the classical counterpart.

Our contributions can be summarized as follows:

- The feedforward neural network is reconstructed based on the high-order quantum system proposed by Hardy et al. [14], [15], which explores a possible method for simulating large-scale quantum systems on classical computers.
- A method is proposed to describe the strength of the correlation between features in the network model by using quantum coherence, which explores a quantitative method for the learning ability of each layer of the neural network.
- Our model performs experimental validation on four classical datasets, which outperforms its classical counterparts.

II. Related Work

In the early days, Menneer et al. [16] proposed a hypothetical neural network model called a quantum-inspired neural network in 1995. Then the quantum revolving gate is introduced into the back-propagation network, and a new quantum-inspired neural network model is proposed. The concept of quantum cellular neural network was proposed by Toth et al. [17] in 1996. They used coupled quantum dot cells in this architecture to build a simulated cellular neural network. Matsui et al. [18] proposed a qubit neuron model which shows quantum learning abilities in 2000. This qubit neuron model has a high efficiency in solving problems like data compression. Kouda et al. [7] also proposed qubit neural network where the interaction between the states of the neurons with other neurons are based on the laws of quantum mechanics. The above model is a preliminary exploration of QNNs. Although the model is not complicated, it is of great significance to future research work. See also Refs. [19]–[21].

In recent years, Rebentrost et al. [10] coded the Hopfield network into the amplitude of a quantum state to realize the storage of large exponential networks in polynomial qubits in 2018. Cong et al. [8] used variable-parameter qubits to build larger-scale quantum systems to implement QNN models and reproduce the functions of classical convolutional neural network in 2019. Bausch [9] constructed a quantum recurrent neural network in which the neurons of the network model are constructed from parameterized qubits in 2020.

Recently, due to the improvement of the computing power and storage capacity of computers, the research work of QNNs has made great progress, and it has been possible to construct larger-scale and complex network models. However, because these QNN models are based on classical qubits, the scale of the models and the difficulty of solving problems still cannot match the classical counterparts. Based on the generalized probability theory proposed by Hardy et al. [14], [15], we will use its high-order elementary quantum system to construct a quantum counterpart that can match the classical neural network model in scale, and solve the difficulties in constructing a larger-scale quantum network model on a classical computer.

III. GPT: Generalized Probability Theory

Following the work of Hardy et al. [14], [15], [22]–[24], we will first derive GPT based on **the instrumentalist framework** [25], [26], that is, the entire cycle of the system will be completed by the preparation, transformation, and measurement devices. The preparation device prepares the system in a certain state, which has a set of switches for changing the state of the generated system. After the preparation device, the system passes through a transformation device, which has a set of switches for varying the transformation applied to the system. The number of transformation devices can be added more than one according to practical needs. Finally, the system enters the measurement device, which also has a set of switches to help the experimenter choose different measurement settings.

Moreover, two natural numbers, d and N, are used to uniformly describe classical, quantum, and general probability theories.

- d is defined as the degree of freedom of the system, that is, the minimum number of real parameters required to fully describe a system.
- N is defined as the maximum number of distinguishable states of the system.

Accordingly, we can establish the functional dependence $d(N)$ between these two natural numbers according to different theories.

- In classical probability theory, there is a linear dependence, $d = N - 1$, which means that a classical bit ($N = 2$) only needs 1 real parameter to describe.
- In quantum probability theory, there is a quadratic dependence, $d = N^2 - 1$, which means that a qubit ($N = 2$) needs 3 real parameters to completely describe.
- For a more general high-order theory, namely GPT, the generalized bit ($N = 2$) will be described as $d = N^r - 1$ where $r \in \mathbb{N}$.

The graphical description is shown in Fig. 1.

A. Formal definition of preparation, transformation, and measurement devices

For a physical system, it is not necessary to provide an exhaustive list of all conceivable measurements, but only a minimal subset of them. We refer to this subset as **the fiducial set**. Therefore, the state of the system will be defined by a list of probabilities corresponding to the fiducial set, which is

$$\mathbf{p} = [p_1, ..., p_d] \tag{1}$$

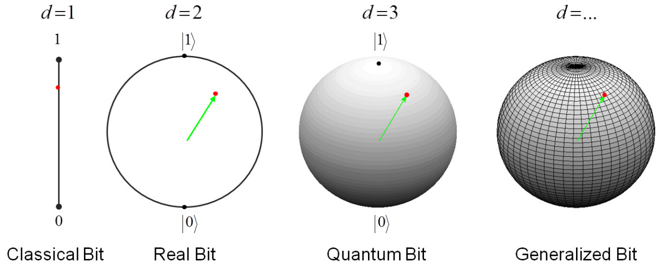

Fig. 1. A classical bit with one parameter, a real bit with two real parameters, a qubit (or quantum bit) with three real parameters, and a generalized bit for which d real parameters are needed to specify the state. The real bit is the case of the qubit in the real number field.

where d is the degree of freedom of the system defined above. From this we can know that the degree of freedom of the system, namely d, is actually the minimum number of measurements required to determine a system.

