• 量子信息专栏 •

基于自适应网络的量子模糊推理系统

闫丽丽*，颜金歌，张仕斌

(成都信息工程大学网络空间安全学院 成都 610225)

【摘要】基于 ANFIS 与量子 BP 神经网络 (QBP) 提出了一种基于自适应网络的量子模糊推理系统 (ANQFIS)。不同于 ANFIS，ANQFIS 以量子门旋转的方式将模糊规则强度与 QBP 相结合，最后以量子态的测量概率作为输出，QBP 的加入使得模型的输出准确率更高，且凭借量子计算的速度优越性提升了模型的计算速度。根据梯度下降法，给出了该系统中参数的学习算法。在仿真实验中，分别使用低维数据和高维数据作为数据集来训练模型，使用攻击算法生成对抗样本进行测试，结果表明 ANQFIS 在输出准确率、鲁棒性方面优于 ANFIS 与 QBP。

关 键 词 量子神经网络； 模糊神经网络； 量子机器学习； 量子计算； 量子模糊机器学习

中图分类号 TP273+.2 **文献标志码** A doi:10.12178/1001-0548.2022220

Adaptive Network-Based Quantum Fuzzy Inference System

YAN Lili*, YAN Jinge, and ZHANG Shibin

(School of Cybersecurity, Chengdu University of Information Technology Chengdu 610225)

Abstract In this paper, a quantum fuzzy inference system based on adaptive network (ANQFIS) is proposed based on ANFIS and quantum BP (QBP) neural network. Different from ANFIS, ANQFIS combines the strength of fuzzy rules with QBP in the way of quantum gate rotation, and finally takes the measurement probability of quantum states as the output. The addition of QBP makes the output accuracy of the model higher, and the calculation speed of the model is improved by virtue of the speed advantage of quantum computing. According to the gradient descent method, the parameters learning algorithm of the system is given. In the simulation experiment, low-dimensional data and high-dimensional data are used as data sets to train the model, and attack algorithms are used to generate adversarial examples for testing. The results show that ANQFIS is superior to ANFIS and QBP in output accuracy and robustness.

Key words fuzzy neural network; quantum computation; quantum fuzzy machine learning; quantum machine learning; quantum neural network

模糊集指边界不明确的集合。1965 年，Zadeh 教授首次提出模糊集的概念，他提出用模糊数学来解决模糊问题[1]。模糊数学在实践中运用数学方法来研究和处理大量的不确定问题。近年来，机器学习领域[2]发展迅速，其中人工神经网络算法[3]得到了广泛的研究和应用。神经网络算法可以模拟人脑神经元对某些事物做出判断的功能。只要用训练数据集对神经网络模型进行训练，对模型参数进行更新和优化，模型就有可能对输入数据做出正确的判断。然而，随着大数据的发展，现实生活中出现了很多模糊数据。传统的神经网络不能很好地处理一些模糊问题，模型输出精度不高。因此，一些研究人员将模糊数学与神经网络相结合。提出了 BP 模糊神经网络[4]、基于自适应网络的模糊推理系统 (a fuzzy inference system based on adaptive network, ANFIS)[5] 和 B-spline 模糊神经网络。这些神经网络吸收了模糊逻辑和神经网络的优点，在处理非线性和模糊问题方面具有一定的优势。

机器学习算法通常需要对大数据进行处理，这使得它们的执行时间很长，这是传统计算机无法企及的。近年来，量子计算领域发展迅速，研究人员利用量子叠加态、量子纠缠和量子测量的特点设计

收稿日期：2022－07－06；修回日期：2022－09－27

基金项目：国家自然科学基金 (62076042, 62102049)；四川省自然科学基金 (2022NSFSC0535)；四川省科技厅重点研发项目 (2021YFSY0012, 2021YFG0332)；四川省量子安全通信创新团队项目 (17TD0009)

作者简介：闫丽丽 (1980－)，女，博士，教授，主要从事量子计算、量子安全通信方面的研究.

