混合量子衍生神经网络模型及算法

李盼池*　　李国蕊
(东北石油大学计算机与信息技术学院　大庆　163318)

摘　要：为提高人工神经网络的逼近能力，该文从研究隐层神经元的映射机制入手，提出基于量子比特在 Bloch 球面的绕轴旋转构造神经网络模型的新思想。首先将样本线性变换为量子比特的相位，并使量子比特在 Bloch 球面上分别绕着 3 个坐标轴旋转，旋转角度即为网络参数。然后通过投影测量可以得到量子比特的球面坐标，将这些坐标值提交到隐层激励函数，可得隐层神经元的输出。输出层采用普通神经元。基于 L-M(Levenberg-Marquardt)算法设计了该模型的学习算法。实验结果表明，该文提出的模型在逼近能力、泛化能力、鲁棒性能方面，均优于采用 L-M 算法的普通神经网络。

关键词：量子计算；量子比特旋转；量子衍生神经元；量子衍生神经网络
中图分类号：TP183　　**文献标识码**：A　　**文章编号**：1009-5896(2016)01-0111-08
DOI: 10.11999/JEIT150444

Hybrid Quantum-inspired Neural Networks Model and Algorithm

LI Panchi　　LI Guorui
(*School of Computer and Information Technology*, *Northeast Petroleum University*, *Daqing* 163318, *China*)

Abstract: To enhance the mapping ability of artificial neural networks, by studying the mapping mechanism of hidden layer neurons, a new idea of designing neural networks model based on rotation of qubits in the Bloch sphere is proposed in this paper. In the proposed approach, the samples are linearly transformed to quantum bit phase, and the qubits are rotated about three coordinate axes, respectively. The network parameters of hidden layer are the rotation angles. The spherical coordinates of qubits can be obtained by the projection measurement. The output of hidden layer neurons can be concluded by submitting these coordinates to excitation functions in hidden layer. The general neurons are applied to the output layer. The learning algorithms of the proposed model are designed based on the Levenberg-Marquardt (L-M) algorithm. The experimental results show that the proposed model is superior to the classical (L-M) algorthm in approximation ability, generalization ability, and robust performance.

Key words: Quantum computation; Qubit ratation; Quantum-inspired neuron; Quantum-inspired neural networks

1　引言

严格说来量子神经网络是完全采用量子计算机制构造的神经网络。由于量子计算在普通计算机上无法实现，所以纯量子神经网络目前尚无法仿真。一般说来，通常量子神经网络也指借用一些量子计算的思路，或受某些量子计算原理的启发，采用经典方法设计的能在普通计算机上运行的神经网络，这种模型也称为量子衍生神经网络。量子神经网络出现于上个世纪 90 年代。1995 年，Kak[1]首次提出了量子神经计算这一全新的计算模型。随后 Gopathy 等人[2]提出了基于多级激励函数的量子神经网络模型，Ventura 等人[3]提出了具有指数级存储容量的量子联想记忆模型。2000 年，基于多宇宙量子理论，文献[4]提出了具有多个宇宙叠加的量子神经网络模型。国内有关量子神经网络的研究是从中国科技大学开始的。2001 年解光军博士和庄镇泉教授首次在国内撰文阐述了量子神经计算的概念[5]，随后解光军等人[6,7]在深入研究通用量子逻辑门工作机理的基础上，提出了基于通用量子逻辑门构造量子神经网络模型的新方法。文献[8]提出一种求解异或问题的量子神经元学习算法。基于量子旋转门和量子受控非门的物理意义，文献[9]提出了一种基于通用量子门的量子 BP 神经网络模型。文献[10]提出了一种基于量子权值和量子活性值的神经网络模型。基于量子门线路的物理意义，文献[11]提出了一种采用量子门节点构造神经网络模型的新方法。文献

收稿日期：2015-04-20；改回日期：2015-08-21；网络出版：2015-11-01
*通信作者：李盼池　lipanchi@vip.sina.com
基金项目：国家自然科学基金(61170132)，黑龙江省自然科学基金(F2015021)，黑龙江省教育厅科学技术研究项目(12541059)
Foundation Items: The National Natural Science Foundation of China (61170132), The Natural Science Foundation of Heilongjiang Province, China (F2015021), The Scientific and Technological Research Project of the Education Department of Heilongjiang Province, China (12541059)

[12-14]分别提出了采用受控门 Hadamard 门和受控旋转门构造输入为多维离散序列的量子神经网络模型的新方法。针对训练多层激励函数量子神经网络时权值与量子间隔的目标函数存在冲突，文献[15]提出了一种改进的量子神经网络的训练算法，降低了目标函数陷入局部极小值的概率。文献[16]在深入分析量子前向对传网模型与基于递归加权最小二乘的量子前向对传算法基础上，提出了量子前向对传网的定义与知识集，并提出了自适应量子前向对传算法。文献[17]提出了用于径向基函数网络训练的量子自适应粒子群算法。目前量子计算与神经计算的融合正逐步成为一个崭新的研究方向。

