

# 一种量子衍生神经网络模型算法及应用

杨淑云<sup>1</sup>, 李盼池<sup>2</sup>

(1. 东北石油大学招生就业处, 黑龙江大庆 163318; 2. 东北石油大学计算机与信息技术学院, 黑龙江大庆 163318)

**摘 要:** 为提高神经网络的逼近和预测能力, 提出一种各维输入为离散序列的量子衍生神经网络模型及算法. 该模型为三层结构, 隐层为量子衍生神经元, 输出层为普通神经元. 量子衍生神经元由量子旋转门和多位受控旋转门组成, 利用多位受控旋转门中目标量子位的输出向输入端的反馈, 实现对输入序列的整体记忆, 利用受控旋转门输出中多位量子比特的纠缠获得量子衍生神经元的输出. 基于量子计算理论设计了该模型的学习算法. 该模型可从宽度和深度两方面获取输入序列的特征. 仿真结果表明, 当输入节点数和序列长度满足一定关系时, 该模型明显优于普通神经网络.

**关键词:** 量子计算; 量子旋转门; 受控旋转门; 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:** 0372-2112 (2014)12-2401-08

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2014.12.010

## Algorithm and Application of the Quantum-Inspired Neural Network Model

YANG Shu-yun<sup>1</sup>, LI Pan-chi<sup>2</sup>

(1. Admission and Employment Office of Northeast Petroleum University, Daqing, Heilongjiang 163318, China;

2. School of Computer & Information Technology, Northeast Petroleum University, Daqing, Heilongjiang 163318, China)

**Abstract:** To enhance the approximation and generalization ability of classical artificial neural networks, a quantum-inspired neural network model, whose input of each dimension is a discrete sequence, is proposed. This model concludes three layers, in which the hidden layer consists of quantum-inspired neurons, and the output layer consists of common neurons. The quantum-inspired neuron consists of the quantum rotation gates and the multi-qubits controlled-rotation gates. By using the information feedback of target qubit from output to input in multi-qubits controlled-rotation gate, the overall memory of input sequences is realized. The output of quantum-inspired neuron is obtained from the entanglements of multi-qubits in controlled-rotation gates. The learning algorithm is designed in detail according to the basic principles of quantum computation. The characteristics of input sequence can be effectively obtained by way of “breadth” and “depth”. The simulation results show that, when the input nodes and the length of the sequence satisfy a certain relations, the proposed model is obviously superior to the common artificial neural networks.

**Key words:** quantum computation; quantum rotation gate; controlled-rotation gate; quantum-inspired neuron; quantum-inspired neural networks

## 1 引言

量子神经网络的概念出现于上个世纪 90 年代. 1995 年, Kak 发表“On Quantum Neural Computing”论文, 首次提出量子神经计算的概念<sup>[1]</sup>. 1997 年, Gopathy 等借用量子理论中量子态叠加的思想, 提出了基于多级激励函数的量子神经网络模型<sup>[2]</sup>. 1998 年, Dan 提出了量子联想存储的算法, 相对于传统存储量子联想存储具有指数级的存储容量<sup>[3]</sup>. 2000 年, Ajit 等从多宇宙的量子理论观点, 提出了构建叠加的多宇宙量子神经网络模型的思

想<sup>[4]</sup>. 2001 年, 中国科学技术大学的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<sup>[5]</sup>. 2004 年, 解光军等深入研究了以通用量子逻辑门组作为计算基函数, 构造量子神经网络模型的新方法<sup>[6,7]</sup>. 2007 年, Matsui 等根据量子线路结构提出一种求解异或问题的量子神经元学习算法<sup>[8]</sup>. 同年, 文献<sup>[9]</sup>提出一种基于量子输入和量子权值的量子自组织特征映射网络模型及聚类算法. 2008 年, 文献<sup>[10]</sup>基于量子旋转门和量子受控非门的物理意义, 提出一种基于通用量子门组演化的量子 BP 网络模型. 2009 年, 文献<sup>[11]</sup>提出一种基

于量子权值及量子活性值的量子神经网络模型. 2011年, 文献[12]基于量子门线路的物理意义, 提出一种量子门节点神经网络模型及算法. 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, 尽管量子计算的融入有效提高了神经计算的性能, 但是, 量子计算与神经计算的融合研究目前还远未成熟, 有必要进一步深入研究二者新的融合方式, 以期进一步提高神经计算的性能. 为此, 本文提出一种基于受控旋转门的量子神经网络(Controlled-Rotation-based Quantum Neural Network, CRQNN)模型, 该模型每个样本的输入为多维离散序列, 利用受控旋转门对目标位的循环控制, 可以实现对输入序列的整体记忆, 基于量子计算原理设计了学习算法. 该模型的核心优势在于信息的并行处理和网络参数的多维调整, 实验结果表明, 在一定条件下, 提出的模型明显优于普通 BP 神经网络.

## 2 量子比特及量子门概述

### 2.1 量子比特

在量子计算中, 量子比特有两个可能的状态 $|0\rangle$ 和 $|1\rangle$ , 与经典比特的区别在于, 量子比特的状态可以落在 $|0\rangle$ 和 $|1\rangle$ 之外, 即可以是状态的线性叠加态.

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

其中 $\alpha, \beta$ 是一对复数, 称为量子比特的概率幅, 量子比特也可用概率幅表示为 $|\varphi\rangle = [\alpha, \beta]^T$ .

### 2.1 量子旋转门

量子门是物理实现量子计算的基础, 它包含了量子计算特点. 量子旋转门定义式为<sup>[13]</sup>

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (2)$$

### 2.3 受控旋转门

在 $n+1$ 比特量子系统 $|x_1\rangle|x_2\rangle\cdots|x_n\rangle|\varphi\rangle$ 中, 当目标比特 $|\varphi\rangle$ 同时受 $n$ 位输入比特 $|x_1\rangle|x_2\rangle\cdots|x_n\rangle$ 的联合控制时, 其系统特性可用如下多比特量子受控旋转门描述, 如图1所示.

