

Applying of Zhang Neural Network in Time-Varying Nonlinear Function Optimization

Elaahe Karami¹, Zeinab Mousavi^{2*}, Kobra Gholaami³

1. MS Student, Department of Mathematics, Bushehr Branch, Islamic Azad University, Bushehr, Iran.

Elahekr12@gmail.com

2. Assistant Professor, Department of Mathematics, Abhar Branch, Islamic Azad University, Abhar, Iran. (*Corresponding Author*) Mousavi.zeinabb@gmail.com

3. Assistant Professor, Department of Mathematics, Bushehr Branch, Islamic Azad University, Bushehr, Iran.

Kobra_gholami@yahoo.com

Abstract

Introduction: Optimization of nonlinear time-varying functions, as a subset of nonlinear programming, has been widely observed in various economic and engineering models. In energy management, one example of optimizing nonlinear functions with time-variable components is the efficient allocation of energy resources and managing changes in demand and supply, leading to increased efficiency and reduced energy waste. In this article, we intend to use Zhang neural networks for optimizing nonlinear functions with time-varying components. By harnessing the parallel processing power of neural networks, Zhang networks search the solution space faster than traditional methods, significantly reducing the required computational time.

Method: In this research, the proposed neural network receives data using MATLAB software. The data is first standardized using standard normalization methods. The data is then divided into four stages: training, testing, experimenting and validation which are further evaluated in five phases. The training data is based on the Luenberger-Madala algorithm for the first layer and a linear function for the second layer. Subsequently, the best network structure is considered with the transformation function and the proposed neural network model is tested in five stages. In this research, the Taylor series is used for data normalization and the zero-stability model of the n discrete time method is used to calculate error which reduces the error. The data in this research examined and evaluated in four stages of training, test, test and validation. Data training is based on Lunberg-Maud algorithm model for the first layer and linear function for the second layer. The reason for using Lunberg-Maud algorithm for research analysis is its convergence speed and higher efficiency due to not being in local minima and small error level.

Results: The best network structure with transformation function was considered and tested in 5 steps based on the proposed neural network model. The mean square of error in the third and fourth experiments has gradually increased compared to the first two stages. This amount of difference in the performance error as well as the coefficient of determination is different in each iteration and is caused by getting stuck in local minima.

Discussion: Due to the results obtained in the five test stages, it can be said that the algorithm based on the proposed neural network improves the performance of the network by increasing the learning rate. However, this algorithm is highly sensitive to local minima. This problem exists even when the learning rate is small and therefore the step of the algorithm is small. To avoid this sensitivity to local minima, the algorithm used in the proposed network was tested with momentum with different learning rates in five stages and the best result was evaluated. Also, at each stage, the process of training, testing and validation was evaluated separately.

Keywords: Neural network, Zhang neural network, Nonlinear optimization, Optimization, Time-varying nonlinear optimization.



انجمن علمی تجارت الکترونیکی ایران

سامانه‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند

Intelligent Multimedia Processing and Communication Systems (IMPCS)



واحد رزجان

استفاده از شبکه عصبی ژانگ با زمان گسسته برای بهینه‌سازی غیر خطی

دوره چهارم، تابستان ۱۴۰۲
شماره دوم، صص: ۳۱-۴۲

تاریخ دریافت: ۱۴۰۲/۰۱/۳۱
تاریخ پذیرش: ۱۴۰۲/۰۳/۱۰

الهه کرمی^۱، زینب موسوی^{۲*}، کبری غلامی^۳

۱. کارشناسی ارشد، گروه ریاضی، واحد بوشهر، دانشگاه آزاد اسلامی، بوشهر، ایران. Elahekr12@gmail.com

۲. استادیار، گروه ریاضی، واحد ابهر، دانشگاه آزاد اسلامی، ابهر، ایران. (نویسنده مسئول) Mousavi.zeinabb@gmail.com

۳. استادیار، گروه ریاضی، واحد بوشهر، دانشگاه آزاد اسلامی، بوشهر، ایران. Kobra_gholami@yahoo.com

چکیده: بهینه‌سازی توابع غیرخطی زمان-متغیر به‌عنوان زیر مجموعه‌ای از برنامه‌ریزی غیرخطی، به‌طور گسترده‌ای در انواع مدل‌های اقتصادی و مهندسی مشاهده شده است. در مدیریت انرژی به یک مثال از مدل توابع غیرخطی با زمان متغیر، می‌توان روش‌های بهینه‌ای را برای تخصیص منابع انرژی و مدیریت تغییرات در تقاضا و عرضه پیدا کرد که باعث بهره‌وری بیشتر و کاهش هدررفت انرژی می‌شود. در این مقاله قصد داریم از شبکه‌های عصبی ژانگ برای بهینه‌سازی توابع غیرخطی با زمان متغیر استفاده کنیم که به دلیل بهره‌گیری از قدرت پردازش موازی شبکه‌های عصبی، سریع‌تر از روش‌های سنتی در فضای حل جستجویی کند که این ویژگی می‌تواند زمان محاسباتی مورد نیاز را به میزان قابل توجهی کاهش دهد. در این پژوهش، با استفاده از نرم‌افزار متلب به منظور ورود داده‌ها به شبکه عصبی پیشنهادی، ابتدا با روش نرمال‌سازی استاندارد، نرمال شده‌اند. داده‌های مورد نظر در پژوهش در چهار مرحله، آموزش، تست، آزمایش و اعتبارسنجی و در پنج فاز مورد بررسی و ارزیابی قرار گرفتند. آموزش داده‌ها بر مبنای مدل الگوریتم لونیبرگ-ماد برای لایه اول و تابع خطی برای لایه دوم انجام شده است. در ادامه بهترین ساختار شبکه با تابع تبدیل در نظر گرفته شده و براساس مدل شبکه عصبی پیشنهادی در پنج مرحله مورد آزمایش قرار گرفته است.

واژه‌های کلیدی: شبکه عصبی، شبکه عصبی ژانگ، بهینه‌سازی غیرخطی، بهینه‌سازی، بهینه‌سازی غیرخطی با زمان متغیر.

۱. مقدمه

به عنوان زیرمجموعه‌ای از برنامه‌ریزی غیرخطی، بهینه‌سازی غیرخطی کاربرد گسترده‌ای داشته و بسیاری از مسائل را می‌توان به عنوان مدل بهینه‌سازی غیرخطی اصلاح کرد. به عنوان مثال، مدل وارپانس میانگین مارکوویتز در امور مالی، مسئله حمل و نقل در مدیریت، کوتاهترین مسیر در مدل شبکه، بهینه‌سازی قابل تفکیک محدب (غیر محدب) در حذف سیگنال ناخواسته از تصویر و غیره از این نوع می‌باشند. به دلیل نقش اساسی آن، بهینه‌سازی غیرخطی در طی قرن‌های گذشته توسط بسیاری از محققان به طور گسترده مورد مطالعه قرار گرفته است، و بسیاری از الگوریتم‌های کارآمد در مقالات توسعه یافته و مورد بررسی قرار گرفته که می‌توان آن‌ها را به دو دسته طبقه‌بندی کرد. دسته اول شامل روش‌های تکرار مرتبه اول است که فقط از اولین اطلاعات مشتق شده از تابع هدف استفاده می‌کند مانند روش نزولی شیب‌دار، روش گرادینان مزدوج و روش گرادینان حافظه‌دار، که این روش‌ها به دلیل ساختار ساده و ذخیره کم برای حل بهینه‌سازی غیرخطی در مقیاس بزرگ مناسب هستند. از نظر عددی، روش گرادینان مزدوج و روش گرادینان حافظه‌دار معمولاً عملکرد بهتری نسبت به روش نزولی شیب‌دار دارند. دسته دوم شامل روش نیوتن و انواع مختلف آن است که نیاز به محاسبه مشتق دوم تابع هدف یا تقریب آن دارد. بنابراین آن‌ها برای حل بهینه‌سازی غیرخطی مقیاس بزرگ مناسب نیستند، اگرچه دارای سرعت همگرایی موضعی بالایی هستند.