在上述所有模型中，量子比特都是采用平面上的单位圆描述的，量子比特的旋转也只是在单位圆上绕着圆心旋转。这种描述方法，量子比特只有一个可调参数，是真实量子模型的一种简化描述，尽管也都不同程度地提高了传统神经网络的性能，但量子特性没有得到充分体现。在真实的量子系统中，量子比特是基于 Bloch 球面描述的，具有两个可调参数，量子比特的旋转是在 Bloch 球面上的绕轴旋转。然而基于量子比特在 Bloch 球面上的绕轴旋转构建神经网络模型的研究，目前在国内外尚未见文献报道。本文将尝试这方面的研究。我们提出的网络为 3 层结构，其中输入为量子比特，隐层为量子衍生神经元，采用量子比特的绕轴旋转构造映射关系，输出层为普通神经元，采用 L-M 算法详细推导了学习算法。实验结果表明，该模型在逼近能力、泛化能力、鲁棒性方面均优于普通神经网络。值得指出，在本文提出的模型中，作为激励函数的量子变换与球形变换密切相关。关于基于球形领域构造神经网络的思想，文献[18,19]做了开创性的工作。在这两篇文献中，应用球形领域覆盖思想，可将神经网络的最优设计问题转化为某种求最优覆盖的问题。本文采用了与此不同的球形变换方法构造隐层神经元激励函数，严格说来本文设计的只是一种量子衍生神经网络，该网络并没有超出经典神经网络的范围，自然也可以按经典神经网络模拟，其核心思想是通过采用与球形变换密切相关的量子比特绕轴旋转，构造新的激励函数，以期提高传统神经网络的性能。

2 量子比特的球面描述及绕轴旋转

2.1 量子比特的球面描述

在量子计算中，一个量子比特是一个可以在2维复希尔伯特空间中描述的两能级量子体系，根据叠加原理，量子比特的任何态都可以写为

$$|\varphi\rangle = \cos(\theta/2)\,|0\rangle + \mathrm{e}^{\mathrm{i}\varphi}\sin(\theta/2)\,|1\rangle = \left[\cos(\theta/2)\ \mathrm{e}^{\mathrm{i}\varphi}\sin(\theta/2)\right]^{\mathrm{T}} \tag{1}$$

其中 $0 \le \theta \le \pi$，$0 \le \varphi \le 2\pi$，$|\,\rangle$ 为狄拉克符号，$|0\rangle$ 和 $|1\rangle$ 为 $|\varphi\rangle$ 的两个基态，$\cos(\theta/2)$ 和 $\mathrm{e}^{\mathrm{i}\varphi}\sin(\theta/2)$ 分别为基态 $|0\rangle$ 和基态 $|1\rangle$ 的概率幅，其模的平方分别为 $|\varphi\rangle$ 处于 $|0\rangle$ 和 $|1\rangle$ 的概率。

由于 θ 和 φ 连续，所以一个量子比特可以描述无穷多个不同的状态。量子比特可以用 3 维 Bloch 球面上的一个点来描述，如图 1 所示，其中 $x = \sin\theta\cos\varphi$，$y = \sin\theta\sin\varphi$，$z = \cos\theta$。此时，在 Bloch 球面上的任意一点 $P(x,y,z)$ 都和一个量子比特 $|\varphi\rangle$ 对应。

2.2 量子比特的绕轴旋转

量子比特在 Bloch 球面上的绕轴旋转，是通过对量子比特施加一个酉算子实现的。算子的酉性能够保证描述量子比特的向量在旋转前后的长度不变。根据量子计算原理，量子比特在 Bloch 球面上绕一个沿单位矢量 $\boldsymbol{n} = [n_x, n_y, n_z]$ 的轴转动 δ 弧度的旋转矩阵为[20]

$$\boldsymbol{R}_{\boldsymbol{n}}(\delta) = \cos\frac{\delta}{2}\boldsymbol{I} - \mathrm{i}\sin\frac{\delta}{2}(\boldsymbol{n}\times\boldsymbol{\sigma}) \tag{2}$$

其中 $\boldsymbol{I}$ 为单位阵，$\boldsymbol{\sigma} = [\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z]$ 为式(3)定义的矩阵。

$$\boldsymbol{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ \boldsymbol{\sigma}_y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix},\ \boldsymbol{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3}$$

在3个坐标轴上分别取单位向量 $\boldsymbol{n}_x = [1,0,0]$，$\boldsymbol{n}_y = [0,1,0]$，$\boldsymbol{n}_z = [0,0,1]$，根据式(2)，式(3)，量子比特绕$X$轴、$Y$轴、$Z$轴旋转 δ 弧度的旋转矩阵分别为

$$R_x(\delta) = \begin{bmatrix} \cos(\delta/2) & -\mathrm{i}\sin(\delta/2) \\ -\mathrm{i}\sin(\delta/2) & \cos(\delta/2) \end{bmatrix} \tag{4}$$

$$R_y(\delta) = \begin{bmatrix} \cos(\delta/2) & -\sin(\delta/2) \\ \sin(\delta/2) & \cos(\delta/2) \end{bmatrix} \tag{5}$$

$$R_z(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\delta} \end{bmatrix} \tag{6}$$

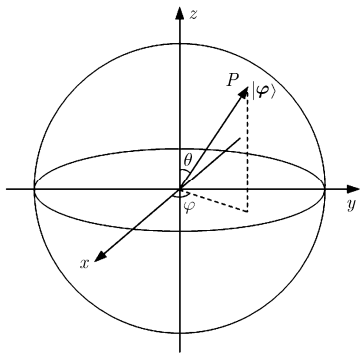

图 1 量子比特的Bloch球面描述

将上述3个旋转矩阵施加到量子比特后，即可使量子比特分别绕着相应的坐标轴旋转 δ 弧度。

3　混合量子衍生神经网络模型

3.1 量子衍生神经元模型

本文提出的量子衍生神经元，输入为量子比特，待调参数为旋转角度，输出为实数，模型如图 2 所示。

该神经元映射关系的设计思路为，对于输入 $|x_i\rangle\,(i=1,2,\cdots,n)$，分别绕 X 轴 Y 轴 Z 轴旋转相同的角度 δ_i，并分别对旋转后的量子比特实施投影测量，以获得量子比特的 Bloch 坐标，然后将这些坐标值求和，再经过激励函数即可得到该神经元的输出。