*通信作者：闫丽丽，E-mail：yanlili@cuit.edu.cn

了一些量子算法。如 Shor 大数分解算法、Grover 搜索算法、HHL 算法[6] 等。因此，使用量子计算来解决机器学习算法[7-8] 的高时间复杂度问题是一个很好的选择。研究人员在经典机器学习算法的基础上设计了一些量子机器学习算法，如量子支持向量机[9]、量子主成分分析[10] 和量子神经网络[11-14]。与经典算法相比，这些算法具有指数加速的优势。结合模糊神经网络在处理模糊问题上的优势和量子神经网络在计算速度上的优势，便可以设计出量子模糊神经网络[15-16]。

ANFIS 将模糊规则强度与样本特征以乘法相结合作为输出，这样能使系统计算速度加快，但却限制了输出的准确度。本文将 ANFIS 与 QBP 相结合，提出了基于自适应网络的量子模糊推理系统 (ANQFIS)，其将样本特征通过一层 QBP 处理后，再将模糊规则强度转化为角度作为量子门的参数，对 QBP 的输出的量子比特做量子门旋转操作，最后以量子态的测量概率作为输出。量子计算的速度优势和神经网络的高准确度使得 ANQFIS 在拥有高计算速度的同时又具有更高的输出准确率。

1　理论知识

1.1　量子比特与量子门

一个量子比特可以是$|0\rangle$、$|1\rangle$或是$|0\rangle$与$|1\rangle$的叠加态，一个叠加态的量子比特可以表示为：

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

式中，α和β是复数，表示振幅，且满足$|\alpha|^2 + |\beta|^2 = 1$，即在$Z$基测量下，$|\varphi\rangle$被测量得到$|0\rangle$的概率为$|\alpha|^2$，被测量得到$|1\rangle$的概率为$|\beta|^2$。

对量子比特的操作变换通常使用量子门来实现，量子比特$|\varphi\rangle$写成向量的形式为$(\cos\theta_0 \ \ \sin\theta_0)^{\mathrm{T}}$，其中$\cos\theta_0 = \alpha$，$\sin\theta_0 = \beta$。而一个量子门可以被写成矩阵的形式，本文用到的量子门定义为：

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

将此量子门作用到$|\varphi\rangle$上则可以看作是对$|\varphi\rangle$做了一个旋转操作：

$$R(\theta)|\varphi\rangle = \begin{pmatrix} \cos(\theta_0+\theta) \\ \sin(\theta_0+\theta) \end{pmatrix} \tag{3}$$

1.2　量子神经元

本文使用的是文献 [11] 提出的一种量子神经元结构的简化版本，如图 1 所示。

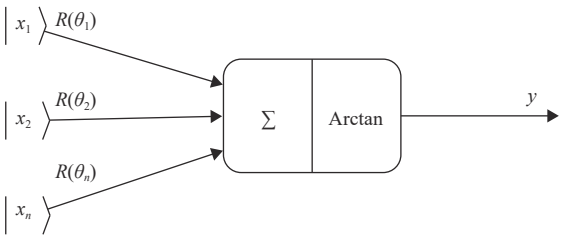

图 1　量子神经元结构

正如经典神经网络中的神经元一样，此量子神经元有 n 个输入和 1 个输出，假设图 1 中的量子比特$|x_j\rangle = (\cos\theta_j \ \ \sin\theta_j)^{\mathrm{T}}$，且$R(\theta)$的定义如 (3) 所示，那么输入 n 个量子比特$|x_1\rangle\cdots|x_n\rangle$后，获得一个新的量子比特$(\cos\theta' \ \ \sin\theta')^{\mathrm{T}}$作为量子神经元的输出，其中：

$$\theta' = \arctan\left(\frac{\sum_{j=1}^{n}\sin(t_j+\theta_j)}{\sum_{j=1}^{n}\cos(t_j+\theta_j)}\right) \tag{4}$$

1.3　模糊规则

对于一个普通集合，一个元素可能属于它或不属于它。而对于模糊集合，元素就不能说属于还是不属于此模糊集合，而是用隶属度来衡量此元素属于它的程度大小。隶属度由隶属度函数来计算得到。如果存在一个模糊集 A，$A(x)$ 为此模糊集的隶属度函数，它的值域为 [0,1]，如果 $A(x)$ 越接近 1，则表明 x 属于模糊集合 A 的程度越大，越接近 0 则程度越小。模糊集 A 可以表示为：

$$A = \{(x, A(x)) | A(x) \in [0,1]\} \tag{5}$$

模糊 if-then 规则或模糊条件语句，是一种以 if A then B 为形式的表达式，其中 A 和 B 都是语言标签 (小、大等)，拥有自己的隶属度函数。下面简要介绍 ANFIS 的模型结构。

