计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(5): 1205−1215, 2025

四元数神经网络的通用近似与逼近优势

吴锦辉　姜　远
（计算机软件新技术全国重点实验室（南京大学）　南京　210023）
（南京大学人工智能学院　南京　210023）
（wujh@lamda.nju.edu.cn）

Universal Approximation and Approximation Advantages of Quaternion-Valued Neural Networks

Wu Jinhui and Jiang Yuan
（*National Key Laboratory for Novel Software Technology* (*Nanjing University*), *Nanjing* 210023）
（*School of Artificial Intelligence*, *Nanjing University*, *Nanjing* 210023）

Abstract　Quaternion-valued neural networks extend real-valued neural networks to the algebra of quaternions. Quaternion-valued neural networks achieve higher accuracy or faster convergence than real-valued neural networks in some tasks, such as singular point compensation in polarimetric synthetic aperture, spoken language understanding, and radar robot control. The performance of quaternion-valued neural networks is widely supported by empirical studies, but there are few studies about theoretical properties of quaternion-valued neural networks, especially why quaternion-valued neural networks can be more efficient than real-valued neural networks. In this paper, we investigate theoretical properties of quaternion-valued neural networks and the advantages of quaternion-valued neural networks compared with real-valued neural networks from the perspective of approximation. Firstly, we prove the universal approximation of quaternion-valued neural networks with a non-split ReLU (rectified linear unit)-type activation function. Secondly, we demonstrate the approximation advantages of quaternion-valued neural networks compared with real-valued neural networks. For split ReLU-type activation functions, we show that one-hidden-layer real-valued neural networks need about 4 times the number of parameters to possess the same maximum number of convex linear regions as one-hidden-layer quaternion-valued neural networks. For the non-split ReLU-type activation function, we prove the approximation separation between one-hidden-layer quaternion-valued neural networks and one-hidden-layer real-valued neural networks, i.e., a quaternion-valued neural network can express a real-valued neural network using the same number of hidden neurons and the same parameter norm, while a real-valued neural network cannot approximate a quaternion-valued neural network unless the number of hidden neurons is exponentially large or the parameters are exponentially large. Finally, simulation experiments support our theoretical findings.

Key words　quaternion-valued neural networks; universal approximation; approximation advantage; maximum number of convex linear regions; approximation separation; neural network theory

摘　要　四元数神经网络将实值神经网络推广到了四元数代数中，其在偏振合成孔径雷达奇异点补偿、口语理解、机器人控制等任务中取得了比实值神经网络更高的精度或更快的收敛速度. 四元数神经网络的

收稿日期：2024-05-31；**修回日期**：2024-10-08
基金项目：国家自然科学基金项目（62176117）；南京大学优秀博士研究生创新能力提升计划项目（202401A13）
This work was supported by the National Natural Science Foundation of China (62176117) and the Program for Outstanding PhD Candidates of Nanjing University (202401A13).
通信作者：姜远（jiangyuan@nju.edu.cn）

性能在实验中已得到广泛验证,但四元数神经网络的理论性质及其相较于实值神经网络的优势研究较少.从表示能力的角度出发,研究四元数神经网络的理论性质及其相较于实值神经网络的优势.首先,证明了四元数神经网络使用一个非分开激活的修正线性单元(rectified linear unit, ReLU)型激活函数时的通用近似定理.其次,研究了四元数神经网络相较于实值神经网络的逼近优势.针对分开激活的ReLU型激活函数,证明了单隐层实值神经网络需要约4倍参数量才能生成与单隐层四元数神经网络相同的最大凸线性区域数.针对非分开激活的ReLU型激活函数,证明了单隐层四元数神经网络与单隐层实值神经网络间的逼近分离:四元数神经网络可用相同的隐层神经元数量与权重模长表示实值神经网络,而实值神经网络需要指数多个隐层神经元或指数大的参数才可能近似四元数神经网络.最后,模拟实验验证了理论.

关键词 四元数神经网络;通用近似;逼近优势;最大凸线性区域数;逼近分离;神经网络理论

中图法分类号 TP18

DOI: 10.7544/issn1000-1239.202440410 **CSTR:** 32373.14.issn1000-1239.202440410

四元数神经网络是实值神经网络在四元数代数中的推广,其输入、输出与权重均可由四元数构成,其激活函数为四元数代数到四元数代数的映射.四元数神经网络在诸多任务中取得成功,如偏振合成孔径雷达奇异点补偿[1]、口语理解[2]、点云配准[3]等.在应用中,四元数网络或是取得了更高的精度[1-2],或是具有更快的收敛速度[4].

基于四元数神经网络的成功应用,许多研究为四元数神经网络的优异性能提供了若干直观解释.其一,高维数据的不同维度间具有相关性,实值神经网络独立地处理不同维度,而四元数神经网络可以更好地刻画不同维度间的相关性,从而学到更好的表示,比如图像数据的RGB值之间的依赖性[5].其二,在描述3维空间旋转时,动态欧拉角会遇到万向锁问题,即损失一个旋转自由度,而四元数代数可以避免万向锁问题,从而更好地刻画3维旋转[6].其三,四元数神经网络可以用更少的参数量完成乘法运算,四元数代数中只需4个实值参数即可完成四元数到四元数的乘法运算,而实数域中则需要16个参数才能完成4维实数到4维实数的乘法运算[7].但四元数神经网络的理论性质研究较少,现有的理论工作主要集中于四元数神经网络使用分开激活的激活函数时的通用近似性质[8-9].使用非分开激活的激活函数的四元数神经网络在一些任务中的表现优于使用分开激活的激活函数的四元数神经网络[10],但其逼近能力尚无理论研究.同时,四元数神经网络相较于实值神经网络的优势尚缺乏理论解释.

本文从表示能力的角度研究四元数神经网络的通用近似性质,及其相较于实值神经网络的逼近优势.本文主要贡献包括3个方面:

1)证明了四元数神经网络使用一个非分开激活的修正线性单元(rectified linear unit, ReLU)型激活函数时具有通用近似性质,即单隐层四元数神经网络能以任意精度逼近任意紧集上的任意连续函数.

2)证明了四元数神经网络的逼近优势.当四元数神经网络使用分开激活的ReLU型激活函数时,单隐层实值神经网络需使用4倍参数量才能生成与单隐层四元数神经网络相同的最大凸线性区域数;当四元数神经网络使用非分开激活的ReLU型激活函数时,单隐层四元数神经网络与实值神经网络之间存在逼近分离,即四元数神经网络可以使用与实值神经网络相同的隐层神经元数目与权重模长表示任意实值神经网络,而实值神经网络需要使用指数多个隐层神经元或指数大的参数才可能逼近特定的四元数神经网络.

3)通过模拟实验验证了四元数神经网络相较于实值神经网络的逼近优势.

1 相关工作

1.1 通用近似

神经网络的通用近似定理证明了神经网络可以以任意精度逼近任意紧集上的任意连续函数,这一性质表明神经网络具有强大的表示能力以处理各种复杂任务.当激活函数满足特定条件时,很多常见结构下的实值神经网络均具有通用近似性质,如前馈神经网络[11-14]、循环神经网络[15-19]、卷积神经网络[20]等.实值神经网络的通用近似性质可以推广到复值神经网络中[21],且已有工作证明了复值神经网络满足通用近似性质的充要条件[22].超复数神经网络的通用近似性质也已有研究,现有工作主要基于分开激活的激活函数进行证明[8-9].四元数是超复数的一种,

本文为使用一个非分开激活的激活函数的四元数神经网络证明了通用近似性质.

1.2 逼近优势

通用近似性质虽然表明神经网络能逼近任意连续函数, 但逼近某些目标函数时所需的参数量是指数多的[23]. 因此, 神经网络的逼近复杂度是一个重要的问题, 并得到了广泛研究.

一类工作聚焦于使用连续分段线性激活的神经网络, 并分析这类神经网络的函数性质. 这类神经网络对应于连续分段线性函数, 且任意连续分段线性函数均可由深度有限的神经网络表示[24]. 因此, 线性区域数可以用来衡量使用连续分段线性激活函数的神经网络的表示能力. 有工作研究了神经网络的宽度、神经网络的深度及激活函数的线性区域数对神经网络线性区域数的影响, 并证明了增加神经网络深度可以指数级增加线性区域数, 而神经网络宽度与激活函数复杂度只能多项式级增加线性区域数[25-26]. 本文分析单隐层实值神经网络与单隐层四元数神经网络的凸线性区域数, 并证明了四元数神经网络可以更高效地生成凸线性区域数.

