

A discrete Hahn wavelet neural network method for Bratu-type equations

Parisa Rahimkhani⁽¹⁾ and Yadollah Ordokhani⁽²⁾ ¹

⁽¹⁾ Faculty of Science, Mahallat Institute of Higher Education, Mahallat, Iran

⁽²⁾ Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran

Received: 23 November 2024

Accepted: 15 April 2026

Published online: 12 June 2026

Abstract: In this paper, a new numerical method based on the neural network and the discrete Hahn wavelets is presented for solving the Bratu-type equations. First, the discrete Hahn wavelets and some of their important properties are expressed to construct the Hahn wavelets neural network. The presented neural network consists of three layers: input layer, hidden layer, output layer. In this method, the Hahn wavelets and sinh functions are used as the activation functions of the hidden layer and output layer, respectively. Then, an analytical optimization method is employed to adjust the neural network weights so that the approximated solution satisfies the given problem. Using this method, the original problem is converted into a system of algebraic equations, which is solved by Newton's iterative method. Finally, the accuracy and efficiency of the proposed method are examined through several numerical examples.

Keywords: Neural network, Hahn wavelets, Bratu equations, Numerical method, Analytical optimization method.



©2026 Kharazmi University, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0 license) (<http://creativecommons.org/licenses/by-nc/4.0/>).

¹Corresponding author

E-mail addresses: (Parisa Rahimkhani) rahimkhani.parisa@mahallat.ac.ir, (Yadollah Ordokhani) ordokhani@alzahra.ac.ir



روش شبکه عصبی موجک هان گسسته برای معادلات براتو

پریسا رحیم خانی^(۱) و یداله اردوخانی^(۲)

^(۱) گروه علوم کامپیوتر، دانشکده علوم، مرکز آموزش عالی محلات، محلات، ایران

^(۲) گروه ریاضی، دانشکده علوم ریاضی، دانشگاه الزهرا(س)، تهران، ایران

تاریخ انتشار: ۱۴۰۵/۳/۲۲

تاریخ پذیرش: ۱۴۰۵/۱/۲۶

تاریخ دریافت: ۱۴۰۳/۹/۲

چکیده: در این مقاله یک روش عددی جدید براساس شبکه عصبی و موجک‌های هان گسسته برای حل معادلات براتو ارائه می‌شود. ابتدا، موجک‌های هان گسسته و برخی ویژگی‌های مهم آن برای ساختن شبکه عصبی موجک‌های هان بیان می‌شود. شبکه عصبی بیان شده شامل سه لایه است: لایه ورودی، لایه خروجی و لایه پنهان. در این روش از موجک‌های هان و تابع سینوس هایپربولیک به ترتیب به عنوان توابع فعال‌ساز لایه پنهان و لایه خروجی استفاده می‌شود. سپس روش بهینه‌سازی تحلیلی برای به دست آوردن وزن‌های شبکه عصبی به گونه‌ای استفاده می‌شود که جواب تقریبی در مساله صدق کند. با استفاده از این روش، مساله اصلی به یک دستگاه از معادلات جبری تبدیل می‌شود، که این دستگاه را می‌توان با روش تکرار نیوتون حل کرد. در انتها با ارائه مثال‌هایی دقت و کارایی روش را مورد بررسی قرار می‌دهیم.

واژه‌های کلیدی: شبکه عصبی، موجک‌های هان، معادلات براتو، روش عددی، روش بهینه‌سازی تحلیلی

رده‌بندی ریاضی: 62M45, 65T60, 74G15.

۱ مقدمه

معادلات براتو به دلیل کاربردهای مهم آن‌ها در علوم و مهندسی توجه و علاقه فزاینده‌ای را به خود جلب کرده است. تاریخچه و اهمیت معادلات نوع براتو را می‌توان در مقاله‌های [۱] و [۲] یافت. فرایندهای شیمیایی و فیزیکی بسیاری با استفاده از معادله براتو مدل‌سازی شده است، برای مثال فرایند واکنش حرارتی در مواد احتراق، مواد غیرقابل تغییر شکل مانند جامدات در مدل احتراق سوخت، فرایند الکتروریسی برای تولید الیاف پلیمری بسیار ریز، نظریه راکتور شیمیایی، و فناوری نانو [۳-۶] را می‌توان بیان کرد. مساله لیوویل-براتو-گلفاند کلاسیک به جواب مثبت معادله زیر می‌پردازد [۲]:

^۱ نویسنده مسئول مقاله

$$\begin{cases} \Delta u + \lambda e^{u(\xi)} = 0, & \xi \in \Omega, \\ u = 0, & \xi \in \partial\Omega, \end{cases}$$

که $\lambda > 0$ و Ω یک دامنه کراندار در $\mathbb{R}^N$ است. این معادله از طریق مطالعه مسئله سهموی شبه‌خطی زیر به دست می‌آید [۲]:

$$\begin{cases} v_\xi = \Delta v + \lambda(1 - \varepsilon v)^{\tilde{m}} e^{v/(1+\varepsilon v)} = 0, & \xi \in \Omega, \\ v = 0, & \xi \in \partial\Omega, \end{cases}$$

که به عنوان مدل احتراق سوخت جامد شناخته می‌شود و به عنوان مدلی برای فرآیند واکنش حرارتی در یک ماده قابل احتراق و تغییر شکل ناپذیر با چگالی ثابت در طول دوره احتراق استخراج شده است. در این مقاله، تمرکز بر حالت $N = 1$ معادله لیوویل-براتو-گلفاند است، که در این حالت معادله به شکل ساده زیر تبدیل می‌شود:

$$u''(\xi) + \lambda e^{u(\xi)} = 0, \quad 0 < \xi \leq 1, \quad (1.1)$$

با شرایط اولیه

$$(i): \quad u(0) = u_0, \quad u'(0) = u_1,$$

یا شرایط مرزی

$$(ii): \quad u(0) = u_0, \quad u(1) = \hat{u}_1.$$

در این حالت، این مسئله به عنوان مسئله کلاسیک براتو [۷] شناخته می‌شود. راه‌حل دقیق مسئله کلاسیک براتو توسط [۷] ارائه شده است:

$$u(\xi) = -2 \operatorname{Ln} \left[\frac{\cosh(\frac{\theta}{4}(\xi - \frac{1}{4}))}{\cosh(\frac{\theta}{4})} \right],$$

که

$$\theta = \sqrt{2\lambda} \cosh\left(\frac{\theta}{4}\right).$$

وقتی $\lambda > \lambda_c$ و $\lambda = \lambda_c$ است، معادله براتو به ترتیب جواب ندارد، یک جواب دارد و یا دو جواب دارد. مقدار بحرانی به صورت زیر تعریف می‌شود:

$$1 = \frac{1}{4} \sqrt{2\lambda_c} \sinh\left(\frac{\theta}{4}\right),$$

که مقدار بحرانی λ_c به صورت $\lambda_c = 3/513830719$ مشخص می‌شود. همان‌طور که گفته شد، معادلات براتو کاربردهای بسیاری در مسائل فیزیکی مختلفی دارند، از جمله می‌توان این معادله را به عنوان یک مدل احتراق سوخت جامد در نظر گرفت که در این حالت u نشان‌دهنده دمای بدون بعد است، λ به عنوان پارامتر فرانک-کامنتسکی شناخته می‌شود، ξ متغیر مکان است و λe^u نشان‌دهنده گرمای احتراق است [۸].

تا کنون کارهای تحقیقاتی بسیاری برای حل عددی معادلات براتو ارائه شده است، از آن جمله می‌توان روش‌های زیر را نام برد: روش تکرار تغییرات اصلاح شده [۹]، روش بی‌اسپلین [۱۰]، روش‌های تکراری-انحرافی [۱۱]، روش موجک لژاندر [۱۲]، روش هم‌مکانی گاوس-ژاکوبی [۱۳]، روش سینک-گالرکین [۱۴]، روش تفاضل متناهی [۱۵]، روش موجک‌های تیلور [۱۶]، روش موجک‌های چبیشف [۱۷] و غیره.