在 ANFIS 中，所考虑的模糊推理系统假设有两个输入和一个输出，分别是x_1、x_2和R，假设规则库中包含有两个 Takagi 和 Sugeno 提出的 if-then 规则：

$$\begin{aligned} &\text{Rule 1: If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1, \text{ then } f_1 = p_1x_1 + q_1x_2 + r_1 \\ &\text{Rule 2: If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2, \text{ then } f_2 = p_2x_1 + q_2x_2 + r_2 \end{aligned} \tag{6}$$

式中，p_i，q_i，r_i是待训练更新的参数，被称作后件参数。并且A_i，B_i的隶属度函数采用的是高斯隶属度函数，假设模糊集A_i的高斯隶属度函数可以表示为：

$$\mu_{A_i}(x) = \exp\left[-\frac{1}{2}\left(\frac{x-c_i}{\sigma_i}\right)^2\right] \tag{7}$$

式中，c_i和σ_i是待训练更新的参数，被称作前件参数。之后根据语言标签的个数，计算出每个节点的输出：

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y) \quad i = 1,2\cdots \tag{8}$$

式中，w_i是模糊规则的强度。之后计算出第i个模糊规则强度占所有模糊规则强度之和的比例：

$$\bar{w}_i = \frac{w_i}{\sum_i w_i} \quad i = 1,2\cdots \tag{9}$$

式中，$\bar{w}_i$被称作归一化强度。最后，根据式 (7) 的规则，计算得到最后的输出：

$$R_i = \bar{w}_i f_i \quad i = 1,2\cdots \tag{10}$$

$$R = \sum_i R_i \tag{11}$$

2 ANQFIS 模型结构

因为一个多输入多输出的系统总是可以由多个多输入单输出的系统组成，所以本文提出的 ANQFIS 模型结构具有 n 个输入和 1 个输出。模型可以看做由两部分组成，一个是量子部分，另一个是模糊部分。量子部分由量子 BP 神经网络启发而来，模糊部分由 ANFIS 启发而来。模型结构如图 2 所示。

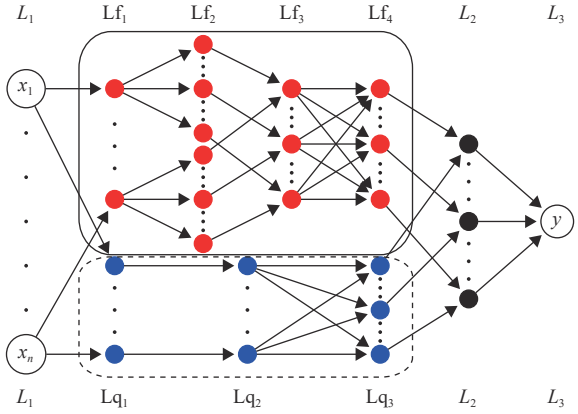

图 2 ANQFIS 结构图

模型的实线框部分为模糊部分，虚线框部分为量子部分。整个模型的第i层表示为L_i，模糊部分的第i层表示为Lf_i，量子部分的第i层表示为Lq_i。对于L_i，Lf_i，Lq_i，它们的第j个节点的输入分别表示为$\mu_{L_i}^j$，$\mu_{\mathrm{Lf}_i}^j$，$\mu_{\mathrm{Lq}_i}^j$，第j个节点的输出分别表示为$o_{L_i}^j$，$o_{\mathrm{Lf}_i}^j$，$o_{\mathrm{Lq}_i}^j$。如果节点有多个输入或输出，则第j个节点的第k个输入分别表示为$\mu_{L_i}^{jk}$，$\mu_{\mathrm{Lf}_i}^{jk}$，$\mu_{\mathrm{Lq}_i}^{jk}$，第j个节点的第k个输出分别表示为$o_{L_i}^{jk}$，$o_{\mathrm{Lf}_i}^{jk}$，$o_{\mathrm{Lq}_i}^{jk}$。一个n维数据$x = (x_1, x_2, \cdots, x_n)$被输入该模型后，分别经过量子部分和模糊部分处理，最后相结合得到输出y。模型各个部分每一层的输入和输出描述如下。