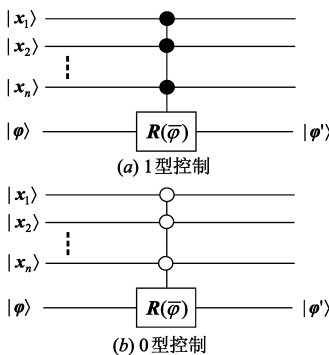


图1 多比特受控旋转门

对于图1(a), 当前 $n$ 个量子比特全为1时, 旋转门作用到最后一位量子比特, 运算关系可描述如下.

$$C^n(R)|x_1x_2\cdots x_n\rangle|\varphi\rangle = |x_1x_2\cdots x_n\rangle R^{x_1x_2\cdots x_n}|\varphi\rangle \quad (3)$$

令 $n$ 个控制比特为 $|x_i\rangle = \cos(\theta_i)|0\rangle + \sin(\theta_i)|1\rangle$ , 目标比特为 $|\varphi\rangle = \cos(\varphi)|0\rangle + \sin(\varphi)|1\rangle$ , 其中 $i = 1, 2, \dots, n$ . 则式(3)可重写为

$$\begin{aligned} C^n(R)|x_1x_2\cdots x_n\rangle|\varphi\rangle &= |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \otimes |\varphi\rangle \\ &+ \prod_{i=1}^n \sin(\theta_i) (\cos(\varphi + \bar{\varphi}) - \cos(\varphi)) \overbrace{|11\cdots 10\rangle} \\ &+ \prod_{i=1}^n \sin(\theta_i) (\sin(\varphi + \bar{\varphi}) - \sin(\varphi)) \overbrace{|11\cdots 11\rangle} \end{aligned} \quad (4)$$

$C^n(R)$ 中目标位的输出端 $|\varphi'\rangle$ 处于状态 $|1\rangle$ 的概率为

$$P = \prod_{i=1}^n \sin^2(\theta_i) (\sin^2(\varphi + \bar{\varphi}) - \sin^2(\varphi)) + \sin^2(\varphi) \quad (5)$$

对于图1(b), 当前 $n$ 个量子比特全为0时旋转门作用到最后一位量子比特.  $C^n(R)$ 运算关系可描述为

$$C^n(R)|x_1x_2\cdots x_n\rangle|\varphi\rangle = |x_1x_2\cdots x_n\rangle R^{x_1+x_2+\cdots+x_n}|\varphi\rangle \quad (6)$$

采用与图1(a)类似的分析方法可知,  $|\varphi'\rangle$ 处于状态 $|1\rangle$ 的概率为

$$P = \prod_{i=1}^n \cos^2(\theta_i) (\sin^2(\varphi + \bar{\varphi}) - \sin^2(\varphi)) + \sin^2(\varphi) \quad (7)$$

此时, 经 $n$ 个输入比特的联合控制后, 目标比特的输出可写为

$$|\varphi'\rangle = \sqrt{1-P}|0\rangle + \sqrt{P}|1\rangle \quad (8)$$

## 3 量子衍生神经元模型

本文提出的量子衍生神经元模型如图2所示. 其输入为量子比特序列 $\{|x_i(t_r)\rangle\}$ , 其中 $t_r \in [0, T]$ ; 输出为在 $[0, T]$ 上的聚合结果 $y$ ; 控制参数为量子旋转门的旋转角度 $\bar{\theta}_i(t_r)$ 和 $\bar{\varphi}(t_r)$ .

与普通神经元不同, 量子衍生神经元的每个输入样本用矩阵描述, 例如, 单个样本可描述为

$$\begin{bmatrix} |x_1(t_r)\rangle \\ |x_2(t_r)\rangle \\ \vdots \\ |x_n(t_r)\rangle \end{bmatrix} = \begin{bmatrix} |x_1(t_1)\rangle & |x_1(t_2)\rangle & \cdots & |x_1(t_q)\rangle \\ |x_2(t_1)\rangle & |x_2(t_2)\rangle & \cdots & |x_2(t_q)\rangle \\ \cdots & \cdots & \cdots & \cdots \\ |x_n(t_1)\rangle & |x_n(t_2)\rangle & \cdots & |x_n(t_q)\rangle \end{bmatrix} \quad (9)$$

设 $|x_i(t_r)\rangle = \cos\theta_i(t_r)|0\rangle + \sin\theta_i(t_r)|1\rangle$ ,  $|\varphi(t_1)\rangle = |0\rangle$ , 记

$$\bar{h}_r = \begin{cases} \prod_{i=1}^n \sin(\theta_i(t_r) + \bar{\theta}_i(t_r)), & \text{对应图 2(a)} \\ \prod_{i=1}^n \cos(\theta_i(t_r) + \bar{\theta}_i(t_r)), & \text{对应图 2(b)} \end{cases} \quad (10)$$

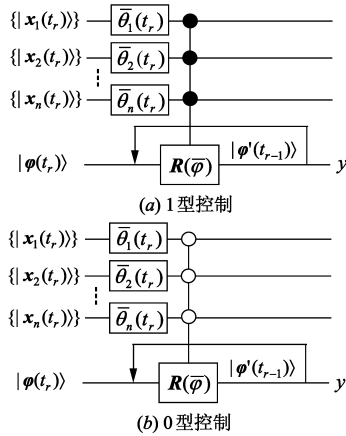


图2 量子衍生神经元模型

由量子旋转门和受控旋转门定义,  $|\varphi'(t_1)\rangle$  可写为

$$|\varphi'(t_1)\rangle = \sqrt{1 - (\bar{h}_1 \sin \bar{\varphi}(t_1))^2} |0\rangle + \bar{h}_1 \sin \bar{\varphi}(t_1) |1\rangle \quad (11)$$

根据  $|\varphi(t_r)\rangle = |\varphi'(t_{r-1})\rangle$ , 量子衍生神经元在  $[0, T]$  上的聚合结果为

$$|y\rangle = |\varphi'(t_q)\rangle = \cos \varphi(t_q) |0\rangle + \sin \varphi(t_q) |1\rangle \quad (12)$$

其中

$$\varphi(t_q) = \arcsin \left[ \left[ (\bar{h}_q)^2 (\sin^2(\varphi(t_{q-1}) + \bar{\varphi}(t_q)) - \sin^2(\varphi(t_{q-1}))) + \sin^2(\varphi(t_{q-1})) \right]^{0.5} \right] \quad (13)$$

在本文中, 我们定义量子衍生神经元的输出为目标比特处于状态  $|1\rangle$  的概率幅, 因此, 其输入输出关系可描述为

$$y = h(t_q) = \sqrt{(\bar{h}_2)^2 U_q + (h(t_{q-1}))^2} \quad (14)$$

$$\text{其中} \begin{cases} U_q = h(t_{q-1}) \sqrt{1 - (h(t_{q-1}))^2} \sin(2\varphi(t_q)) \\ \quad + (1 - 2(h(t_{q-1}))^2) \sin^2(\varphi(t_q)) \\ h(t_1) = \bar{h}_1 \sin(\bar{\varphi}(t_1)) \end{cases}$$

## 4 量子衍生神经网络模型

本文提出的基于受控旋转门的量子衍生神经网络模型由三层组成, 其中隐层为量子衍生神经元, 第 1, 3, 5, ... 个神经元采用 1 型控制, 第 2, 4, 6, ... 个神经元采用 0 型控制, 输出层为普通神经元, 如图 3 所示. 输出层采用 Sigmoid 函数作为激励函数.