امروزه روش‌های عددی مختلفی برای حل مسائل دنیای واقعی ارائه شده است [۱۸-۲۰]، که از جمله جدیدترین روش‌ها در این زمینه می‌توان روش‌های یادگیری ماشین را نام برد. در حقیقت این روش‌ها با یادگیری از داده‌های دریافتی، ساختار اساسی مساله را استخراج می‌کنند. کاربرد روش‌های یادگیری ماشین در حال حاضر هم در مسائل مهندسی و هم در مسائل ریاضی گسترش یافته است. به عنوان مثال، تجزیه و تحلیل کماتش در ستون‌های تیر [۲۱]،

دینامیک ارتعاش ماشین‌های الکتریکی دوار [۲۲]، مسئله برنامه‌نویسی [۲۳]، مسئله سیستماتیک معکوس [۲۴]، و نوسان ساز دافینگ [۲۵]. بنابراین از این روش‌ها برای حل مسائل مختلفی استفاده می‌شود از جمله این روش‌ها عبارتند از: راجا و همکاران [۲۶] دستگاه مرتبه کسری معادله باگلی توروک [۱] را با استفاده از یک روش شبکه عصبی و ژنتیکی پیشرو حل کردند. راجا و همکارانش در [۲۷] روشی مبتنی بر شبکه‌های عصبی مصنوعی و برنامه‌ریزی درجه دوم متوالی برای به‌دست آوردن حل عددی معادلات دیفرانسیل درجه دوم غیرخطی ریکاتی [۲] با مرتبه کسری، ارائه کردند. همچنین، آن‌ها [۲۸] یک شبکه عصبی کسری را معرفی کردند که از الگوریتم‌های نقطه درونی [۳] به عنوان بهینه‌ساز برای حل یک دستگاه معادلات باگلی-توروک استفاده کردند. رحیم‌خانی و حیدری [۲۹] روش شبکه عصبی مورگان-ویس [۴] انتقال‌یافته کسری را برای حل عددی معادلات دیفرانسیل کسری-فرکتال به‌کار بردند. همچنین رحیم‌خانی [۳۰] معادلات دیفرانسیل تصادفی غیرخطی با حرکت براونی کسری را با روش شبکه عصبی عمیق جنوکی [۵] کسری حل کرد. پرند و همکاران [۳۱] روش رگرسیون بردار پشتیبان حداقل مربعات را برای حل معادلات انتگرال ولترا به‌کار بردند. هادیان-رسانان و همکاران [۳۲] روش شبکه عصبی لژاندر کسری را برای حل نوع‌های مختلف معادلات لان-امدن [۶] به‌کار بردند. رحیم‌خانی و همکارانش در مقاله [۳۳] روش رگرسیون بردار پشتیبان حداقل مربعات موجک‌های برنولی را برای حل عددی دستگاه معادلات دیفرانسیل کسری ارائه دادند. پرند و همکارانش در مقاله [۳۴] روش ماشین بردار پشتیبان حداقل مربعات را برای حل مدل جمعیت ولترا مرتبه کسری به‌کار بردند. طاهری و همکارانش [۳۵] روش یادگیری ماشین و باقیمانده وزن‌دار را برای حل معادلات دیفرانسیل تاخیری مرتبه کسری به‌کار بردند. عبادی و همکارانش در مقاله [۳۶] روش شبکه عصبی بازگشتی یک لایه را برای حل عددی مسائل بهینه‌سازی غیرهموار با نابرابری‌های غیرخطی و محدودیت‌های برابری خطی به‌کار بردند.

در این مقاله در نظر داریم ترکیبی از روش شبکه عصبی و موجک‌های هان را برای حل معادلات براتو استفاده کنیم. مزیت اصلی تقریب یک تابع پیوسته توسط چندجمله‌ای‌های هان به صورت زیر خلاصه می‌شود [۳۷، ۳۸]:

- تعامد گسسته روی نقاط یکنواخت: چندجمله‌ای‌های هان دارای خاصیت تعامد گسسته هستند. این ویژگی امکان محاسبه ضرایب بسط را تنها از طریق جمع‌های متناهی و بدون نیاز به انتگرال‌گیری عددی فراهم می‌کند و در نتیجه، منبع مهمی از خطاهای عددی حذف می‌شود.
 - پایداری عددی مناسب: تعامد گسسته منجر به ساخت ماتریس‌های عملیاتی خوش‌وضع شده و پایداری عددی روش را به‌ویژه در حل معادلات غیرخطی بهبود می‌بخشد.
 - دقت بالا با درجات آزادی کمتر: تقریب مبتنی بر چندجمله‌ای‌های هان قادر است با استفاده از تعداد محدودی از توابع پایه، جواب‌هایی با دقت بالا ارائه دهد که این امر کارایی محاسباتی روش را افزایش می‌دهد.
 - همواری مناسب تقریب: برخلاف پایه‌هایی نظیر موجک‌های هار که تنها پیوستگی قطعه‌ای دارند، چندجمله‌ای‌های هان تقریب‌هایی هموار و مشتق‌پذیر تولید می‌کنند که برای حل معادلات دیفرانسیل نظیر معادله براتو بسیار مناسب هستند.
 - برتری نسبت به موجک‌های چبیشف و لژاندر: در مقایسه با موجک‌های مبتنی بر چندجمله‌ای‌های متعامد پیوسته مانند چبیشف و لژاندر، روش مبتنی بر چندجمله‌ای‌های هان ضمن حفظ دقت طیفی، سازگاری طبیعی‌تری با فرمول‌بندی‌های گسسته داشته و منجر به پایداری عددی بهتر، کاهش خطا و ساده‌تر شدن ساختار ماتریس‌های عملیاتی می‌شود.
- لازم به ذکر است که حل‌های تقریبی به‌دست‌آمده با استفاده از شبکه‌های عصبی مصنوعی دارای مزایای متعددی هستند [۳۹]. برخی از ویژگی‌های اصلی چارچوب مبتنی بر شبکه‌های عصبی مصنوعی به شرح زیر فهرست می‌شوند [۴۰]:
- پیوستگی و مشتق‌پذیری جواب: جواب تقریبی حاصل از شبکه عصبی به‌طور ذاتی پیوسته و مشتق‌پذیر است، که این امر آن را برای حل معادلات دیفرانسیل، مسائل کنترلی و مسائل شامل عملگرهای مشتق مناسب می‌سازد.

- قابلیت تعمیم‌پذیری بالا: تقریب شبکه عصبی این امکان را فراهم می‌کند که جواب مسئله در هر نقطه دلخواه از دامنه، حتی خارج از مجموعه نقاط آموزشی، ارزیابی شود. این ویژگی باعث می‌شود روش پیشنهادی نسبت به روش‌های مبتنی بر شبکه‌بندی، انعطاف‌پذیری بیشتری در بازسازی پاسخ داشته باشد.
- قابلیت افزایش دقت: با افزایش تعداد نورون‌ها یا توابع پایه، دقت تقریب به صورت خودکار قابل بهبود است و می‌توان بدون تغییر در ساختار کلی روش، به جواب‌های با دقت بالاتر دست یافت.
- ارتباط با روش‌های کلاسیک عددی: در حالتی که تابع فعال‌ساز به صورت $AF(\xi) = \xi$ انتخاب شود، روش پیشنهادی به تقریب‌های کلاسیک عددی تبدیل می‌شود. از این رو، روش‌های کلاسیک را می‌توان به عنوان حالت‌های خاصی از چارچوب شبکه عصبی هان در نظر گرفت و روش پیشنهادی یک رویکرد تعمیم‌یافته است که آن‌ها را نیز در بر می‌گیرد.
- سهولت پیاده‌سازی: ساختار ساده شبکه عصبی و عدم نیاز به شبکه‌بندی پیچیده یا الگوریتم‌های تکراری سنگین، پیاده‌سازی روش را آسان کرده و آن را از نظر محاسباتی کارآمد می‌سازد.
- گسترش‌پذیری چارچوب: چارچوب ارائه‌شده به طور طبیعی قابل توسعه به حل معادلات انتگرالی، دیفرانسیلی، تصادفی، تاخیری، مرتبه‌کسری و همچنین مسائل معکوس است، بدون آنکه نیاز به تغییرات اساسی در روش وجود داشته باشد.

به طور خلاصه، در ادبیات موجود، روش‌های عددی مبتنی بر یادگیری ماشین برای حل معادلات دیفرانسیل غیرخطی و کسری عمدتاً در قالب چند چارچوب اصلی توسعه یافته‌اند که از جمله مهم‌ترین آن‌ها می‌توان به شبکه‌های عصبی آگاه از فیزیک، مدل‌های مبتنی بر موجک، و روش‌های ترکیبی طیفی-یادگیری اشاره کرد. با وجود موفقیت‌های قابل توجه این روش‌ها، بسیاری از آن‌ها با محدودیت‌هایی از جمله همگرایی کند در نواحی غیرخطی، حساسیت بالا نسبت به انتخاب نقاط آموزشی، و عدم بهره‌برداری مؤثر از توابع پایه متناسب با ساختار مسئله مواجه هستند. به‌ویژه در بسیاری از روش‌های مبتنی بر شبکه‌های عصبی آگاه از فیزیک، استفاده از توابع فعال‌سازی عمومی موجب می‌شود رفتارهای موضعی و چندمقیاسی جواب به‌خوبی بازنمایی نشود. در این راستا، در پژوهش حاضر یک چارچوب عصبی ترکیبی مبتنی بر موجک‌های گسسته هان ارائه می‌شود که با ترکیب توان تقریب بالای پایه‌های موجکی متعامد و انعطاف‌پذیری شبکه‌های عصبی، این کاستی‌ها را برطرف می‌کند. روش پیشنهادی با بهره‌گیری از ساختار موجکی متناسب با مسئله، دقت بالاتر، پایداری عددی بهتر و نرخ همگرایی سریع‌تری را نسبت به روش‌های عددی مبتنی بر یادگیری ماشین موجود، به‌ویژه برای معادلات غیرخطی از نوع براتو، فراهم می‌آورد.