الگوریتم‌های فوق ذاتاً برای حل بهینه‌سازی غیرخطی ایستا طراحی شده‌اند. بنابراین ممکن است برای حل بهینه‌سازی غیرخطی با زمان متغیر با تابع هدف $f(x(t), t)$ متغیر تصمیم و t متغیر زمان، به اندازه کافی مؤثر نباشند. در نتیجه با الگوریتم‌های ذکر شده فوق می‌توان آن را حل کرد. با این حال این روش به سه دلیل زیر توصیه نمی‌شود:

۱. معمولاً به دلیل نیاز به حل دنباله بهینه‌سازی غیرخطی ساکن ناکارآمد بوده و دارای دقت کم می‌باشد.
۲. در فرآیند حل آنلاین بهینه‌سازی غیرخطی زمان گسسته متغیر با زمان، داده‌های موجود و یا قبلی با توجه به t_i ($i \leq k$) باید به اندازه کافی برای تولید متغیر تصمیم مجهول x_{k+1} استفاده شود.
۳. به طور عمده، در لحظه t_k ، اطلاعات آینده، از قبیل مقادیر تابع $f(x(t_{k+1}), t_{k+1})$ ، برای ما معلوم نیست و فقط از اطلاعات فعلی و گذشته برای نمونه‌های زمانی t_j با $j \leq k$

k استفاده می‌شود. بنابراین، روش‌های استاتیک متداول و شبکه‌های عصبی که بر پایه اطلاعات آینده می‌باشند نمی‌توانند بهینه‌سازی غیرخطی متغیر با زمان را حل کنند.

از طرف دیگر، در طول دهه‌های گذشته شبکه عصبی به دلیل ویژگی‌های خوب خود، از جمله فضای ذخیره‌سازی توزیع شده، پردازش موازی با سرعت بالا، برنامه‌های سخت‌افزاری و عملکرد برتر در برنامه‌های آنلاین در مقیاس بزرگ، توجه محققان را به خود جلب کرده است. برخی از شبکه‌های عصبی برای حل بهینه‌سازی غیرخطی طی دهه‌های اخیر توسعه یافته‌اند.

جین و همکاران [1] یک مدل یک مرحله‌ای زمان گسسته شبکه عصبی ژانگ^۱ (ZNN) برای بهینه‌سازی غیرخطی متغیر با زمان ارائه دادند که حداکثر خطای باقیمانده آن از نظر تئوری $O(\tau^2)$ است. در ادامه، گوو و همکاران [2] دو مدل زمان گسسته شبکه عصبی ژانگ برای بهینه‌سازی غیرخطی متغیر با زمان پیشنهاد داده‌اند که متعلق به زمان گسسته شبکه عصبی ژانگ سه مرحله‌ای است که خطای باقیمانده حالت پایدار به صورت $O(\tau^3)$ تغییر می‌کند. سپس، به تازگی ژانگ و همکاران [3] یک مدل پویایی مشتق زمان گسسته چهار مرحله و یک مدل کلی چهار مرحله‌ای زمان گسسته شبکه عصبی ژانگ برای بهینه‌سازی غیرخطی متغیر با زمان ارائه داده‌اند. هر دو مدل حاوی یک پارامتر آزاد هستند که می‌تواند هر مقداری از بازه $(\frac{1}{12}, \frac{1}{6})$ را بگیرد و با خطای برش $O(\tau^4)$ همگرا است.

از آنجا که، مدل پویایی مشتق در زمان مداوم برای به دست آوردن راه‌حل‌های زمان واقعی مسائل توسعه یافته است. بنابراین در این مقاله با هدف رفع نقاط ضعف مدل، یک مدل پویایی صفر مداوم ارائه و بررسی شده است. همچنین برای تحقق سخت‌افزار دیجیتال بالقوه، با استفاده از تبدیل دو خطی، یک فرمول عمومی چهار مرحله‌ای ژانگ و همکاران برای گسسته‌سازی هر دو مدل پیشنهاد شده و اعمال می‌شود. یک پویایی مشتق چهار مرحله‌ای گسسته کلی چهار مرحله‌ای و یک مدل کلی چهار مرحله‌ای پویایی زمان گسسته کلی چهار مرحله‌ای بر اساس این فرمول کلی چهار مرحله‌ای ارائه شده است.

هدف این مقاله، تعیین اندازه گام بهینه می‌باشد و سعی شده است تا مطالعه دو مدل کلی چند مرحله‌ای زمان گسسته شبکه عصبی ژانگ گسترش یابد. علاوه بر این، مرز فوقانی محکم و اندازه گام بهینه مورد تجزیه و تحلیل قرار گیرد و سپس پارامتر آزاد در کمترین زمان ممکن حل شود. مطابق با معیار پایداری راث^۲، فرمول کلی پنج مرحله‌ای مدل گسسته‌سازی ژانگ با خطای کوتاه‌سازی برش $O(\tau^5)$ ، با استفاده از نرم‌افزار متلب، به دست آمده است. در این رابطه می‌بایست دو مدل کلی

چندمرحله‌ای و رابطه پارامتر و اندازه گام زمان گسسته شبکه عصبی ژانگ را به دست آوریم.

معیارهای عملکرد شبکه برای ارائه نتایج عبارتند از: میانگین مربعات خطا، مجذور میانگین مربعات خطا، انحراف معیار، میزان رگرسیون، میانگین و انحراف معیار استاندارد می‌باشند.

در این مقاله از سری تیلور برای نرمال‌سازی داده‌ها و از مدل پایداری صفر روش زمان گسسته Π برای محاسبه خطا استفاده می‌شود. استفاده از سری تیلور برای همگن کردن داده‌ها دارای این مزیت است که می‌توان رگرسیون داده‌ها را بهبود بخشید که این خود موجب کاهش خطای می‌گردد. در این پژوهش به منظور ورود داده‌ها به شبکه عصبی پیشنهادی، با استفاده از نرمال‌سازی استاندارد، نرمال شده‌اند. در ادامه این مقاله، مسئله بهینه‌سازی غیرخطی و مدل شبکه عصبی ژانگ معرفی می‌گردد و سپس در بخش دوم الگوریتم پیشنهادی ارائه شده است. در بخش آخر نیز نتایج عددی و برنامه متلب گنجانده شده است. از آنجاکه در این پژوهش برای حل مسئله بهینه‌سازی غیرخطی، از شبکه عصبی ژانگ استفاده خواهیم کرد ضروری است این دو را معرفی کنیم.

۲. بهینه‌سازی غیرخطی

هدف مسائل برنامه‌ریزی غیرخطی در شکل کلی آن، پیدا کردن مقادیر $x = (x_1, x_2, \dots, x_n)$ است به گونه‌ای که:

$$(1) \text{Max } f(x)$$

$$g_i(x) \leq b_i \quad i = 1, 2, \dots, m$$

$$x \geq 0$$

که $f(x)$ ، $g_i(x)$ توابع n متغیره معلوم هستند.

الگوریتمی وجود ندارد که بتواند همه مسائلی که در چارچوب برنامه‌ریزی غیرخطی به صورت فوق می‌گنجد را حل کنند. لیکن، چنانچه فرضیات مشخصی در مورد این توابع به کار گرفته شود، آنگاه حالت‌های خاصی از برنامه‌ریزی غیرخطی به دست می‌آیند که در رابطه با حل آن‌ها پیشرفت‌های چشمگیری حاصل شده است [4].