L_1(输入层)：n维数据$x = (x_1, x_2, \cdots, x_n)$被输入到 ANQFIS 模型。

Lf_1(模糊部分−输入层)：n维数据$x = (x_1, x_2, \cdots, x_n)$被输入到模糊部分，这一层中第$j$个节点的第$k$个输出为：

$$o_{L_1}^{jk} = x_j \tag{12}$$

Lf_2(模糊部分−模糊化层)：设规则库有r个规则，则易得该层有nr个节点。结合式 (13) 从图 2 中可以得知这一层中每一个节点的输入。每一个节点都定义了一个高斯隶属度函数，第j个节点的输出为：

$$o_{\mathrm{L}_2}^j : g_j = \exp\left[-\frac{1}{2}\left(\frac{\mu_{\mathrm{Lf}_2}^j - c_j}{\sigma_j}\right)^2\right] \tag{13}$$

式中，c_j和σ_j是待训练参数，$j = 1,2,\cdots,nr$。

Lf_3(模糊部分−模糊规则层)：在这一层中，每一个节点可以看作是一个模糊规则，规则库有r个规则，故该层有r个节点。根据模糊规则，该层的第j个节点取Lf_2中的$g_j, g_{j+r}, \cdots, g_{j+nr}$作为其输入$\mu_{\mathrm{Lf}_3}^{j1}, \mu_{\mathrm{Lf}_3}^{j2}, \cdots, \mu_{\mathrm{Lf}_3}^{jr}$，将这些输入相乘获得输出。第$j$个节点的输出就是其规则强度：

$$o_{\mathrm{Lf}_3}^j : w_j = \prod_{k=1}^{n} \mu_{\mathrm{Lf}_3}^{jk} \tag{14}$$

式中，$j = 1,2,\cdots,r$。

Lf_4(模糊部分−归一化层)：在这一层中，第j个节点的输出为Lf_3中第j个模糊规则强度占所有模糊规则强度之和的比例，再转换为角度：

$$o_{\mathrm{Lf}_4}^j : \bar{w}_j = 2\pi \frac{w_j}{\sum_{i=1}^{r} w_i} \tag{15}$$

Lq_1(量子部分−角度化层)：在这一层中，n维数据$x = (x_1, x_2, \cdots, x_n)$被输入到量子部分，$x_1, x_2, \cdots, x_n$被转化为$t_1, t_2, \cdots, t_n$，第$j$个节点的输出为：

$$o_{\mathrm{Lq}_1}^j : t_j = \frac{2\pi}{\left[1 + \exp(-x_j)\right]} \tag{16}$$

Lq_2(量子部分−量子化层)：在这一层中，根据$t_1, t_2, \cdots, t_n$制备相应的量子比特$|x_1\rangle, |x_2\rangle, \cdots, |x_n\rangle$，并将其输出给下一层的每一个节点。该层第$j$个节点的第$k$个输出为：

$$o_{\mathrm{Lq}_2}^{jk}:\left|x_j\right\rangle=\cos t_j|0\rangle+\sin t_j|1\rangle \tag{17}$$

Lq_3(量子部分−量子神经元层)：在这一层中有r个节点。第j个节点的第k个输入为：

$$\mu_{\mathrm{Lq}_3}^{jk}=\left|x_j\right\rangle=\cos t_j|0\rangle+\sin t_j|1\rangle \tag{18}$$

式中，$j=1,2,\cdots,n,k=1,2,\cdots,r$。该层中的节点实现神经网络中神经元的功能，节点的每一个$\mu_{\mathrm{Lq}_3}^{jk}$都对应着一个以随机值为初值的角度θ_{jk}，它们会在模型学习时被训练更新。对于第k个节点 (注意此处用k而不是用j作为节点下标)，使用式 (2) 中提到的量子门和θ_{jk}对其输入的所有量子比特进行旋转，并根据式 (4) 得到新的量子比特$|a_k\rangle$，并将其作为该节点的输出：

$$\varphi_k=\arctan\left(\frac{\sum_{j=1}^{n}\sin(t_j+\theta_{jk})}{\sum_{j=1}^{n}\cos(t_j+\theta_{jk})}\right) \tag{19}$$

$$o_{\mathrm{Lq}_3}^{k}:|a_k\rangle=\cos\varphi_k|0\rangle+\sin\varphi_k|1\rangle \tag{20}$$

式中，$k=1,2,\cdots,r$。

L_2(量子模糊层)：在这一层中，第j个节点接受Lf_4和Lq_3的输出作为其两个输入：

$$\mu_{\mathrm{L}_2}^{j1}=o_{\mathrm{Lf}_3}^{j}=\bar{w}_j \tag{21}$$

$$\mu_{\mathrm{L}_2}^{j2}=o_{\mathrm{Lq}_3}^{j}=\left|a_j\right\rangle=\cos\varphi_j|0\rangle+\sin\varphi_j|1\rangle \tag{22}$$