另一类工作对比 2 种不同结构的神经网络, 研究 2 种神经网络逼近特定目标函数所需的参数量或参数大小, 并得到一种神经网络比另一种更高效的结论. 逼近分离是一种常见的对比结论, 神经网络A与神经网络B之间具有逼近分离是指神经网络A可以高效地逼近神经网络B, 而神经网络B需要指数多个参数或指数大的参数才能逼近神经网络A. 比如深层神经网络与浅层神经网络之间具有逼近分离[27-28]以及复值神经网络与实值神经网络之间具有逼近分离等[29-30]. 本文证明了单隐层四元数神经网络与单隐层实值神经网络之间的逼近分离.

2 预备知识

令x_i表示向量$\boldsymbol{x}$的第i个分量, $[N]$表示集合$\{1,2,\cdots,N\}$. 记四元数代数为$\mathbb{H}$, i,j,k为四元数的 3 个虚部单位. 对于任意四元数$q\in\mathbb{H}$, 令q_{R}为q的实部, q_{I}, q_{J}, q_{K}为q的 3 个虚部, 即

$$q=q_{\mathrm{R}}+q_{\mathrm{I}}\mathrm{i}+q_{\mathrm{J}}\mathrm{j}+q_{\mathrm{K}}\mathrm{k}.$$

令$I(e)$表示指示函数, 当事件e为真时函数值为 1, 否则函数值为 0. 令$poly(d)$表示所有d的多项式构成的集合, 即

$$poly(d)=\left\{\sum_{i=0}^{n}a_i d^i\,\middle|\,n\in\mathbb{N},a_i\in\mathbb{R}\right\}.$$

本文使用标准的渐进符号$O(\cdot)$与$\Omega(\cdot)$, 令$f(d)$与$g(d)$为$\mathbb{N}$到$\mathbb{R}^+$的函数, $f(d)=O(g(d))$定义为

$$\exists k>0,d_0\in\mathbb{R},\ \text{s.t. } f(d)\leqslant kg(d),\forall d\geqslant d_0.$$

$f(d)=\Omega(g(d))$定义为

$$\exists k>0,d_0\in\mathbb{R},\ \text{s.t. } f(d)\geqslant kg(d),\forall d\geqslant d_0.$$

记实值神经网络的输入为$\boldsymbol{x}_{\mathbb{R}}\in\mathbb{R}^{4d}$, 其中不失一般性地假设输入维度为 4 的倍数. 考虑 1 维的实值输出空间, 则具有n个隐层神经元的单隐层实值神经网络可表示为

$$f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})=\boldsymbol{\alpha}_{\mathbb{R}}\sigma_{\mathbb{R}}(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}}),$$

其中$\boldsymbol{\alpha}_{\mathbb{R}}\in\mathbb{R}^{1\times n},\boldsymbol{W}_{\mathbb{R}}\in\mathbb{R}^{n\times 4d},\boldsymbol{b}_{\mathbb{R}}\in\mathbb{R}^{n}$为神经网络参数, $\sigma_{\mathbb{R}}:\mathbb{R}\to\mathbb{R}$为逐元素使用的激活函数. 记四元数神经网络的输入为$\boldsymbol{x}_{\mathbb{H}}\in\mathbb{H}^{d}$, 其中

$$\boldsymbol{x}_{\mathbb{H}}=(\cdots;x_{4i-3}+x_{4i-2}\mathrm{i}+x_{4i-1}\mathrm{j}+x_{4i}\mathrm{k};\cdots),i\in[d]$$

为实值输入$\boldsymbol{x}_{\mathbb{R}}=(x_1;x_2;\cdots;x_{4d})$的四元数表示, $\boldsymbol{x}_{\mathbb{R}}$称为$\boldsymbol{x}_{\mathbb{H}}$的实值表示. 则具有$m$个隐层神经元的单隐层四元数神经网络可表示为

$$f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=\mathrm{Re}[\boldsymbol{\alpha}_{\mathbb{H}}\sigma_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})],$$

其中$\boldsymbol{\alpha}_{\mathbb{H}}\in\mathbb{H}^{1\times m},\boldsymbol{W}_{\mathbb{H}}\in\mathbb{H}^{m\times d},\boldsymbol{b}_{\mathbb{H}}\in\mathbb{H}^{m}$为神经网络参数, $\sigma_{\mathbb{H}}:\mathbb{H}\to\mathbb{H}$为逐元素使用的激活函数, $\mathrm{Re}[q]=q_{\mathrm{R}}$返回一个四元数的实部部分.

3 四元数神经网络的逼近理论

本节研究四元数神经网络的逼近理论, 我们先研究四元数神经网络使用一个非分开激活的 ReLU 型激活函数时的通用近似性质, 再证明四元数神经网络使用分开激活与非分开激活的激活函数时相较于实值神经网络的逼近优势.

3.1 四元数神经网络的通用近似

本节研究使用四元数神经网络的通用近似性质. 四元数神经网络使用分开激活的激活函数时的通用近似性质已有广泛研究[8-9], 我们主要关注如下定义的非分开激活的 ReLU 型激活函数

$$\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0).$$

这一非分开激活的激活函数虽然简单, 但其具备通用近似与逼近分离性质, 表明非分开激活的激活函数具有强大的表示能力. 下面我们证明使用该激活函数的四元数神经网络具有通用近似性质, 逼近分离性质将在 3.3 节中讨论.

定理 1. 令 K 为 $\mathbb{H}^d$ 中任一紧集，$g:K\to\mathbb{R}$ 为一个连续函数. 则对任意 $\varepsilon>0$，存在 $\boldsymbol{\alpha}_{\mathbb{H}}\in\mathbb{H}^{1\times m},\boldsymbol{W}_{\mathbb{H}}\in\mathbb{H}^{m\times d},\boldsymbol{b}_{\mathbb{H}}\in\mathbb{H}^m$ 与使用激活函数 $\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$ 的四元数神经网络 $f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=\mathrm{Re}[\boldsymbol{\alpha}_{\mathbb{H}}\sigma_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})]$，使得

$$|g(\boldsymbol{x}_{\mathbb{H}})-f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})|\leqslant\varepsilon$$

对任意 $\boldsymbol{x}_{\mathbb{H}}\in K$ 均成立.

定理 1 证明了使用一个非分开激活的激活函数的四元数神经网络的通用近似性质，即四元数神经网络能以任意精度逼近任意紧集上的连续函数. 该定理的证明对四元数神经网络的参数进行特殊赋值，使四元数神经网络退化为一个实值神经网络，从而利用实值神经网络的通用近似性质证明四元数神经网络的通用近似性质. 这一证明方法可以用来证明四元数神经网络使用许多其他非分开激活的激活函数时的通用近似性质.

证明. 根据实值神经网络的通用近似定理，知存在参数 $\boldsymbol{\alpha}_{\mathbb{R}}\in\mathbb{R}^{1\times n},\boldsymbol{W}_{\mathbb{R}}\in\mathbb{R}^{n\times 4d},\boldsymbol{b}_{\mathbb{R}}\in\mathbb{R}^n$ 与实值神经网络 $f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})=\boldsymbol{\alpha}_{\mathbb{R}}\sigma_{\mathbb{R}}(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}})$，使得

$$|g(\boldsymbol{x}_{\mathbb{H}})-f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})|\leqslant\varepsilon$$

对任意 $\boldsymbol{x}_{\mathbb{H}}\in K$ 均成立，其中 $\sigma_{\mathbb{R}}$ 为 ReLU 激活函数，$\boldsymbol{x}_{\mathbb{R}}$ 为 $\boldsymbol{x}_{\mathbb{H}}$ 的实值表示.

构造如下参数

$$\begin{aligned}&\boldsymbol{\alpha}_{\mathbb{H},\mathrm{R}}=\boldsymbol{\alpha}_{\mathbb{R}},\boldsymbol{\alpha}_{\mathbb{H},\mathrm{I}}=\boldsymbol{\alpha}_{\mathbb{H},\mathrm{J}}=\boldsymbol{\alpha}_{\mathbb{H},\mathrm{K}}=\boldsymbol{0},\\&\boldsymbol{b}_{\mathbb{H},\mathrm{R}}=\boldsymbol{b}_{\mathbb{R}},\boldsymbol{b}_{\mathbb{H},\mathrm{I}}=\boldsymbol{b}_{\mathbb{H},\mathrm{J}}=\boldsymbol{b}_{\mathbb{H},\mathrm{K}}=\boldsymbol{0},\\&\boldsymbol{w}_{\mathbb{H},\mathrm{R},j}=\boldsymbol{w}_{\mathbb{R},4j-3},\boldsymbol{w}_{\mathbb{H},\mathrm{I},j}=-\boldsymbol{w}_{\mathbb{R},4j-2},\\&\boldsymbol{w}_{\mathbb{H},\mathrm{J},j}=-\boldsymbol{w}_{\mathbb{R},4j-1},\boldsymbol{w}_{\mathbb{H},\mathrm{K},j}=-\boldsymbol{w}_{\mathbb{R},4j},j\in[d],\end{aligned}$$

其中 $\boldsymbol{w}_j$ 表示矩阵 $\boldsymbol{W}$ 的第 j 列. 则四元数神经网络 $f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=\mathrm{Re}[\boldsymbol{\alpha}_{\mathbb{H}}\sigma_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})]$ 满足

$$\begin{aligned}f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=&\boldsymbol{\alpha}_{\mathbb{H},R}(\boldsymbol{\sigma}_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}}))_{\mathrm{R}}=\\&\boldsymbol{\alpha}_{\mathbb{H},\mathrm{R}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})_{\mathrm{R}}I((\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})_{\mathrm{R}}\geqslant 0)=\\&\boldsymbol{\alpha}_{\mathbb{R}}(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}})I(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}}\geqslant 0)=\\&f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}}).\end{aligned}$$

即上述构造的四元数神经网络与使用 ReLU 激活的实值神经网络具有相同表达式. 将上述结论代入实值神经网络的通用近似中，可得

$$|g(\boldsymbol{x}_{\mathbb{H}})-f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})|\leqslant\varepsilon$$

对任意 $x_{\mathbb{H}}\in K$ 均成立. 证毕.