ادامه این مقاله به صورت زیر سازمان‌دهی می‌شود: در بخش ۲، ساختار شبکه عصبی موجک‌های هان گسسته بیان می‌شود. در بخش ۳، روش عددی مبتنی بر شبکه عصبی موجک‌های هان گسسته ارائه می‌شود. در بخش ۴، نتایج عددی روی چهار مثال با شرایط اولیه و مرزی گزارش می‌شوند. در بخش ۵، یک نتیجه‌گیری کلی از روش بیان می‌شود.

۲ ساختار شبکه عصبی موجک‌های هان گسسته

شبکه عصبی موجک‌های هان گسسته برای حل معادلات براتو در نظر گرفته می‌شود. شکل ۱ ساختار شبکه عصبی موجک‌های هان گسسته شامل سه لایه را نشان می‌دهد.

- اولین لایه، لایه ورودی شبکه است که دارای یک گره ξ است که این گره دارای q تعداد داده

$$\{\xi_1, \xi_2, \dots, \xi_q\}$$

است.

- لایه دوم، لایه پنهان شبکه است که در این لایه یک کلاس از توابع متعامد به عنوان تابع فعال‌ساز انتخاب می‌شود. در حقیقت، موجک‌های هان گسسته به عنوان توابع فعال‌ساز اعمال می‌شوند.

• لایه سوم، لایه خروجی شبکه است که ورودی این لایه ترکیبی خطی از موجک‌های هان گسسته از لایه دوم است و خروجی آن حاصل اعمال تابع فعال‌ساز $AF(\cdot)$ بر روی ورودی آن است. در این لایه تابع سینوس هایپربولیک به عنوان تابع فعال‌ساز در نظر گرفته می‌شود.

بنابراین، خروجی شبکه عصبی موجک‌های هان گسسته با داده ورودی ξ و پارامتر c به صورت زیر است:

$$N(\xi, c) = AF(\Theta) = \sinh(\Theta). \quad (1.2)$$

در اینجا Θ یک ترکیب خطی از موجک‌های هان است که به صورت زیر نوشته می‌شود:

$$\Theta = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k) = c^T \Psi^*(\xi, a, b, M, L, k), \quad (2.2)$$

که

$$c_{n,m}$$

وزن‌های مجهول شبکه هستند که در فرآیندهای آموزشی برای کاهش تابع خطا تعیین می‌شوند. بردار به صورت زیر مشخص می‌شود:

$$c = [c_{1,0}, c_{1,1}, \dots, c_{1,M-1}, c_{2,0}, \dots, c_{2,M-1}, \dots, c_{2^{k-1}, M-1}]^T.$$

موجک‌های هان گسسته روی بازه $[0, L]$ به صورت زیر تعریف می‌شوند [۴۱]:

$$\psi_{nm}^*(\xi, a, b, M, L, k) = \begin{cases} 2^{\frac{k-1}{r}} \varphi_m^*\left(\frac{2^{k-1}}{L} \xi - n + 1, a, b, M, 1\right), & \frac{n-1}{2^{k-1}} L \leq \xi < \frac{n}{2^{k-1}} L, \\ 0, & \text{Otherwise} \end{cases} \quad (3.2)$$

که در رابطه (۳.۲)، ضریب φ_m با استفاده از خواص چندجمله‌ای‌های هان [۴۱] به صورت زیر بدست می‌آید:

$$\varphi_m = \sqrt{\frac{(2m + a + b + 1)(a + 1)_m (-M + 1)_m (M - 1)!}{(-1)^m (m + a + b + 1)_M (b + 1)_m j!}},$$

و همچنین داریم:

$$m = 0, 1, \dots, M - 1, \quad n = 1, 2, \dots, 2^{k-1}, \quad k = 1, 2, \dots,$$

علاوه بر این، چندجمله‌ای‌های هان انتقال‌یافته از مرتبه m روی بازه $[0, L]$ را با $h_m^*(\xi, a, b, M, L)$ نشان می‌دهیم که به صورت زیر تعریف می‌شوند [۴۱]:

$$h_m^*(\xi, a, b, M, L) = \sum_{j=0}^m \sum_{r=0}^j (-1)^j \frac{(-m)_j (m + a + b + 1)_j}{(a + 1)_j (-M)_j j!} \left(\frac{M}{L}\right)^r S_j^{(r)} \xi^r, \quad (4.2)$$

که

$$m = 0, 1, \dots, M - 1.$$

و S_j^r اعداد استرلینگ [۱] نوع اول هستند. در جدول ۱، پارامترهای به کاررفته در شبکه عصبی موجک هان به تفصیل تشریح شده‌اند.

جدول ۱: تعریف و نقش پارامترها در تعریف شبکه عصبی موجک‌های هان

پارامتر	تعریف	نقش در روش عددی	دامنه انتخاب شده در مثال‌ها عددی
a	پارامتر وزن چپ	کنترل توزیع وزن در نزدیکی $\xi = 0$	در تمامی مثال‌ها $a = 1$
b	پارامتر وزن راست	کنترل توزیع وزن در نزدیکی $\xi = L$	در تمامی مثال‌ها $b = 2$
M	تعداد توابع پایه در هر زیربازه	کنترل مرتبه و دقت تقریب محلی	مقادیر صحیح کوچک مانند $M = 2, 3, 4, \dots, 13$
L	طول دامنه محاسباتی	نرمال‌سازی دامنه و تعریف چندجمله‌ای‌های انتقال یافته	در تمامی مثال‌ها $L = 1$
k	تعداد زیربازه‌ها	کنترل تفکیک مکانی و تقریب موضعی	مقادیر صحیح $k = 0, 1$
$AF(\cdot)$	تابع فعال‌ساز	ایجاد غیرخطی بودن در ساختار شبکه و افزایش توان تقریب برای حل معادلات غیرخطی	در تمامی مثال‌ها $AF(\cdot) = \sinh(\cdot)$

تشریح معماری شبکه عصبی

تشریح معماری شبکه عصبی شبکه عصبی موجک‌های هان گسسته به صورت یک شبکه پیشرو سه لایه طراحی شده است که شامل لایه ورودی، لایه پنهان و لایه خروجی می‌باشد. در این ساختار، متغیر مستقل به عنوان ورودی شبکه در نظر گرفته می‌شود و هدف شبکه، تقریب تابع مجهول $u(\xi)$ است.

لایه ورودی شامل یک نورون است که مقدار ξ را بدون اعمال هیچ گونه تبدیلی به لایه پنهان منتقل می‌کند. لایه پنهان شامل $2^{k-1}M$ نورون است که هر نورون متناظر با یک موجک هان گسسته می‌باشد. در این لایه، موجک‌های هان گسسته به عنوان توابع فعال‌ساز انتخاب شده‌اند و خروجی لایه پنهان به صورت بردار

$$\Psi^*(\xi, a, b, M, L, k) = [\psi_{1,0}^*(\xi, a, b, M, L, k), \psi_{1,1}^*(\xi, a, b, M, L, k), \dots, \psi_{1,M-1}^*(\xi, a, b, M, L, k), \dots, \psi_{2^{k-1},M-1}^*(\xi, a, b, M, L, k)]^T,$$

تعریف می‌شود.

خروجی لایه پنهان از طریق یک ترکیب خطی با ضرایب قابل آموزش

$$c = [c_{1,0}, c_{1,1}, \dots, c_{1,M-1}, c_{2,0}, \dots, c_{2,M-1}, \dots, c_{2^{k-1},M-1}]^T,$$

به لایه خروجی منتقل می‌شود. بنابراین، ورودی لایه خروجی به صورت

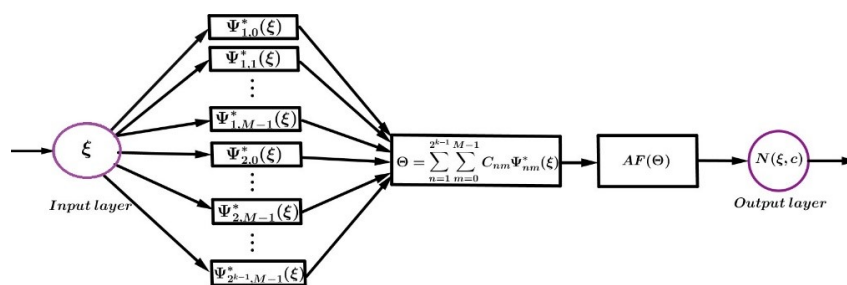
$$\Theta = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k),$$

خواهد بود. در لایه خروجی، تابع فعال‌سازی سینوس هایپربولیک اعمال می‌شود و خروجی نهایی شبکه، به شکل

$$N(\xi, c) = AF\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) = \sinh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) = \sinh(\Theta),$$