We further subdivide a physical system into pure and mixed states. If a state is a pure state, it cannot be expressed as a convex mixed form of other states. If a state is not a pure state, it must be a mixed state. For example, if a mixed state is prepared with the probability of λ as $\mathbf{p}_1$ and the probability of $1-\lambda$ as $\mathbf{p}_2$, the mixed state is defined as

$$\mathbf{p} = \lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2. \tag{2}$$

Preparation Device: The output of the preparation device is a list of probabilities of any conceivable measurement of the system, including two forms of pure and mixed states.

Transformation Device: If the system in state $\mathbf{p}$ is input to the transformation device, its state will be transformed to a new state $O(\mathbf{p})$. The transformation $O(\cdot)$ is a linear function, so it is necessary to ensure the linear structure of the mixed state. For example, consider the mixed state $\mathbf{p}$ which is generated by preparing state $\mathbf{p}_1$ with probability λ and $\mathbf{p}_2$ with probability $1-\lambda$. Then, in each single run, either $\mathbf{p}_1$ or $\mathbf{p}_2$ is transformed and thus one has:

$$\begin{aligned} O(\mathbf{p}) &= O(\lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2)) & (3)\\ &= \lambda O(\mathbf{p}_1) + (1-\lambda) O(\mathbf{p}_2). & (4)\end{aligned}$$

Measurement Device: Like the transformation device, the measurement $M(\cdot)$ cannot change the mixing coefficients of the mixed state $\mathbf{p}$ and therefore the measured probability is a linear function:

$$\begin{aligned} M(\mathbf{p}) &= M(\lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2)) & (5)\\ &= \lambda M(\mathbf{p}_1) + (1-\lambda) M(\mathbf{p}_2). & (6)\end{aligned}$$

Given a measurement setting, the outcome probabiltiy P_{meas} can be computed by $M(\cdot)$,

$$P_{meas} = M(\mathbf{p}). \tag{7}$$

B. Example: high-order elementary quantum system

An elementary system, namely system of information capacity of 1 bit, has two distinguishable outcomes which can be identified by a pair of basis states $\{p_i, p_i^{\perp}\}, i \in \{1, 2, ..., d\}$. The state is specified by d probabilities for d fiducial measurements,

$$\mathbf{p} = [p_1, ..., p_d] \tag{8}$$

where p_i is probability for a particular outcome of the i-th fiducial measurement. The dependent probabilties for the opposite outcomes, $p_i^{\perp} = 1 - p_i$, are omitted in the state description. We usually replace the probability vector $\mathbf{p}$ with the geometric representation of **the Bloch sphere** [27],

$$\mathbf{x} = [x_1, ..., x_d] \tag{9}$$

where $x_i = 2p_i - 1$. The mapping between the two different representations is an linear map and therefore preserves the structure of the mixture

$$\lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2 \mapsto \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2. \tag{10}$$

Therefore, **the Bloch vector** of the totally mixed state is the zero-vector $\mathbf{x} = \vec{0}$.

The transformation $O(\cdot)$ does not change the totaly mixed state, hence $O(\vec{0}) = \vec{0}$. This condition together with the linearity condition Eq. (3) implies that any transformation can be represented by a $d \times d$ real matrix. The measured probability is given by the formula:

$$P_{meas} = M(\mathbf{x}) = \frac{1}{2}(1 + \mathbf{r}^T \mathbf{x}). \tag{11}$$

$\mathbf{r}$ represents the state of the system to be measured under a given measurement setting. For example, $\mathbf{r} = [1, 0, 0, ...]$ is to measure the first state of the fiducial set.

C. The superiority of GPT

Hardy [14] pointed out that classical probability theory is a special case of quantum one, and he also derived a more general probability theory, namely GPT. Quantum probability theory and GPT are not only generalizations of classical probability theory, but also reveal to some extent the non-classical features of nature, that is, the quantum features of the micro world. To enable readers to better accept the universality or superiority of GPT, here we review Cabello's work [22], [24].