定理 1 是对使用非分开激活的激活函数的四元数神经网络通用近似性质的初步尝试，未来还有许多重要的问题可以研究. 其一，使用完全四元值函数(fully quaternion-valued functions)作为激活函数的四元数神经网络是否具有通用近似性质还有待研究. 比如完全四元值双曲正切激活函数

$$\sigma_{\mathbb{H}}(q)=\frac{\mathrm{e}^{2q}-1}{\mathrm{e}^{2q}+1},$$

$$\mathrm{e}^q=\mathrm{e}^{q_{\mathrm{R}}}\left(\cos|v|+\frac{v}{|v|}\sin|v|\right),$$

其中 $v=q_{\mathrm{I}}\mathrm{i}+q_{\mathrm{J}}\mathrm{j}+q_{\mathrm{K}}\mathrm{k}$. $\sigma_{\mathbb{H}}(q)$ 是一个常用的完全四元值激活函数，且在一些任务中取得了比分开激活的激活函数更好的结果[10]. 但完全四元值双曲正切激活函数存在奇点，从而函数值在有界范围内无界. 此外，复值神经网络的通用近似理论表明使用完全复值激活函数的复值神经网络不具备通用近似性质[22]，而四元数作为复数的推广，使用完全四元值激活函数的神经网络的通用近似性质是否成立还有待研究. 其二，可以考虑使用其他输出形式的四元数神经网络的通用近似. 本文考虑的取实部作为实值输出并非唯一的输出形式，还有许多常见的输出形式，如将四元数输出以拟合高维输出[8]以及对四元数逐元素使用归一化指数函数(softmax)以得到后验概率[2]等.

3.2 分开激活时的最大凸线性区域数

本节研究使用 ReLU 型激活函数时实值神经网络与四元数神经网络的最大凸线性区域数. 对于实值神经网络，本节使用的 ReLU 型激活函数包括 ReLU[31]，leakyReLU(leaky ReLU)[32]，PReLU(parametric ReLU)[33]，这些激活函数均为具有 2 个分段的连续分段线性函数且具有如下形式：

$$\sigma_{\mathbb{R}}(y)=\max\{0,y\}+a\min\{0,y\},y\in\mathbb{R},$$

其中 $a\in\mathbb{R}$ 为 ReLU 型激活函数的固定或可学参数. 对于四元数神经网络，本节使用分开激活的 ReLU 型激活函数，即对于任意四元数 $q(q\in\mathbb{H})$，

$$\sigma_{\mathbb{H}}(q)=\sigma_{\mathbb{R}}(q_{\mathrm{R}})+\sigma_{\mathbb{R}}(q_{\mathrm{I}})\mathrm{i}+\sigma_{\mathbb{R}}(q_{\mathrm{J}})\mathrm{j}+\sigma_{\mathbb{R}}(q_{\mathrm{K}})\mathrm{k},$$

其中 $\sigma_{\mathbb{R}}$ 为 $\sigma_{\mathbb{H}}$ 对应的实值激活函数. 由于 ReLU 型激活函数与分开激活的 ReLU 型激活函数均为连续分段线性函数，从而使用这些激活函数的神经网络也是连续分段线性函数. 下面我们回顾连续分段线性函数的数学定义[26].

定义 1. 令 $f:\mathbb{R}^d\to\mathbb{R}$ 为一个映射. 如果 f 连续，且存在若干线性映射 $\{f_k:k\in[K]\}$、若干内部非空且两两内部不相交的闭集 $\{\Omega_k:k\in[K]\}$，满足

$$\bigcup_{k=1}^{K}\Omega_k=\mathbb{R}^d\text{且}f(\boldsymbol{x})=f_k(\boldsymbol{x}),\forall\boldsymbol{x}\in\Omega_k,$$

则称 f 是连续分段线性函数. 其中 f_k 是 f 的线性分段，Ω_k 是 f_k 对应的投影区域.

定义 1 可以直接推广到四元数代数上的连续分段线性函数，只需将 $\mathbb{H}^d$ 视为 $\mathbb{R}^{4d}$ 即可. 投影区域的数量可以衡量连续分段线性函数的复杂度，投影区域

越多意味着函数越复杂. 但投影区域未必是凸集, 甚至不一定是连通的. 因此, 具有良好几何性质的凸线性区域数被提出并用来衡量连续分段线性函数的复杂度[26], 下面我们回顾凸线性区域数的定义.

定义 2. 令 $f:\mathbb{R}^d\to\mathbb{R}$或 $f:\mathbb{H}^d\to\mathbb{R}$为一个连续分段线性函数, $\{f_k:k\in[K]\}$与$\{\Omega_k:k\in[K]\}$为其线性分段与投影区域. 若所有Ω_k均为凸集, 则称$\{\Omega_k:k\in[K]\}$为f的线性凸分割. f的凸线性区域数κ_f为f的线性凸分割基数的最小值, 即

$$\kappa_f=\min\left\{|A|\,\middle|\,A\text{为函数}f\text{的线性凸分割}\right\}.$$

令$\boldsymbol{\theta}$为神经网络的可学参数, Θ表示可学参数的定义域, 并记神经网络为$f(\boldsymbol{x};\boldsymbol{\theta})$. 一个使用连续分段线性函数且结构固定的神经网络对应于一个连续分段线性函数族, 该函数族中函数的凸线性区域数的最大值称为该网络结构的最大凸线性区域数κ, 即

$$\kappa=\max\left\{\kappa_{f(\boldsymbol{x};\boldsymbol{\theta})}\,\middle|\,\boldsymbol{\theta}\in\Theta\right\}.$$

最大凸线性区域数刻画了神经网络能表达的连续分段线性函数的复杂度的上界, 更大的最大凸线性区域数对应了更强的表示能力. 下面我们研究实值神经网络与四元数神经网络的最大凸线性区域数.

定理 2. 设输入空间为$\mathbb{R}^{4d}$或$\mathbb{H}^d$. 单隐层实值神经网络使用参数为a的 ReLU 型激活函数, 且具有n个隐层神经元. 单隐层四元数神经网络使用分开激活的参数为a的 ReLU 型激活函数, 且具有m个隐层神经元. 记$\kappa_{\mathbb{R}}(n)$与$\kappa_{\mathbb{H}}(m)$分别为单隐层实值神经网络与单隐层四元数神经网络的最大凸线性区域数. 则

$$\kappa_{\mathbb{R}}(n)\leqslant\sum_{k=0}^{\min\{4d,n\}}\binom{n}{k},\kappa_{\mathbb{H}}(m)\leqslant\sum_{k=0}^{\min\{4d,4m\}}\binom{4m}{k},$$

且当且仅当$a\neq1$时上界是紧的.

定理 2 给出了单隐层实值神经网络与单隐层四元数神经网络的最大凸线性区域数上界, 并给出了上界是紧的充要条件. 当$n=4m$时, 最大凸线性区域数$\kappa_{\mathbb{R}}(n)$与$\kappa_{\mathbb{H}}(m)$的上界具有相同的形式. 此时, 实值神经网络的参数量为

$$N_{\mathbb{R}}=n\times4d+n+n=4m(4d+2),$$

四元数神经网络的实值参数量为

$$N_{\mathbb{H}}=4(m\times d+m+m)=4m(d+2).$$

当输入维度d较大时, $N_{\mathbb{R}}\approx4N_{\mathbb{H}}$, 即在高维任务中, 实值神经网络需约 4 倍参数量才能产生与四元数神经网络相同的最大凸线性区域数. 也可以类似地证明在高维任务中, 复值神经网络需约 2 倍参数量才能产生与四元数神经网络相同的最大凸线性区域数. 这表明四元数神经网络具有更强的表示能力.