设  $|x_i^l(t_r)\rangle = \cos \theta_i^l(t_r) |0\rangle + \sin \theta_i^l(t_r) |1\rangle$  为第  $l$  个样本. 令  $|\varphi_j(t_1)\rangle = |0\rangle$ , 记

$$\bar{h}_j^l = \begin{cases} \prod_{i=1}^n \sin(\theta_i^l(t_r) + \theta_{ij}(t_r)), j = 1, 3, 5, \dots \\ \prod_{i=1}^n \cos(\theta_i^l(t_r) + \theta_{ij}(t_r)), j = 2, 4, 6, \dots \end{cases} \quad (15)$$

根据量子神经元的输入输出关系, 隐层第  $j$  个量子衍生神经元的输出为

$$h_j^l = h_j^l(t_q) = \sqrt{(\bar{h}_j^l)^2 U_j^l + (h_j^l(t_{q-1}))^2} \quad (16)$$

$$\text{其中} \begin{cases} U_{ji}^l = h_j^l(t_{q-1}) \sqrt{1 - (h_j^l(t_{q-1}))^2} \sin(2\varphi_j(t_q)) \\ \quad + (1 - 2(h_j^l(t_{q-1}))^2) \sin^2(\varphi_j(t_q)) \\ h_j^l(t_1) = \bar{h}_{j1}^l \sin(\bar{\varphi}(t_1)) \end{cases}$$

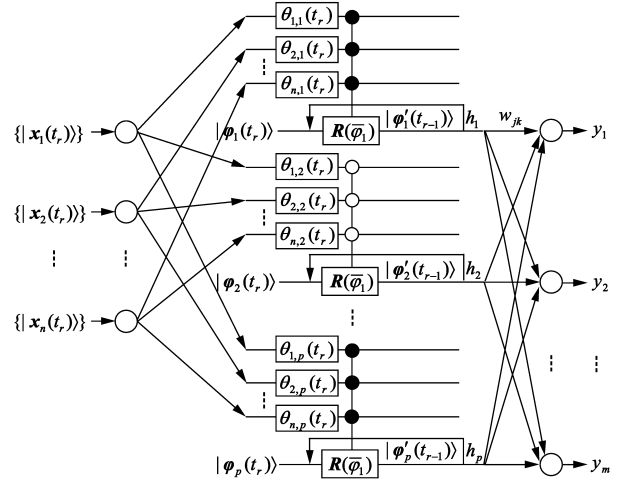


图3 量子衍生神经网络模型

输出层第  $k$  个普通神经元的输出为

$$y_k^l = \frac{1}{1 + e^{-\sum_{j=1}^p w_{jk} h_j^l}} \quad (17)$$

其中  $i = 1, 2, \dots, n, j = 1, 2, \dots, p, k = 1, 2, \dots, m, l = 1, 2, \dots, L, L$  表示样本总数.

## 5 量子衍生神经网络算法

### 5.1 样本量子态描述

令样本  $\{\bar{X}^l(t_r)\} = [\{x_1^l(t_r)\}, \{x_2^l(t_r)\}, \dots, \{x_n^l(t_r)\}]^T$ , 其中  $r = 1, 2, \dots, q, l = 1, 2, \dots, L$ , 记

$$\begin{cases} \text{Max}_{i,r} = \max(x_i^1(t_r), x_i^2(t_r), \dots, x_i^L(t_r)) \\ \text{Min}_{i,r} = \min(x_i^1(t_r), x_i^2(t_r), \dots, x_i^L(t_r)) \end{cases} \quad (18)$$

$$\theta_i^l(t_r) = \frac{\bar{x}_i^l(t_r) - \text{Min}_{i,r} \pi}{\text{Max}_{i,r} - \text{Min}_{i,r} 2} \quad (19)$$

这些样本可用量子态形式描述如下:

$$\{|\mathbf{X}^l(t_r)\rangle\} = [\{|\mathbf{x}_1^l(t_r)\rangle\}, \{|\mathbf{x}_2^l(t_r)\rangle\}, \dots, \{|\mathbf{x}_n^l(t_r)\rangle\}]^T \quad (20)$$

其中  $|\mathbf{x}_i^l(t_r)\rangle = \cos(\theta_i^l(t_r)) |0\rangle + \sin(\theta_i^l(t_r)) |1\rangle, i = 1, 2, \dots, n$ .

### 5.2 网络参数调整方法

网络参数包括隐层量子旋转门旋转角度  $\theta_{ij}(t_r)$  和  $\bar{\varphi}_j(t_r)$ , 输出层连接权  $w_{jk}$ . 设归一化后的期望输出为  $\bar{y}_1^l, \bar{y}_2^l, \dots, \bar{y}_m^l$ , 对应的实际输出为  $y_1^l, y_2^l, \dots, y_m^l$ , 误差函数定义为

$$E = \max_{1 \leq l \leq L} \max_{1 \leq k \leq m} |e_k^l| = \max_{1 \leq l \leq L} \max_{1 \leq k \leq m} |\bar{y}_k^l - y_k^l| \quad (21)$$