شبکه عصبی الهام گرفته شده از ویژگی‌های مغز و سیستم عصبی بدن انسان است. نورون، اصلی‌ترین بخش شبکه عصبی انسان می‌باشد. عملیات دریافت و پردازش اطلاعات و همچنین انتقال نتایج به دست آمده، توسط نورون^۳ صورت می‌گیرد. نورون بیولوژیکی که پایه مکتب شبکه‌های عصبی بر این بنیان استوار است، در واقع خود یک موتور ساده است، اما نحوه عملکرد نورون‌ها برای ما کاملاً روشن نیست و هر نورون تقریباً به ۱۰,۰۰۰ نورون دیگر متصل است و از سه قسمت تشکیل شده است:

جسم سلولی (مرکز نورون)

آکسون (رشته خروجی هر نورون)

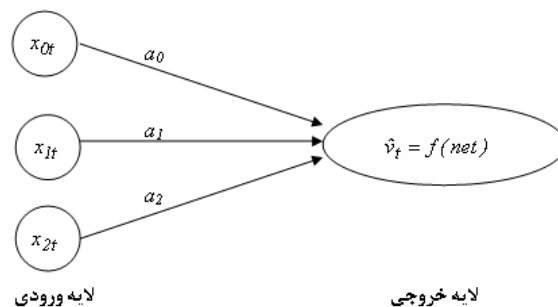
دندریت (شاخه‌های کوتاه درخت مانند انتهای هر آکسون)

به نقاطی که آکسون‌های نورون به یکدیگر متصل می‌شوند سیناپس گفته می‌شود. از طریق سیناپس‌ها علائم یا سیگنال‌های یک آکسون به نورون‌های دریافت‌کننده آن علائم منتقل می‌شوند. میزان علائم بستگی به نیروی سیناپتیک در محل اتصال دارد که دارای ماهیت شناسایی است. محققین در این باره معتقدند که کلید اصلی درک رفتار مغز انسان به عنوان یک سیستم پردازش اطلاعات در چگونگی ارتباط و اتصال نورون‌ها با یکدیگر نهفته است [5]. یک شبکه عصبی مصنوعی، مجموعه‌ای از نورون‌های به هم متصل در لایه‌های مختلف است که اطلاعاتی را برای یکدیگر ارسال می‌کنند. در بدن انسان، نورون‌ها ساده‌ترین واحد ساختاری شبکه عصبی هستند. در واقع بافت‌هایی که عصب نامیده می‌شوند، اجتماعی از نورون‌ها هستند. این نورون‌ها، اطلاعات و پیام‌ها را از یک قسمت بدن به قسمت دیگر منتقل می‌کنند. کلید اصلی درک رفتار مغز انسان به عنوان یک سیستم پردازش اطلاعات، در چگونگی ارتباط و اتصال نورون‌ها با یکدیگر است. عنصر اصلی شبکه‌های عصبی مصنوعی نیز نورون‌های مصنوعی به عنوان عناصر ریاضی پردازشگر هستند. در ادبیات شبکه‌های عصبی در بسیاری از موارد از کلمه واحد^۴ یا گره^۵ به جای نورون استفاده می‌شود [6].

ساده‌ترین شکل شبکه عصبی تنها دو لایه دارد، لایه ورودی^۶ و لایه خروجی^۷. شبکه عصبی همچون یک سیستم ورودی-خروجی عمل می‌کند و ارزش نورون‌های ورودی را برای محاسبه ارزش نورون‌های خروجی مورد استفاده قرار می‌دهد. هر نورون، به وسیله یک دایره و ارتباط میان نورون‌ها به وسیله یک فلش نشان داده می‌شود. خروجی y و ورودی‌های x_0 و x_1 و x_2 بردارهای $1 \times N$ هستند که در آن، N تعداد مشاهدات است. اطلاعات منحصرأ از ورودی‌ها به خروجی‌ها حرکت می‌کنند، بنابراین، مدل مورد بحث به شبکه عصبی پیش‌خور معروف است. ارتباط میان یک ورودی و یک خروجی به وسیله یک وزن a که بیانگر اهمیت نسبی ورودی مذکور در محاسبه ارزش خروجی است، مشخص می‌شود.

به این ترتیب ارزش نورون خروجی مشاهده t از رابطه زیر به دست می‌آید.

$$Net_t = a_0 x_{0t} + a_1 x_{1t} + a_2 x_{2t} = \sum_{i=0}^2 a_i x_{it} \quad (2)$$



شکل ۱: مدل شبکه عصبی ساده پیش‌خور

سپس نورون خروجی ارزش به‌دست‌آمده را با استفاده از یک تابع تبدیل یا فعال‌ساز که با $f(x)$ نشان داده می‌شود، پردازش می‌کند. در ساده‌ترین شکل شبکه عصبی پیش‌خور، تابع فعال‌ساز خطی است (به عنوان نمونه تابع $f(x) = x$).

در رابطه زیر، ارزش به‌دست‌آمده و یک تابع فعال‌ساز خطی، خروجی نهایی شبکه را برای مشاهده t به صورت زیر می‌سازد [6].

$$\begin{aligned} \hat{y}_t &= f(\text{Net}_t = a_0x_{0t} + a_1x_{1t} + a_2x_{2t}) \\ &= a_0x_{0t} + a_1x_{1t} + a_2x_{2t} \end{aligned} \quad (3)$$

در این گونه معادلات، معمولاً یکی از ورودی‌ها برای مشاهدات دارای یک است و جمله اریب^۱ نامیده می‌شود. در این رابطه اگر در نظر گرفته شود که x_0 جمله اریب باشد، در آن صورت می‌توان بیان داشت که خروجی شبکه از رابطه زیر حاصل می‌شود:

$$\hat{y}_t = a_0 + a_1x_{1t} + a_2x_{2t} \quad (4)$$

در معادلات شبکه عصبی متداول مانند یک شبکه عصبی پیش‌خور با دو لایه و تابع فعال‌ساز خطی مشابه مدل رگرسیون خطی چندمتغیره است. در این گونه شبکه‌ها، نورون‌های ورودی همان متغیرهای مستقل یا رگرسیون‌ها هستند و نورون خروجی همان برآورد متغیر وابسته است. وزن‌های مختلف شبکه نیز، مشابه پارامترهای مدل رگرسیون و جمله اریب نیز همان عرض از مبدأ یا جمله ثابت در مدل رگرسیون است. در صورتی که وقفه‌های متغیر وابسته به مجموعه ورودی‌ها اضافه شود، در آن صورت شبکه‌ای مشابه با مدل اتورگرسیون خطی حاصل می‌شود. رایج‌ترین تابع فعال‌ساز مورد استفاده در ادبیات شبکه عصبی، تابع لجستیک^۲ می‌باشد که به صورت زیر تعریف می‌شود.

$$f(x) = \frac{1}{1 + e^{-x}}$$

مقدار تابع لجستیک در محدوده صفر و یک قرار دارد. توابع دیگری مانند تابع گوسین، تابع آستانه‌ای و... نیز مورد استفاده قرار می‌گیرند.