定理 2 的证明将神经网络的最大凸线性区域数转化为高维空间中的超平面排列问题: 神经元对应于输入空间中的超平面, 所有神经元对应的超平面在输入空间中分割而成的区域数即为最大凸线性区域数的上界. 1 个实值神经元对应 1 个超平面, 而 1 个四元数神经元对应 4 个超平面, 从而四元数神经元具有更强的表示能力. 上界紧的充要条件很直观: 当$a=1$时, 激活函数为线性函数, 此时整个神经网络是一个线性函数, 因而最大凸线性区域数恒为 1; 当$a\neq1$时, 可以选取适当的参数, 使得所有神经元对应的超平面分割而成的每个区域的函数表达式两两不同, 因而这些超平面分割而成的区域数即为神经网络的最大凸线性区域数.

证明. 在单隐层实值神经网络中, 1 个实值神经元具有 2 个凸线性区域, 且凸线性区域的边界为输入空间中的 1 个超平面. 因此, 1 个实值神经元对应于输入空间中的 1 个超平面, 这个超平面将输入空间分为 2 个区域, 每个区域中该神经元均为线性函数且 2 个区域中的线性函数不同. 从而具有n个隐层神经元的实值神经网络对应于n个超平面, 这些超平面分割输入空间而成的每个区域中实值神经网络均为线性函数, 即这些超平面将空间分割而成的区域数即为最大凸线性区域数的上界. 由于n个超平面最多将d维空间分成

$$\sum_{k=0}^{\min\{d,n\}}\binom{n}{k}$$

个连通区域[34], 因此具有n个隐层神经元的单隐层实值神经网络的最大凸线性区域数满足

$$\kappa_{\mathbb{R}}(n)\leqslant\sum_{k=0}^{\min\{4d,n\}}\binom{n}{k}.$$

在四元数神经网络中, 1 个四元数神经元具有 16 个凸线性区域, 且凸线性区域的边界构成输入空间中的 4 个互相垂直的超平面. 因此, 1 个四元数神经元对应于输入空间中的 4 个互相垂直的超平面, 这些超平面将输入空间分为 16 个区域, 每个区域中该神经元均为线性函数且 16 个区域中的线性函数两两不同. 从而具有m个隐层神经元的四元数神经网络对应于输入空间中的m组超平面, 每组中的 4 个超平面互相垂直. 因此, 具有m个隐层神经元的四元数神经网络的最大凸线性区域数满足

$$\kappa_{\mathbb{H}}(m)\leqslant\sum_{k=0}^{\min\{4d,4m\}}\binom{4m}{k}.$$

当$a=1$时, 激活函数为线性函数. 由于线性函数

的线性组合依旧是线性函数, 可知实值神经网络与四元数神经网络均为线性函数. 从而最大凸线性区域数恒为 1, 即 $a=1$ 时上界不紧.

当 $a\neq 1$ 时, 令 $H_1,H_2,\cdots,H_n$ 为实值神经网络 $f_{\mathbb{R}}$ 对应的 n 个超平面. 对任意 $i\in[n]$, 记满足 $\boldsymbol{w}_{\mathbb{R},i}\boldsymbol{x}_{\mathbb{R}}+b_{\mathbb{R},i}\geqslant 0$ 的区域为超平面 H_i 的正区域, 满足 $\boldsymbol{w}_{\mathbb{R},i}\boldsymbol{x}_{\mathbb{R}}+b_{\mathbb{R},i}\leqslant 0$ 的区域为超平面 H_i 的负区域, 其中 $\boldsymbol{w}_{\mathbb{R},i}$ 为矩阵 $\boldsymbol{W}_{\mathbb{R}}$ 的第 i 行, $b_{\mathbb{R},i}$ 为向量 $\boldsymbol{b}_{\mathbb{R}}$ 的第 i 行. 令 $R_1,R_2,\cdots,R_N$ 为这 n 个超平面分割输入空间而成的区域. 定义

$$\boldsymbol{C}=(c_{ij})_{i\in[n],j\in[N]}\in\{0,1\}^{n\times N},$$

其中 $c_{ij}=I(R_j$ 在超平面 H_i 的正区域中$)$. 在区域 R_j 中, 实值神经网络 $f_{\mathbb{R}}$ 满足

$$f_{\mathbb{R}|R_j}(\boldsymbol{x}_{\mathbb{R}})=\sum_{i=1}^{n}c_{ij}\alpha_{\mathbb{R},i}(\boldsymbol{w}_{\mathbb{R},i}\boldsymbol{x}_{\mathbb{R}}+b_{\mathbb{R},i}),$$

其中 $\alpha_{\mathbb{R},i}$ 为向量 $\boldsymbol{\alpha}_{\mathbb{R}}$ 的第 i 列. 则实值神经网络 $f_{\mathbb{R}}$ 在 2 个不同的区域 R_j 与 R_k 上表达式相同当且仅当

$$\sum_{i=1}^{n}(c_{ij}-c_{ik})\alpha_{\mathbb{R},i}(\boldsymbol{w}_{\mathbb{R},i},b_{\mathbb{R},i})=\boldsymbol{0}. \tag{1}$$

由区域 $R_1,R_2,\cdots,R_N$ 定义可知, 矩阵 $\boldsymbol{C}$ 的列向量 $\boldsymbol{c}_j$ 两两不同. 从而当 $(\boldsymbol{w}_{\mathbb{R},i},b_{\mathbb{R},i})$ 全不为零向量时, 式(1)是关于变量 $\boldsymbol{\alpha}_{\mathbb{R}}$ 的系数不全为 0 的线性方程组, 故满足式(1)的 $\boldsymbol{\alpha}_{\mathbb{R}}$ 是 $\mathbb{R}^n$ 中的零测集. 由于有限个零测集的并为零测集, 可知使得实值神经网络 $f_{\mathbb{R}}$ 在区域 $R_1,R_2,\cdots,R_N$ 上存在相同表达式的 $\boldsymbol{\alpha}_{\mathbb{R}}$ 构成 $\mathbb{R}^n$ 中的零测集. 因此, 存在 $\boldsymbol{\alpha}_{\mathbb{R}}$ 的取值, 使得实值神经网络 $f_{\mathbb{R}}$ 在区域 $R_1,R_2,\cdots,R_N$ 上两两不同, 即网络的最大凸线性区域数与超平面分割输入空间的连通区域数相等. 从而 $a\neq 1$ 时, 实值神经网络的最大凸线性区域数上界是紧的.

四元数神经网络对应的超平面虽然不是任意的, 但每组超平面的互相垂直性质不会影响分割得到的区域数. 同理可证 $a\neq 1$ 时四元数神经网络的最大凸线性区域数上界是紧的. 证毕.

定理 2 表明使用分开激活的激活函数的单隐层四元数神经网络比单隐层实值神经网络表示能力更强, 其证明依赖于超平面排列得到的紧的最大凸线性区域数上界. 在深层神经网络中, 只有第 1 隐层的凸线性区域数可由超平面的排列得到, 更深层的凸线性区域数需要研究凸线性区域的排列. 文献 [26] 给出了深层神经网络的最大凸线性区域的上下界, 但直接使用该上下界会得出四元数神经网络的最大凸线性区域数下界小于实值神经网络的最大凸线性区域数上界的结论, 无法比较 2 种神经网络的表示能力. 因此, 从最大凸线性区域数的角度比较深层神经网络的表示能力还需更紧的上下界.

3.3 非分开激活时的逼近分离

本节对比使用 ReLU 激活函数的单隐层实值神经网络与使用非分开激活的激活函数 $\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$ 的单隐层四元数神经网络的表达能力. 我们先回顾 (ε,D)-逼近的定义[30].

定义 3. 令 $g:\mathbb{R}^d\to\mathbb{R}$ 为一个映射, F 为一些 $\mathbb{R}^d$ 到 $\mathbb{R}$ 的函数构成的集合, D 为 $\mathbb{R}^d$ 上的分布. 如果存在函数 $f\in F$, 使得

$$E_{\boldsymbol{x}\sim D}(|f(\boldsymbol{x})-g(\boldsymbol{x})|)\leqslant\varepsilon,$$

则称函数集合 F 能 (ε,D)-逼近函数 g.

定义 3 使用函数值之差的绝对值的期望, 而非函数值之差的平方的期望[30] 作为 2 个函数差距的度量. 2 种定义形式在一定条件下是可以相互转换的, 这里为了证明的简洁采用了绝对值的形式. 此外, 定义 3 可以直接推广到四元数代数上的连续分段线性函数, 只需将 $\mathbb{H}^d$ 视为 $\mathbb{R}^{4d}$ 即可. 取函数集合 F 是一个结构固定的神经网络构成的函数空间, 函数 g 为目标函数. 此时, (ε,D)-逼近刻画了神经网络对目标函数的逼近能力. 下面我们证明本节的第 1 个结论.