根据梯度下降法, 隐层旋转角的梯度计算式为

$$\frac{\partial e_k^l}{\partial \theta_{ij}(t_r)} = \begin{cases} \frac{-y_k^l(1-y_k^l)w_{jk} \prod_{s=r+1}^q (V_{js}^l + \bar{V}_{js}^l) (\bar{h}_{jr}^l)^2 (U_{jr}^l + \bar{U}_{jr}^l) \cot(\theta_i^l(t_r) + \theta_{ij}(t_r))}{h_j^l(t_r) \prod_{s=r+1}^q 2h_j^l(t_s) \sqrt{1-(h_j^l(t_{s-1}))^2}}, j = 1, 3, 5, \dots \\ \frac{y_k^l(1-y_k^l)w_{jk} \prod_{s=r+1}^q (V_{js}^l + \bar{V}_{js}^l) (\bar{h}_{jr}^l)^2 (U_{jr}^l + \bar{U}_{jr}^l) \tan(\theta_i^l(t_r) + \theta_{ij}(t_r))}{h_j^l(t_r) \prod_{s=r+1}^q 2h_j^l(t_s) \sqrt{1-(h_j^l(t_{s-1}))^2}}, j = 2, 4, 6, \dots \end{cases} \quad (22)$$

$$\frac{\partial e_k^l}{\partial \varphi_j(t_r)} = \frac{-y_k^l(1-y_k^l)w_{jk} \prod_{s=r+1}^q (V_{js}^l + \bar{V}_{js}^l) (\bar{h}_{jr}^l)^2 (W_{jr}^l + \bar{W}_{jr}^l)}{h_j^l(t_r) \prod_{s=r+1}^q 2h_j^l(t_s) \sqrt{1-(h_j^l(t_{s-1}))^2}} \quad (23)$$

其中

$$\begin{cases} \bar{h}_{jr}^l = \begin{cases} \prod_{i=1}^n \sin(\theta_i^l(t_r) + \theta_{ij}(t_r)), j = 1, 3, 5, \dots \\ \prod_{i=1}^n \cos(\theta_i^l(t_r) + \theta_{ij}(t_r)), j = 2, 4, 6, \dots \end{cases} \\ U_{jr}^l = h_j^l(t_{r-1}) \sqrt{1-(h_j^l(t_{r-1}))^2} \sin(2\varphi_j(t_r)) \\ \bar{U}_{jr}^l = (1-2(h_j^l(t_{r-1}))^2) \sin^2(\varphi_j(t_r)) \\ V_{jr}^l = (1-2(h_j^l(t_{r-1}))^2) (\bar{h}_{jr}^l)^2 \sin(2\varphi_j(t_r)) \\ \bar{V}_{jr}^l = 2h_j^l(t_{r-1}) (1-2(\bar{h}_{jr}^l)^2 \sin^2(\varphi_j(t_r))) \sqrt{1-(h_j^l(t_{r-1}))^2} \\ W_{jr}^l = h_j^l(t_{r-1}) \sqrt{1-(h_j^l(t_{r-1}))^2} \cos(2\varphi_j(t_r)) \\ \bar{W}_{jr}^l = 0.5(1-2(h_j^l(t_{r-1}))^2) \sin(2\varphi_j(t_r)) \end{cases} \quad (24)$$

输出层连接权的梯度计算式为

$$\frac{\partial e_k^l}{\partial w_{jk}} = -y_k^l(1-y_k^l)h_j^l(t_q) \quad (25)$$

因梯度计算式较为复杂,故梯度下降法收敛较慢或不易收敛,因此本文采用 L-M 算法.记  $\mathbf{X}$  为参数向量,  $\mathbf{V}$  为误差向量,  $\mathbf{J}$  为雅可比矩阵,分别定义如下.

$$\mathbf{X}^T = [\theta_{11}(t_1), \dots, \theta_{mp}(t_q), \bar{\varphi}_1(t_1), \dots, \bar{\varphi}_p(t_q), w_{11}, \dots, w_{pm}] \quad (26)$$

$$\mathbf{V}^T(\mathbf{X}) = [e_1^1, \dots, e_m^1, e_1^2, \dots, e_m^2, \dots, e_1^l, \dots, e_m^l] \quad (27)$$

$$\mathbf{J}(\mathbf{X}) = \begin{bmatrix} \frac{\partial e_1^1}{\partial \theta_{11}(t_1)}, \dots, \frac{\partial e_1^1}{\partial \theta_{mp}(t_q)}, \frac{\partial e_1^1}{\partial \varphi_1(t_1)}, \dots, \frac{\partial e_1^1}{\partial \varphi_p(t_q)}, \frac{\partial e_1^1}{\partial w_{11}}, \dots, \frac{\partial e_1^1}{\partial w_{pm}} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_m^1}{\partial \theta_{11}(t_1)}, \dots, \frac{\partial e_m^1}{\partial \theta_{mp}(t_q)}, \frac{\partial e_m^1}{\partial \varphi_1(t_1)}, \dots, \frac{\partial e_m^1}{\partial \varphi_p(t_q)}, \frac{\partial e_m^1}{\partial w_{11}}, \dots, \frac{\partial e_m^1}{\partial w_{pm}} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_1^l}{\partial \theta_{11}(t_1)}, \dots, \frac{\partial e_1^l}{\partial \theta_{mp}(t_q)}, \frac{\partial e_1^l}{\partial \varphi_1(t_1)}, \dots, \frac{\partial e_1^l}{\partial \varphi_p(t_q)}, \frac{\partial e_1^l}{\partial w_{11}}, \dots, \frac{\partial e_1^l}{\partial w_{pm}} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_m^l}{\partial \theta_{11}(t_1)}, \dots, \frac{\partial e_m^l}{\partial \theta_{mp}(t_q)}, \frac{\partial e_m^l}{\partial \varphi_1(t_1)}, \dots, \frac{\partial e_m^l}{\partial \varphi_p(t_q)}, \frac{\partial e_m^l}{\partial w_{11}}, \dots, \frac{\partial e_m^l}{\partial w_{pm}} \end{bmatrix} \quad (28)$$

根据 L-M 算法, CRQNN 参数调整的迭代方程为

$$\mathbf{X}_{t+1} = \mathbf{X}_t - [\mathbf{J}^T(\mathbf{X}_t)\mathbf{J}(\mathbf{X}_t) + \mu_t \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{X}_t) \mathbf{V}(\mathbf{X}_t) \quad (29)$$

其中  $t$  为迭代步数,  $\mathbf{I}$  为单位矩阵,  $\mu_t$  为一小正数, 以使  $\mathbf{J}^T(\mathbf{X}_t)\mathbf{J}(\mathbf{X}_t) + \mu_t \mathbf{I}$  可逆.

### 5.3 网络的收敛性

1989 年, 关于两层前馈误差反向传播神经网络的收敛性, Hecht-Nielsen 证明了如下引理<sup>[14]</sup>.