با هدف حل مشکلات متغیر در زمان و برنامه درجه دوم متغیر در زمان، نوع خاصی ماتریس متغیر در زمان و همکاران پیشنهاد شده است. از مارس ۲۰۰۱ طبق روش از شبکه‌های عصبی مکرر با نام شبکه عصبی ژانگ (ZNN) به‌طور رسمی توسط ژانگ و همکاران پیشنهاد شده است. از مارس ۲۰۰۱ طبق روش طراحی ژانگ و همکاران، طراحی ZNN بر اساس یک تابع ژانگ^۱ نامحدود (ZF) است، با کلمه "نامحدود" به این معنی که چنین ZF می‌تواند مثبت، صفر، منفی یا حتی کمتر نامحدود باشد با بهره‌گیری روش شناختی از اطلاعات مشتق شده از زمان ضرایب متغیر زمان درگیر در مشکلات متغیر زمان، مدل‌های ZNN حاصل می‌توانند مشکلات متغیر زمان را به‌طور مؤثر و کارآمد حل کنند (از نظر اجتناب از خطاهای عقب‌افتاده توسط روش‌های متداول ایجاد شده است). برای درک بهتر و ایجاد زمینه برای تحقیقات بیشتر، مفاهیم ZNN و ZF به شرح زیر ارائه شده است [8].

شبکه عصبی ژانگ که نوع خاصی از شبکه‌های عصبی مکرر است، از سال ۲۰۰۱ توسعه یافته و مورد مطالعه قرار گرفته است. این شبکه از تحقیقات شبکه‌های عصبی نوع هاپفیلد نشأت گرفته و روشی سیستماتیک برای حل مشکلات متغیر در زمان است. مشکلاتی که در شبکه عصبی ژانگ باید حل شود، عملکرد خطای نامحدود، فرمول طراحی نوع نما، معادله پویا و استفاده از اطلاعات مشتق از زمان، با شبکه عصبی شیب معمولی متفاوت است [9], [10].

تابع ژانگ (ZF) اساس طراحی ZNN است. این تابع با توابع خطا/انرژی رایج در مطالعه روش‌های متداول تفاوت دارد. به‌طور خاص، در مقایسه با تابع انرژی مثبت مقیاس‌دار با ارزش پله‌ای یا حداقل پایین‌تر، ZF می‌تواند محدود، نامحدود یا حتی پایین‌تر از حد (در یک کلمه، نامحدود) باشد. علاوه بر این، مطابق با مسئله بردار یا ماتریس که باید حل شود، ZF می‌تواند بردار یا ماتریس باشد تا روند حل را به‌طور کامل کنترل کند.

۳. الگوریتم پیشنهادی

اساس طراحی برای مدل‌های ZNN برای حل مسائل LMVI^۱ و LMI^۲ متغیر در زمان است. بنابراین، برای راحتی ارائه، در این بخش، ZF با $E(t)$ نشان داده می‌شود که $\dot{E}(t)$ مشتق زمانی $E(t)$ است. بر اساس ZF، روش طراحی یک مدل ZNN برای حل LMVI/LMI متغیر در زمان به شرح زیر ارائه شده است.

در مرحله اول، یک ZF نامحدود به عنوان عملکرد نظارت بر خطا برای نظارت بر روند حل LMVI/LMI متغیر در زمان تعریف شده است.

به‌طور خاص، مشکل LMVI متغیر در زمان در این فصل مورد توجه قرار گرفته‌است:

$$A(t)x(t) \leq b(t)$$

که در آن $A(t) \in \mathbb{R}^{n \times n}$ و $b(t) \in \mathbb{R}^n$ به ترتیب ماتریس، بردار متغیر زمان و $x(t) \in \mathbb{R}^n$ بردار متغیر زمان ناشناخته‌ای است که باید به دست آید. هدف یافتن راه‌حل عملی $x(t)$ است که (۴) برای هر لحظه $t \geq 0$ صادق باشد. توجه داشته باشید که برای بحث بیشتر، فرض می‌شود که $A(t)$ در هر زمان $t \in [0, +\infty)$ در این فصل نامنفرد (معکوس‌پذیر) است.

تابع بهینه‌سازی غیرخطی متغیر با زمان¹³ (TVNO) به صورت رابطه زیر در نظر گرفته می‌شود:

$$\min_{x(t) \in \mathbb{R}^n} f(x(t), t) \in \mathbb{R}, \quad t \in [0, t_f] \quad (5)$$

در این رابطه، تابع هدف $f(x, y)$ به صورت رابطه زیر تعریف می‌شود:

$$f(x, y) : \mathbb{R}^n \times [0, t_f] \rightarrow \mathbb{R} \quad (6)$$

به عنوان تابع مرتبه دوم قابل تمایز و محدود در نظر گرفته می‌شود.

تابع مرتبه دوم قابل تمایز و محدود به حل مقادیر $x(t) \in \mathbb{R}^n$ در تابع هدف $f(x(t), t)$ کمک می‌کند که به جهت مینیمم‌سازی مقادیر در نظر گرفته شده در هر $t \geq 0$ ضروری است.

در این معادلات یافتن راه‌حل بهینه به منظور بهینه‌سازی تابع غیرخطی ثابت با زمان، توسط الگوریتم‌های عددی سنتی دشوار است. بنابراین به یافتن نقطه ثابت بهینه‌سازی غیرخطی متغیر با زمان متوسل می‌شویم. در این صورت معادله (۵) ارائه شده، به یافتن نقطه ثابت با در نظر گرفتن معادلات غیرخطی تبدیل می‌شود:

$$g(x(t), t) = \frac{\partial f(x(t), t)}{\partial x(t)} = 0 \in \mathbb{R}^n, \quad \forall t \in [0, t_f] \quad (7)$$

به منظور رسیدن به جواب بهینه، پارامتر $e(t) = g(x(t), t)$ در رابطه شبکه عصبی ژانگ به صورت رابطه زیر بیان می‌شود:

$$\dot{e}(t) = \frac{de(t)}{dt} = -\gamma e(t), \quad \gamma > 0 \quad (8)$$

در این حالت مدل زمان پیوسته در شبکه عصبی ژانگ به صورت رابطه زیر حاصل می‌شود:

$$\dot{x}(t) = -H(x(t), t)^{-1} (\gamma g(x(t), t) + g_t(x(t), t)) \quad (9)$$

ثانیاً، مجبور کردن ZF یعنی $E(t)$ که به صفر همگرا شوند. بدین معنی که از طریق فرمول طراحی ZNN مشتق زمانی آن را انتخاب می‌کنیم یعنی $\dot{E}(t)$.

سرانجام، با گسترش فرمول طراحی ZNN، معادله پویا یک مدل ZNN برای حل LMVI/LMI متغیر در زمان ایجاد می‌شود. به منظور استخراج مدل‌های مختلف ZNN برای حل LMVI و LMI متغیر در زمان، دو فرمول طراحی زیر (که بخش مهمی در روش طراحی ZNN فوق هستند) در این فصل مورد استفاده قرار گرفته‌است:

$$\begin{aligned} \dot{E}(t) &= -\gamma F(E(t)) \\ \dot{E}(t) &= -\gamma SGN(\dot{E}) \odot F(E(t)), \end{aligned}$$

به طوری که

$$\gamma > 0 \in \mathbb{R}$$

یک پارامتر ظرفیت است که برای مقیاس‌بندی نرخ همگرایی استفاده می‌شود و F نشان‌دهنده آرایه تابع فعال‌سازی است. توجه داشته باشید که به طور کلی، پارامتر طراحی γ باید به اندازه سیستم سخت افزاری اجازه دهد، یا به طور مناسب برای اهداف شبیه‌سازی انتخاب شود. علاوه بر این، تابع f_0 ، به عنوان یک عنصر پردازش F ، می‌تواند هر تابع فعال‌کننده یکنواختی را افزایش دهد، به عنوان مثال، توابع فعال‌ساز خطی قدرت سیگموئید و هیپربولیک سینوس. علاوه بر این $E_0 = E(t=0)$ خطای اولیه را نشان می‌دهد و تابع علامت تک‌قطبی، $sgn(\cdot)$ که عنصری از $sgn(0)$ است، به صورت زیر تعریف می‌شود:

$$sgn(c) = \begin{cases} 1, & \text{if } c > 0, \\ 0, & \text{if } c \leq 0. \end{cases}$$

علاوه بر این، عملگر \odot ضرب هادامارد است و به شرح زیر تعریف شده است:

$$U \odot V = \begin{bmatrix} u_{11}v_{11} & \cdots & u_{1n}v_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1}v_{m1} & \cdots & u_{mn}v_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

در مورد فرمول‌های طراحی ارائه شده (۲) و (۳)، فرمول اصلی فرمول طراحی ZNN است که توسط ژانگ و همکاران پیشنهاد شده است. برای حل معادله متغیر زمان سیلوستر، در حالی که دومین نوع چنین فرمول طراحی ZNN است که به طور مفصل برای حل نابرابری‌های خطی متغیر در زمان ساخته شده است. بنابراین، برای سهولت ارائه و درک بهتر، (۲) فرمول اصلی طراحی، در حالی که (۳) فرمول طراحی متنوع برای حل LMVI و LMI متغیر در زمان در این فصل نامیده می‌شود.