定理 3. 令 $g:\mathbb{R}^{4d}\to\mathbb{R}$ 为任意映射, D 为 $\mathbb{R}^{4d}$ 上的任意分布, $\varepsilon>0$ 为任意正实数. 若函数 g 能被使用 n 个隐层神经元与 ReLU 激活函数的单隐层实值神经网络 (ε,D)-逼近, 则函数 g 也能被使用 n 个隐层神经元、激活函数 $\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$、相同权重模长的单隐层四元数神经网络 (ε,D)-逼近.

定理 3 表明实值神经网络能逼近的函数也能被四元数神经网络逼近, 且四元数神经网络与实值神经网络具有相同的隐层大小和权重模长. 相同的隐层大小意味着四元数神经网络的参数量与实值神经网络同阶. 这表明四元数神经网络的表示能力不弱于实值神经网络. 定理 3 的证明是构造性的, 对于任意给定的实值神经网络, 都可以选取合适的四元数神经网络参数, 使得 2 个网络的输出一致.

证明. 由于函数 g 能被具有 n 个隐层神经元的实值神经网络 (ε,D)-逼近, 可知存在参数 $\boldsymbol{\alpha}_{\mathbb{R}}\in\mathbb{R}^{1\times n},\boldsymbol{W}_{\mathbb{R}}\in\mathbb{R}^{n\times 4d},\boldsymbol{b}_{\mathbb{R}}\in\mathbb{R}^n$ 以及单隐层实值神经网络 $f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})=\boldsymbol{\alpha}_{\mathbb{R}}\sigma_{\mathbb{R}}(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}})$, 使得

$$E_{\boldsymbol{x}_{\mathbb{R}}\sim D}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g(\boldsymbol{x}_{\mathbb{R}})|)\leqslant\varepsilon,$$

其中 $\sigma_{\mathbb{R}}$ 为 ReLU 激活函数. 令

$$\boldsymbol{\alpha}_{\mathbb{H},\mathrm{R}}=\boldsymbol{\alpha}_{\mathbb{R}},\boldsymbol{\alpha}_{\mathbb{H},\mathrm{I}}=\boldsymbol{\alpha}_{\mathbb{H},\mathrm{J}}=\boldsymbol{\alpha}_{\mathbb{H},\mathrm{K}}=\boldsymbol{0},$$

$$\boldsymbol{b}_{\mathbb{H},\mathrm{R}}=\boldsymbol{b}_{\mathbb{R}},\boldsymbol{b}_{\mathbb{H},\mathrm{I}}=\boldsymbol{b}_{\mathbb{H},\mathrm{J}}=\boldsymbol{b}_{\mathbb{H},\mathrm{K}}=\boldsymbol{0},$$
$$\boldsymbol{w}_{\mathbb{H},\mathrm{R},j}=\boldsymbol{w}_{\mathbb{R},4j-3},\boldsymbol{w}_{\mathbb{H},\mathrm{I},j}=-\boldsymbol{w}_{\mathbb{R},4j-2},$$
$$\boldsymbol{w}_{\mathbb{H},\mathrm{J},j}=-\boldsymbol{w}_{\mathbb{R},4j-1},\boldsymbol{w}_{\mathbb{H},\mathrm{K},j}=-\boldsymbol{w}_{\mathbb{R},4j},j\in[d],$$

其中$\boldsymbol{w}_j$表示矩阵$\boldsymbol{W}$的第j列. 则单隐层四元数神经网络$f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=\mathrm{Re}[\boldsymbol{\alpha}_{\mathbb{H}}\sigma_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})]$具有$n$个隐层神经元、与实值神经网络相同的权重模长, 且满足

$$f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}}),\forall\boldsymbol{x}_{\mathbb{H}}\in\mathbb{H}^d,$$

其中$\boldsymbol{x}_{\mathbb{R}}$为$\boldsymbol{x}_{\mathbb{H}}$的实值表示. 从而

$$E_{\boldsymbol{x}_{\mathbb{R}}\sim D}(|f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})-g(\boldsymbol{x}_{\mathbb{R}})|)\leqslant\varepsilon,$$

即函数g能被使用激活函数$\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$、$n$个隐层神经元、相同权重模长的四元数神经网络$(\varepsilon,D)$-逼近. 证毕.

下面我们介绍一个重要引理.

引理 1. 令$g=I(y\geqslant 0)$为阶跃函数, D为$[-a,a]$上的均匀分布($a>0$), f为一个L-Lipschitz 连续函数. 则

$$E_{y\sim D}(|f(y)-g(y)|)\geqslant\min\{8^{-1},(16La)^{-1}\}.$$

引理 1 给出了 Lipschitz 连续函数逼近阶跃函数的误差下界, 该下界与 Lipschitz 连续函数的 Lipschitz 常数成反比.

证明. 不妨设$f(0)\leqslant 0.5$. 则对于任意$y\geqslant 0$, 三角不等式与函数f的L-Lipschitz 连续性表明

$$|f(y)-g(y)|\geqslant|f(0)-g(y)|-|f(0)-f(y)|\geqslant 0.5-Ly.$$

当$L\leqslant 0.5a^{-1}$时, 有

$$E_{y\sim D}(|f(y)-g(y)|)\geqslant\int_0^a(0.5-Ly)(2a)^{-1}\mathrm{d}y\geqslant 8^{-1}.$$

当$L\geqslant 0.5a^{-1}$时, 有

$$E_{y\sim D}(|f(y)-g(y)|)\geqslant\int_0^{(2L)^{-1}}(0.5-Ly)(2a)^{-1}\mathrm{d}y\geqslant(16La)^{-1}.$$

综上, 对任意 Lipschitz 常数L, 有

$$E_{y\sim D}(|f(y)-g(y)|)\geqslant\min\{8^{-1},(16La)^{-1}\}.$$

$f(0)\geqslant 0.5$的情形同理可证. 证毕.

下面我们证明本节的第 2 个结论.

定理 4. 存在$\mathbb{R}^{4d}\to\mathbb{R}$的映射$g_d$与$\mathbb{R}^{4d}$上的分布$D_d$, 使得下列结论成立:

1)对于任意$\varepsilon_d>0$, 单隐层四元数神经网络可以用激活函数$\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$、1 个隐层神经元以及模长为$O(1)$的参数$(\varepsilon_d,D_d)$-逼近函数$g_d$.

2)对于任意多项式函数$poly_0(d)$, 存在d_0以及$\varepsilon_d=\Omega(1)$, 对于任意$d\geqslant d_0$, 单隐层实值神经网络使用 ReLU 激活函数、$poly_0(d)$个隐层神经元以及模长为$poly_0(d)$的参数无法使用(ε_d,D_d)-逼近函数g_d.

定理 4 表明存在一列函数g_d与分布D_d, 单隐层四元数神经网络可以用有界的隐层神经元数目与有界的权重以任意精度逼近函数g_d; 而实值神经网络在高维输入时无法用多项式级的隐层神经元数目与多项式大小的权重以常数精度逼近函数g_d, 即实值神经网络只有使用指数多个隐层神经元或是指数大的参数才有可能以任意精度逼近函数g_d, 而指数依赖性在高维任务中是不可接受的. 这一结果表明四元数神经网络在逼近特定函数时具有更强的表示能力.

定理 4 的证明是构造性的. 函数g_d是一列特殊的非连续函数, 使得使用具有间断点的激活函数的四元数神经网络可以很好地表示函数g_d. 而实值神经网络使用的激活函数是连续的, 当隐层神经元数目与参数权重均有限时, 实值神经网络是一个 Lipschitz 常数有限的连续函数, 因而无法表示非连续函数. 分布D_d是集中于函数g_d间断点附近的分布, 随着输入维度增加, 分布向间断点收缩, 使得实值神经网络逼近函数g_d更加困难.

证明. 对$\boldsymbol{x}_{\mathbb{R}}\in\mathbb{R}^{4d}$, 定义映射$g_d$为

$$g_d(\boldsymbol{x}_{\mathbb{R}})=x_{\mathbb{R},2}I(x_{\mathbb{R},1}\geqslant 0),$$

其中$x_{\mathbb{R},i}$为向量$\boldsymbol{x}_{\mathbb{R}}$的第$i$个分量. 定义分布$D_d=U(A_d)$为集合$A_d$上的均匀分布, 其中

$$A_d=[-2^{-d},2^{-d}]\times[-1,1]^{d-1}.$$

在四元数神经网络$f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=\mathrm{Re}[\boldsymbol{\alpha}_{\mathbb{H}}\sigma_{\mathbb{H}}(\boldsymbol{W}_{\mathbb{H}}\boldsymbol{x}_{\mathbb{H}}+\boldsymbol{b}_{\mathbb{H}})]$中, 选取参数:

$$\boldsymbol{\alpha}_{\mathbb{H}}=(-\mathrm{i})\in\mathbb{H}^{1\times 1},$$
$$\boldsymbol{b}_{\mathbb{H}}=(0)\in\mathbb{H}^{1\times 1},$$
$$\boldsymbol{W}_{\mathbb{H}}=(1,0,0,\cdots,0)\in\mathbb{H}^{1\times d}.$$

则该四元数神经网络具有 1 个隐层神经元, 参数的最大模长为 1, 且满足

$$\begin{aligned}f_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}})=&\mathrm{Re}[-\mathrm{i}\sigma_{\mathbb{H}}(x_{\mathbb{H},1})]=\\&\mathrm{Re}[-\mathrm{i}x_{\mathbb{H},1}I(x_{\mathbb{H},\mathrm{R},1}\geqslant 0)]=\\&x_{\mathbb{R},2}I(x_{\mathbb{R},1}\geqslant 0)=\\&g_d(\boldsymbol{x}_{\mathbb{R}}),\end{aligned}$$

其中$\boldsymbol{x}_{\mathbb{R}}$是$\boldsymbol{x}_{\mathbb{H}}$的实值表示. 因此, 函数$g_d$能被使用 1 个隐层神经元与模长不超过 1 的参数的单隐层四元数神经网络表示, 从而(ε_d,D_d)-逼近.