**引理** 具有两层连接权值的误差反向传播神经网络, 在隐层神经元节点足够多的情况下能够逼近  $[0, 1]$  上的任意  $L_2$  函数.

关于本文基于受控旋转门的量子衍生神经网络的收敛性, 我们给出如下结论.

**定理** 当隐层神经元足够多时, CRQNN 能够逼近  $[0, 1]$  上的任意  $L_2$  函数.

**证明** 由 CRQNN 隐层量子衍生神经元输出式(16)可知, 隐层输出  $h_j^l = h_j^l(t_q)$  为旋转角度的连续可微函数. 由于  $h_j^l$  为目标比特处于状态  $|1\rangle$  的概率幅, 所以  $-1 \leq h_j^l \leq 1$ , 即  $h_j^l$  有界. 因此量子神经元可视为传统神经元在激励函数、聚合方式两方面的推广. CRQNN 的输出层就是普通神经元. 因此 CRQNN 可视为一种特殊形式的两层神经网络. 当  $\mu_t$  较大时, L-M 算法接近于如下有小的学习速度的最速下降算法

$$\mathbf{X}_{t+1} \cong \mathbf{X}_t - \frac{1}{\mu_t} \mathbf{J}^T(\mathbf{X}_t) \mathbf{V}(\mathbf{X}_t) = \mathbf{X}_t - \frac{1}{\mu_t} \nabla E(\mathbf{X}_t) \quad (30)$$

而当  $\mu_t$  较小时, L-M 算法接近于高斯牛顿算法, 该算法较标准牛顿算法的优点是不需要计算二阶导数. 根据最优化理论, 最速下降法和牛顿法均可用于神经网络训练, 实质上是误差反向传播算法中参数调整量两种具体计算形式, 区别在于牛顿法比最速下降法收敛速度快, 但梯度计算较为复杂. 因此, L-M 算法兼有最速下降法和牛顿法的优点, 从而 CRQNN 本质上也是一种误差反向传播神经网络. 根据引理, 定理得证.

### 5.4 序列样本的构造方法

对于适合传统神经网络的向量式样本, 必须首先转化为矩阵式的序列样本, 才能提交 CRQNN 训练. 我们给出的转化方法是: 对于传统神经网络的一个  $n$  维向量样本  $\mathbf{X}$ , 第一步: 如果  $n$  是素数, 令  $m = n + 1$ , 并通过使  $\mathbf{X}(m) = \mathbf{X}(n)$  将该样本扩展到  $m$  维, 否则, 令  $m = n$ ; 第二步: 将  $m$  分解为两因子  $m_1$  和  $m_2$  的乘积, 并使  $m_1$  和  $m_2$  尽可能接近. 此时, 一个  $n$  维向量样本即可被转化为  $m_1$  维序列样本, 每维序列长度为  $m_2$ , 或  $m_2$  维序列样本, 每维序列长度为  $m_1$ , 进而可以提交

CRQNN 进行训练.

## 6 实际应用及分析

本节通过两个实验并与采用 L-M 算法的 BP 神经网络(BP neural network, BPNN)对比,验证 CRQNN 的优越性.为便于对比,首先定义如下预测指标.

**预测正确数** 设  $\bar{y}^1, \bar{y}^2, \dots, \bar{y}^M$  表示  $M$  个样本的期望输出,  $y^1, y^2, \dots, y^M$  表示对应的实际输出,  $M$  表示训练集样本数,训练过程独立重复  $N$  次.训练集的预测正确数定义为

$$N_{tr} = \sum_{n=1}^N \sum_{m=1}^M \delta(\bar{y}^m - [y^m]) / N \quad (31)$$

同理,测试集的预测正确数定义为

$$N_{te} = \sum_{n=1}^N \sum_{m=1}^M \delta(\bar{y}^m - [y^m]) / N \quad (32)$$

其中  $\bar{M}$  表示预测集样本数,当  $x = 0$  时  $\delta(x) = 0$ , 否则  $\delta(x) = 1$ .  $[y^m]$  为将  $y^m$  按四舍五入取整.

**预测正确率** 训练集和预测集正确率分别定义为

$$R_{tr} = 100 N_{tr} / M \quad (33)$$

$$R_{te} = 100 N_{te} / \bar{M} \quad (34)$$

**平均运行时间** 设  $N$  次独立的训练过程,运行时间分别为  $T_1, T_2, \dots, T_N$ , 平均运行时间定义为

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i \quad (35)$$

### 6.1 乳腺癌复发预测

本实验用 CRQNN 预测乳腺癌患者手术后的复发情况.样本数据来源于文献[15].全部样本分为两类,术

后 24 个月内复发患者为第一类(标记为 1),未复发患者为第二类(标记为 2).输入样本由患者的 33 个症状特征值组成,第 1 个为复发月数,第 2 个为被切除的肿瘤直径,第 3 个为阳性淋巴结数量,第 4 ~ 33 个特征来源于对乳腺肿块实施细针穿刺的数字化图像.样本总数为 194,其中复发 46 例;不复发 148 例.随机挑选 29 例复发和 91 例不复发组成 120 个样本的训练集,余下的 74 个样本(17 例复发和 57 例不复发)作为测试集.对于 CRQNN 和 BPNN 的输入节点,我们采取表 1 给出的 4 种情形.

表 1 两种模型输入节点和序列长度设置

模型标识	CRQNN		BPNN	
	输入节点	序列长度	模型标识	输入节点
CRQNN1_33	1	33	BPNN	33
CRQNN3_11	3	11	BPNN	33
CRQNN11_3	11	3	BPNN	33
CRQNN33_1	33	1	BPNN	33

对于训练过程,我们不设置误差精度,仅设置最大迭代步数为  $N = 50$ , L-M 参数为  $\mu_l = 0.10$ .为充分对比两种模型的预测能力,两种模型的隐层分别取 5, 10,  $\dots$ , 15 个节点.对于输入节点和隐层神经元节点的每种组合,两种模型分别用训练集训练 10 次,每次训练之后立即用测试集检验模型的预测能力,预测结果对比如表 2 所示.