در این بخش، ما دو ZF مختلف را معرفی کرده و مدل‌های حاصل از ZNN را برای حل نابرابری ماتریس بردار خطی متغیر در زمان (LMVI) توسعه می‌دهیم. سپس، نتایج و تجزیه و تحلیل‌های نظری ارائه می‌شود تا عملکرد محاسباتی چنین دو مدل ZNN را نشان دهد [3].

که در این رابطه مؤلفه $g_t(x(t), t)$ مشتق جزئی به دست آمده از $g(x(t), t)$ می باشد که به صورت رابطه زیر به دست می آید:

$$g_t(x(t), t) = \frac{\partial g(x(t), t)}{\partial t} = \frac{\partial^2 f(x(t), t)}{\partial x(t) \partial t} = \left[\frac{\partial^2 f}{\partial x_1 \partial t}, \frac{\partial^2 f}{\partial x_2 \partial t}, \dots, \frac{\partial^2 f}{\partial x_n \partial t} \right]^T \quad (10)$$

ماتریس کلی به صورت رابطه زیر در نظر گرفته می شود:

$$H(x(t), t) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \quad (11)$$

تفاوت اصلی مدل شبکه عصبی ژانگ و مدل های شبکه عصبی دیگر در ارزیابی اطلاعات مشتق زمانی $g_t(x(t), t)$ برای به دست آوردن راه حل سریع و دقیق در تابع بهینه سازی غیرخطی متغیر با زمان (TVNO) می باشد. معادله حرکت در مدل های شبکه عصبی اخیر برای سایر مسائل بهینه سازی، مانند نابرابری های متغیر و مسائل مکمل به صورت رابطه زیر تعریف می شود:

$\dot{x}(t) = F(x)$
 که در این رابطه $F(x): \mathbb{R}^n \rightarrow \mathbb{R}^n$ که یک نگاشت با توجه به متغیر تصمیم x و یک تابع ضمنی با توجه به t است.

$$\dot{x}(t) = -\gamma Hx^{-1} g(x) \quad (12)$$

مجموعه x بر مبنای تابع هدف به صورت رابطه زیر نمایش داده می شود.

$$\dot{f}_k = \frac{1}{\tau} \left(\sum_{i=1}^{n+1} a_i f_{k-n+i} \right) + O(\tau^p) \quad (13)$$

که در این رابطه n تعداد راه حل های بهینه به منظور رسیدن به جواب نهایی، $a_i \in R (i = 1, 2, \dots, n+1)$ ضریب اطمینان در مجموعه، $O(\tau^p)$ بیانگر میزان خطای کوتاه سازی، f_k تابع هدف $f(t)$ با در نظر گرفتن ثابت زمانی t_k می باشد. در این رابطه، k بیانگر به روزرسانی مجموعه با در نظر گرفتن شاخص ها است.

حال گام های روش ارائه شده در این مقاله را به صورت خلاصه زیر بیان می کنیم.

مرحله اول: در این مرحله، مدل پایداری صفر روش زمان گسسته n مرحله ای، به صورت رابطه زیر حاصل می شود:

$$x_{k+1} + \sum_{i=1}^n a_i x_{k+1-i} = \tau \sum_{i=0}^n \beta_i v_{k+1-i} \quad (14)$$

پایداری در نظر گرفته شده به صورت حاصل ریشه های معادله ی زیر حاصل می شود:

$$P(v) = v^n + \sum_{i=1}^n a_i v^{n-i} \quad (15)$$

مرحله دوم: یک روش زمان گسسته n مرحله ای بر مبنای تابع هدف مسئله در نظر گرفته می شود. در این روش خطای تابع $P(v)$ به صورت $O(\tau^p)$ محاسبه می شود و هر چقدر این خطا کمتر شود، تابع بهینه تر شده و به سمت هدف متمایل تر می شود.

مرحله سوم: پس از تعیین میزان خطای بهینه، هر مجموعه X به سمت مجموعه هدف خود میل می کند (۱۶). جواب نهایی زمانی حاصل می شود که چنین الگوریتمی به سمت صفر پایدار و یا سازگار میل نماید. به عبارت دیگر، ثبات و میل به سمت صفر پایدار موجب به همگرایی می شود. به طور کلی یک روش سازگار با کمینه شدن خطای کوتاه سازی آن، همگرا می شود.

$$x_{[(t-t_0)/\tau]} \rightarrow x^*(t) \quad (16)$$

در این مرحله مجموعه لایه ها به منظور کمینه سازی خطا، بر مبنای سری تیلور به صورت رابطه ی (۱۷) محاسبه می شود.

$$f_{k+1} = f_k + \tau \dot{f}_k + \frac{\tau^2}{2} \ddot{f}_k + \dots + \frac{\tau^m}{m!} f_k^{(m)} + O(\tau^{m+1}) \quad (17)$$

در این رابطه m یک عدد صحیح غیرمنفی می باشد. فرآیند دو مرحله ای به منظور یافتن میزان خطای بهینه کوتاه سازی در سری تیلور به صورت رابطه (۱۸) محاسبه می شود:

$$\dot{f}_k = \frac{f_{k+1} - f_{k-1}}{2\tau} \quad (18)$$

در این رابطه اگر تعداد نوروں ها مجموعه $(-\frac{1}{2}, +\infty)$ در نظر گرفته شود، در این صورت رابطه (۱۸) به صورت رابطه (۱۹) بیان می شود:

$$\dot{f}_k = \frac{(1+a_1)f_{k+1} - (1+2a_1)f_k + a_1f_{k-1}}{\tau} \quad (19)$$

مجموعه X نیز بر مبنای تکرار مؤلفه k به صورت رابطه (۲۰) محاسبه می شود:

$$x_{k+1} = x_k - H(x_k, t_k)^{-1} (hg(x_k, t_k) + \tau g_t(x_k, t_k)) \quad (20)$$

برای هر نمونه گیری ثابت به منظور تعیین لایه های پنهان در هر دور، خطای کوتاه سازی به عنوان پارامتر کاهش می شود.

۴. نتایج

۴.۱. تجزیه و تحلیل داده ها

معیارهای عملکرد شبکه برای ارائه نتایج عبارتند از: میانگین مربعات خطا، مجذور میانگین مربعات خطا، انحراف معیار، میزان رگرسیون، میانگین و انحراف معیار استاندارد می باشند. نمودارهای رگرسیونی مربوط به نتایج خروجی، در ادامه نمایش داده شده است.

و بهترین نتیجه مورد ارزیابی قرار گرفت. همچنین در هر مرحله، فرآیند آموزش، تست و اعتبارسنجی به صورت جداگانه مورد ارزیابی قرار گرفت.