使用实值神经网络$f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})=\boldsymbol{\alpha}_{\mathbb{R}}\sigma_{\mathbb{R}}(\boldsymbol{W}_{\mathbb{R}}\boldsymbol{x}_{\mathbb{R}}+\boldsymbol{b}_{\mathbb{R}})$逼近函数$g_d$时, 误差满足

$$\begin{aligned}&E_{\boldsymbol{x}_{\mathbb{R}}\sim D_d}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g_d(\boldsymbol{x}_{\mathbb{R}})|)=\\&E_{x_3,\cdots,x_{4d}\sim D_2}E_{x_2\sim D_2}E_{x_1\sim D_1}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g_d(\boldsymbol{x}_{\mathbb{R}})|),\end{aligned}$$

其中$x_i=x_{\mathbb{R},i}$表示实值输入的第i维, x_1服从区间$[-2^{-d},2^{-d}]$上的均匀分布$D_1=U(-2^{-d},2^{-d})$, $x_2,x_3,\cdots,x_{4d}$均服从区间$[-1,1]$上的均匀分布$D_2=U(-1,1)$. 当x_2固定时, 函数g_d为x_2乘以关于x_1的阶跃函数. 由于实值神经网络$f_{\mathbb{R}}$的隐层神经元数为$poly_0(d)$且参数模长为

$poly_0(d)$，ReLU 激活函数是 1-Lipschitz 连续的，可知该网络 $f_{\mathbb{R}}$ 是 $poly(d)$-Lipschitz 连续的. 因此

$$\begin{aligned}&E_{\boldsymbol{x}_{\mathbb{R}}\sim D_d}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g_d(\boldsymbol{x}_{\mathbb{R}})|)=\\&E_{x_3,\cdots,x_{4d}}E_{x_2}E_{x_1}(|x_2\|x_2^{-1}f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-I(x_1\geqslant 0)|)\geqslant\\&E_{x_3,\cdots,x_{4d}}E_{x_2}(|x_2|\min\{8^{-1},|x_2|2^d poly(d)^{-1}\}),\end{aligned}\tag{2}$$

其中第 2 行中括号中的随机变量在零测集 $x_2=0$ 上定义为 0，不等式使用了引理 1 的结论并将常数 16 合并到了 $poly(d)$ 中. 令 $c_d=2^{-(d+3)}poly(d)$，当 d 足够大时，$c_d\leqslant 2^{-1}$，此时式(2)中关于 x_2 的期望为

$$\begin{aligned}&E_{x_2}(|x_2|\min\{8^{-1},|x_2|2^d poly(d)^{-1}\})=\\&\int_{|x_2|=0}^{c_d}x_2^2 2^d poly(d)^{-1}\mathrm{d}x_2+\int_{|x_2|=c_d}^{1}8^{-1}|x_2|\mathrm{d}x_2=\\&\frac{2^{d+1}poly(d)^{-1}c_d^3}{3}+\frac{1-c_d^2}{8}>\frac{1}{32},\end{aligned}$$

其中不等式使用了 $c_d\leqslant 2^{-1}$. 将上述结果代入到期望误差中，可知 d 足够大时，期望误差满足

$$\begin{aligned}&E_{\boldsymbol{x}_{\mathbb{R}}\sim D_d}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g_d(\boldsymbol{x}_{\mathbb{R}})|)>\\&E_{x_3,\cdots,x_{4d}}(32^{-1})=32^{-1},\end{aligned}$$

取 $\varepsilon_d=32^{-1}=\Omega(1)$，即有

$$E_{\boldsymbol{x}_{\mathbb{R}}\sim D_d}(|f_{\mathbb{R}}(\boldsymbol{x}_{\mathbb{R}})-g_d(\boldsymbol{x}_{\mathbb{R}})|)>\varepsilon_d$$

从而，函数 g_d 不能被使用 $poly_0(d)$ 个隐层神经元与模长不超过 $poly_0(d)$ 的实值神经网络 (ε_d,D_d)-逼近. 证毕.

本节的定理 3 与定理 4 分别从一般情形与特殊情形分析了四元数神经网络相比于实值神经网络的表示能力. 一般情形下四元数神经网络的逼近效率不差于实值神经网络，且存在特殊情形使得四元数神经网络能更高效地逼近目标函数. 这 2 方面的结论构成了四元数神经网络与实值神经网络的逼近分离.

本节的逼近分离结果依赖于特殊构造的四元数神经网络激活函数. 逼近分离定理在常用的非分开激活函数上是否成立是一个值得研究的问题. 此外，定理 4 的构造虽然简单，但构造的目标函数是不连续的，分布的定义域随着维度增加是指数级收缩的. 通过更复杂的证明，或许可以将逼近分离结论推广到 Lipschitz 连续的目标函数[27]，或是单位立方体上的均匀分布[28].

4 实　　验

本节通过模拟实验来验证四元数神经网络更强的表示能力.4.1 节与 4.2 节分别对 3.1 节与 3.2 节分析的四元数神经网络与实值神经网络进行对比.

4.1 使用分开激活的四元数神经网络

本节对比使用分开激活的 ReLU 激活函数的四元数神经网络与使用 ReLU 激活的实值神经网络在 1 维子空间上的线性区域数.1 维子空间上的线性区域数直观地体现了神经网络的复杂度与表示能力.

我们随机初始化一个四元数神经网络或一个实值神经网络，并在输入空间中随机选取 100 个经过原点的 1 维子空间，每个 1 维子空间在 $[-10,10]$ 的范围内以 0.01 为间隔生成样本集. 我们计算神经网络在 100 个样本集上归一化后的函数值，及使用分段线性函数拟合这些函数值所需的线性区域数. 这些线性区域数的最大值作为衡量该神经网络在 1 维子空间上的线性区域数的指标.

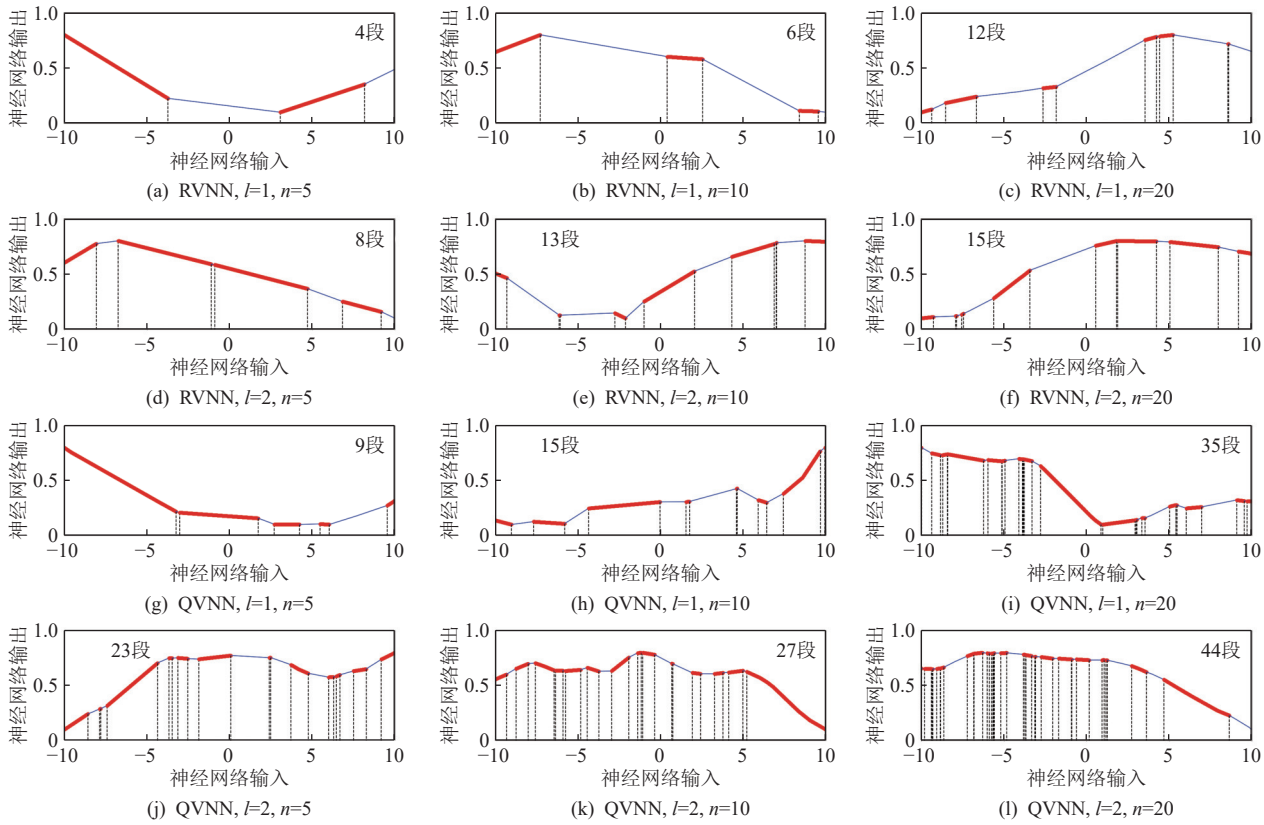
我们对不同隐层数量 l、不同隐层大小 n 的实值神经网络(real-valued neural network, RVNN)与四元数神经网络(quaternion-valued neural network, QVNN)进行上述实验，并在图 1 中展示了最大线性区域数及其对应的函数曲线，其中相邻的线性区域由粗线与细线区分，并在转折点添加了竖直虚线. 实验结果表明隐层数量与隐层大小相同时，四元数神经网络具有更大的最大线性区域数.