表 2 CRQNN 与 BPNN 预测结果对比

CRQNN	指标	隐层神经元										
		10	11	12	13	14	15	16	17	18	19	20
1_33	$R_{tr}(\%)$	50.00	60.33	60.33	44.83	65.50	50.00	55.17	39.67	44.83	55.17	50.00
	$R_{te}(\%)$	50.00	60.81	60.81	44.59	66.22	50.00	55.41	39.19	44.59	55.41	50.00
	$T_{avg}(s)$	4.330	5.420	6.620	8.100	9.580	11.21	13.04	15.01	16.47	18.63	21.09
3_11	$R_{tr}(\%)$	78.83	80.08	81.83	82.75	80.08	81.83	84.92	84.00	85.42	82.83	84.08
	$R_{te}(\%)$	76.35	76.89	77.30	76.35	76.49	76.49	73.65	76.49	75.14	77.30	76.76
	$T_{avg}(s)$	3.250	4.080	5.050	6.210	7.490	8.880	10.47	12.27	14.15	16.27	18.33
11_3	$R_{tr}(\%)$	78.42	77.33	80.08	79.67	80.50	79.67	80.33	82.00	83.17	80.42	82.08
	$R_{te}(\%)$	74.19	67.70	75.81	72.70	74.32	74.73	72.16	71.89	70.14	72.70	72.43
	$T_{avg}(s)$	2.920	3.670	4.560	5.610	6.760	8.050	9.490	11.11	12.93	14.85	17.05
33_1	$R_{tr}(\%)$	47.50	50.10	46.80	50.70	51.30	49.10	48.90	52.40	47.80	50.10	53.70
	$R_{te}(\%)$	50.11	51.38	49.04	50.64	50.53	49.15	45.74	47.02	45.64	50.43	52.55
	$T_{avg}(s)$	5.230	6.610	8.260	10.19	12.39	14.87	17.64	20.83	24.34	28.10	32.01
BPNN	$R_{tr}(\%)$	77.58	73.42	70.33	74.08	71.75	75.75	72.17	81.58	74.50	80.08	66.42
	$R_{te}(\%)$	69.86	65.00	61.08	63.92	53.92	68.24	66.62	72.03	65.68	74.86	58.11
	$T_{avg}(s)$	1.800	2.340	2.990	3.790	4.720	5.920	7.080	8.530	9.960	11.64	13.60

根据表 2, CRQNN3\_11 和 CRQNN11\_3 的两种预测正确率明显优于 BPNN, 而 CRQNN1\_33 和 CRQNN33\_1 的两种预测正确率明显劣于 BPNN. 预测结果表明, 当输入节点和序列长度差距较大时, CRQNN 劣于 BPNN, 而当输入节点和序列长度较接近时, CRQNN 优于 BPNN. 另一方面, 四种 CRQNN 的平均运行时间明显长于 BPNN. 这是由于 CRQNN 的输出及梯度计算都比较复杂, 在提高期预测能力的同时, 必然会导致计算效率下降的缘故.

## 6.2 波形识别

本实验采用神经网络方法识别带有噪声的三类波形(分别用 0, 1, 2 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 200 个作为训练集(三类分别为 63, 68, 69), 后 300 个作为测试集(三类分别为 101, 98, 101). 对于 CRQNN 和 BPNN 的输入节点, 我们采取表 3 给出的 8 种情形.

表 3 两种模型输入节点和序列长度设置

CRQNN			BPNN	
模型标识	输入节点	序列长度	模型标识	输入节点
CRQNN1_40	1	40	BPNN	40
CRQNN2_20	2	20	BPNN	40
CRQNN4_10	4	10	BPNN	40
CRQNN5_8	5	8	BPNN	40
CRQNN8_5	8	5	BPNN	40
CRQNN10_4	10	4	BPNN	40
CRQNN20_2	20	2	BPNN	40
CRQNN40_1	40	1	BPNN	40

设置误差精度为 0.25, 最大迭代步数为  $N = 100$ , L-M 参数为  $\mu_i = 0.10$ . 为充分对比两种模型的预测能力, 两种模型的隐层分别取 10, 11,  $\dots$ , 20 个节点. 对于输入节点和隐层神经节点的每种组合, 两种模型分别用训练集训练 10 次, 每次训练之后立即用测试集检验模型的识别能力, 识别结果对比如表 4 所示.

根据表 4, CRQNN4\_10、CRQNN5\_8、CRQNN8\_5、CRQNN10\_4 的识别正确率明显高于 BPNN, 而 CRQNN1\_40 和 CRQNN40\_1 的识别正确率明显低于 BPNN. 识别结果再次表明, 当输入节点和序列长度差距较大时, CRQNN 劣于 BPNN, 而当输入节点和序列长度较接近时, CRQNN 优于 BPNN. 另一方面, CRQNN4\_10、CRQNN5\_8、CRQNN8\_5、CRQNN10\_4 的平均运行时间明显少于 BPNN. 这是由于我们设置了误差精度, 四种

CRQNN 能够较快达到精度要求, 从而使迭代步数明显少于 BPNN 的缘故.

下面, 我们对以上两个实验结果给出必要的理论分析.

(1) 与 BPNN 比较, CRQNN 的核心优势在于信息的并行处理和网络参数的多维调整. 信息的并行处理, 可以使 CRQNN 同时接收多维空间中序列式的矩阵信息, 而 BPNN 只能同时接收多维空间中几何点式的向量信息, 因此 CRQNN 具有更强的信息获取能力. 网络参数的多维调整, 可以使 CRQNN 更为准确的把握网络训练的收敛方向和更为精细的计算网络参数的搜索步长, 从而可以提高网络的逼近能力.

(2) CRQNN 和 BPNN 对输入信息采取了完全不同的处理方式, CRQNN 直接接收离散序列, 利用量子信息处理机制, 将输入序列中的数据循环地映射为隐层量子受控旋转门的输出, 由于多位受控旋转门中目标位的受控关系远比 BPNN 的 Sigmoid 函数复杂, 所以这种映射具有高度的非线性, 并使 CRQNN 有更强的预测能力.