式中，$j=1,2,\cdots,r$。再将$\bar{w}_j$作为角度使用式 (2) 中提到的量子门作用到$\left|a_j\right\rangle$上，得到新的量子比特$\left|b_j\right\rangle$，并将其作为输出：

$$o_{\mathrm{L}_2}^{j}:\left|b_j\right\rangle=\cos\psi_j|0\rangle+\sin\psi_j|1\rangle \tag{23}$$

式中，$\psi_j=\varphi_j+\bar{w}_j,\ j=1,2,\cdots,r$。总之，这一层按照模糊部分输出的角度值来对量子部分输出的量子比特进行旋转，得到新的量子比特。

L_3(输出层)：这一层是整个模型的输出层。将上一层输出的所有量子比特按照式 (4) 的方式融合得到新的量子比特$|y\rangle$：

$$\gamma=\arctan\left(\frac{\sum_{j=1}^{r}\sin\psi_j}{\sum_{j=1}^{r}\cos\psi_j}\right) \tag{24}$$

$$|y\rangle=\cos\gamma|0\rangle+\sin\gamma|1\rangle \tag{25}$$

最后，取对$|y\rangle$测量得到$|1\rangle$的概率作为整个模型的输出：

$$y=\sin^2\gamma \tag{26}$$

3　ANQFIS 学习算法

在 ANQFIS 模型中，有一些参数需要在模型训练学习中被更新优化，它们分别是Lq_3中的$\theta_{jk}(j=1,2,\cdots,n,k=1,2,\cdots,r)$以及$\mathrm{Lf}_2$中的$x_j$，$\sigma_j(j=1,2,\cdots,nr)$。模型的损失函数定义为：

$$L=\frac{1}{2}(y-\hat{y})^2 \tag{27}$$

式中，y是模型的真实输出；$\hat{y}$是标签 (0 或 1)。为了能够正确的表示节点，令$J=j\%r$，便可得到这些参数对损失函数的梯度为：

$$\nabla_{\theta_{jk}}=\frac{\partial L}{\partial\theta_{jk}}=\frac{\partial L}{\partial y}\frac{\partial y}{\partial\gamma}\frac{\partial\gamma}{\partial\psi_k}\frac{\partial\psi_k}{\partial\varphi_k}\frac{\partial\varphi_k}{\partial\theta_{jk}} \tag{28}$$

$$\nabla_{c_j}=\frac{\partial L}{\partial c_j}=\frac{\partial L}{\partial y}\frac{\partial y}{\partial\gamma}\frac{\partial\gamma}{\partial\psi_J}\frac{\partial\psi_J}{\partial\bar{w}_j}\frac{\partial\bar{w}_J}{\partial w_J}\frac{\partial w_J}{\partial g_j}\frac{\partial g_j}{\partial c_j} \tag{29}$$

$$\nabla_{\sigma_j}=\frac{\partial L}{\partial\sigma_j}=\frac{\partial L}{\partial y}\frac{\partial y}{\partial\gamma}\frac{\partial\gamma}{\partial\psi_J}\frac{\partial\psi_J}{\partial\bar{w}_J}\frac{\partial\bar{w}_J}{\partial w_J}\frac{\partial w_J}{\partial g_j}\frac{\partial g_j}{\partial\sigma_j} \tag{30}$$

定义变量：

$$\nabla_0=2(y-\hat{y})\sin\gamma\cos\gamma \tag{31}$$

$$\nabla_{\theta_{jk}1}=\frac{\cos\psi_k\mathrm{Tc}_0+\sin\psi_k\mathrm{Ts}_0}{\mathrm{Ts}_0{}^2+\mathrm{Tc}_0{}^2} \tag{32}$$

式中，$\mathrm{Ts}_0=\sum_{i=1}^{r}\sin\psi_i,\mathrm{Tc}_0=\sum_{i=1}^{r}\cos\psi_i$

$$\nabla_{\theta_{jk}2}=\frac{\cos(t_j+\theta_{jk})\mathrm{Tc}_1+\sin(t_j+\theta_{jk})\sin\psi_k\mathrm{Ts}_1}{(\mathrm{Ts}_1{}^2+\mathrm{Tc}_1{}^2)} \tag{33}$$