令 $|x_i\rangle=[\cos(\theta_i/2)\ \ \mathrm{e}^{\mathrm{i}\phi_i}\sin(\theta_i/2)]^{\mathrm{T}}$，$|x_i\rangle$ 绕 3 个坐标轴旋转之后分别为 $|\hat{x}_i\rangle$，$|\hat{y}_i\rangle$，$|\hat{z}_i\rangle$，则 $|x_i\rangle$ 绕着 3 个坐标轴旋转 δ_i 弧度的旋转操作可表述为

$$
\begin{aligned}
|\hat{x}_i\rangle &= R_x(\delta_i)\,|x_i\rangle \\
&= \begin{bmatrix} \cos\dfrac{\delta_i}{2} & -\mathrm{i}\sin\dfrac{\delta_i}{2} \\ -\mathrm{i}\sin\dfrac{\delta_i}{2} & \cos\dfrac{\delta_i}{2} \end{bmatrix}\begin{bmatrix} \cos\dfrac{\theta_i}{2} \\ \mathrm{e}^{\mathrm{i}\phi_i}\sin\dfrac{\theta_i}{2} \end{bmatrix} \\
&= \begin{bmatrix} \cos\dfrac{\delta_i}{2}\cos\dfrac{\theta_i}{2}-\mathrm{i}\mathrm{e}^{\mathrm{i}\phi_i}\sin\dfrac{\delta_i}{2}\sin\dfrac{\theta_i}{2} \\ -\mathrm{i}\sin\dfrac{\delta_i}{2}\cos\dfrac{\theta_i}{2}+\mathrm{e}^{\mathrm{i}\phi_i}\cos\dfrac{\delta_i}{2}\sin\dfrac{\theta_i}{2} \end{bmatrix}
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
|\hat{y}_i\rangle &= R_y(\delta_i)\,|x_i\rangle \\
&= \begin{bmatrix} \cos\dfrac{\delta_i}{2} & -\sin\dfrac{\delta_i}{2} \\ \sin\dfrac{\delta_i}{2} & \cos\dfrac{\delta_i}{2} \end{bmatrix}\begin{bmatrix} \cos\dfrac{\theta_i}{2} \\ \mathrm{e}^{\mathrm{i}\phi_i}\sin\dfrac{\theta_i}{2} \end{bmatrix} \\
&= \begin{bmatrix} \cos\dfrac{\delta_i}{2}\cos\dfrac{\theta_i}{2}-\mathrm{e}^{\mathrm{i}\phi_i}\sin\dfrac{\delta_i}{2}\sin\dfrac{\theta_i}{2} \\ \sin\dfrac{\delta_i}{2}\cos\dfrac{\theta_i}{2}+\mathrm{e}^{\mathrm{i}\phi_i}\cos\dfrac{\delta_i}{2}\sin\dfrac{\theta_i}{2} \end{bmatrix}
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
|\hat{z}_i\rangle = R_z(\delta_i)\,|x_i\rangle &= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\delta_i} \end{bmatrix}\begin{bmatrix} \cos\dfrac{\theta_i}{2} \\ \mathrm{e}^{\mathrm{i}\phi_i}\sin\dfrac{\theta_i}{2} \end{bmatrix} \\
&= \begin{bmatrix} \cos\dfrac{\theta_i}{2} \\ \mathrm{e}^{\mathrm{i}(\delta_i+\phi_i)}\sin\dfrac{\theta_i}{2} \end{bmatrix}
\end{aligned}
\tag{9}
$$

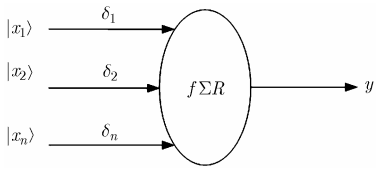

图 2　量子衍生神经元模型

利用泡利矩阵实施投影测量可以得到 $|\hat{x}_i\rangle$，$|\hat{y}_i\rangle$，$|\hat{z}_i\rangle$ 各自对应的 Bloch 球面坐标，如式(10)～式(12)所示。其中 $\langle\bullet|$ 为 $|\bullet\rangle$ 的共轭转置。

$$
\left.\begin{aligned}
x\_\hat{x}_i &= \langle\hat{x}_i\,|\,\sigma_x\,|\,\hat{x}_i\rangle = \sin\theta_i\cos\phi_i \\
y\_\hat{x}_i &= \langle\hat{x}_i\,|\,\sigma_y\,|\,\hat{x}_i\rangle = \sin\theta_i\sin\phi_i\cos\delta_i - \cos\theta_i\sin\delta_i \\
z\_\hat{x}_i &= \langle\hat{x}_i\,|\,\sigma_z\,|\,\hat{x}_i\rangle = \sin\theta_i\sin\phi_i\sin\delta_i + \cos\theta_i\cos\delta_i
\end{aligned}\right\}
\tag{10}
$$

$$
\left.\begin{aligned}
x\_\hat{y}_i &= \langle\hat{y}_i\,|\,\sigma_x\,|\,\hat{y}_i\rangle = \sin\theta_i\cos\phi_i\cos\delta_i + \cos\theta_i\sin\delta_i \\
y\_\hat{y}_i &= \langle\hat{y}_i\,|\,\sigma_y\,|\,\hat{y}_i\rangle = \sin\theta_i\sin\phi_i \\
z\_\hat{y}_i &= \langle\hat{y}_i\,|\,\sigma_z\,|\,\hat{y}_i\rangle = \cos\theta_i\cos\delta_i - \sin\theta_i\cos\phi_i\sin\delta_i
\end{aligned}\right\}
\tag{11}
$$

$$
\left.\begin{aligned}
x\_\hat{z}_i &= \langle\hat{z}_i\,|\,\sigma_x\,|\,\hat{z}_i\rangle = \cos(\phi_i+\delta_i)\sin\theta_i \\
y\_\hat{z}_i &= \langle\hat{z}_i\,|\,\sigma_y\,|\,\hat{z}_i\rangle = \sin(\phi_i+\delta_i)\sin\theta_i \\
z\_\hat{z}_i &= \langle\hat{z}_i\,|\,\sigma_z\,|\,\hat{z}_i\rangle = \cos\theta_i
\end{aligned}\right\}
\tag{12}
$$