4.2 使用非分开激活的四元数神经网络

本节对比使用激活函数 $\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$ 的四元数神经网络与使用 ReLU 激活的实值神经网络在模拟实验中的泛化性能.

图 2 展示了使用实值神经网络与四元数神经网络学习一个实值神经网络时的测试损失. 其中输入是 $\mathbb{R}^{4d}$ 或 $\mathbb{H}^d$ 中的向量，目标实值神经网络由 1 个隐层神经元与 1 个输出神经元构成，学习的神经网络具有 n 个隐层神经元与 1 个输出神经元. 所有实值神经网络使用 ReLU 激活函数，四元数神经网络均使用激活函数 $\sigma_{\mathbb{H}}(q)=qI(q_{\mathrm{R}}\geqslant 0)$. 训练集包含 7 000 个样本，测试集包含 3 000 个样本，这些样本的特征从高斯分布中随机采样得到，标记是目标实值神经网络的输出. 2 个学习神经网络均使用随机初始化，并用步长为 0.01 的梯度下降算法对均方误差优化 100 轮. 所有实验重复 10 次并绘制了测试损失的均值与标准差. 图 3 展示了使用实值神经网络与四元数神经网络学习一个四元数神经网络的测试损失. 该实验与图 2 的设定相似，唯一区别是将目标神经网络替换为由 1 个隐层神经元与 1 个输出神经元构成的四元数神经网络.

实验结果表明学习实值神经网络时，四元数神经网络可以取得与实值神经网络相似的测试损失. 同时，学习四元数神经网络时，四元数神经网络取得了比实值神经网络更好的测试损失. 这些现象表明使用非分开激活的激活函数的四元数神经网络具有



(a) RVNN, l=1, n=5　(b) RVNN, l=1, n=10　(c) RVNN, l=1, n=20

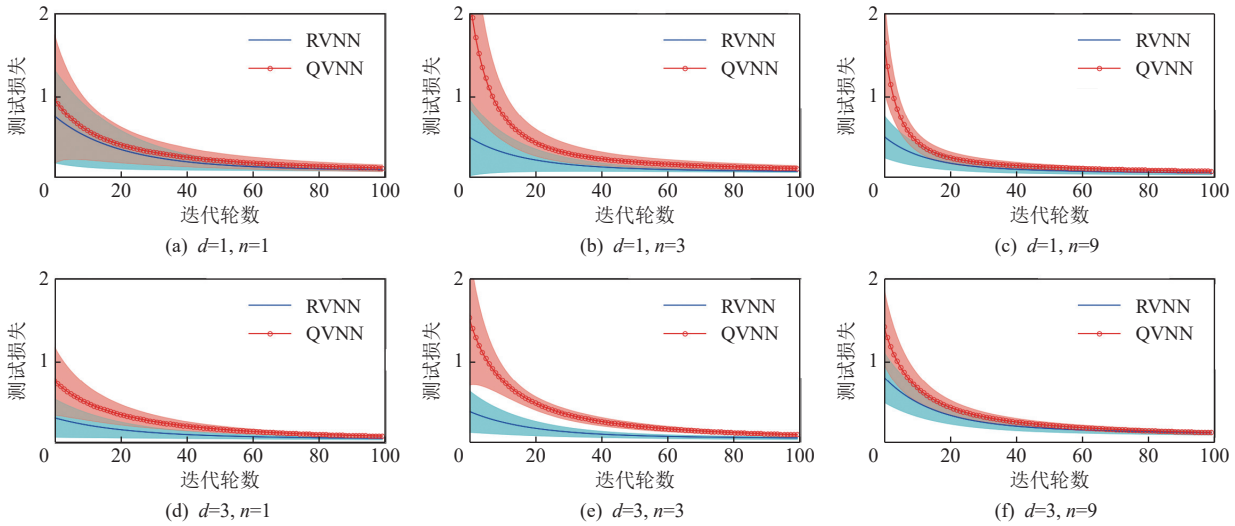
(d) RVNN, l=2, n=5　(e) RVNN, l=2, n=10　(f) RVNN, l=2, n=20

(g) QVNN, l=1, n=5　(h) QVNN, l=1, n=10　(i) QVNN, l=1, n=20

(j) QVNN, l=2, n=5　(k) QVNN, l=2, n=10　(l) QVNN, l=2, n=20

Fig. 1　Number of linear regions on a 1-dimensional subspace

图 1　1 维子空间上的线性区域数



(a) d=1, n=1　(b) d=1, n=3　(c) d=1, n=9

(d) d=3, n=1　(e) d=3, n=3　(f) d=3, n=9

Fig. 2　Test loss of learning a real-valued neural network

图 2　学习实值神经网络的测试损失

比实值神经网络更强的学习能力，也验证了四元数神经网络具有更强的表示能力.

5　总　　结

四元数神经网络是一种重要的神经网络模型，在许多任务中被成功应用并取得了比实值神经网络更好的表现. 本文从表示理论的角度为四元数神经网络提供理论解释. 首先，我们证明了使用非分开激活的激活函数的四元数神经网络的通用近似性质. 以往四元数神经网络的通用近似定理只关注分开激活的激活函数，我们的理论提供了首个使用非分开

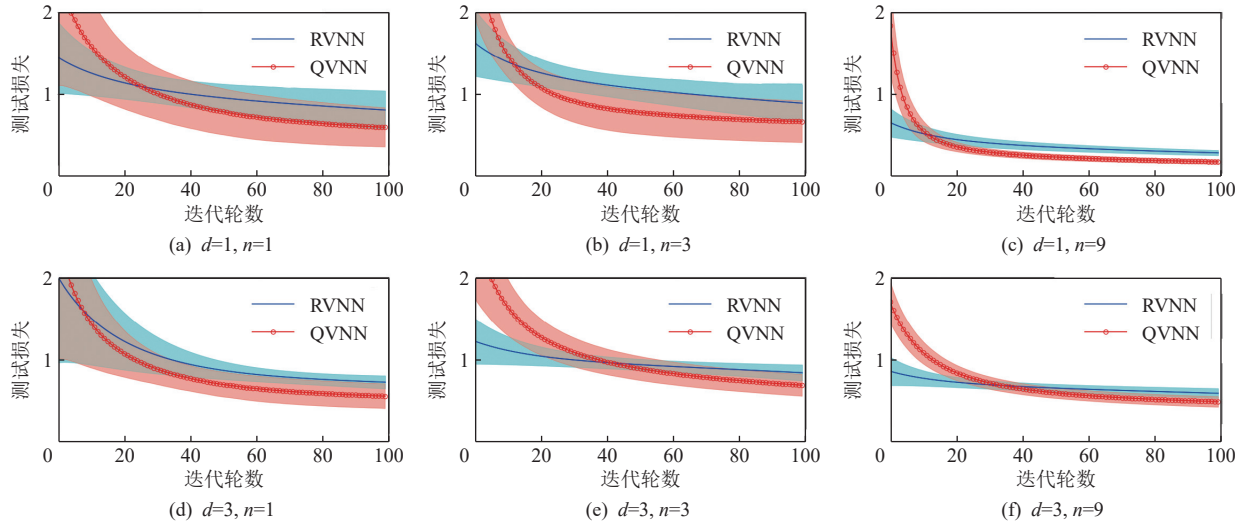


Fig. 3 Test loss of learning quaternion-valued neural network

图 3 学习四元数神经网络的测试损失

激活的激活函数的四元数神经网络的通用近似性质. 其次, 我们证明了四元数神经网络具有比实值神经网络更强的表示能力. 在使用分开激活的激活函数时, 我们证明了单隐层实值神经网络需要约 4 倍参数才能生成与单隐层四元数神经网络相同的最大凸线性区域数. 在使用非分开激活的激活函数时, 我们证明了单隐层四元数神经网络与单隐层实值神经网络的逼近分离: 四元数神经网络只需相同的隐层大小与参数模长即可表示实值神经网络, 而实值神经网络需要指数多个隐层神经元或指数大的参数才可能逼近四元数神经网络.