式中，$\mathrm{Ts}_1=\sum_{i=1}^{n}\sin(t_i+\theta_{ik})$，$\mathrm{Tc}_1=\sum_{i=1}^{n}\cos(t_i+\theta_{ik})$。

$$\nabla_1=\frac{\cos\psi_J\mathrm{Tc}_0+\sin\psi_J\mathrm{Ts}_0}{\mathrm{Ts}_0{}^2+\mathrm{Tc}_0{}^2} \tag{34}$$

$$\nabla_2=2\pi\frac{\mathrm{Tw}-w_J}{\mathrm{Tw}^2} \tag{35}$$

式中，$\mathrm{Tw}=\sum_{i=1}^{r}w_i$。

$$\nabla_3=\frac{\prod_{i=0}^{i=n-1}g_{J=ir}}{g_j} \tag{36}$$

$$\nabla_{c_j 1} = \exp\left[-\frac{1}{2}\left(\frac{\mu_{\mathrm{Lf}_2}^{j} - c_j}{\sigma_j}\right)^2\right]\frac{\mu_{\mathrm{Lf}_2}^{j} - c_j}{{\sigma_j}^2} \tag{37}$$

$$\nabla_{o_j 1} = -\exp\left[-\frac{1}{2}\left(\frac{\mu_{\mathrm{Lf}_2}^{j} - c_j}{\sigma_j}\right)^2\right]\frac{\left(\mu_{\mathrm{Lf}_2}^{j} - c_j\right)^2}{{\sigma_j}^3} \tag{38}$$

那么根据链式求导法则可得这些参数的梯度的准确值：

$$\nabla_{\theta_{jk}} = \nabla_0 \nabla_{\theta_{jk}1} \nabla_{\theta_{jk}2} \tag{39}$$

$$\nabla_{c_j} = \nabla_0 \nabla_1 \nabla_2 \nabla_3 \nabla_{c_j 1} \tag{40}$$

$$\nabla_{o_j} = \nabla_0 \nabla_1 \nabla_2 \nabla_3 \nabla_{o_j 1} \tag{41}$$

在模型训练时，只需要按照下面的方式更新这些参数，即可优化模型：

$$\theta_{jk}(t+1) = \theta_{jk}(t) - \eta \nabla_{\theta_{jk}} \tag{42}$$

$$c_j(t+1) = c_j(t) - \eta \nabla_{c_j} \tag{43}$$

$$\sigma_j(t+1) = \sigma_j(t) - \eta \nabla_{\sigma_j} \tag{44}$$

式中， t 是指当前训练轮数，t+1 则是下一个训练轮数；η是学习率，其决定了模型训练的每一步的步长，在模型训练一开始时被定义为一个固定值。

4 仿真实验对比

由于 ANQFS 执行时需要大量的量子比特对高维数据进行编码，而目前网络量子云平台所提供的量子设备是难以实现的，因此本文采用 Pennylane[16] 仿真框架构建了 ANQFIS 模型来验证其学习算法的正确性，并且对 ANQFIS、ANFIS 和 QBP 的输出准确率和鲁棒性进行了分析和比较。由于仿真模型是使用经典计算机来模拟量子计算机，并不能体现出真实量子设备的速度优越性，所以本文没有在实验中对比模型之间的时间消耗。

下面分别使用低维数据集和高维数据集来训练和验证模型，让其实现二分类功能。对于低维数据集，选择 IRIS 鸢尾花卉数据集 (4 维)中的所有 0 类和 1 类数据作为训练集与验证集，学习率η选择 0.1。对 ANQFIS、ANFIS、QBP 模型设置同样的学习率进行训练，每次训练 100 轮，一共分别训练 50 次。每一次训练中，每当训练 10 轮之后，使用验证集来测试模型的准确率并记录结果。图 3 是通过对 50 次训练结果求平均值，从而得到模型的平均准确率与训练轮数之间的关系图。