由式(10)～式(12)可知，$|x_i\rangle$ 绕 X 轴旋转前后的 x 坐标不变，绕 Y 轴旋转前后的 y 坐标不变，绕 Z 轴旋转前后的 z 坐标不变。所以我们在对坐标求和时不考虑这些坐标值。

令 $S_i=0.5(y\_\hat{x}_i+z\_\hat{x}_i+x\_\hat{y}_i+z\_\hat{y}_i+x\_\hat{z}_i+y\_\hat{z}_i)$，由式(10)～式(12)得

$$
S_i = \sin\theta_i\cos\phi_i + \sin\theta_i\sin\phi_i + \cos\theta_i)\cos\delta_i \tag{13}
$$

采用 Sigmoid 函数作为激励函数，隐层量子神经元的输出为

$$
y = 1\Big/\left(1+\mathrm{e}^{-\sum_{i=1}^{n}S_i}\right) \tag{14}
$$

3.2 混合量子衍生神经网络模型

采用量子衍生神经元为隐层，普通神经元为输出层，本文提出的 3 层混合量子衍生神经网络模型如图 3 所示。其中输入层、隐层、输出层分别有 n，p，m 个神经元。输出层激励函数也采用 Sigmoid 函数。

该模型的隐层输出和输出层输出分别按式(15)，式(16)计算。

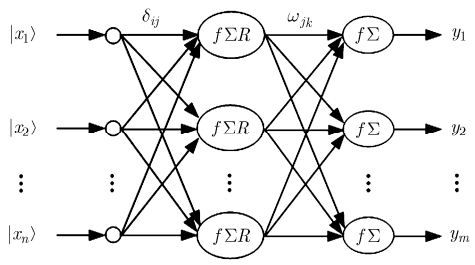

图 3　混合量子衍生神经网络模型

$$h_j = f\left(\sum_{i=1}^{n} S_{ij}\right) = \frac{1}{1+\mathrm{e}^{-\sum_{i=1}^{n}(\sin\theta_i\cos\phi_i+\sin\theta_i\sin\phi_i+\cos\theta_i)\cos\delta_{ij}}} \tag{15}$$

$$y_k = f\left(\sum_{j=1}^{p} \omega_{jk} h_j\right) = \frac{1}{1+\mathrm{e}^{-\sum_{j=1}^{p}\omega_{jk}h_j}} \tag{16}$$

4 混合量子衍生神经网络算法

4.1 样本量子态描述

首先，将全部样本数据归一化为[0,1]之间的数值。虽然采取将每个样本除以该样本各维绝对值的最大值的方法，可以实现归一化，但这种方法归一化后的每个样本中均含有数值1，从而必然会改变各个样本之间的数量关系。

令 L 个 n 维样本为 $\boldsymbol{X}_l = [x_{l1}, x_{l2}, \cdots, x_{ln}]^{\mathrm{T}}$，其中 $l = 1, 2, \cdots, L$。为了确保各维样本之间的相对大小保持不变，本文采用式(17)的归一化方法。

$$x_{li} = \frac{x_{li} - \mathrm{MIN}_i}{\mathrm{MAX}_i - \mathrm{MIN}_i} \tag{17}$$

其中 MAX_i 和 MIN_i 分别是所有样本第 i 维的最大值和最小值。

令归一化后的样本为 $\boldsymbol{X}_l = [x_{l1}, x_{l2}, \cdots, x_{ln}]^{\mathrm{T}}$，利用式(18)，式(19)将其转化为量子比特的相位。

$$\boldsymbol{\theta}_l = [\theta_{l1}, \theta_{l2}, \cdots, \theta_{ln}]^{\mathrm{T}} = [\pi x_{l1}, \pi x_{l2}, \cdots, \pi x_{ln}]^{\mathrm{T}} \tag{18}$$

$$\boldsymbol{\varphi}_l = [\varphi_{l1}, \varphi_{l2}, \cdots, \varphi_{ln}]^{\mathrm{T}} = [2\pi x_{l1}, 2\pi x_{l2}, \cdots, 2\pi x_{ln}]^{\mathrm{T}} \tag{19}$$

至此，$\widetilde{\boldsymbol{X}}_l$ 可映射为 Bloch 球面上的量子比特，如式(20)所示。

$$|\boldsymbol{X}_l\rangle = [|x_{l1}\rangle, |x_{l2}\rangle, \cdots, |x_{ln}\rangle]^{\mathrm{T}} \tag{20}$$

其中 $|x_{li}\rangle = \cos(\theta_{li}/2)|0\rangle + \mathrm{e}^{\mathrm{i}\varphi_{li}}\sin(\theta_{li}/2)|1\rangle$。

4.2 网络各层参数的调整

网络的待调参数为隐层旋转角度 δ_{ij} 和输出层权值 ω_{jk}。由于传统BP算法采用误差函数的一阶导数，因此收敛速度较慢，虽然牛顿法采用误差函数的二阶导数调整网络参数，具有较快的收敛速度，但该方法比较苛刻，其二阶导数矩阵常常不可逆。L-M 算法是一阶梯度下降法和二阶牛顿法之间的一种折中算法。收敛速度较快，对参数要求也比较宽松，因此本文采用 L-M 算法训练网络参数。