۲.۴. پردازش داده‌ها و اعتبارسنجی به کمک مدل شبکه عصبی پیشنهادی

شبکه عصبی مورد استفاده در این پژوهش از نوع شبکه عصبی LNF- FE بوده و از مجموع داده‌های پژوهش، ۷۰ درصد از داده‌ها در مرحله آموزش، ۱۵ درصد از داده‌ها در مرحله اعتبارسنجی و ۱۵ درصد دیگر از داده‌ها در مرحله تست مورد ارزیابی قرار گرفتند.

نتایج حاصله در این مرحله شامل ارائه معادله کلی در مرحله تست، آموزش، اعتبارسنجی و حالت کلی و ضریب رگرسیون می‌باشد. در مرحله پردازش اطلاعات با استفاده از داده‌های ورودی اولیه و تابع انتقال، قانون آموزش و تعداد لایه پنهان مناسب، که در مراحل قبل انتخاب شدند، استفاده شده‌است. در صورتی که پارامتر ورودی در این مرحله به سیستم را به عنوان سیگنال در نظر بگیریم، در این صورت می‌توان ادعا کرد که رفتار سیگنال در زمان t می‌تواند وابسته به رفتار سیگنال در زمان $t-1$ باشد. همچنین لایه پنهان به گونه‌ای که شبکه کمترین خطا را داشته باشد، به روش آزمون و خطا تعیین می‌شود. نمودار رگرسیونی بر مبنای شبکه عصبی پیشنهادی در چهار تکرار مورد ارزیابی قرار گرفته‌است و در هر قسمت داده‌های حاصل بر مبنای حالت آموزش، تست، اعتبارسنجی و حالت کلی بیان شده‌اند. در ادامه نتایج حاصل بر مبنای نمودارهای مورد ارزیابی ارائه شده‌است. مطابق شکل ۲ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی، معادله حاصل در آموزش برابر با معادله ارائه شده در (۲۱) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (21)$$

میزان رگرسیون حاصل در مرحله آموزش برابر با $R = 0.79213$ می‌باشد.

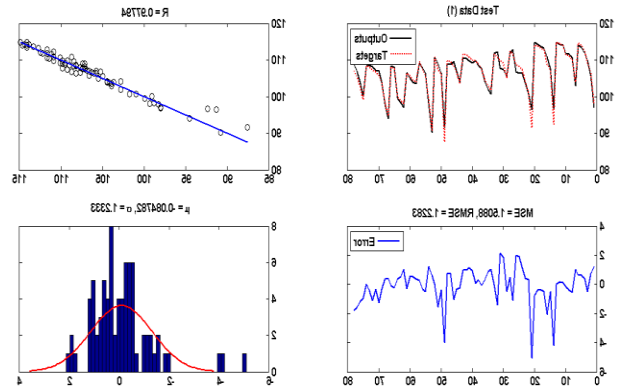
طبق شکل ۶ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی، معادله حاصل در تست برابر با معادله ارائه شده در (۲۲) می‌باشد.

$$output = 0.63 * Target + 7.9 \quad (22)$$

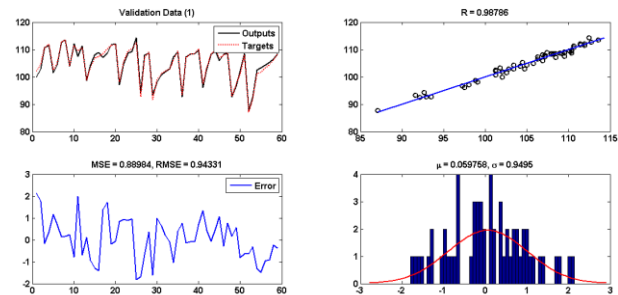
میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.7898$ می‌باشد.

مطابق شکل ۶، معادله حاصل در اعتبارسنجی برابر با معادله ارائه شده در (۲۳) می‌باشد.

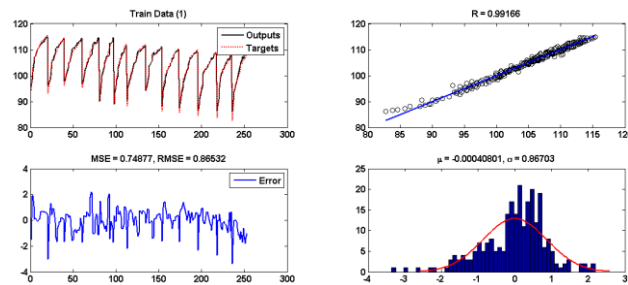
$$output = 0.62 * Target + 8 \quad (23)$$



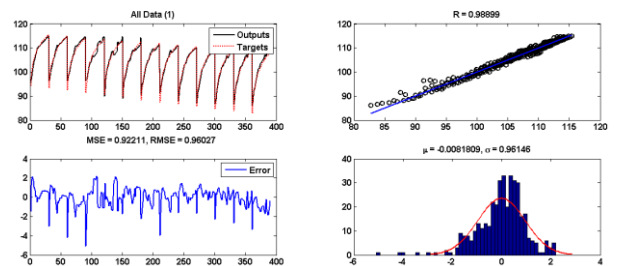
شکل ۲: نمودار رگرسیونی حالت تست



شکل ۳: نمودار رگرسیونی حالت اعتبارسنجی



شکل ۴: نمودار رگرسیونی حالت آموزش



شکل ۵: نمودار رگرسیونی حالت کلی

بر مبنای نتایج در پنج مرحله آزمایش می‌توان بیان داشت که الگوریتم مورد استفاده بر مبنای شبکه عصبی پیشنهادی، با افزایش نرخ یادگیری عملکرد شبکه بهبود می‌یابد. این الگوریتم به شدت نسبت به مینیمم‌های محلی حساس است، حتی در مواقعی که نرخ یادگیری کوچک و در نتیجه گام حرکت الگوریتم کم است، این مشکل نیز وجود دارد. برای جلوگیری از این امر، الگوریتم مورد استفاده در شبکه پیشنهادی، همرا با ممنتم با نرخ یادگیری متفاوت در پنج مرحله مورد آزمایش قرار گرفت

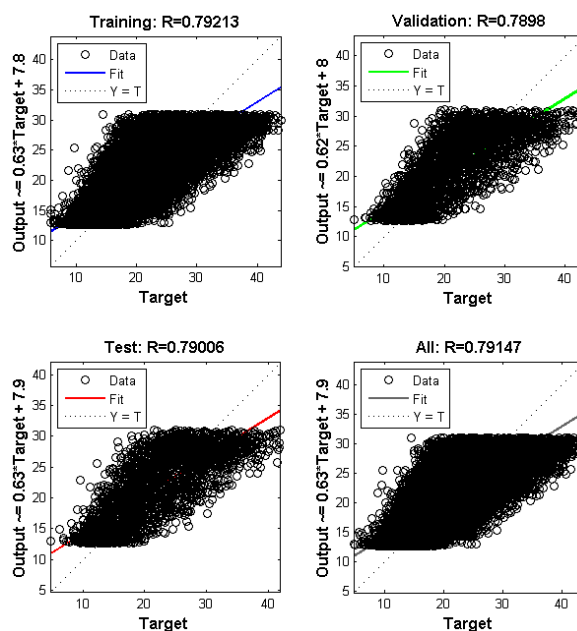
میزان رگرسیون در مرحله اعتبارسنجی برابر با $R = 0.78117$ می‌باشد.