To verify the quantum features revealed by quantum theory, some inequalities derived from classical theories have been proposed, such as the Clauser-Horne-Shimony-Holt (CHSH) inequality [28] to verify **quantum nonlocality** [29],

$$|E(a,b) - E(a,b') + E(a',b) + E(a',b')| \leq 2 \tag{12}$$

where $a, a' \in \{+1, -1\}$ are detector settings on side A, $b, b' \in \{+1, -1\}$ on side B, and $E(\cdot,\cdot)$ denote expectation value, and the Klyachko-Can-Binicioğlu-Shumovsky (KCBS) inequality [30] to verify **quantum contextuality** [31],

$$\sum_{i=0}^{4} \langle x_i x_{i+1} \rangle \geq -3 \tag{13}$$

where

$$\langle x_i x_j \rangle = \sum_{x_i, x_j = \pm 1} x_i x_j P(x_i, x_j). \tag{14}$$

The correlations in the CHSH and KCBS inequalities are expressed as a linear combination of probabilities of a subset of events of the corresponding experiment. The fact that the sum of probabilities of outcomes of a test is 1 can be used to express these correlations as a positive linear combination of probabilities of events e_i,

$$S = \sum_i w_i P(e_i) \tag{15}$$

with $w_i > 0$. Therefore, the CHSH and KCBS inequalities can be expressed, respectively, as

$$S_{CHSH} = \sum_{i=0}^{3} \sum_{a,b} P(a,b|i,i+1) \overset{LHV}{\leq} 3 \tag{16}$$

where $a, b \in \{0,1\}$ with $a = b$ if $i \neq 2$ and $a \neq b$ if $i = 2$, the sum in $i+1$ is taken modulo 4 and LHV denote local hidden variable theory (namely classical theory), and

$$S_{KCBS} = \sum_{i=0}^{4} P(0,1|i,i+1) \overset{NCHV}{\leq} 2 \tag{17}$$

where the sum in $i+1$ is taken modulo 5 and NCHV denote noncontextual hidden variable theory (namely classical theory). Although in these inequalities all probabilities have weight 1, each probability $P(e_i)$ may have a different weight w_i. The inequalities can be described as a vertex-weighted graph (G, w), that is, a graph G with a vertex set V and a weight assignment $w : V \to \mathbb{R}^+$.

From classical theory (e.g., LHV and NCHV), Quantum Theory (QT), and GPT, we can get different degrees of violations for the inequalities, that is, the following inequality relationship

$$S \overset{LHV,NCHV}{\leq} \alpha(G,w) \overset{QT}{\leq} \vartheta(G,w) \overset{GPT}{\leq} \beta(G,w) \tag{18}$$

where $\alpha(G, w)$ is the independence number of (G, w) [32], $\vartheta(G, w)$ is the Lovász number of (G, w) [32]–[34], and $\beta(G, w)$ is the fractional packing number of (G, w) [32]. The detailed proof of this conclusion can be found in Refs. [22], [24].

Based on the above conclusions, we can get the understanding that GPT has the same ability to reveal non-classical features of things like quantum theory, and it perfectly generalizes classical and quantum probability theories. Therefore, it has the ability to describe the classic and non-classical characteristics of things, and can be used as a research tool with universal characteristics.

IV. The Network Model under GPT

Broadly speaking, each layer of the FNN completes the vector-to-vector mapping and the FNN as a whole describes the entire transformation process of the vector in a fine-grained manner. In fact, the elementary system of GPT is a vector built on or within the Bloch sphere, that is, the pure state is a vector on the sphere and the mixed state is a vector in the sphere, which has natural advantages in vector operations. This not only brings the possibility of reconstructing the FNN in GPT, but also gives realistic physical meaning to the transformation process of vectors. In this section, we will leverage the elementary system of GPT to reconstruct the FNN, called the network model under GPT.

A. The preparation process

Here we use the geometric representation of the Bloch sphere to describe the elementary system, which means that the preparation process of the elementary system will output a state vector of the Bloch sphere, $\mathbf{x} = [x_1, x_2, ..., x_d]$. We assume that the preparation device only generates pure state, i.e., $\|\mathbf{x}\| = 1$. In fact, this assumption is not harsh. In physics experiments, we can create an ideal elementary system by adding some specific restrictions.