记误差函数为

$$E = \frac{1}{2}\sum_{k=1}^{m}(\overline{y}_k - y_k)^2 = \frac{1}{2}\sum_{k=1}^{m}(e_k)^2 \tag{21}$$

误差函数 e_k 对 δ_{ij} 和 ω_{jk} 的梯度计算式为

$$\frac{\partial e_k}{\partial \delta_{ij}} = \sum_{k=1}^{m} y_k(1-y_k)\omega_{jk}h_j(1-h_j) \cdot (\sin\theta_i\cos\phi_i + \sin\theta_i\sin\phi_i + \cos\theta_i)\sin\delta_{ij} \tag{22}$$

$$\frac{\partial e_k}{\partial \omega_{jk}} = -y_k(1-y_k)h_j \tag{23}$$

记 $\boldsymbol{P}$ 表示参数向量，$\boldsymbol{V}$ 表示误差向量，$\boldsymbol{J}$ 表示雅可比矩阵，它们分别定义为

$$\boldsymbol{P}^{\mathrm{T}} = \left[\delta_{11}, \delta_{21}, \cdots, \delta_{np}, \omega_{11}, \omega_{21}, \cdots, \omega_{pm}\right] \tag{24}$$

$$\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{P}) = \left[e_1, e_2, \cdots, e_m\right] \tag{25}$$

$$\boldsymbol{J}(\boldsymbol{P}) = \begin{bmatrix} \frac{\partial e_1}{\partial \delta_{11}} & \cdots & \frac{\partial e_1}{\partial \delta_{np}} & \frac{\partial e_1}{\partial \omega_{11}} & \cdots & \frac{\partial e_1}{\partial \omega_{pm}} \\ \frac{\partial e_2}{\partial \delta_{11}} & \cdots & \frac{\partial e_2}{\partial \delta_{np}} & \frac{\partial e_2}{\partial \omega_{11}} & \cdots & \frac{\partial e_2}{\partial \omega_{pm}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_m}{\partial \delta_{11}} & \cdots & \frac{\partial e_m}{\partial \delta_{np}} & \frac{\partial e_m}{\partial \omega_{11}} & \cdots & \frac{\partial e_m}{\partial \omega_{pm}} \end{bmatrix} \tag{26}$$

根据 L-M 算法，参数调整的迭代方程为

$$\boldsymbol{P}_{t+1} = \boldsymbol{P}_t - \alpha\left[\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{P}_t)\boldsymbol{J}(\boldsymbol{P}_t) + \mu_t\boldsymbol{I}\right]^{-1}\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{P}_t)\boldsymbol{V}(\boldsymbol{P}_t) \tag{27}$$

其中 t 为迭代步数，α 为学习速率，$\boldsymbol{I}$ 为单位矩阵，μ_t 为一小正数，以使 $\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{P}_t)\boldsymbol{J}(\boldsymbol{P}_t) + \mu_t\boldsymbol{I}$ 可逆。

5 对比实验

为验证混合量子衍生神经网络(Hybrid Quantum-inspired Neural Networks, HQNN)的优良性能，本实验采用国际通用的标准样本进行仿真，并与普通人工神经网络(Artificial Neural Networks, ANN)对比。为使对比公平，ANN 采用与 HQNN 相同的网络结构，并且也采用 L-M 算法训练网络。我们将从映射能力、泛化能力、鲁棒性能 3 个方面考察两种网络的对比。

5.1 人体姿势识别

本实验使用的样本数据为人体的 5 种姿势，来源于如下网址：http://archive.ics.uci.edu/ml/machine-learning-databases/00302/。该数据集由从事视频处理的研究人员，根据连续动作的时间采样建立。建立过程为：请 3 位候选人各读一本连环画，然后在传感器前讲述连环画的内容。利用传感器记录 6 个关节点坐标。6 个关节点分别是：左手、右手、左腕、右腕、头和脊柱。每个样本有 18 个特征值，其含义分别是：左手位置(x, y, z)，右手位置(x, y, z)，头位置(x, y, z)，脊柱位置(x, y, z)，左腕位置(x, y, z)，右腕位置(x, y, z)。5 种姿势及样本个数分别为休息(rest)698 个、预备(preparation)163 个、中风(stroke)656 个、紧握(hold)39 个、退缩(retraction)191 个。为简便，我们以样本数最少的第 4 类为准，每类随机选取 39 个，并将每类的前 20 个作为训练集，后 19 个作为测试集。因此，训练集有 100 个样本，测试集有 95 个样本。

5.2 算法设置

根据样本数据，输入样本为 18 维行向量。因此，网络的输入节点 $n=18$，网络输出为每个样本所属的类别，用 1~5 的自然数表示，所以输出层节点数 $m=1$。对于 L-M 算法，控制参数 μ_t 取 0.1。对于 HQNN，隐层旋转角度均在 $(-1,1)$ rad 内随机取值，输出层权值均在 $(-1,1)$ 内随机取值。对于 ANN，两层权值均在 $(-1,1)$ 内随机取值。算法的终止条件为达到限定迭代步数，且两种网络的限定迭代步数均取 1000。为使对比充分，采用两种方案。第 1 种方案将隐层神经元数固定为 $p=10$，使学习速率从 $\alpha=0.1$ 以 0.1 为步长逐步增大到 $\alpha=2.0$。第 2 种方案将学习速率固定为 $\alpha=1.0$，使隐层神经元数从 $p=1$ 逐步增大到 $p=20$。