به دست می‌آید که منطبق با روابط (۱.۲) و (۲.۲) می‌باشد. در شکل ۱ برای سادگی در ترسیم قرار دادیم

$$\psi_{nm}^*(\xi, a, b, M, L, k) = \psi_{nm}^*(\xi)$$



شکل ۱: ساختار شبکه عصبی موجک‌های هان

۳ روش عددی

در اینجا روش شبکه عصبی موجک‌های هان برای حل عددی معادلات براتو (۱.۱) ارائه می‌شود. بدین منظور، ابتدا تابع $u(\xi)$ با شبکه عصبی موجک‌های هان به صورت زیر تقریب زده می‌شود. برای معادله براتو با شرط اولیه، داریم:

$$(i) : \quad u(\xi) \simeq \tilde{u}(\xi) = u_0 + u_1 \xi + \xi^\gamma N(\xi, c), \quad (1.3)$$

که

$$\tilde{u}(\xi) = u_0 + u_1 \xi + \xi^\gamma \sinh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right),$$

همچنین داریم:

$$\begin{aligned} \tilde{u}'(\xi) &= u_1 + \gamma \xi \sinh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) \\ &+ \xi^\gamma \left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k)\right) \cosh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right), \end{aligned}$$

و

$$\begin{aligned} \tilde{u}''(\xi) &= [\gamma + \xi^\gamma \left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k)\right)] \sinh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) \\ &+ [(\gamma \xi \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k) + \xi^\gamma \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*''}(\xi, a, b, M, L, k))] \\ &\cosh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right). \end{aligned}$$

برای معادله براتو با مقادیر مرزی قرار داده می‌شود:

$$(ii) : \quad u(\xi) \simeq \tilde{u}(\xi) = u_0 + (\hat{u}_1 - u_0)\xi + \xi(\xi - 1)N(\xi, c), \quad (2.3)$$

که

$$\tilde{u}(\xi) = u_0 + (\hat{u}_1 - u_0)\xi + \xi(\xi - 1) \sinh\left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right),$$

همچنین داریم:

$$\begin{aligned}\tilde{u}'(\xi) &= (\hat{u}_1 - u_0) + (\mathfrak{z}\xi - 1) \sinh\left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) \\ &+ (\xi^{\mathfrak{z}} - \xi) \left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k)\right) \cosh\left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right),\end{aligned}$$

و

$$\begin{aligned}\tilde{u}''(\xi) &= [\mathfrak{z} + (\xi^{\mathfrak{z}} - \xi) \left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k)\right)^{\mathfrak{z}}] \sinh\left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right) \\ &+ [((\mathfrak{z}\xi - 2) \sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi, a, b, M, L, k) + (\xi^{\mathfrak{z}} - \xi) \sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*''}(\xi, a, b, M, L, k))] \\ &\cosh\left(\sum_{n=1}^{\mathfrak{z}^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi, a, b, M, L, k)\right).\end{aligned}$$

با جایگذاری تقریب‌های فوق در معادله (۱.۱) و با استفاده از تعریف تابع باقیمانده $Res(\xi, c)$ رابطه زیر به دست می‌آید:

$$Res(\xi, c) = \tilde{u}''(\xi) + \lambda e^{\tilde{u}(\xi)}. \quad (3.3)$$

برای به دست آوردن تقریبی از تابع مجهول، تابع باقیمانده مذکور باید به صفر میل کنند. برای این منظور مجموعه داده‌های آموزشی $\{\xi_1, \xi_2, \dots, \xi_{n^*}\}$ در نظر گرفته می‌شود. اگر مربع تابع باقیمانده به ازای تمام نقاط آموزشی صفر شوند، آن‌گاه تابع خطای باقیمانده به صفر میل می‌کند. در نهایت مساله بهینه‌سازی زیر به دست می‌آید که جواب آن، وزن‌های بهینه شبکه عصبی است. این مساله بهینه‌سازی به صورت زیر است:

$$C^* = \min_C \frac{1}{\mathfrak{z}} \sum_{i=1}^{n^*} Res^{\mathfrak{z}}(\xi_i, c). \quad (4.3)$$

حال بردار C به گونه‌ای به دست می‌آید که C^* مینیم شود. شرایط لازم برای مینیم کردن C^* عبارت است از:

$$\frac{\partial}{\partial c_{i,j}} C^* = 0, \quad i = 1, 2, \dots, \mathfrak{z}^{k-1}, \quad j = 0, 1, \dots, M-1. \quad (5.3)$$

این مشتق به صورت زیر محاسبه می‌شود:

$$\begin{aligned}
 \frac{\partial}{\partial c_{i,j}} C^* &= \sum_{i=1}^{n^*} \left([\Psi + \xi_i^\Psi \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi_i, a, b, M, L, k) \right)^\Psi] \right. \\
 &\sinh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) \\
 &+ [(\Psi \xi_i \sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi_i, a, b, M, L, k) \\
 &+ \xi_i^\Psi \sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*''}(\xi_i, a, b, M, L, k))] \\
 &\cosh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) \\
 &+ \lambda e^{u_0 + u_1 \xi_i + \xi_i^\Psi \sinh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right)} \\
 &\times \left([\Psi \xi_i^\Psi \psi_{ij}^{*'}(\xi_i, a, b, M, L, k) \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi_i, a, b, M, L, k) \right)] \right. \\
 &\sinh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) \\
 &+ [\Psi + \xi_i^\Psi \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi_i, a, b, M, L, k) \right)^\Psi] \psi_{ij}^*(\xi_i, a, b, M, L, k) \\
 &\cosh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) \\
 &+ [(\Psi \xi_i \psi_{ij}^{*'}(\xi_i, a, b, M, L, k) + \xi_i^\Psi \psi_{ij}^{*''}(\xi_i, a, b, M, L, k))] \\
 &\cosh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) \\
 &+ [(\Psi \xi_i \sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*'}(\xi_i, a, b, M, L, k) \\
 &+ \xi_i^\Psi \sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^{*''}(\xi_i, a, b, M, L, k))] \psi_{ij}^*(\xi_i, a, b, M, L, k) \\
 &\sinh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) + \lambda \xi_i^\Psi \psi_{ij}^*(\xi_i, a, b, M, L, k) \\
 &\cosh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right) e^{u_0 + u_1 \xi_i + \xi_i^\Psi \sinh \left(\sum_{n=1}^{\Psi^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{nm}^*(\xi_i, a, b, M, L, k) \right)} = 0.
 \end{aligned}$$

یک دستگاه از معادلات با $\Psi^{k-1} M$ معادله و $\Psi^{k-1} M$ مجهول نتیجه گرفته می‌شود. این دستگاه از معادلات با روش تکرار نیوتون حل می‌شود. دستگاه معادلات برای مساله (ii) نیز به روش قبل بدست می‌آید.

ملاحظه ۱.۳. شایان ذکر است که روش‌های متعددی برای انتخاب مجموعه داده‌های آموزشی وجود دارد. از جمله نقاطی که تاکنون توسط پژوهشگران به عنوان نقاط آموزشی مورد استفاده قرار گرفته‌اند، می‌توان به نقاط متساوی‌فاصله روی یک بازه مشخص و ریشه‌های چندجمله‌ای‌های لژاندر اشاره کرد. در این مطالعه، ریشه‌های چندجمله‌ای‌های لژاندر به عنوان نقاط آموزشی انتخاب شده‌اند؛ زیرا این نقاط در تقریب‌های طیفی و روش‌های پایه‌ای مرتبه بالا از خواص مطلوبی برخوردار بوده و موجب کمینه‌سازی خطای تقریب می‌شوند. به بیان دیگر، انتخاب این نقاط امکان کاهش خطای ناشی از تقریب با ترکیب خطی توابع پایه را فراهم کرده و در نتیجه، تقریب با دقت بالاتری حاصل می‌شود. در این مقاله، تعداد نقاط آموزشی برابر با $n^* = \Psi M$ در نظر گرفته شده است.

ملاحظه ۲.۳. برای حل مساله بهینه‌سازی ایجاد شده در معادله (۵.۳) روش‌های عددی مختلفی وجود دارد [۳۰].

در این مطالعه، روش بهینه‌سازی تحلیلی به کار گرفته شده است، به طوری که نقطه بهینه با محاسبه مشتق تابع هدف نسبت به متغیرها و برابر صفر قرار دادن آن‌ها تعیین می‌شود.

ملاحظه ۳.۳. برای حل دستگاه معادلات ناشی از تابع باقیمانده، از دستور *FindRoot* در نرم‌افزار ممتیکا استفاده شده است. برای تعیین معیار توقف، این دستور از شرط پیش‌فرض ممتیکا بهره گرفته شده است؛ به طور پیش‌فرض، *FindRoot* تا زمانی ادامه می‌یابد که مقدار تابع در ریشه تقریبی نزدیک صفر شود یا تغییرات مقادیر متغیرها بین دو تکرار متوالی کوچک گردد. این شرط پیش‌فرض به طور خودکار دقت و پایداری حل عددی را کنترل می‌کند.

الگوریتم: روش شبکه عصبی موجک‌های هان برای حل عددی معادلات براتو

ورودی: معادله براتو (۱.۱)، شرط اولیه یا شرط مرزی، پارامترهای موجک‌های هان a, b, M, L, k ، پارامترهای شبکه عصبی $AF(\cdot)$.