Since the preparation device needs a set of switches to change the generated state vector, e.g., $\mathbf{v} = [v_1, v_2, ..., v_d]$, we can define the preparation process as

$$\mathbf{x} = Normal(\mathbf{v}) = \frac{1}{\mathcal{N}}[v_1, v_2, ..., v_d] \tag{19}$$

where $\mathcal{N} = \sqrt{\sum_{i=1}^{d} v_i^2}$. A more pure understanding is that the input of the preparation device is $\mathbf{v}$ and the output is $\mathbf{x}$, and the function of the preparation process is $Normal(\cdot)$, that is, a normalization operation.

B. The transformation process

The input of the transformation device can be either the output of the preparation device, that is, the pure state, $\|\mathbf{x}\| = 1$, or the output of the previous transformation device, that is, the pure or mixed states, $\|\mathbf{x}\| \leq 1$. In other words, the transformation process can contain more than one transformation device. For a given input state $\mathbf{x}$, a strict rotation transformation $SO(\cdot)$ can be used to represent the operation of the transformation device,

$$\mathbf{x}' = SO(\mathbf{x}). \tag{20}$$

A more general formal representation can be described as

$$\mathbf{x}' = Normal(\mathbf{W}\mathbf{x}) \tag{21}$$

where $Normal(\cdot)$ is the normalization function defined above, and $\mathbf{W}$ is a real number matrix whose dimensions are determined by the input $\mathbf{x}$ and the output $\mathbf{x}'$. Since the output of the transformation device can be a mixed state, it is represented as a convex combination of pure states,

$$\mathbf{x}' = \sum_{i=1}^{n} \lambda_i \mathbf{x}'_i = \sum_{i=1}^{n} \lambda_i Normal(\mathbf{W}_i \mathbf{x}) \tag{22}$$

where λ_i is the mixing coefficient, i.e., $\lambda_i \geq 0$ and $\sum_{i=1}^{n} \lambda_i = 1$.

C. The measurement process

The input of the measuring device is a pure or mixed state vector $\mathbf{x}$, and the output is a probability vector P about the fiducial set. According to GPT, there is a linear correspondence between the geometric representation of the Bloch sphere of the state vector and the probability representation about the fiducial set, that is, Eq. (11). Therefore, we can define the function of the measurement device as

$$P_{meas} = \frac{1}{2}(\mathbf{1} + \mathbf{r} \odot \mathbf{x}), \tag{23}$$

where $\mathbf{1}$ is a vector of all ones, $\odot$ is bitwise multiplication, and $\mathbf{r}$ can be considered as a parameter that needs to be input for the switch of the measurement device. Since in specific applications, we only need to observe a limited number of states of the system to complete the task, so we can set the measurement vector according to practical needs. For example, $\mathbf{r} = [0, ..., 0, 1, 1, 1, 0]$ refers to measuring the status of the fourth, third, and second from the bottom of the fiducial set.

What needs to be emphasized here is that because GPT is based on complementary measurable principle, that is, for $\{p_i, p_i^{\perp}\}, i \in \{1, ..., d\}$, $p_i + p_i^{\perp} = 1$, so

$$Sum(P_{meas}) = \sum_{i=1}^{d} p_i \neq 1. \tag{24}$$

D. The loss function and optimization method

For the optimization method, this paper cannot propose a better or more suitable optimization method for Hilbert space, so readers can choose the classical optimization method to optimize the parameters of the network model according to their needs. In this paper, the cross-entropy loss function is used as the loss function, and *Adam* [35], which is widely adopted, is used as an optimization method.

V. Experiments and Analysis

A. Datasets and evaluation metrics

The experiments were conducted on four commonly used classification datasets, namely MNIST, Fashion-MNIST [36], Cifar-10, and Cifar-100 [37]. These data sets are often used for testing new models or methods. The statistical information of the datasets is shown in Table I. Here we leverage the accuracy rate, which is often selected in the deep learning model, as the evaluation metric.

TABLE I
Dataset Statistics: The number of training set, validation set and test set and the number of categories are shown in the table. The ratio of the training set to the validation set is 8 : 2. The dataset Fashion-MNIST is abbreviated as F-MNIST.

Dataset	Training	Validation	Test	Categories
MNIST	48000	12000	10000	10
F-MNIST	48000	12000	10000	10
Cifar-10	40000	10000	10000	10
Cifar-100	40000	10000	10000	100

B. Reconstructed FNN vs. traditional FNN

Before conducting experimental verification, we first formally analyze the similarities and differences between the Network Model reconstructed from GPT (called NM-GPT) and the FNN.