مطابق شکل ۶ نمودار رگرسیونی، معادله حاصل در حالت کلی برابر با معادله ارائه‌شده در (۲۴) می‌باشد.

$$output = 0.63 * Target + 7.9 \quad (24)$$

میزان رگرسیون در مرحله کلی برابر با $R = 0.79147$ می‌باشد. مطابق شکل ۷، معادله حاصل در آموزش برابر با معادله ارائه‌شده در (۲۵) می‌باشد.

$$output = 0.52 * Target + 9.8 \quad (25)$$



شکل ۶: نمودار رگرسیونی شبکه عصبی پیشنهادی در

تکرار اول

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.78238$ می‌باشد.

طبق شکل ۷ نمودار رگرسیونی، معادله حاصل در تست برابر با معادله ارائه‌شده در (۲۶) می‌باشد.

$$output = 0.52 * Target + 9.9 \quad (26)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.78033$ می‌باشد.

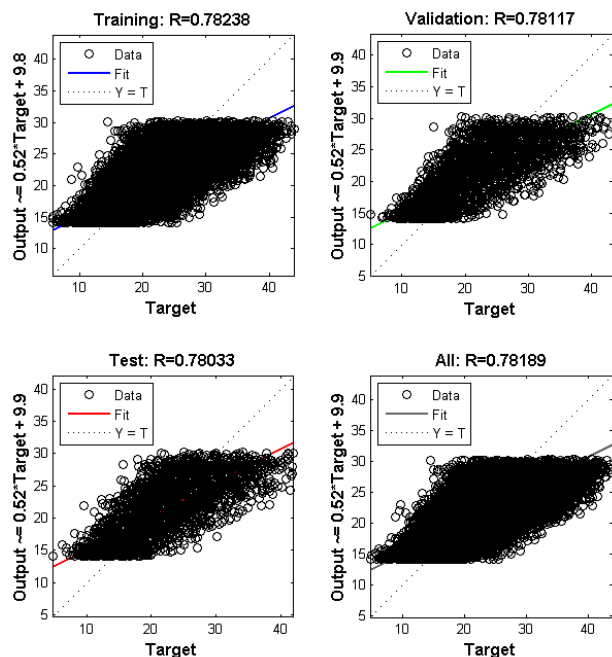
مطابق شکل ۷، نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی، معادله حاصل در اعتبارسنجی برابر با معادله ارائه‌شده در (۲۷) می‌باشد.

$$output = 0.52 * Target + 9.9 \quad (27)$$

میزان رگرسیون در مرحله اعتبارسنجی برابر با $R = 0.78117$ می‌باشد.

مطابق شکل ۷، معادله حاصل در حالت کلی برابر با معادله ارائه‌شده در (۲۸) می‌باشد.

$$output = 0.52 * Target + 9.9 \quad (28)$$



شکل ۷: نمودار رگرسیونی شبکه عصبی پیشنهادی در

تکرار دوم

میزان رگرسیون در مرحله کلی برابر با $R = 0.78189$ می‌باشد. مطابق شکل ۸ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در

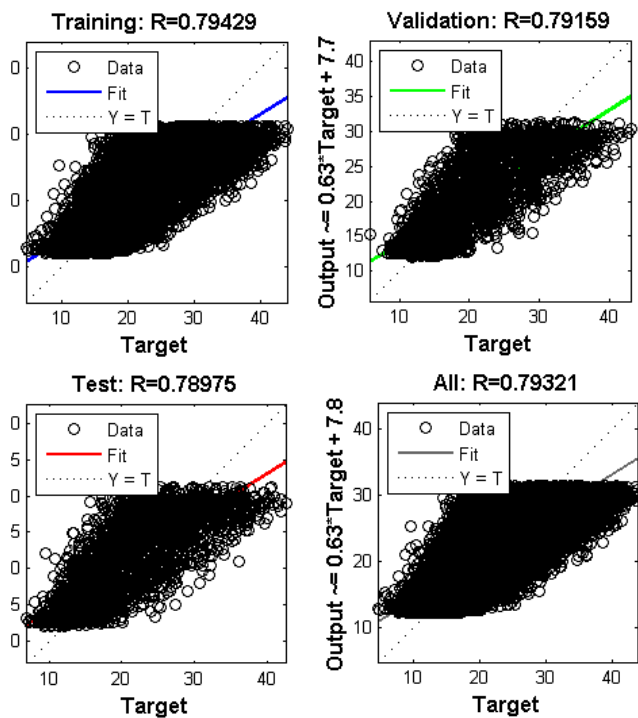
شبکه عصبی پیشنهادی در تکرار سوم، معادله حاصل در آموزش برابر با معادله ارائه‌شده در (۲۹) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (29)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.79313$ می‌باشد.

مطابق شکل ۸، معادله حاصل در تست برابر با معادله ارائه‌شده در (۳۰) می‌باشد.

$$output = 0.63 * Target + 7.9 \quad (30)$$



شکل ۹: نمودار رگرسیونی شبکه عصبی پیشنهادی در

تکرار چهارم

$$output = 0.63 * Target + 7.8 \quad (34)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.78975$ می‌باشد.

مطابق شکل ۹، معادله حاصل در اعتبارسنجی برابر با معادله ارائه‌شده در (۳۵) می‌باشد.

$$output = 0.63 * Target + 7.7 \quad (35)$$

میزان رگرسیون در مرحله اعتبارسنجی برابر با $R = 0.79159$ می‌باشد.

مطابق شکل ۹ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی، معادله حاصله در حالت کلی برابر با معادله ارائه‌شده در (۳۶) می‌باشد.

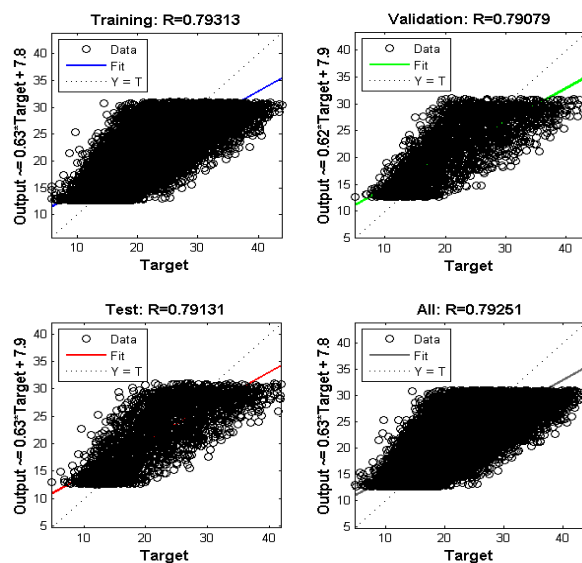
$$output = 0.63 * Target + 7.8 \quad (36)$$

میزان رگرسیون در مرحله کلی برابر با $R = 0.79321$ می‌باشد. مطابق شکل ۱۰، معادله حاصل در آموزش برابر با معادله ارائه‌شده در (۳۷) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (37)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.79131$ می‌باشد.

مطابق شکل ۸، معادله حاصل در اعتبارسنجی برابر با



شکل ۸: نمودار رگرسیونی شبکه عصبی پیشنهادی در

تکرار سوم

معادله ارائه‌شده در (۳۱) می‌باشد.

$$output = 0.62 * Target + 7.9 \quad (31)$$

میزان رگرسیون در مرحله اعتبارسنجی برابر با $R = 0.79079$ می‌باشد.

مطابق شکل ۸، معادله حاصل در حالت کلی برابر با معادله ارائه‌شده در (۳۲) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (32)$$

میزان رگرسیون در مرحله کلی برابر با $R = 0.79251$ می‌باشد. مطابق با شکل ۹ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی در تکرار چهارم، معادله حاصل در آموزش برابر با معادله ارائه‌شده در (۳۳) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (33)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.79429$ می‌باشد.

مطابق شکل ۹، معادله حاصل در تست برابر با معادله ارائه‌شده در (۳۴) می‌باشد.