گام دوم: ساخت شبکه عصبی: تقریب تابع $u(\xi)$ با ترکیب موجک‌های هان و شبکه عصبی

گام سوم: اعمال شرایط اولیه و مرزی: اصلاح ضرایب شبکه به طوری که تقریب با شرایط مسئله سازگار باشد

گام چهارم: تشکیل تابع باقیمانده: تعریف تابع باقیمانده $Res(\xi, c)$ با جایگذاری تقریب در معادله اصلی

گام پنجم: تعیین نقاط آموزشی: به کاربردن ریشه‌های چندجمله‌ای لژاندر درجه $n^* = 2M$ به عنوان نقاط آموزشی

گام ششم: تشکیل مساله بهینه‌سازی C^*

گام هفتم: استفاده از روش بهینه‌سازی تحلیلی برای حل ضرایب مجهول $c_{i,j}; i = 1, 2, \dots, 2^{k-1}, j = 0, 1, \dots, M-1$

گام هشتم: حل دستگاه بدست آمده از گام هفتم با استفاده از روش تکرار نیوتون: در این روش مقدار اولیه را صفر در نظر می‌گیریم و برای شرط توقف از شرط پیش‌فرض ممتیکا استفاده می‌کنیم.

خروجی: به دست آوردن تقریب نهایی تابع مجهول: استفاده از وزن‌های بهینه شبکه عصبی برای ساخت تقریب نهایی $u(\xi)$

۴ مثال های عددی

در این بخش برای نشان دادن کارایی و کاربرد روش پیشنهاد شده در بخش قبل، روش را روی چهار مثال مختلف پیاده‌سازی می‌شود و نتایج حاصل نیز گزارش می‌شود.

مثال ۱.۴. معادله براتو زیر در نظر می‌گیریم:

$$\begin{aligned} u''(\xi) - 2e^{u(\xi)} &= 0, \\ u(0) = u'(0) &= 0, \quad 0 \leq \xi < 1, \end{aligned}$$

جواب دقیق برای این معادله برابر است با

$$u(\xi) = -2Ln[\cos \xi].$$

در جدول ۲، خطای مطلق نتایج عددی روش حاضر برای روش‌های الگوریتم ژنتیک (GA)، روش مجموعه فعال (ASM)، ترکیب الگوریتم ژنتیک و مجموعه فعال (GA-ASM) و روش موجک تیلور (TWM) که در مقاله [۱۴] گزارش شده‌اند، مقایسه می‌شود. نتایج نشان می‌دهد که روش پیشنهادی در تمامی نقاط محاسباتی دارای خطای کمتری نسبت به روش‌های ذکر شده است، به ویژه در نواحی داخلی دامنه که دقت روش حاضر به طور محسوسی افزایش یافته است. در جدول ۳، نتایج عددی روش حاضر برای روش‌های موجک‌های تیلور و موجک‌های لژاندر مقایسه (LWM) می‌شوند. این جدول برتری روش حاضر را نسبت به روش‌های دیگر نشان می‌دهد. همچنین گراف نتایج عددی برای در شکل ۲ توضیح داده می‌شود. نتایج عددی ارائه شده در شکل ۲ نشان می‌دهد که با افزایش تعداد توابع پایه، جواب عددی به تدریج به مقدار دقیق نزدیک‌تر می‌شود که این امر بیانگر همگرایی مناسب و پایداری عددی روش پیشنهادی است.

جدول ۲: مقایسه خطای مطلق روش حاضر با سایر روش‌ها (مثال ۱.۴)

روش حاضر	TWM	GA - ASM	ASM	GA	ξ
$۱/۲۹ \times ۱۰^{-۱۴}$	$۷/۸۰ \times ۱۰^{-۸}$	$۵/۷۱ \times ۱۰^{-۸}$	$۱/۰۵ \times ۱۰^{-۸}$	$۵/۸۵ \times ۱۰^{-۴}$	۰/۱
$۲/۰۵ \times ۱۰^{-۱۴}$	$۶/۶۲ \times ۱۰^{-۸}$	$۸/۴۳ \times ۱۰^{-۸}$	$۵/۲۲ \times ۱۰^{-۸}$	$۶/۱۹ \times ۱۰^{-۴}$	۰/۲
$۳/۰۷ \times ۱۰^{-۱۴}$	$۵/۳۵ \times ۱۰^{-۸}$	$۷/۱۸ \times ۱۰^{-۸}$	$۲/۱۶ \times ۱۰^{-۷}$	$۷/۰۸ \times ۱۰^{-۴}$	۰/۳
$۱/۰۲ \times ۱۰^{-۱۴}$	$۱/۷۷ \times ۱۰^{-۸}$	$۴/۳۲ \times ۱۰^{-۸}$	$۹/۰۹ \times ۱۰^{-۸}$	$۸/۰۳ \times ۱۰^{-۴}$	۰/۴
$۵/۹۵ \times ۱۰^{-۱۴}$	$۲/۷۹ \times ۱۰^{-۸}$	$۱/۳۰ \times ۱۰^{-۷}$	$۱/۸۰ \times ۱۰^{-۷}$	$۸/۷۳ \times ۱۰^{-۴}$	۰/۵
$۲/۰۶ \times ۱۰^{-۱۴}$	$۵/۲۹ \times ۱۰^{-۸}$	$۲/۵۹ \times ۱۰^{-۷}$	$۸/۹۷ \times ۱۰^{-۸}$	$۹/۴۲ \times ۱۰^{-۴}$	۰/۶
$۳/۱۳ \times ۱۰^{-۱۴}$	$۴/۸۹ \times ۱۰^{-۸}$	$۵/۸۹ \times ۱۰^{-۸}$	$۱/۶۸ \times ۱۰^{-۷}$	$۱/۰۵ \times ۱۰^{-۳}$	۰/۷
$۱/۵۰ \times ۱۰^{-۱۰}$	$۳/۸۰ \times ۱۰^{-۸}$	$۳/۱۳ \times ۱۰^{-۷}$	$۳/۳۱ \times ۱۰^{-۸}$	$۱/۲۱ \times ۱۰^{-۳}$	۰/۸
$۳/۳۰ \times ۱۰^{-۱۰}$	$۵/۷۱ \times ۱۰^{-۸}$	$۱/۴۳ \times ۱۰^{-۷}$	$۵/۵۳ \times ۱۰^{-۸}$	$۱/۳۹ \times ۱۰^{-۳}$	۰/۹

جدول ۳: مقایسه خطای مطلق روش حاضر با روش‌های دیگر (مثال ۱.۴)

روش حاضر با ۵ لایه ($K=1, M=7$)	روش حاضر با ۳ لایه ($K=1, M=7$)	روش حاضر با ۲ لایه ($K=1, M=7$)	TWM ($K=1, M=7$)	LWM ($K=1, M=7$)	ξ
$۲/۳۲۲۰۱ \times ۱۰^{-۷}$	$۳/۰۲۸۳۱ \times ۱۰^{-۷}$	$۴/۴۰۲۵۵ \times ۱۰^{-۷}$	$۲/۶۹۶۱۱ \times ۱۰^{-۵}$	$۹/۰۲۳۷۸ \times ۱۰^{-۸}$	۰/۱
$۳/۰۵۲۳۶ \times ۱۰^{-۷}$	$۳/۹۰۵۸۸ \times ۱۰^{-۷}$	$۵/۵۳۴۵۹ \times ۱۰^{-۷}$	$۲/۳۸۹۶۸ \times ۱۰^{-۵}$	$۱/۵۰۶۴۱ \times ۱۰^{-۷}$	۰/۲
$۲/۸۴۱۶۳ \times ۱۰^{-۷}$	$۳/۷۱۱۲۷ \times ۱۰^{-۷}$	$۵/۴۱۳۸۵ \times ۱۰^{-۷}$	$۱/۰۱۳۲۹ \times ۱۰^{-۵}$	$۶/۱۴۰۱۸ \times ۱۰^{-۷}$	۰/۳
$۱/۲۴۰۲۶ \times ۱۰^{-۷}$	$۱/۷۸۹۷۱ \times ۱۰^{-۷}$	$۲/۹۱۲۹۸ \times ۱۰^{-۷}$	$۲/۱۲۴۰۸ \times ۱۰^{-۵}$	$۸/۸۰۰۶۴ \times ۱۰^{-۶}$	۰/۴
$۱/۰۰۴۷۵ \times ۱۰^{-۶}$	$۱/۳۰۹۴۴ \times ۱۰^{-۶}$	$۱/۸۹۸۵۹ \times ۱۰^{-۶}$	$۱/۱۵۳۱۶ \times ۱۰^{-۵}$	$۵/۶۷۱۲۴ \times ۱۰^{-۵}$	۰/۵
$۷/۱۵۴۷۶ \times ۱۰^{-۷}$	$۸/۹۷۵۵۹ \times ۱۰^{-۷}$	$۱/۲۳۱۸۲ \times ۱۰^{-۶}$	$۱/۸۵۱۳۶ \times ۱۰^{-۵}$	$۲/۵۵۷۷۶ \times ۱۰^{-۴}$	۰/۶
$۱/۵۷۹۴۱ \times ۱۰^{-۹}$	$۱/۵۴۶۴۴ \times ۱۰^{-۸}$	$۶/۱۲۰۶۸ \times ۱۰^{-۸}$	$۱/۱۵۴۷۳ \times ۱۰^{-۵}$	$۹/۲۴۶۰۷ \times ۱۰^{-۴}$	۰/۷
$۶/۱۸۵۶ \times ۱۰^{-۷}$	$۸/۱۱۷۳۸ \times ۱۰^{-۷}$	$۱/۲۰۱۱۲ \times ۱۰^{-۶}$	$۲/۲۶۴۹۴ \times ۱۰^{-۵}$	$۲/۸۶۱۹۸ \times ۱۰^{-۳}$	۰/۸
$۱/۰۸۸۳۴ \times ۱۰^{-۶}$	$۱/۴۰۴۷۸ \times ۱۰^{-۶}$	$۲/۰۰۳۶۵ \times ۱۰^{-۶}$	$۱/۱۲۹۳۳ \times ۱۰^{-۵}$	$۷/۹۱۲۴۸ \times ۱۰^{-۳}$	۰/۹
۳۸۹۷	۲/۲۳۵	۱/۰۹۴	۰/۹۳۶	۱/۴۶۵	CPU