1) The preparation process vs. the input layer: The input layer of the FNN receives the data $\mathbf{x}$, and does not do any processing, $\mathbf{x}' = \mathbf{x}$, while the preparation process of the NM-GPT adds a normalization constraint Eq. 19,

$$\mathbf{x}' = Normal(\mathbf{x}). \tag{25}$$

2) The transformation process vs. the hidden layer: The FNN adds an activation function $\sigma(\cdot)$ to each neuron in the hidden layer, which is a mapping function that is independently adjusted from other neurons,

$$\mathbf{x}' = \sigma(\mathbf{W}\mathbf{x} + b); \tag{26}$$

while the NM-GPT adds a normalized constraint to the entire layer, that is, the neurons in the hidden layer are not independent of each other, and there is no bias term b,

$$\mathbf{x}' = Normal(\mathbf{W}\mathbf{x}). \tag{27}$$

Moreover, for complex situations, each hidden layer of the NM-GPT is equivalent to a linear combination of multiple hidden layers,

$$\mathbf{x}' = \sum_{i=1}^{n} \lambda_i Normal(\mathbf{W}_i \mathbf{x}'), \tag{28}$$

that is, a linear combination of multiple pure states into a mixed state.

3) The measurement process vs. the output layer: The output layer of the FNN is similar to the hidden layer, but the added activation function is mostly $Softmax(\cdot)$,

$$\mathbf{x}' = Softmax(\mathbf{W}\mathbf{x} + b); \tag{29}$$

while the NM-GPT is a simple linear mapping,

$$\mathbf{x}' = \frac{1}{2}(\mathbf{1} + \mathbf{r} \odot \mathbf{x}). \tag{30}$$

4) Experimental verification: From the mathematical analysis, it can be seen that the NM-GPT and the FNN are very similar. In order to effectively verify the performance of the NM-GPT, we first construct a control group model and then add modified parts to the control group model to obtain the experimental group model.

Control Group Model: On the datasets MNIST and Fashion-MNIST, we use a five-layer FNN as the control group model. The number of neurons in the input layer is determined by the number of features in the dataset; the second and third layers have 512 neurons; the fourth layer has 128 neurons; the number of neurons in the output layer is determined by the number of categories in the dataset. The activation function is $ReLU$ (Rectified Linear Unit) uniformly, the optimizer is $Adam$ [38], and the loss function is cross-entropy loss function [39]. This model is called the basic model, and subsequent changes will focus on this structure.

On the datasets Cifar-10 and Cifar-100, we add a three-layer convolutional layer to the basic model to cope with the complex structure of the datasets. The number of convolution kernels of the three convolutional layers are 32, 64, and 64 respectively. The size of the convolution kernels is 3×3. The

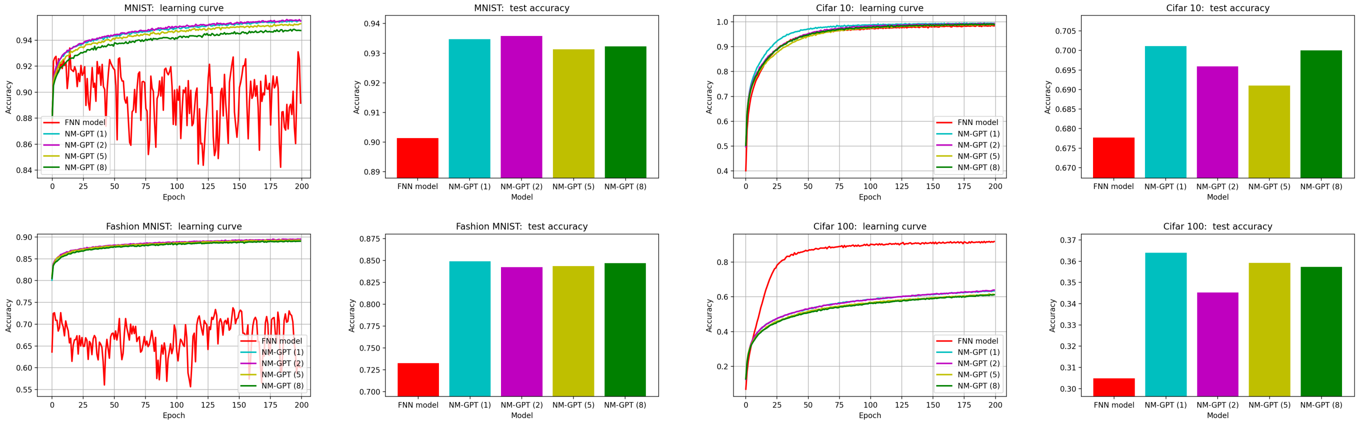

Fig. 2. Experimental results under MNIST, Fashion-MNIST, Cifar-10, and Cifar-100. The number in parentheses indicates the number of pure states that constitute the mixed state, which is n in Eq. 28.

pooling layers are added between the convolutional layers, and the size of the pooling kernels is 2×2. The activation function used by the convolutional layer is $ReLU$.