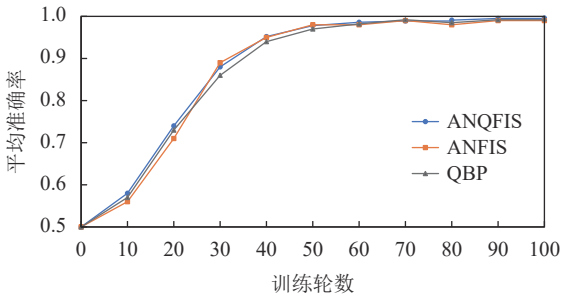

图 3 使用 IRIS 鸢尾花卉数据集时平均准确率与训练轮数的关系图

根据图 3 可以看出，在使用低维数据集作为训练集和验证集时，ANQFIS 在准确率上并没有更优异的表现。下面采用高维数据集进行实验，选择 MNIST 手写数字数据集 (784 维) 中的所有标签为 0 和 1 的数据作为高维数据集，对 ANQFIS、ANFIS、QBP 模型设置同样的学习率$\eta = 0.1$进行训练，每次训练 500 轮，一共分别训练 50 次。在一次训练中，每当训练 50 轮之后，使用验证集来测试模型准确率并记录结果。图 4 为通过对 50 次训练的结果求平均值，从而得到模型平均准确率与训练轮数之间的关系图。

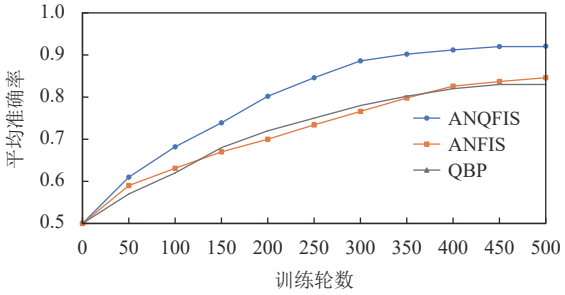

图 4 使用 MNIST 数据集时平均准确率与训练轮数的关系图

根据图 4 可以看出，在使用高维数据集作为训练集和验证集时，500 轮之后，ANFIS 和 QBP 的准确率基本稳定在 85% 左右，而 ANQFIS 的准确率稳定在 92% 左右，这说明 ANQFIS 在准确率上明显优于 ANFIS 和 QBP。

为了测试 ANQFIS 模型的鲁棒性，使用 FGSM 攻击算法[17] 分别对经过 MNIST 数据集 500 轮训练后的 3 种模型生成对抗样本来测试。FGSM 算法首先求出模型损失函数相对于模型输入 $\boldsymbol{x}$ 的梯度$\nabla_{\boldsymbol{x}}$，对其取符号，再定义一个扰动系数 δ，便可得到一个扰动值，最后将此扰动添加到合法样本上，便可

得到对抗样本$\boldsymbol{x}'$：

$$\boldsymbol{x}' = \boldsymbol{x} + \delta \text{sign} \nabla_{\boldsymbol{x}} \tag{45}$$

通过调整扰动系数 δ，便可以调整对抗样本的对抗性强弱。在 MNIST 数据集中选取一个标签为 0 的数据，如图 5a 所示，且 ANQFIS、ANFIS、QBP 都能对其准确分类。再设置 δ=0，使其每次增加 0.1，并生成相应的对抗样本，然后输入 3 个模型，直到模型对其错误分类，此时 δ 的值称之为该模型对该样本的“扰动系数阈值”，此时的对抗样本称为“阈值样本”。ANQFIS、ANFIS、QBP 的阈值样本分别如图 5b、5c、5d 所示。

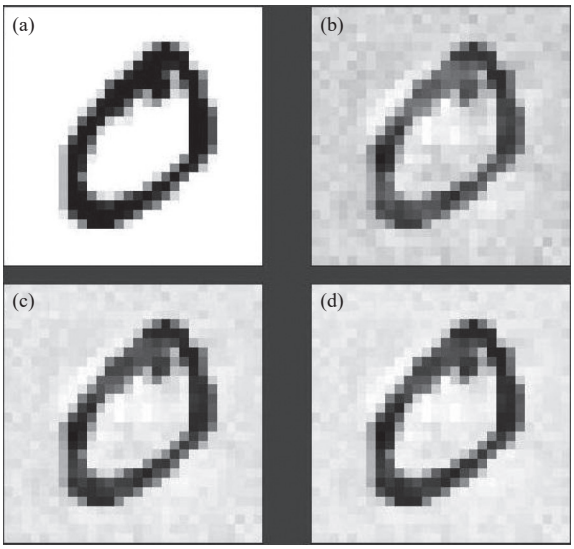

图 5　阈值样本对比图

图 5a 为合法样本，图 5b 为 ANQFIS 的阈值样本，扰动系数阈值为 2.61，图 5c 为 ANFIS 的阈值样本，扰动系数阈值为 1.89，图 5d 为 QBP 的阈值样本，扰动系数阈值为 1.75。可看出，仅对这一个样本而言，ANQFIS 的扰动系数阈值更高，ANQFIS 具有更高的鲁棒性。