5.3 结果对比

对比指标为训练集样本归一化之后的最大误差、训练集正确分类数、测试集正确分类数。为减小训练结果的随机性，对于两种方案下 α 和 p 的每种取值，HQNN 和 ANN 分别独立训练 30 次，然后取平均结果进行对比。两种网络的训练结果对比如表 1，表 2 所示，训练集和测试集的正确分类数随参数 α 和 p 的变化情况分别如图 4，图 5 所示，训练误差对比如图 6，图 7 所示。

由以上实验结果可知，就逼近能力和泛化能力而言，在学习速率的各种设置下，HQNN 都优于 ANN；在隐层神经元数的各种设置下，除神经元数取 1~5 外，HQNN 也都优于 ANN。就鲁棒性能而言，当固定学习速率，改变隐层神经元数时，HQNN

表 1 不同学习速率下的平均训练结果对比(隐层节点 $p=10$)

α	HQNN			ANN		
	训练集正确数	测试集正确数	最大训练误差	训练集正确数	测试集正确数	最大训练误差
0.1	94.6333	67.1000	0.2381	93.7333	63.1667	0.2395
0.2	97.0000	66.0333	0.1775	94.9667	62.4667	0.1940
0.3	97.5667	66.3000	0.1622	96.0333	62.5000	0.1635
0.4	97.7667	68.5667	0.1492	96.6667	63.5333	0.1399
0.5	98.7333	68.7333	0.1327	97.5667	63.5333	0.1314
0.6	99.2667	70.4000	0.1271	98.0000	63.5000	0.1255
0.7	99.7333	71.7333	0.1145	98.0333	64.1667	0.1234
0.8	99.8333	71.7333	0.1090	98.0000	64.0333	0.1222
0.9	99.9333	71.4667	0.1089	97.9000	64.3000	0.1209
1.0	100.0000	73.3333	0.1008	98.1667	66.2333	0.1204
1.1	99.7667	71.5667	0.1067	98.0667	64.4667	0.1189
1.2	99.9000	71.0333	0.1007	98.1333	64.6667	0.1173
1.3	99.9667	71.1000	0.0975	92.9000	61.3333	0.2052
1.4	99.9667	72.3000	0.0944	84.8000	58.4000	0.2049
1.5	99.9000	72.5667	0.1000	63.7667	46.9333	0.4160
1.6	99.9667	71.5667	0.0972	50.9000	39.9667	0.7438
1.7	99.9667	71.6333	0.0947	31.2333	36.3667	0.8918
1.8	99.4000	72.0000	0.1076	25.9333	30.3333	0.8839
1.9	99.4000	69.9000	0.1105	21.0333	28.8667	0.9722
2.0	87.7333	66.7333	0.3390	20.4000	30.7000	0.9998

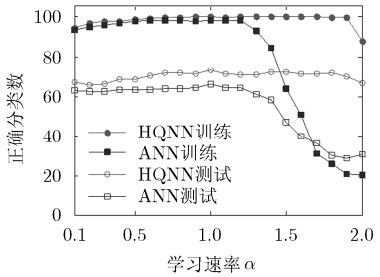

图 4 正确分类数随学习速率的变化

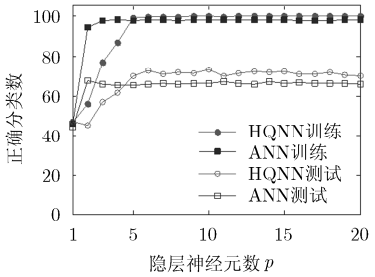

图 5 正确分类数随隐层神经元数的变化

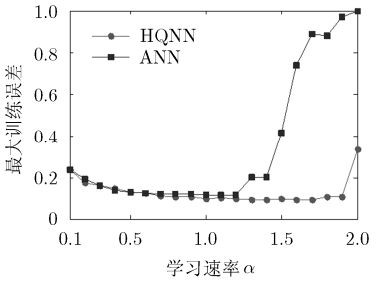

图 6 最大训练误差随学习速率的变化

表 2 不同隐层神经元数下的平均训练结果对比(学习速率 $\alpha = 1.0$)

p	HQNN			ANN		
	训练集正确数	测试集正确数	最大训练误差	训练集正确数	测试集正确数	最大训练误差
1	46.7667	47.1667	0.5014	46.1000	44.0667	0.5102
2	55.9000	45.2000	0.4881	94.3667	67.9667	0.1724
3	77.1000	56.8000	0.3199	97.7333	65.6667	0.1376
4	86.8333	61.5333	0.2412	98.0667	65.4667	0.1323
5	99.0667	70.2667	0.1321	97.9667	65.2000	0.1191
6	99.6667	72.7333	0.1130	98.0667	65.7333	0.1208
7	99.5667	71.1333	0.1165	98.1333	66.2333	0.1198
8	99.6000	71.8333	0.1124	97.8667	66.0667	0.1191
9	99.8667	71.4000	0.1069	98.0333	66.3667	0.1197
10	100.0000	73.3333	0.1008	98.1667	66.2333	0.1204
11	99.7000	69.9000	0.1127	98.3000	67.0667	0.1200
12	99.8667	71.6667	0.1031	98.0667	66.5000	0.1204
13	99.9000	72.5333	0.1074	98.2333	65.6667	0.1210
14	99.9667	72.1333	0.1000	98.2000	67.2000	0.1209
15	100.0000	72.2667	0.1026	98.0667	66.5333	0.1211
16	99.9667	71.1667	0.0990	97.9667	66.6667	0.1218
17	99.9333	70.8667	0.0994	97.8667	66.3667	0.1218
18	99.9667	71.8667	0.0991	97.9667	66.2000	0.1214
19	100.0000	70.7000	0.0983	98.0667	66.1000	0.1217
20	99.9667	69.8667	0.0955	98.0000	65.9667	0.1220