مثال ۲.۴. معادله براتو زیر را در نظر می‌گیریم:

$$u''(\xi) + \lambda e^{u(\xi)} = 0, \\ u(0) = u(1) = 0, \quad 0 \leq \xi < 1, \quad \lambda = 0/0001, 0/0001, 1, 2, 3/5.$$

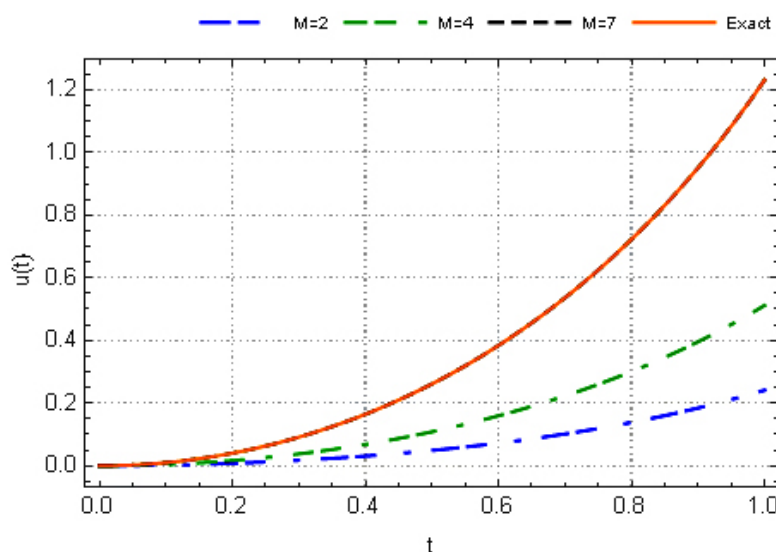
جواب دقیق در مقاله [۱۴] مشخص شده است. نتایج عددی روش حاضر ($k=1, M=10$) و روش GA-ASM برای $\lambda=1, 2, 3/5$ همراه با جواب دقیق در جدول ۴ گزارش شده است. همچنین در شکل ۳، نتایج عددی روش حاضر ($k=1, M=5$) با روش TWM برای $\lambda=1, 2, 3/5$ مقایسه می‌شوند. مقایسه خطای مطلق روش بیان شده ($k=1, M=5$) با روش TWM برای $\lambda=0/0001, 0/0001$ در شکل ۴ نشان داده شده است. در شکل‌های ۳ و ۴ نشان داده شده است که نتایج روش عددی هم‌راستا با روش‌های عددی دیگر است و این امر قابل اعتماد بودن روش پیشنهادی را تأیید می‌کند.

جدول ۴: نتایج عددی برای مقادیر متفاوت λ (مثال ۲.۴)

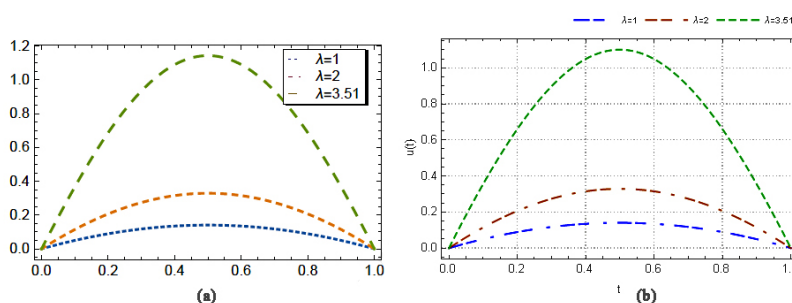
روش حاضر			GA-ASM			جواب دقیق			ξ
$\lambda=3/5$	$\lambda=2$	$\lambda=1$	$\lambda=3/5$	$\lambda=2$	$\lambda=1$	$\lambda=3/5$	$\lambda=2$	$\lambda=1$	
۰/۳۶۴۳۳۵۶	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۷	۰/۳۶۴۳۳۶۱	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۷	۰/۳۶۴۳۳۵۸	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۷	۰/۱
۰/۶۷۷۸۶۹۳	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۶۷۷۸۶۹۶	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۶۷۷۸۶۹۷	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۲
۰/۹۲۲۲۱۳۶	۰/۲۷۳۸۷۹۳	۰/۱۱۷۶۰۹۰	۰/۹۲۲۲۱۳۹	۰/۲۷۳۸۷۹۳	۰/۱۱۷۶۰۹۱	۰/۹۲۲۲۱۴۲	۰/۲۷۳۸۷۹۳	۰/۱۱۷۶۰۹۰	۰/۳
۱/۰۷۸۶۳۳۸	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۱/۰۷۸۶۳۳۸	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۱/۰۷۸۶۳۴۲	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۰/۴
۱/۱۳۲۶۱۷۱	۰/۳۲۸۹۵۲۴	۰/۱۴۰۵۳۹۲	۱/۱۳۲۶۱۷۶	۰/۳۲۸۹۵۲۴	۰/۱۴۰۵۳۹۲	۱/۱۳۲۶۱۸۰	۰/۳۲۸۹۵۲۴	۰/۱۴۰۵۳۹۲	۰/۵
۱/۰۷۸۶۳۳۸	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۱/۰۷۸۶۳۳۵	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۱/۰۷۸۶۳۴۲	۰/۳۱۵۰۸۹۳	۰/۱۳۴۷۹۰۲	۰/۶
۰/۹۲۲۲۱۳۶	۰/۲۷۳۸۷۹۳	۰/۱۱۷۶۰۹۰	۰/۹۲۲۲۱۳۸	۰/۲۷۳۸۷۹۳	۰/۱۱۶۰۹۱۰	۰/۹۲۲۲۱۴۲	۰/۲۷۳۸۷۹۳	۰/۱۱۷۶۰۹۰	۰/۷
۰/۶۷۷۸۶۹۳	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۶۷۷۸۶۹۱	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۶۷۷۸۶۹۷	۰/۲۰۶۴۱۹۱	۰/۰۸۹۱۸۹۹	۰/۸
۰/۳۶۴۳۳۵۶	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۷	۰/۳۶۴۳۳۵۵	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۸	۰/۳۶۴۳۳۵۸	۰/۱۱۴۴۱۰۷	۰/۰۴۹۸۴۶۷	۰/۹
۵/۴۸۵	۳/۲۸۱	۲/۱۴۰	-	-	-	-	-	-	CPU

مثال ۳.۴. معادله براتو زیر را در نظر می‌گیریم:

$$u''(\xi) - \pi^2 e^{u(\xi)} = 0, \\ u(0) = u(1) = 0, \quad 0 \leq \xi < 1,$$



شکل ۲: نتایج عددی و جواب دقیق برای $k = 1$ و مقادیر متفاوت M (مثال ۱.۴)



شکل ۳: مقایسه نتایج عددی با روش TWM (مثال ۲.۴)

جواب دقیق برای این معادله برابر است با

$$u(\xi) = -2Ln[1 - \cos((\circ, \xi + \xi)\pi)].$$

در جدول ۵، خطای مطلق روش حاضر برای $M = 11$ ، $k = 1$ با خطای مطلق روش‌های $GA-TWM$ ، RSM ، ASM ، مقایسه می‌شود. نتایج این جدول برتری روش حاضر را نسبت به روش‌های دیگر نشان می‌دهد. نتایج عددی روش حاضر برای $M = 3, 5, 7$ ، $k = 1$ در شکل ۵ نشان داده می‌شود. همان‌طور که مشاهده می‌شود، با افزایش تعداد توابع پایه، خطای عددی روش حاضر کاهش می‌یابد و همگرایی روش به مقدار دقیق به وضوح قابل مشاهده است.