جدول ۱: نتایج رگرسیون حاصله از مراحل آموزش، تست،

اعتبارسنجی در پنج مرحله آموزش

شماره آزمایش	مرحله آموزش	مرحله تست	مرحله اعتبارسنجی	حالت کلی
اول	۰,۷۹۲۱۳	۰,۷۹۰۰۶	۰,۷۸۹۸	۰,۷۹۱۴۷
دوم	۰,۷۸۲۳۸	۰,۷۸۰۳۳	۰,۷۸۱۱۷	۰,۷۸۱۸۹
سوم	۰,۷۹۳۱۳	۰,۷۹۱۳۱	۰,۷۹۰۷۹	۰,۷۹۲۵۱
چهارم	۰,۷۹۴۲۹	۰,۷۸۹۷۵	۰,۷۹۱۵۹	۰,۷۹۳۲۱
پنجم	۰,۷۹۴۳۹	۰,۷۸۹۸۸	۰,۷۹۱۴۸	۰,۷۹۳۲۸

۵. نتیجه‌گیری

در این مقاله به منظور ورودی، داده‌ها به شبکه عصبی پیشنهادی، با استفاده از نرمال‌سازی استاندارد، نرمال شده‌اند. داده‌های موردنظر در پژوهش در چهار مرحله، آموزش، تست، آزمایش و اعتبارسنجی و در پنج فاز بررسی و ارزیابی شدند. آموزش داده‌ها بر مبنای مدل الگوریتم لونیبرگ-ماد برای لایه اول و تابع خطی برای لایه دوم انجام شد. دلیل استفاده از الگوریتم لونیبرگ-ماد، سرعت همگرایی و کارایی بالاتر آن به دلیل نیفتادن در مینیمم‌های موضعی و کوچکی سطح خطا می‌باشد. در ادامه بهترین ساختار شبکه با تابع تبدیل در نظر گرفته شده و براساس مدل شبکه عصبی پیشنهادی در ۵ مرحله شد. میزان میانگین مربعات خطا و مجذور میانگین مربعات خطا برای مراحل اول تا پنجم به ترتیب عبارتند از: ۱,۵۰۸۸ و ۱,۲۲۸۳ برای آزمایش اول، ۰,۸۸۹۸۴ و ۰,۹۴۳۳۱ برای آزمایش دوم، ۰,۷۴۸۷۷ و ۰,۸۶۵۳۲ برای آزمایش سوم، ۰,۹۲۲۱۱ و ۰,۹۶۰۲۷ و همچنین میزان همگرایی داده‌ها در پیش‌بینی انجام شده در چهار مرحله آموزش به ترتیب ۰,۹۷۷۹۴ برای آزمایش اول، ۰,۹۸۷۸۶ برای آزمایش دوم، ۰,۹۹۱۶۶ برای آزمایش سوم، ۰,۹۸۸۹۹ برای آزمایش چهارم می‌باشد. میزان میانگین مربعات خطا در آزمایش سوم و چهارم به تدریج نسبت به دو مرحله اول افزایش داشته‌است. این مقدار اختلاف در خطای عملکرد و همچنین ضریب تعیین در هر تکرار متفاوت و حاصل گیر افتادن در مینیمم‌های موضعی است.

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.79439$ می‌باشد.

مطابق شکل ۱۰، معادله حاصل در تست برابر با معادله ارائه شده در (۳۸) می‌باشد.

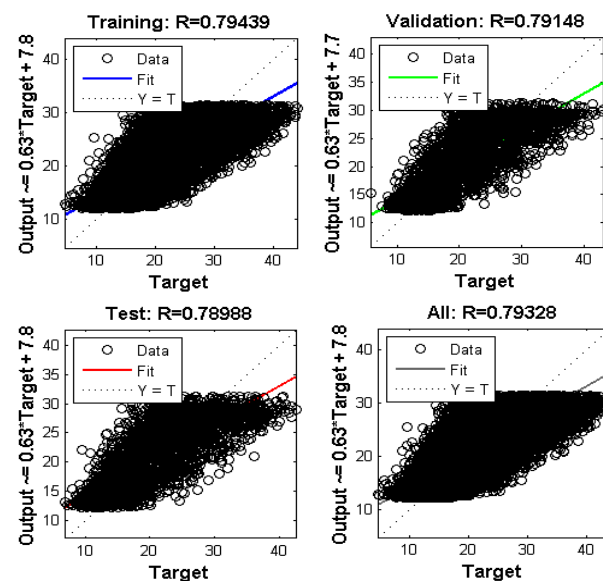
$$output = 0.63 * Target + 7.8 \quad (38)$$

میزان رگرسیون حاصل در مرحله تست برابر با $R = 0.78988$ می‌باشد.

مطابق شکل ۱۰ نمودار رگرسیونی حاصل از تجزیه و تحلیل اطلاعات در شبکه عصبی پیشنهادی، معادله حاصل در اعتبارسنجی برابر با معادله ارائه شده در (۳۹) می‌باشد.

$$output = 0.63 * Target + 7.7 \quad (39)$$

میزان رگرسیون در مرحله اعتبارسنجی برابر با $R = 0.79148$ می‌باشد.



شکل ۱۰: نمودار رگرسیونی شبکه عصبی پیشنهادی در

تکرار پنجم

مطابق شکل ۱۰، معادله حاصل در حالت کلی برابر با معادله ارائه شده در (۴۰) می‌باشد.

$$output = 0.63 * Target + 7.8 \quad (40)$$

میزان رگرسیون در مرحله کلی برابر با $R = 0.79328$ می‌باشد.

1. Zhang Neural Network
2. Routh
3. Neuron
4. Unit
5. Node
6. Input Layer
7. Output Layer
8. Bias
9. Logistic
10. Zhang Function
11. Time-Varying Linear Matrix-Vector Inequality
12. Linear Matrix Inequality
13. Time-Varying Nonlinear Optimization

- [1] L. Jin and Y. Zhang "Discrete-time Zhang neural network for online time-varying nonlinear optimization with application to manipulator motion generation," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 26, no. 7, pp. 1525-1540, 2015.
- [2] D. Guo and et. al., "Design and analysis of two discrete-time ZD algorithms for time-varying nonlinear minimization," *Numerical Algorithms*, vol. 77, pp. 23-36, 2018.
- [3] Y. Zhang and et. al., "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *Journal of Computational and Applied Mathematics*, pp. Vol. 347, pp: 314-329, 2019.
- [4] Y. Wang and J. Xiu, *Nonlinear Programming Theory and Algorithm*, Shaanxi Science and Technology Press, 2008.
- [5] F. Uhlig and Y. Zhang, "Neural Networks for fast and accurate computations of the field of values," *Linear and Multilinear Algebra*, vol. Vol. 13, pp. 1-18, 2019.
- [6] W. Bian and et. al., "Neural network for nonsmooth pseudoconvex optimization with general convex constraints," *Neural Networks*, vol. Vol. 101, pp. 1-14, 2018.
- [7] A. Hosseini, J. Wang and S.M. hosseini, "A recurrent neural network for solving a class of generalized convex optimization problems," *Neural Network*, vol. Vol. 44, pp. 78-86, 2013.
- [8] Y. Zhang, "ZFD formula 4I g SFD_Y applied to future minimization," *Physics Letters A*, vol. Vol. 381, no. No. 19, p. 1677-1681, 2017.
- [9] A. Joze Ashori, Sh. Karimi, "QoS Management Solution in Software Defined Networking using Ryu Controller," *IMPCS*, 2021, 3(1 9-23).
- [10] J. Balakudehi, M. Tahghighi Sharabyan, "Providing a New Approach to Identify and Detect Credit Card Fraud Using ANN-ICA," *IMPCS*, 2022, 3(2), 51-62.