和 ANN 的适应能力几乎相等；而当固定隐层神经元数，改变学习速率时，HQNN 在各种速率下的逼近和泛化能力具有良好的稳定性，而当学习速率超过 1.2 后，ANN 的逼近和泛化能力快速下降。这表明 HQNN 的鲁棒性能也优于 ANN。就训练误差而言，也反映出相同的趋势。

5.4 波形识别

本实验采用神经网络方法识别带有噪声的 3 类波形(分别用 1,2,3 标识)。数据来源于如下网址：http://archive.ics.uci.edu/ml/machine-learning-databases/waveform/。每类波形由 40 个属性值来描述，其中前 21 个属性值为标准波形数据，后 19 个属性值为理想白噪声数据。该数据集共有 5000 个样本，为简便，我们仅取前 500 个样本作为仿真对象。其中前 300 个作为训练集(3 类分别为 100,100,100)，后 200 个作为测试集(3 类分别为 64, 66, 70)。

根据样本数据，输入样本为 40 维行向量。因此，网络的输入节点 $n = 40$，网络输出为每个波形所属的类别，用自然数 0, 1, 2 表示，所以输出层节点数 $m = 1$。对于 L-M 算法，控制参数 $\mu_t = 0.1$。对于 HQNN，隐层旋转角度均在 $(-\pi/2, \pi/2)$ rad 内随机取值，输出层权值均在 $(-1,1)$ 内随机取值。对于 ANN，两层权值均在 $(-1, 1)$ 内随机取值。算法的终止条件为达到限定迭代步数，且两种网络的限定迭代步数均取 1500。采用与人体姿势识别相同的对比方案，首先将隐层神经元数固定为 $p = 15$，学习速率 α 依次从集合 $\{0.1, 0.2, \cdots, 2.0\}$ 取值。然后将学习速率固定为 $\alpha = 0.8$，使隐层神经元数从 $p = 1$ 逐步增大到 $p = 25$。

为减小对比结果的随机性，对于 α 和 p 的每种取值，HQNN 和 ANN 分别独立训练 30 次，然后取平均结果进行对比。两种网络训练集和测试集的正确分类数随参数 α 和 p 的变化情况分别如图 8，图 9 所示。

从图 8，图 9 可知，在学习速率及隐层节点的各种设置下，HQNN 几乎都优于 ANN；当固定学习速率，改变隐层神经元数时，由于 ANN 在隐层神经元数为 22 和 25 时有较大的波动，因此 HQNN 的适应能力略好于 ANN；当固定隐层神经元数，改变学习速率时，HQNN 的性能比较稳定，而当学习速率超过 1.3 后，ANN 的性能下降较快。实验结果再次验证了 HQNN 的优越性。

5.5 结果分析

对于上述实验结果，我们给出如下理论分析。首先，在 HQNN 中，隐层神经元采用了完全不同的映射机制，将传统神经元中对输入的加权，改变为对输入的旋转；ANN 中对每个权值的搜索过程是使该权值在数轴上左右移动，HQNN 中对转角的搜索，

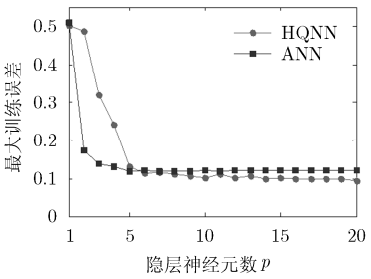

图 7 最大训练误差随隐层神经元数的变化

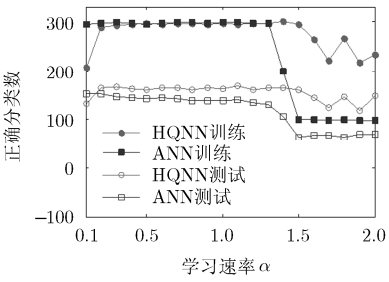

图 8 正确分类数随学习速率的变化

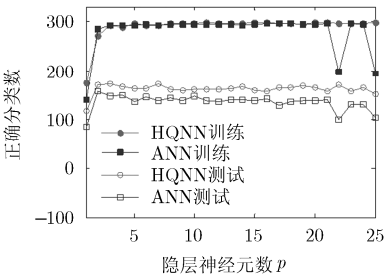

图 9 正确分类数随隐层神经元数的变化

是使该转角在 Bloch 球面上绕 3 个坐标轴转动；这样的搜索方式对解空间的遍历更为精细。第二，从 HQNN 的输入输出关系看，其输出式中，隐层的转角参数 δ_{ij} 以余弦函数形式出现，根据三角函数的周期性，若 δ_{ij} 最优，则 $\delta_{ij}+2k\pi\,(k=\pm1,\pm2,\cdots)$ 也最优，这将导致在隐层神经元中存在大量多重吸引子，从而可增强其逼近和泛化能力，当参数改变时，也能使 HQNN 有较好的适应能力。另外由于 HQNN 隐层的计算量比 ANN 略有增大，因此其计算效率略有降低，这是 HQNN 的缺点，但 HQNN 正是以牺牲计算效率换取网络性能提高的，这与无免费午餐定理是一致的。

6　结论

本文提出了一种混合量子衍生神经网络模型及算法。该模型输入为量子比特，隐层为量子衍生神经元，输出层为普通神经元。采用量子比特在 Bloch 球面上的绕轴旋转实现对输入的加权聚合，基于 L-M 算法导出了学习算法。该模型的隐层蕴含着多重吸引子，实验结果表明其逼近和泛化能力优于普通神经网络。本文研究结果揭示出采用量子比特在 Bloch 球面的绕轴旋转构造神经网络的思路是可行的。

参 考 文 献

[1] KAK S. On quantum neural computing[J]. *Information Sciences*, 1995, 83(3): 143–160.