مثال ۴.۴. معادله براتو دوبعدی زیر را در نظر می‌گیریم [۸]:

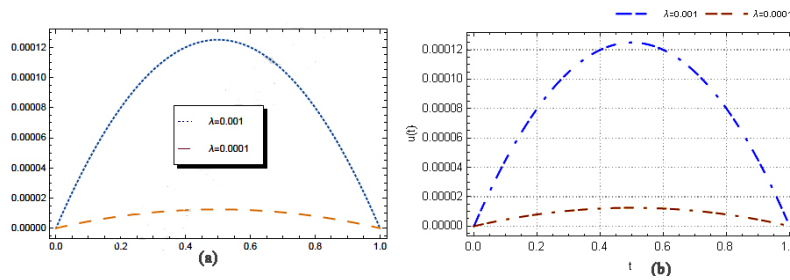
$$\frac{\partial^2 u(\xi, \eta)}{\partial \xi^2} + \frac{\partial^2 u(\xi, \eta)}{\partial \eta^2} + \lambda e^{u(\xi, \eta)} = 0,$$

$$u(\circ, \eta) = u(1, \eta) = u(\xi, \circ) = u(\xi, 1) = 0, \quad 0 \leq \xi, \eta < 1,$$

برای حل معادله براتو دوبعدی فوق با استفاده از شبکه عصبی موجک هان گسسته تابع مجهول $u(\xi, \eta)$ را به صورت زیر تقریب می‌زنیم:

$$u(\xi, \eta) \simeq N(\xi, \eta, U),$$

که U یک ماتریس مجهول $2^{k-1}M \times 2^{k-1}M$ است که باید بدست بیاید.



شکل ۴: مقایسه نتایج عددی با روش TWM (مثال ۲.۴)

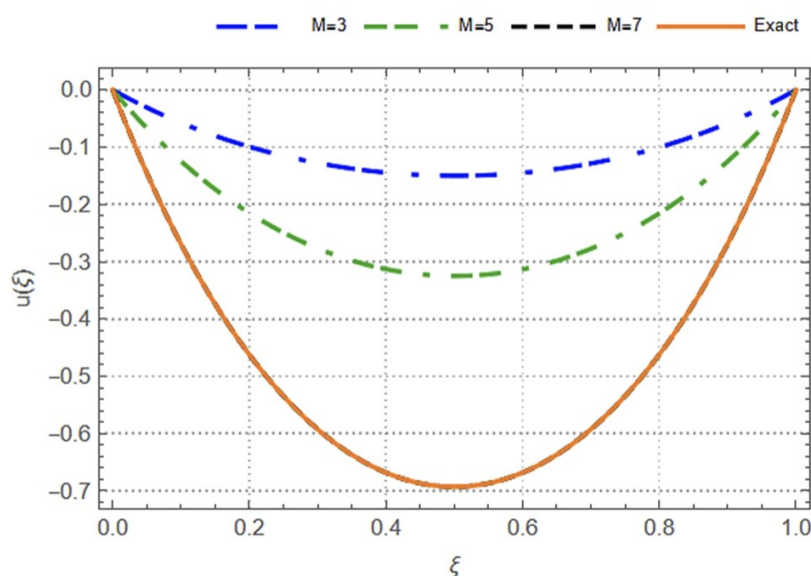
جدول ۵: مقایسه خطای مطلق با روش‌های دیگر (مثال ۳.۴)

روش حاضر	TWM	GA-ASM	RSM	GA	ξ
$1/19 \times 10^{-9}$	$5/59 \times 10^{-8}$	$1/63 \times 10^{-7}$	$1/49 \times 10^{-6}$	$8/23 \times 10^{-3}$	۰/۱
$3/68 \times 10^{-9}$	$7/46 \times 10^{-8}$	$2/53 \times 10^{-7}$	$8/64 \times 10^{-7}$	$5/95 \times 10^{-3}$	۰/۲
$8/08 \times 10^{-10}$	$1/34 \times 10^{-8}$	$3/09 \times 10^{-8}$	$9/89 \times 10^{-7}$	$4/15 \times 10^{-3}$	۰/۳
$9/15 \times 10^{-9}$	$8/99 \times 10^{-8}$	$6/82 \times 10^{-7}$	$5/20 \times 10^{-7}$	$3/06 \times 10^{-3}$	۰/۴
$1/40 \times 10^{-8}$	$1/43 \times 10^{-7}$	$6/44 \times 10^{-7}$	$5/41 \times 10^{-7}$	$2/40 \times 10^{-3}$	۰/۵
$8/57 \times 10^{-9}$	$8/99 \times 10^{-8}$	$1/56 \times 10^{-7}$	$5/49 \times 10^{-7}$	$1/76 \times 10^{-3}$	۰/۶
$1/13 \times 10^{-10}$	$1/34 \times 10^{-8}$	$7/05 \times 10^{-7}$	$2/64 \times 10^{-7}$	$9/46 \times 10^{-4}$	۰/۷
$3/65 \times 10^{-9}$	$7/46 \times 10^{-8}$	$9/73 \times 10^{-7}$	$3/35 \times 10^{-7}$	$6/33 \times 10^{-5}$	۰/۸
$1/70 \times 10^{-9}$	$5/59 \times 10^{-8}$	$1/29 \times 10^{-6}$	$2/61 \times 10^{-7}$	$7/62 \times 10^{-4}$	۰/۹
۳/۰۲۳	—	—	—	—	CPU

جدول ۶ خطای باقیمانده روش حاضر را برای $k = 1, M = 2$ و مقادیر مختلف λ گزارش می‌دهد. این جدول کارایی روش مطرح شده را برای معادلات براتو دوبعدی نشان می‌دهد.

۵ نتیجه‌گیری

در این مقاله، عملکرد روش شبکه عصبی مبتنی بر موجک‌های هان برای حل عددی معادلات دیفرانسیل براتو مورد بررسی قرار گرفت. ساختار شبکه شامل سه لایه ورودی، پنهان و خروجی است که در آن، به ترتیب از موجک‌های هان و تابع سینوس هایپربولیک به عنوان توابع فعال‌ساز لایه پنهان و لایه خروجی استفاده شده است. وزن‌های شبکه با بهره‌گیری از یک روش بهینه‌سازی تحلیلی تعیین شدند، به گونه‌ای که پاسخ تقریبی شبکه با شرایط مسئله سازگار باشد. نتایج حاصل از اجرای روش پیشنهادی بر روی چند مثال عددی نشان داد که این رویکرد از دقت مناسب و همگرایی مطلوب برخوردار است و قادر است پاسخ‌های عددی پایداری را با تعداد نسبتاً کمی پارامتر به دست آورد. استفاده از موجک‌های هان به عنوان تابع فعال‌ساز، به دلیل خواص تعامدی و قابلیت تقریب مناسب، منجر به کاهش پیچیدگی محاسباتی و افزایش کارایی روش در مقایسه با برخی روش‌های عددی متداول گردیده است. همچنین، به کارگیری روش بهینه‌سازی تحلیلی، نیاز به الگوریتم‌های یادگیری تکراری مبتنی بر گرادیان را کاهش داده و باعث سادگی پیاده‌سازی و کاهش هزینه محاسباتی شده است. به عنوان چشم‌انداز پژوهشی آینده، می‌توان توسعه روش حاضر را برای حل معادلات براتو با شرایط مرزی پیچیده‌تر، معادلات غیرخطی با مرتبه بالاتر، و نیز تعمیم آن به مسائل کسری و معادلات دیفرانسیل جزئی مورد توجه قرار داد. علاوه بر این، ترکیب شبکه عصبی موجک‌های هان با روش‌های هوشمند بهینه‌سازی و یا بررسی عملکرد آن در مسائل مهندسی و فیزیکی واقعی می‌تواند زمینه‌های تحقیقاتی ارزشمندی را فراهم سازد.



شکل ۵: نتایج عددی برای مقادیر متفاوت M (مثال ۳.۴)

سپاسگزاری

نویسندگان مراتب سپاس و قدردانی خود را از داوران گرامی مقاله که با ارائه دیدگاه‌های علمی دقیق و پیشنهادهای اصلاحی مؤثر، نقش مهمی در بهبود ساختار، شفافیت و محتوای این پژوهش ایفا کردند، اعلام می‌دارند.

مراجع

- [1] McGough J., "Numerical continuation and the Gelfand problem," Appl. Math. Comput., 89 (1998) 225-239.
- [2] Jacobsen J., and Schmitt K., "The Liouville-Bratu-Gelfand problem for radial operators," J. Differential, 184 (2002) 283-298.
- [3] He J., "Some asymptotic methods for strongly nonlinear equations," Internat. J. Modern Phys., 20 (2006) 1141-1199.
- [4] Jalilian R., "Non-polynomial spline method for solving Bratu's problem," Comput. Phys. Comm., 181 (2010) 1868-1872.
- [5] Mohsen A., "A simple solution of the Bratu problem," Comput. Math. Appl., 67(1) (2014) 26-33.
- [6] Kafri H., and Khuri, S., "Bratu's problem: A novel approach using fixed-point iterations and," Comput. Phys. Comm., 198 (2016) 97-104.
- [7] Ascher U.M., Matheij R.M., and Russel R.D., "Numerical solution of boundary value problems for ordinary differential equations", SIAM, Philadelphia, PA, 1995.
- [8] Yang Z., and Liao S., "A HAM-based wavelet approach for nonlinear partial differential equations: two dimensional Bratu problem as an application", Commun. Nonlinear Sci. Numer. Simul., 53 (2017) 249-62.