(3) 设  $n$  表示输入节点数,  $q$  表示序列长度, CRQNN 的每个样本可以描述为一个  $n$  行  $q$  列的矩阵, 从 CRQNN 算法可以看出, 对于  $n$  和  $q$  的不同组合, 隐层量子神经元的计算结果也是不同的. 事实上, 输入节点个数  $n$  可以视为模式记忆的维数, 主要体现为提取信息的多路性; 而离散序列长度  $q$  可以视为对每一维模式记忆的强化次数, 主要体现为通过反馈不断对各路信息实施强化记忆. 当维数和强化次数适当匹配时, CRQNN 呈现出明显优于 BPNN 的性能. 当  $q = 1$  时, 输入节点是最多的, 隐层量子神经元输出的计算式也是最简单的, 但此时网络只能以多维方式提取模式信息, 而不能对每一维信息实施强化记忆, 因此, 直接导致了逼近能力的降低; 当  $n = 1$  时, 隐层量子神经元输出的计算是最复杂的, 虽然此时可以提高网络的映射能力, 然而此时网络参数调整的计算式也是最复杂的, 过长的链式求导计算会使梯度趋于零或无穷, 从而阻碍收敛进程, 并导致逼近能力的降低. 因此, 当  $q = 1$  或  $n = 1$  时, CRQNN 的逼近能力弱于 BPNN, 而当  $q > 1$  和  $n > 1$  时, CRQNN 的逼近能力趋于改进, 并且在一定条件下必然会超越 BPNN. 关于  $q$  和  $n$  究竟满足怎样的精确关系, CRQNN 的逼近能力才是最强的, 这个问题尚需进一步深入研究.

(4) 对于 BPNN, 由于不能直接处理序列式输入, 因此每个样本只能表述为一个  $n$  维向量, 即只能以多维方式获取样本信息而不能对每一维实施强化记忆, 因此在 BPNN 的信息处理过程中, 其信息提取能力不可避免地受到影响, 从而降低了网络的逼近及预测能力.

表 4 CRQNN 与 BPNN 识别结果对比

CRQNN	指标	隐层神经元										
		10	11	12	13	14	15	16	17	18	19	20
1_40	$R_{tr}(\%)$	33.65	34.10	33.45	34.10	33.35	34.05	33.20	34.40	34.00	33.35	33.05
	$R_{te}(\%)$	32.90	32.83	32.87	32.80	34.60	33.17	33.30	32.97	33.73	32.27	33.33
	$T_{avg}(s)$	96.54	112.5	128.5	146.3	165.5	189.5	215.9	245.9	276.1	308.2	344.7
2_20	$R_{tr}(\%)$	100.0	100.0	100.0	100.0	100.0	93.45	100.0	93.45	93.45	100.0	100.0
	$R_{te}(\%)$	60.17	60.83	60.13	59.47	61.13	58.50	59.50	57.63	57.27	59.80	61.23
	$T_{avg}(s)$	10.42	9.087	11.79	10.61	13.59	20.85	12.93	25.68	27.76	15.95	19.27
4_10	$R_{tr}(\%)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	$R_{te}(\%)$	59.33	58.27	57.33	58.73	57.97	57.13	58.90	58.27	57.80	58.27	57.77
	$T_{avg}(s)$	2.565	3.182	3.282	4.454	4.546	5.824	5.591	5.863	6.934	7.524	7.653
5_8	$R_{tr}(\%)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	$R_{te}(\%)$	57.47	55.73	55.17	56.17	56.80	57.77	56.43	55.47	56.73	56.73	56.77
	$T_{avg}(s)$	2.097	2.558	3.296	3.367	3.727	4.610	4.229	5.532	6.627	6.575	6.650
8_5	$R_{tr}(\%)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	$R_{te}(\%)$	60.03	60.43	60.10	60.07	60.90	60.90	60.63	60.57	60.13	60.67	60.60
	$T_{avg}(s)$	2.691	2.984	5.309	3.903	4.597	4.916	4.897	5.215	6.476	6.859	7.614
10_4	$R_{tr}(\%)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	$R_{te}(\%)$	60.67	61.03	59.77	61.57	60.30	61.83	60.23	62.30	61.37	62.80	60.90
	$T_{avg}(s)$	2.796	3.505	3.923	3.880	5.660	5.047	6.092	5.806	8.245	8.441	8.569
20_2	$R_{tr}(\%)$	41.60	51.40	44.45	47.90	49.70	57.85	67.10	58.00	63.45	74.45	90.90
	$R_{te}(\%)$	39.57	41.90	40.37	41.50	43.23	47.40	48.20	44.87	47.23	50.07	55.97
	$T_{avg}(s)$	14.30	17.00	19.81	23.76	28.02	31.61	35.34	39.84	45.87	50.44	57.41
40_1	$R_{tr}(\%)$	34.00	34.00	34.00	34.00	34.00	34.00	34.00	34.00	34.00	34.00	34.00
	$R_{te}(\%)$	32.67	32.67	32.67	32.67	32.67	32.67	32.67	32.67	32.67	32.67	32.67
	$T_{avg}(s)$	14.49	16.99	19.90	23.71	27.52	31.33	34.98	39.03	44.27	49.24	55.90
BPNN	$R_{tr}(\%)$	59.10	72.75	59.80	73.45	59.70	53.20	52.80	40.15	59.80	53.15	53.55
	$R_{te}(\%)$	46.80	54.03	46.63	50.93	48.13	42.50	43.67	36.70	45.67	41.70	43.70
	$T_{avg}(s)$	8.231	8.697	9.181	9.731	15.81	18.40	23.55	27.25	22.46	35.51	32.07

## 7 结论

本文基于量子计算原理提出了一种新型量子衍生神经网络模型,不同于普通量子神经网络,该模型各维输入为量子比特序列.实验结果表明,利用量子受控旋转门的信息处理机制,该模型可以从多维度和对每一维实施强化记忆两方面高效的获取样本信息,当输入节点和序列长度比较接近时可明显提高其逼近和泛化能力.从而揭示出量子计算与神经计算的融合切实能够改进神经网络的性能.关于该模型的连续性、计算复杂度等是我们下一步将要深入研究的问题.

### 参考文献

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

[2] Gopathy P, Nicolaos B. Quantum neural networks (QNNs) inherently fuzzy feedforward neural networks[J]. IEEE Transactions on Neural Networks, 1997, 8(3): 679 - 693.

[3] Ventura D, Tony M. Quantum associative memory with exponential capacity [A]. Proceedings of the IEEE International Joint Conference on Computational Intelligence [C]. Piscat-

away, NJ: IEEE, 1998. 509 - 513.