[2] GOPATHY P and NICOLAOS B. Quantum Neural Networks (QNNs) inherently fuzzy feed-forward neural networks[J]. *IEEE Transactions on Neural Networks*, 1997, 8(3): 679–693.

[3] VENTURA D and TONY M. Quantum associative memory with exponential capacity[C]. Proceedings of the IEEE International Joint Conference on Computational Intelligence, Piscataway, NJ, 1998: 509–513.

[4] AJIT N and TAMMY M. Quantum artificial neural network architectures and components[J]. *Information Sciences*, 2000, 128(3): 231–255.

[5] 解光军, 庄镇泉. 量子神经网络[J]. 计算机科学, 2001, 28(7): 1–6.
XIE G J and ZHUANG Z Q. Quantum neural network[J]. *Computer Science*, 2001, 28(7): 1–6.

[6] 解光军, 范海秋, 操礼程. 一种量子神经计算网络模型[J]. 复旦学报(自然科学版), 2004, 43(5): 700–703.
XIE G J, FAN H Q, and CAO L C. A quantum neural computational network model[J]. *Journal of Fudan University* (*Natural Science*), 2004, 43(5): 700–703.

[7] 解光军, 周典, 范海秋. 基于量子门组单元的神经网络及其应用[J]. 系统工程理论与实践, 2005, 25(5): 113–117.
XIE G J, ZHOU D, and FAN H Q. A neural network model based on quantum gates cell and its applications[J]. *Systems Engineering – Theory & Practice*, 2005, 25(5): 113–117.

[8] MAEDA M, SUENAGA M, and MIYAJIMA H. Qubit neuron according to quantum circuit for XOR problem[J]. *Applied Mathematics and Computation*, 2007, 185(2): 1015–1025.

[9] LI P C and LI S Y. Learning algorithm and application of quantum BP neural networks based on universal quantum gates[J]. *Journal of Systems Engineering and Electronics*, 2008, 19(1): 167–174.

[10] 李盼池. 一种量子神经网络模型学习算法及应用[J]. 控制理论与应用, 2009, 26(5): 531–534.
LI P C. Learning algorithm and applications of the quantum neural networks model[J]. *Control Theory & Application*, 2009, 26(5): 531–534.

[11] LI P C, SONG K P, and YANG E L. Model and algorithm of neural networks with quantum gated nodes[J]. *Neural Network World*, 2010, 20(2): 189–206.

[12] LI P C and XIAO H. A hybrid quantum-inspired neural networks with sequence inputs[J]. *Neurocomputing*, 2013, 117: 81–90.

[13] LI P C and XIAO H. Model and algorithm of quantum-inspired neural network with sequence input based on controlled rotation gates[J]. *Application Intelligence*, 2014, 40(1): 107–126.

[14] 李盼池, 周红岩. 基于受控 Hadamard 门的量子神经网络模型及算法[J]. 计算机研究与发展, 2015, 52(1): 211–220.

LI P C and ZHOU H Y. Model and algorithm of quantum neural network based on the controlled Hadamard gates[J]. *Journal of Computer Research and Development*, 2015, 52(1): 211–220.

[15] 张翼鹏, 陈亮, 郝欢. 一种改进的量子神经网络训练算法[J]. 电子与信息学报, 2013, 35(7): 1630–1635. doi: 10.3724/SP.J.1146.2012.01417.

ZHANG Y P, CHEN L, and HAO H. An improved training algorithm for quantum neural networks[J]. *Journal of Electronics & Information Technology*, 2013, 35(7): 1630–1635. doi: 10.3724/SP.J.1146.2012.01417.

[16] 李楠, 侯旋. 自适应量子前向对传算法研究[J]. 电子与信息学报, 2013, 35(11): 2778–2783. doi: 10.3724/SP.J.1146.2013.00101.

LI N and HOU X. Research on adaptive quantum forward counter propagation algorithm[J]. *Journal of Electronics & Information Technology*, 2013, 35(11): 2778–2783. doi: 10.3724/SP.J.1146.2013. 00101.

[17] 郭通, 兰巨龙, 李玉峰. 基于量子自适应粒子群优化径向基函数神经网络的网络流量预测[J]. 电子与信息学报, 2013, 35(9): 2220–2226. doi: 10.3724/SP.J.1146.2012.01343.

GUO T, LAN J L, and LI Y F. Network traffic prediction with radial basis function neural network based on quantum adaptive particle swarm optimization[J]. *Journal of Electronics & Information Technology*, 2013, 35(9): 2220–2226. doi: 10.3724/SP.J.1146.2012.01343.

[18] 张铃, 张钹. M-P 神经元模型的几何意义及其应用[J]. 软件学报, 1998, 9(5): 334–338.

ZHANG L and ZHANG B. A geometrical representation of M-P neural model and its applications[J]. *Journal of Software*, 1998, 9(5): 334–338.

[19] 张铃, 张钹, 殷海风. 多层前向网络的交叉覆盖设计算法[J]. 软件学报, 1999, 10(7): 737–742.

ZHANG L, ZHANG B, and YIN H F. An alternative covering design algorithm of multi-layer neural networks[J]. *Journal of Software*, 1999, 10(7): 737–742.

[20] GIULIANO B, GIULIO C, and GIULIANO S. Principles of Quantum Computation and Information Volume I: Basic Concepts[M]. Singapore: World Scientific, 2004: 108–112.

李盼池： 男，1969 年生，教授，博士生导师，研究方向为量子智能优化和量子神经网络.

李国蕊： 女，1989 年生，硕士，研究方向为神经网络和智能优化算法.