جدول ۶: مقایسه خطای باقیمانده برای مقادیر متفاوت λ (مثال ۴.۴)

$\lambda = ۵$	$\lambda = ۳/۵۱$	$\lambda = ۲$	$\lambda = ۱$	ξ
$۴/۴۴ \times ۱۰^{-۱۶}$	$۴/۴۴ \times ۱۰^{-۱۶}$	$۴/۰۸ \times ۱۰^{-۴۳}$	$۱/۱۱ \times ۱۰^{-۱۶}$	۰/۱
$۴/۴۴ \times ۱۰^{-۱۶}$	$۷/۰۰ \times ۱۰^{-۴۵}$	$۲/۰۸ \times ۱۰^{-۴۳}$	$۲/۲۲ \times ۱۰^{-۱۶}$	۰/۲
$۴/۴۴ \times ۱۰^{-۱۶}$	$۳/۵۶ \times ۱۰^{-۴۴}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	۰/۳
$۸/۸۸ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۸/۸۹ \times ۱۰^{-۴۵}$	$۱/۱۱ \times ۱۰^{-۱۶}$	۰/۴
$۹/۲۸ \times ۱۰^{-۴۵}$	$۶/۶۶ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۱/۱۱ \times ۱۰^{-۱۶}$	۰/۵
$۱/۵۲ \times ۱۰^{-۴۴}$	$۶/۶۶ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۱/۱۱ \times ۱۰^{-۱۶}$	۰/۶
$۴/۴۴ \times ۱۰^{-۱۶}$	$۶/۶۶ \times ۱۰^{-۱۶}$	$۴/۴۴ \times ۱۰^{-۱۶}$	$۱/۱۱ \times ۱۰^{-۱۶}$	۰/۷
$۱/۳۳ \times ۱۰^{-۱۵}$	$۴/۴۴ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	۰/۸
$۸/۸۸ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۲/۲۲ \times ۱۰^{-۱۶}$	$۱/۶۷ \times ۱۰^{-۱۶}$	۰/۹
۰/۱۸۸	۰/۲۱۹	۰/۲۱۹	۰/۲۱۹	CPU

- [9] Jin L., "Application of modified variational iteration method to the Bratu-type problems," Int. J. Contemp. Math. Sciences, 5 (2010) 153-158.
- [10] Caglar H., Caglar N., Ozer M., Valaristos A., and Anagnostopoulos, A., "B-spline method for solving Bratu's problem," Int. J. Comput. Math., 87(8) (2010) 1885-1891.
- [11] Aksoy A., and Pakdemirli M., "New perturbation-iteration solutions for Bratu-type equations," Comput. Math. Appl., 59 (2010) 2802-2808.
- [12] Venkatesh S., Ayyaswamy S., and Raja Balachandar, S., "The Legendre wavelet method for solving initial value problems of Bratu-type," Comput. Math. Appl., 63 (2012) 1287-1295.
- [13] Doha E., Bhrawy A., Baleanu D., and Hafez R., "Efficient Jacobi-Gauss collocation method for solving initial value problems of Bratu type," Comput. Math. Math. Phys., 53(9) (2013) 1292-1302.
- [14] Rashidinia J., Maleknejad K., and Taheri N., "Sinc-Galerkin method for numerical solution of the Bratu's problem," Numer. Algorithms, 62 (2013) 1-11.
- [15] Temimi H., and Ben-Romdhane M., "An iterative finite difference method for solving Bratu's problem," J. Comput. Appl. Math., 292 (2016) 76-82.
- [16] Keshavarz E., Ordokhani Y., and Razzaghi M., "The Taylor wavelets method for solving the initial and boundary value problems of Bratu-type equations," Appl. Numer. Math., 128 (2018) 205-216.
- [17] Yang C., and Hou J., "Chebyshev wavelets method for solving Bratu's problem," Bound. Value Probl., 2013 (2013) (142).
- [18] Abdollahi Z., Mohseni Moghadam M., Saeedi H., and Ebadi M., "A computational approach for solving fractional Volterra integral equations based on two-dimensional Haar wavelet method," Int. J. Comput. Math., 99(7) (2022) 1488-1504.
- [19] Schieber B., and Vahidi S., "Approximating connected maximum cuts via local search," 31st Annual European Symposium on Algorithms (ESA 2023) (2023).

- [20] Alam M., Haq S., Ali I., Ebadi M., and Salahshour S., "Radial basis functions approximation method for time-fractional FitzHugh-Nagumo equation," *Fractal Fract.*, 7(12) (2023) 882.
- [21] Kumar M., Yadav N., "Buckling analysis of a beam-column using multilayer perceptron," *J. Frankl. Inst.*, 50(10) (2013) 3188–3204.
- [22] Raja M., Niazi S., and Butt S., "An intelligent computing technique to analyze," *Neurocomputing*, 219 (2017) 280–299.
- [23] Effati S., Mansoori A., and Eshaghnezhad M., "An efficient projection neural network for solving bilinear programming problems," *Neurocomputing*, 168 (2015) 1188–1197.
- [24] Momani S., Abo-Hammour Z., and Alsmadi O., "Solution of inverse kinematics problem using genetic algorithms," *Appl. Math. Inf. Sci.*, 10(1) (2016) 225-233.
- [25] Mall S., and Chakraverty S., "Hermite functional link neural network for solving the Van der Pol-duffing oscillator equation," *Neural. Comput.*, 28(8) (2016) 1574-1598.
- [26] Raja M., Khan J., and Qureshi I., "Evolutionary computational intelligence in solving the fractional differential equations," In: *Asian Conference on Intelligent Information and Database Systems* Springer, (2010) 231-240.
- [27] Raja M., Manzar M., and Samar R., "An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP," *Appl. Math. Model.*, 39(0-11) (2015) 3075-3093.
- [28] Raja M., Samar R., Manzar M., and Shah S., "Design of unsupervised fractional neural network model optimized with interior point algorithm for solving Bagley-Torvik," *Math. Comput. Simul.*, 132 (2017) 139-158.
- [29] Rahimkhani P., and Heydari M., "Fractional shifted Morgan–Voyce neural networks for solving fractal-fractional pantograph differential equations," *Chaos, Solitons & Fractals*, 175 (2023) 114070.
- [30] Rahimkhani P., "Numerical solution of nonlinear stochastic differential equations with fractional Brownian motion using fractional-order Genocchi deep neural networks," *Commun. Nonlinear Sci. Numer. Simul.*, 126 (2023) 107466.
- [31] Parand K., Razzaghi M., Sahleh R., and Jani M., "Least squares support vector regression for solving Volterra integral equations," *Eng. Comput.*, 38 (2022) 789 –796.
- [32] Hadian-Rasanan A.H., Rahmati D., Gorgin S., and Parand K., "A single layer fractional orthogonal neural network for solving various types of Lane-Emden equation," *New Astron.*, 75 (2020) 101307.
- [33] Rahimkhani, P., Ordokhani Y., and Sabermahani S., "Bernoulli wavelet least squares support vector regression: robust numerical method for systems of fractional differential equations," *Math. Method. Appl. Sci.*, 46(17) (2023) 17641-17659.
- [34] Parand K., Aghaei A., Jani M., and Ghodsi A., "Parallel LS-SVM for the numerical simulation of fractional Volterra's population model," *Alex. Eng. J.*, 60 (2021) 5637-5647.
- [35] Taheri T., Afzal Aghaei A., and Parand K., "Bridging machine learning and weighted residual methods for delay differential equations of fractional order," *Appl. Soft Comput.*, 149 (2023) 110936.

- [36] Ebadi M., Hosseini A., and Hosseini M., "A projection type steepest descent neural network for solving a class of nonsmooth optimization problems ," Neurocomputing, 235 (2017) 164-181.
- [37] Salehi F., Saeedi H., and Moghadam M.M., "Discrete Hahn polynomials for numerical solution of two-dimensional variable-order fractional Rayleigh-Stokes problem", Comput. Appl. Math., 37(4) (2018) 5274-92.
- [38] Salehi F., Saeedi H., and Mohseni Moghadam M., "A Hahn computational operational method for variable order fractional mobile-immobile advection-dispersion equation", Math. Sci., 12(2) (2018) 91-101.
- [39] Hadian-Rasanan A.H., Rahmati D., Gorgin S., and Parand K., "A single layer fractional orthogonal neural network for solving various types of Lane-Emden equation", New Astron., 75 (2020) 101307.
- [40] Pakdaman M., Ahmadian A., Effati S., Salahshour S., and Baleanu D., "Solving differential equations of fractional order using an optimization technique based on training artificial neural network", Appl. Math. Comput., 293 (2017) 81-95 .
- [41] Rahimkhani P., and Ordokhani Y., "Hahn wavelets collocation method combined with Laplace transform method for solving fractional integro-differential equations," Math. Sci., 18(3) (2024) 463-477.