Experimental Group Model: The experimental group model is obtained by modifying the basic model of the control group model. Here we propose two modifications, one is to reconstruct the FNN with a pure state quantum system, and the other is to reconstruct the FNN with a mixed state quantum system. In the pure state, the number of parameters in NM-GPT will be exactly the same as the number of traditional FNN. In the illustrations of the figures and tables, we use NM-GPT(1) to indicate the pure state and NM-GPT(x) to indicate the mixed state, where x indicates the mixed state composed of x pure states, that is, n in Eq. 28. In the experiment, the values of x are 1, 2, 4, and 8. The statistics of the number of parameters of each model are shown in Table II.

TABLE II
THE STATISTICAL INFORMATION OF THE TOTAL PARAMETERS OF DIFFERENT MODELS UNDER MNIST, FASHION-MNIST (F-MNIST), CIFAR-10, AND CIFAR-100 DATASETS. NM-GPT IS ABBREVIATED AS NGPT. THE NUMBER IN PARENTHESES INDICATES THE NUMBER OF PURE STATES THAT CONSTITUTE THE MIXED STATE, WHICH IS n IN EQ. 28.

Model	MNIST	F-MNIST	Cifar-10	Cifar-100
FNN	798,474	731,530	910,730	529,124
NGPT (1)	798,474	731,530	910,730	529,124
NGPT (2)	1,594,384	1,461,776	1,763,856	989,034
NGPT (5)	3,982,105	3,652,505	4,323,225	2,368,755
NGPT (8)	6,369,826	5,843,234	6,882,594	3,748,476

Hyper-parameter settings: The models of the experimental group and the control group adopt the same hyper-parameter settings, the learning rate is 0.0001, the batch is 32, and the epoch is 200. Other hyper-parameters have been given during the model building process. The parameters not mentioned use the default values of the framework.

Experimental results: The experimental results under MNIST, Fashion-MNIST, Cifar-10, and Cifar-100 datasets are shown in Fig. 2. As shown in the figure, the performance of the NM-GPT is significantly better than the FNN and the NM-GPT in the pure state, namely NM-GPT(1), is better than others. The explanation we give is that NM-GPT in the mixed state has more parameters and is more difficult to learn. From mathematical analysis and experimental verification, it can be known that the traditional FNN is not the best performing network structure, but has a simple design, fewer parameters, and easy parameter learning process; the NM-GPT has a complex design, more parameters, and difficult parameter learning process, but it has better physical interpretability and stronger learning ability.

C. Ablation experiments

For readers, there will be the question about whether the superior performance of the NM-GPT will be fully benefited from the normalization constraint $normal(\cdot)$ rather than the overall design. We use ablation experiments to answer this question.

Under Cifar-10 and Cifar-100 datasets, we use different activation functions on the traditional FNN to get the experimental results, and only use $normal(\cdot)$ as the activation function on the traditional FNN to get the experimental results. The results are shown in Fig. 3. It can be seen from the diagram that the best performance is not the result of $normal(\cdot)$ as the activation function, which means that just using $normal(\cdot)$ is not the reason for the superior performance of the NM-GPT.

VI. NECESSITY OF GPT FOR NNS

Regarding the necessity of introducing the physical model or physical meaning of NNs (mainly referring to feedforward NNs), here is our analysis. NNs are a mathematical model constructed by imitating the human brain nervous system, that is, the form of linear mapping plus activation function, which can well simulate the operating mechanism of the nervous system. Since this model is a mathematical model driven by data, it is often called a “black box” model, which lacks clear physical principles.