[4] Ajit N, 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 Guangjun, Zhuang Zhenquan. Quantum neural networks [J]. Computer Science, 2001, 28(7): 1 - 6. (in Chinese)

[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 Guangjun, Fan Haiqiu, Cao Licheng. A quantum neural computational network model[J]. Journal of Fudan University (Natural Science), 2004, 43(5): 700 - 703. (in Chinese)

[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 Guangjun, Zhou Dian, Fan Haiqiu. A neural network model based on quantum gates cell and its Applications[J]. Systems Engineering Theory & Practice, 2005, 25(5): 113 - 117. (in Chinese)

[8] Maeda M, Suenaga M, Miyajima H. Qubit neuron according to quantum circuit for XOR problem[J]. Applied Mathematics and Computation, 2007, 185(2): 1015 - 1025.

- [9] 李盼池, 李士勇. 一种量子自组织特征映射网络模型及聚类算法[J]. 量子电子学报, 2007, 24(4): 463 – 468.
- Li Panchi, Li Shiyong. A quantum self-organization feature mapping networks and clustering algorithm[J]. Chinese Journal of Quantum Electronics, 2007, 24(4): 463 – 468. (in Chinese)
- [10] Li P C, 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.
- [11] 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 Panchi. A learning algorithm and its applications to the quantum neural network model[J]. Control Theory & Applications, 2009, 26(5): 531 – 534. (in Chinese)
- [12] Li P C, Song K P, Yang E L. Model and algorithm of neural networks with quantum gated nodes [J]. Neural Network world, 2010, 20(2): 189 – 206.
- [13] Giuliano B, Giulio C, Giuliano S. Principles of Quantum Com-

putation and Information Volume I: Basic Concepts[M]. Singapore: World Scientific, 2004.

- [14] Hecht-Nielsen R. Theory of the back propagation neural network[A]. Proceedings of the International Joint Conference on Neural Networks[C]. New York: IEEE Communications Society, 1989. 593 – 605.
- [15] Mangasarian O L, Street W N, Wolberg W H. Breast cancer diagnosis and prognosis via linear programming[J]. Operations Research, 1995, 43(4): 570 – 577.

#### 作者简介

杨淑云 女, 1968 年 09 月生, 黑龙江省安达市人, 2013 年毕业于东北石油大学成人教育学院, 获学士学位, 现为讲师, 研究方向为神经网络和优化算法.

E-mail: yangshuyun68@126.com

李盼池 男, 1969 年 11 月生, 河北省大城县人, 2009 年毕业于哈尔滨工业大学, 获博士学位, 现为教授, 研究方向为量子搜索、量子神经网络、量子优化算法.

E-mail: lipanchi@vip.sina.com

# 基于图形电磁学雷达散射截面计算方法之改进

崔俊伟, 杨 颢

(大连理工大学船舶工程学院, 辽宁大连 116024)

**摘 要:** 图形电磁学(Graphical Electromagnetic Computing, GRECO)利用图形加速卡和 Z-Buffer 技术可较为有效地解决传统电磁计算方法中存在的消隐困难和非可视化难题,是求解高频电大尺寸目标特性最有效的方法之一.但传统 GRECO 算法存在着无法精确提取目标法矢信息、计算精度依赖屏幕分辨率和多次反射计算困难等缺点,限制了这种方法的使用.本文针对 GRECO 方法就精确提取像素几何信息方法进行了简要改进,将其与基于帧缓存对象(Frame-Buffer Object, FBO)的离屏渲染技术相结合,提出了改进的 GRECO 算法,克服了传统 GRECO 算法无法精确提取像素法矢信息和计算精度依赖屏幕分辨率的缺点.进而,采用 AP/PO(Area Projection/Physical Optics)法,并对传统的多次散射面元对判别方法进行了适当改进,实现了对产生多次反射目标的雷达截面计算.

**关键词:** 雷达截面积; 图形电磁学; 多次反射; 帧缓存对象; 离屏渲染

**中图分类号:** TN911.23 **文献标识码:** A **文章编号:** 0372-2112 (2014)12-2409-06

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2014.12.011

## Improvement of Radar Cross Section Computing Method Based on GRECO

CUI Jun-wei, YANG Yang

(Department of Naval Architecture, Dalian University of Technology, Dalian, Liaoning 116024, China)

**Abstract:** Based on 3-D graphics hardware accelerator and Z-Buffer technique, graphic electromagnetic computing (GRECO) algorithm can efficiently resolve blanking difficulties and visualization problems of traditional electromagnetic calculation procedures. Therefore, GRECO algorithm is considered as one of the most efficient methods to acquire characteristics of high-frequency and electricity large-sized target. However, there are disadvantages of traditional GRECO algorithm, as follows: normal vector of target cannot be extracted accurately, calculation accuracy is affected by screen resolution greatly and multiple reflections cannot be calculated directly. As a result, traditional GRECO algorithm is limited for this reason and cannot be used in some region widely. The traditional GRECO is improved in this paper, so that the geometric information of pixel can be extracted accurately. Technique of off-screen rendering based on frame buffer object (FBO) is used for improving the algorithm. Then the normal vector of target can be obtained precisely and effectively. Traditional discriminated method of facet pairs is improved by using area projection/physical optics to adapt the computation of RCS multiple scattering.

**Key words:** radar cross section (RCS); graphical electromagnetic computing; multi-reflection; FBO; off-screen rendering

## 1 引言

自西班牙学者 J. M. Rius 等人<sup>[1]</sup>于 1993 年提出“图形电磁计算方法”以来,由于其在求解高频电大尺寸目标散射特性方面无可比拟的优势,而受到人们的青睐.图形电磁算法将传统高频算法和计算机图形学结合起来,通过使用图形显卡的快速消隐功能来获取目标的可见部分,省去了传统高频算法中极为耗时的可见面判别过程,极大地提高了计算效率.在此基础上,利用物理光学法(Physical Optics, PO)、利用物理绕射法(Physical The-

ory of Diffraction, PTD)等算法完成对目标 RCS 的仿真工作.

然而,传统 GRECO 算法也存在很多不足:(1)无法精确获取目标的几何参数;(2)传统算法中积分是在屏幕显示的可见面元上进行的,导致其计算精度严重依赖于屏幕分辨率的大小;(3)目标多次反射计算困难.为了解决上述不足,众多学者提出了相应的改进措施,刘佳<sup>[2]</sup>提出了一种精确提出像素几何参数的新方法,并将 GO/PO 法引入到 GRECO 法中,初步解决了传统 GRECO 算法多次散射计算困难的问题.严靖峰和徐鹏根<sup>[3]</sup>提出