本文初步探讨了四元数神经网络的表示能力, 四元数神经网络的理论性质还有许多方向可以研究. 四元数的表示能力还存在许多局限. 首先, 实值神经网络与复值神经网络的通用近似性质均已给出激活函数需满足的充要条件[14,22], 而四元数神经网络的通用近似性质还只有充分条件. 其次, 四元数神经网络的逼近优势建立在特殊激活函数以及浅层神经网络之上. 这些结论在更一般情形下是否成立值得研究. 此外, 四元数神经网络的逼近能力与其实际表现有很大差距, 还需要其他角度的理论探索, 比如四元数神经网络的优化性质、泛化能力等. 最后, 非分开激活的激活函数相比分开激活的激活函数能为四元数神经网络带来更强的逼近分离性质, 但常用的四元数神经网络结构及其优化算法都建立在实值神经网络的基础上且未考虑激活函数可能带来的间断点或奇点. 因此, 优化算法如何处理激活函数带来的间断点与奇点, 以及设计理论性质与实验表现更好的激活函数和网络结构都是未来可以研究的方向.

作者贡献声明: 吴锦辉负责理论证明、完成实验和撰写论文; 姜远负责写作指导和修改审定.

参 考 文 献

[1] Oyama K, Hirose A. Phasor quaternion neural networks for singular point compensation in polarimetric-interferometric synthetic aperture radar[J]. IEEE Transactions on Geoscience and Remote Sensing, 2018, 57(5): 2510−2519

[2] Parcollet T, Morchid M, Linares G. Deep quaternion neural networks for spoken language understanding [C] //Proc of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop. Piscataway, NJ: IEEE, 2017: 504−511

[3] Wu Yue, Yuan Yongzhe, Yue Mingyu, et al. Feature mining method of multi-dimensional information fusion in point cloud registration[J]. Journal of Computer Research and Development, 2022, 59(8): 1732−1741 (in Chinese)
(武越, 苑咏哲, 岳铭煜, 等. 点云配准中多维度信息融合的特征挖掘方法[J]. 计算机研究与发展, 2022, 59(8): 1732−1741)

[4] Bayro-Corrochano E, Lechuga-Gutiérrez L, Garza-Burgos M. Geometric techniques for robotics and HMI: Interpolation and haptics in conformal geometric algebra and control using quaternion spike neural networks[J]. Robotics and Autonomous Systems, 2018, 104: 72−84

[5] Parcollet T, Ravanelli M, Morchid M, et al. Quaternion recurrent neural networks [C/OL] //Proc of the 7th Int Conf on Learning Representations. 2019 [2024-07-14]. https://openreview.net/pdf?id=ByMHvs0cFQ

[6] Shoemake K. Animating rotation with quaternion curves [C] //Proc of the 12th Annual Conf on Computer Graphics and Interactive Techniques. New York: ACM, 1985: 245−254

[7] Parcollet T, Morchid M, Linarès G. A survey of quaternion neural networks[J]. Artificial Intelligence Review, 2020, 53(4): 2957−2982

[8] Arena P, Fortuna L, Muscato G, et al. Multilayer perceptrons to approximate quaternion valued functions[J]. Neural Networks, 1997, 10(2): 335−342

[9] Valle M E, Vital W L, Vieira G. Universal approximation theorem for vector- and hypercomplex-valued neural networks [J]. arXiv preprint, arXiv: 2401.02277, 2014

[10] Ujang B C, Took C C, Mandic D P. Quaternion-valued nonlinear adaptive filtering[J]. IEEE Transactions on Neural Networks, 2011, 22(8): 1193−1206

[11] Cybenko G. Approximation by superpositions of a sigmoidal function[J]. Mathematics of Control, Signals and Systems, 1989, 2(4): 303−314

[12] Funahashi K I. On the approximate realization of continuous mappings by neural networks[J]. Neural Networks, 1989, 2(3): 183−192

[13] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359−366

[14] Leshno M, Lin V Y, Pinkus A, et al. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function[J]. Neural Networks, 1993, 6(6): 861−867

[15] Seidl D R, Lorenz R D. A structure by which a recurrent neural network can approximate a nonlinear dynamic system [C] //Proc of the 1991 Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 1991: 709−714

[16] Funahashi K I, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks[J]. Neural Networks, 1993, 6(6): 801−806

[17] Chow T W, Li Xiaodong. Modeling of continuous time dynamical systems with input by recurrent neural networks[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2000, 47(4): 575−578

[18] Li Xiaodong, Ho J K, Chow T W. Approximation of dynamical time-variant systems by continuous-time recurrent neural networks[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2005, 52(10): 656−660

[19] Schäfer A M, Zimmermann H G. Recurrent neural networks are universal approximators [C] //Proc of the 16th Int Conf of Artificial Neural Networks. Berlin: Springer, 2006: 632−640

[20] Zhou Dingxuan. Universality of deep convolutional neural networks[J]. Applied and Computational Harmonic Analysis, 2020, 48(2): 787−794

[21] Arena P, Fortuna L, Re R, et al. On the capability of neural networks with complex neurons in complex valued functions approximation [C] //Proc of the 1993 IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 1993: 2168−2171

[22] Voigtlaender F. The universal approximation theorem for complex-valued neural networks[J]. Applied and Computational Harmonic Analysis, 2023, 64: 33−61

[23] Barron A R. Approximation and estimation bounds for artificial neural networks[J]. Machine Learning, 1994, 14(1): 115−133

[24] Arora R, Basu A, Mianjy P, et al. Understanding deep neural networks with rectified linear units [C/OL] //Proc of the 6th Int Conf on Learning Representations. 2018 [2024-07-14]. https://openreview.net/pdf?id=B1J_rgWRW

[25] Montufar G F, Pascanu R, Cho K, et al. On the number of linear regions of deep neural networks [C] //Advances in Neural Information Processing Systems 27. Cambridge, MA: MIT, 2014: 2924−2932

[26] Goujon A, Etemadi A, Unser M. On the number of regions of piecewise linear neural networks[J]. Journal of Computational and Applied Mathematics, 2024, 441: 115667

[27] Eldan R, Shamir O. The power of depth for feedforward neural networks [C] //Proc of the 29th Conf on Learning Theory. NewYork: PMLR, 2016: 907−940

[28] Telgarsky M. Benefits of depth in neural networks [C] //Proc of the 29th Conf on Learning Theory. New York: PMLR, 2016: 1517−1539

[29] Zhang Shaoqun, Gao Wei, Zhou Zhihua. Towards understanding theoretical advantages of complex-reaction networks[J]. Neural Networks, 2022, 151: 80−93

[30] Wu Jinhui, Zhang Shaoqun, Jiang Yuan, et al. Theoretical exploration of flexible transmitter model[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 35(3): 3674−3688

[31] Fukushima K. Visual feature extraction by a multilayered network of analog threshold elements[J]. IEEE Transactions on Systems Science and Cybernetics, 1969, 5(4): 322−333

[32] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models [C/OL] //Proc of the 2013 ICML Workshop on Deep Learning for Audio, Speech, and Language Processing. 2013 [2024-07-15]. http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

[33] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification [C] //Proc of the 2015 IEEE Int Conf on Computer Vision. Piscataway , NJ: IEEE, 2015: 1026−1034

[34] Zaslavsky T. Facing up to Arrangements: Face-count Formulas for Partitions of Space by Hyperplanes [M]. Providence, RI: American Mathematical Society, 1975

Wu Jinhui, born in 1998. PhD candidate. His main research interests include neural network theories and machine learning.

吴锦辉, 1998 年生. 博士研究生. 主要研究方向为神经网络理论、机器学习.

Jiang Yuan, born in 1976. PhD, professor, PhD supervisor. Her main research interests include artificial intelligence, machine learning, and intelligent medical applications.

姜　远, 1976 年生. 博士, 教授, 博士生导师. 主要研究方向为人工智能、机器学习、智能医学应用.