Based on GPT, this paper deconstructs the overall operation process of NNs into the transformation process of the state

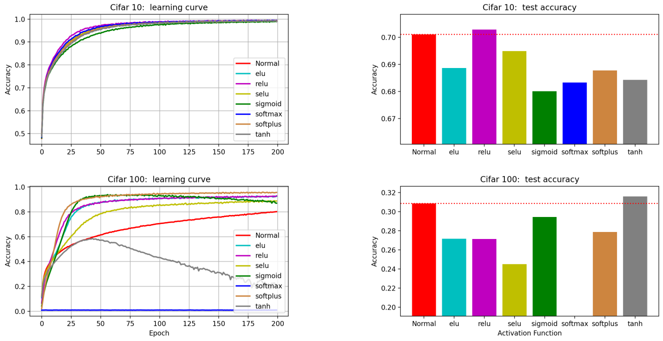

Fig. 3. The experimental results of the traditional FNN model under Cifar-10 and Cifar-100 datasets using different activation functions.

vector in Bloch representation, so that the output vector of each layer in NNs has a clear physical meaning, that is, the state vector in Bloch representation. An obvious benefit of this correspondence is to explain the role of the hidden layer of NNs, that is, to find an intuitive explanation. Moreover, it also found a way to study the contribution and function of each layer of NNs in detail.

In addition, based on the different situations of the state vector, the pure or the mixed states, the relationship between the state vector and the environment can be well described. The expression method of system and environment can correspond to the quantum system and environment in quantum theory. The reason why the quantum system exhibits some abnormal phenomena (contrary to classical theory) is that quantum system will establish a connection with environment, such as entanglement. In this article, we provide the possibility to describe the pure and the mixed states, which is to provide a novel idea for explaining NNs. For example, the generalization ability of NNs can be analyzed by the purity of the state vector. The greater the purity of the state vector, the weaker the connection between the system and the environment, that is, the knowledge contained in the system is sufficient, and the information contained in the environment is less, which has little effect on decision-making.

Here we try to use the explanation system of GPT to answer some NN-related questions (**Q**) that the classical explanation system cannot or cannot easily answer.

Q1: It is mentioned in the article that NM-GPT is a network model with stronger constraints than NNs. If $Normal(\cdot)$ proposed by NM-GPT is used as the activation function of NNs to increase the strength constraint, will there be the same effect as NM-GPT?

Here we illustrate the problem through a set of experiments. We use the NN model under Cifar 10 and Cifar100 as the experimental model, and replace its activation function with $Normal(\cdot)$ and some other commonly used activation functions to test whether simply adding a strong constraint can improve the effect of the network model. The experimental results are shown in Fig. 3. From the experimental results, the effect of the model under $Normal(\cdot)$ is not the best. It shows that NM-GPT is a complete system, not simply explained by a strong constraint $Normal(\cdot)$.

Q2: Can the generalization problem of NNs be better explained under GPT?

In NNs, regularization can improve the generalization ability of the model. The explanation is that adding regularization will make the weight matrix sparse or scattered, and the weight matrix directly acts on the input vector, making the network more inclined to use all input features. The model under GPT normalizes the input vector to a pure state or linear combination of multiple pure states into a mixed state. The realistic result is to smooth the input vector so that all features can have a certain contribution. It can be seen that the two are the same in this respect. Explained from a broader perspective, when the state vector is a pure state, the contribution of each feature of the feature vector is very different compared to the mixed state, which indicates that all features have not been fully learned; when the state vector is in the mixed state, the contribution of each feature is not much different, which indicates that each feature is fully utilized.

VII. Conclusions

This paper reconstructs the Feedforward Neural Network (FNN) based on the high-order quantum system revealed by the generalized probability theory proposed by Hardy et al. [14], [15]. Specifically, we construct each layer of the FNN as an elementary quantum system, which can be a pure state system or a mixed state system, and then leverage quantum coherence to describe the complex relationship between the features extracted by each neuron in each layer. The significance is that we propose a new method for constructing large-scale, high-order quantum systems on classical computers and explores a quantitative method for the learning ability of each layer of the neural network.

Our current work only leverages the high-order elementary quantum system, which means that we can only leverage quantum coherence between the superposition states of a single system, and does not involve the more important or valuable quantum entanglement. Because this theory is in the initial stage of research, there are still certain difficulties in constructing composite systems, which is also a research direction that we will need to explore later.

References

[1] A. Peres, *Quantum theory: concepts and methods*. Springer Science & Business Media, 2006, vol. 57.

[2] S. K. Jeswal and S. Chakraverty, “Recent developments and applications in quantum neural network: A review,” *Archives of Computational Methods in Engineering*, pp. 1–15, 2018.

[3] M. Hayashi, *Quantum information*. Springer, 2006.

[4] M. M. Wilde, *Quantum information theory*. Cambridge University Press, 2013.