为了使结果更有说服力，从 MNIST 数据集中分别选取了 100 个标签为 0 的数据和 100 个标签为 1 的数据，并且再引入一个 FGM 攻击算法[18-19]。此算法生成对抗样本的方式是：

$$\boldsymbol{x}' = \boldsymbol{x} + \delta \nabla_{\boldsymbol{x}} \tag{46}$$

在这两种攻击算法下，分别计算出这 3 个模型对这 200 个样本的扰动系数阈值。由于对于不同的样本而言，扰动系数阈值往往不在同一个数量级，因此将扰动系数阈值进行归一化处理，3 个模型在两种攻击算法下对这 200 个合法样本的平均归一化扰动系数阈值如表 1 所示。

表 1　平均归一化扰动系数阈值

模型	FGSM攻击算法	FGM攻击算法
ANQFIS	0.418	0.395
ANFIS	0.301	0.311
QBP	0.281	0.294

根据表 1 可以看出，ANQFIS 在鲁棒性上也明显优于 ANFIS 和 QBP。

5　结 束 语

本文将 ANFIS 与 QBP 相结合，提出了一种基于自适应网络的量子模糊推理系统 ANQFIS。经过仿真实验得出，该模型对于高维数据集在准确率方面明显优于 ANFIS 和 QBP 和鲁棒性。

参　考　文　献

[1] ZADEH L A. Fuzzy sets[J]. Information and Control, 1965, 8(3): 338-353.

[2] JORDAN M I, MITCHELL T M. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255-260.

[3] SCHMIDHUBER J. Deep learning in neural networks: An overview[J]. Neural Networks, 2015, 61: 85-117.

[4] LEE H M, LU B H. Fuzzy BP: A neural network model with fuzzy inference[C]//Proceedings of 1994 IEEE International Conference on Neural Networks. Orlando: IEEE, 1994: 1583-1588.

[5] JANG R J S. ANFIS: Adaptive-Network-Based fuzzy inference system[J]. IEEE Transactions on Systems Man & Cybernetics, 1993, 37(4): 446-461.

[6] HARROW A W, HASSIDIM A, LLOYD S. Quantum algorithm for linear systems of equations[J]. Physical Review Letters, 2009, 103(15): 150502.

[7] DAS S S, DENG D L, DUAN L M. Machine learning meets quantum physics[J]. Physics Today, 2019, 72(3): 48-54.

[8] BIAMONTE J, WITTEK P, PANCOTTI N, et al. Quantum machine learning[J]. Nature, 2017, 549(7671): 195-202.

[9] REBENTROST P, MOHSENI M, LLOYD S. Quantum support vector machine for big data classification[J]. Physical Review Letters, 2014, 113(13): 130503.

[10] CONG I, DUAN L. Quantum discriminant analysis for dimensionality reduction and classification[J]. New Journal of Physics, 2016, 18(7): 073011.

[11] 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.

[12] BENEDETTI M, LLOYD E, SACK S, et al. Parameterized quantum circuits as machine learning models[J]. Quantum Science and Technology, 2019, 4(4): 019601.

[13] CONG I, CHOI S, LUKIN M D. Quantum convolutional neural networks[J]. Nature Physics, 2019, 15(12): 1273-1278.

[14] DALLAIRE-DEMERS P L, KILLORAN N. Quantum generative adversarial networks[J]. Physical Review A, 2018, 98(1): 012324.

[15] CHEN C H, LIN C J, LIN C T. An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation[J]. Soft Computing, 2008, 12(6): 567-583.

[16] MIAO F Y, XIONG Y, CHEN H H, et al. A fuzzy quantum neural network and its application in pattern recognition[J]. Chinese Journal of Electronics, 2005, 14(3): 524-528.

[17] VILLE B. A open-source software framework for quantum machine learning[EB/OL]. [2021-10-11]. https://github.com/PennyLaneAI/pennylane.

[18] GOODFELLOW I, SHLENS J, SZEGEDY C, et al. Explaining and harnessing adversarial examples[EB/OL]. [2021-10-15]. https://arxiv.org/abs/1412.6572v1.

[19] MIYATO T, DAI A M, GOODFELLOW I. Adversarial training methods for semi-supervised text classification [EB/OL]. [2021-12-11]. https://arxiv.org/abs/1605.07725v2.

编 辑 叶 芳