[5] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. American Association of Physics Teachers, 2002.

[6] D. Ventura and T. Martinez, “Quantum associative memory,” *Inf. Sci.*, vol. 124, pp. 273–296, 2000.

[7] N. Kouda, N. Matsui, H. Nishimura, and F. Peper, “Qubit neural network and its efficiency,” in *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 2003, pp. 304–310.

[8] I. Cong, S. Choi, and M. Lukin, "Quantum convolutional neural networks," *Nature Physics*, pp. 1–6, 2018.

[9] J. Bausch, "Recurrent quantum neural networks," *ArXiv*, vol. abs/2006.14619, 2020.

[10] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, "Quantum hopfield neural network," *Physical Review A*, vol. 98, p. 042308, 2018.

[11] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, "Quantum graph neural networks," *ArXiv*, vol. abs/1909.12264, 2019.

[12] A. Streltsov, G. Adesso, and M. Plenio, "Colloquium: quantum coherence as a resource," *Reviews of Modern Physics*, vol. 89, p. 041003, 2017.

[13] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Reviews of modern physics*, vol. 81, no. 2, p. 865, 2009.

[14] L. Hardy, "Quantum theory from five reasonable axioms," *arXiv: Quantum Physics*, 2001.

[15] B. Dakic and C. Brukner, "Quantum theory and beyond: Is entanglement special?" *arXiv preprint arXiv:0911.0695*, 2009.

[16] T. Menneer and A. Narayanan, "Quantum-inspired neural networks," *Tech. Rep. R329*, 1995.

[17] G. Tóth, C. Lent, P. D. Tougaw, Y. Brazhnik, W. Weng, W. Porod, R. Liu, and Y. Huang, "Quantum cellular neural networks," *Superlattices and Microstructures*, vol. 20, pp. 473–478, 1996.

[18] N. Matsui, M. Takai, and H. Nishimura, "A network model based on qubitlike neuron corresponding to quantum circuit," *Electronics and Communications in Japan Part Iii-fundamental Electronic Science*, vol. 83, pp. 67–73, 2000.

[19] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, and P. S. Yu, "Deep learning for community detection: Progress, challenges and opportunities," in *IJCAI*, 2020.

[20] X. Ma, J. Wu, S. Xue, J. Yang, Q. Z. Sheng, and H. Xiong, "A comprehensive survey on graph anomaly detection with deep learning," *ArXiv*, vol. abs/2106.07178, 2021.

[21] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q. Z. Sheng, and P. S. Yu, "A comprehensive survey on community detection with deep learning," *ArXiv*, vol. abs/2105.12584, 2021.

[22] A. Cabello, S. Severini, and A. Winter, "(non-)contextuality of physical theories as an axiom," *arXiv: Quantum Physics*, 2010.

[23] L. Masanes and M. P. Müller, "A derivation of quantum theory from physical requirements," *New Journal of Physics*, vol. 13, no. 6, p. 063001, 2011.

[24] A. Cabello, S. Severini, and A. Winter, "Graph-theoretic approach to quantum correlations," *Physical review letters*, vol. 112, no. 4, p. 040401, 2014.

[25] M. Otto, "Instrumentalism," *The Monist*, pp. 577–593, 1926.

[26] R. Torretti, *The philosophy of physics*. Cambridge University Press, 1999.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2004.

[28] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Physical review letters*, vol. 23, no. 15, p. 880, 1969.

[29] M. Rowe, D. Kielpinski, V. Meyer, C. Sackett, W. Itano, C. Monroe, and D. Wineland, "Experimental violation of a bell's inequality with efficient detection," *Nature*, vol. 409, pp. 791–794, 2001.

[30] A. A. Klyachko, M. A. Can, S. Binicioğlu, and A. S. Shumovsky, "Simple test for hidden variables in spin-1 systems," *Physical review letters*, vol. 101, no. 2, p. 020403, 2008.

[31] J. S. Bell, "On the problem of hidden variables in quantum mechanics," *Reviews of Modern Physics*, vol. 38, pp. 447–452, 1966.

[32] M. Grötschel, L. Lovász, and A. Schrijver, "Relaxations of vertex packing," *Journal of Combinatorial Theory, Series B*, vol. 40, no. 3, pp. 330–343, 1986.

[33] L. Lovász, "On the shannon capacity of a graph," *IEEE Transactions on Information theory*, vol. 25, no. 1, pp. 1–7, 1979.

[34] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[36] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[37] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.