مجله کنترل
جلد ۶، شماره ۱، بهار ۱۳۹۱، صفحه ۵۰-۴۱

# پیاده‌سازی شبکه‌های عصبی راف با یادگیری احتمالاتی جهت شناسایی سیستم‌های غیرخطی

سید محمد جواد آل‌هاشر[۱]، محمد تشنه‌لب[۲]

[۱] دانشجوی دکتری مهندسی کامپیوتر- هوش مصنوعی، گروه کامپیوتر، دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران،
soheil.alehasher@gmail.com
[۲] استاد، قطب علمی کنترل صنعتی، دانشکدهٔ مهندسی برق، گروه کنترل، دانشگاه خواجه نصیر، teshnehlab@kntu.ac.ir

(تاریخ دریافت مقاله ۱۳۹۰/۱۲/۳، تاریخ پذیرش مقاله ۱۳۹۱/۳/۱)

چکیده: در این مقاله یک شبکه عصبی راف بهبود یافته به منظور شناسایی سیستم‌های غیرخطی آشوبی ارائه شده است. شبکه‌های عصبی راف نوعی از ساختارهای عصبی هستند که براساس نرون‌های راف طراحی می‌شوند. یک نرون راف را می‌توان بصورت زوجی از نرون‌ها در نظر گرفت، که به نرون‌های کرانه بالا و کرانه پایین موسوم هستند. رویکرد نرون راف استفاده از محاسبات بازه‌ای در شبکه عصبی را امکان پذیر می‌سازد، بنابراین می‌توان آنرا به عنوان نگرشی نوین در زمینه طراحی شبکه‌های عصبی تلقی نمود. همانند شبکه‌های پرسپترون چندلایه، شبکه‌های عصبی راف نیز می‌توانند با استفاده از الگوریتم پس انتشار خطا مبتنی بر گرادیان نزولی آموزش داده شوند، با این حال این الگوریتم دارای مشکلاتی مانند کمینه محلی است. در این مقاله روش یادگیری با نظارت جدیدی براساس احتمال خطای موثر نرون‌ها، برای آموزش شبکه‌های عصبی ارائه شده است که آنرا یادگیری احتمالاتی می‌نامیم. جهت ارزیابی این تحقیق، کارایی شبکه عصبی راف بهبود یافته و الگوریتم یادگیری پیشنهادی برحسب خطای شناسایی سری‌های زمانی آشوبی مورد بررسی قرار گرفته است.

**کلمات کلیدی:** نرون راف، شبکه عصبی راف، یادگیری احتمالاتی، شناسایی سیستم غیرخطی و سری‌های زمانی آشوبی.

# Implementation of Rough Neural Networks with Probabilistic Learning for Nonlinear System Identification

**Seyyed Mohammad Javad Alehasher, Mohammad Teshnehlab**

**Abstract:** In this paper an improved rough neural network is presented for identification of chaotic system. Rough neural networks are a type of neural stractures that they are designed based on rough neurons. A rough neuron is considered as a pair of neurons that called lower boandry neuron and upper boandry neuron. Rough neuron approach, allows use of interval computing in neural networks, therefore it can be considered as a new opinion in designing neural networks. The same as multilayer perceptron, rough neural networks also can be trained using by back propagation algorithm based on gradient descending, however, this algorithm has problems such as local minima. In this paper, a new supervised learning method based on effective error of neuron is presented for training of neural networks, which it is called probabilistic learning. To evaluate this study, performance of rough neural network improved, and proposed learning algorithm have been examined in terms of error detection of chaotic time series.

**Keywords:** Rough Neural Networks, Probabilistic Learning, Nonlinear System Identification.



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]

[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]
[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]

## ۱- مقدمه

مساله شناسایی از نقطه نظر مدلسازی را می‌توان یک مساله بهینه سازی دانست [۱]. در این گونه از مسائل هدف از شناسایی، تقریب تابع ناشناخته‌ای است که بر سیستم حاکم است. در این مقاله از یک شبکه عصبی راف[1] پیشنهادی به عنوان شناساگر سیستم‌های غیرخطی استفاده می‌شود. شبکه‌های عصبی راف، ساختارهای عصبی هستند که در آن از نرون‌های راف[2] استفاده می‌شود. اصطلاح نرون‌های راف اولین بار توسط لینگراس در سال ۱۹۹۶ در [۲] معرفی شده است. یک نرون راف برحسب کرانه‌های پایین و بالای الگوهایی که معمولا مقادیر بازه‌ای دارند، تعریف می‌شود؛ بنابراین، این نوع از نرون‌ها را نرون‌های راف بازه‌ای[3] نیز می‌گویند. شبکه عصبی طراحی شده از چنین نرون‌هایی ابتدا در پیش بینی حجم ترافیک شهری استفاده شده است [۴-۲]. همچنین از جمله مطالعاتی که در زمینه شبکه‌های عصبی راف انجام شده است، می‌توان به کلاس بندی تصاویر [۵]، سیستم پشتیبان تشخیص پزشکی [۶] و پیش‌بینی شیوع بیماری سل [۷] اشاره کرد.

در این مقاله از یک شبکه عصبی راف با ساختاری مبتنی بر شبکه‌های پیشرو چندلایه استفاده می‌شود که شامل یک لایه ورودی از مقادیر حقیقی، یک لایه پنهان و یک لایه خروجی است که تمامی آنها شامل نرون‌های راف هستند. هر نرون راف زوجی از دو نرون است که یکی مقدار کرانه پایین و دیگر مقدار کرانه بالای سیگنال تولید شده توسط نرون راف را به عنوان خروجی‌های آن از خود عبور می‌دهند؛ بدین ترتیب بین این دو نرون همواره تبادل اطلاعاتی وجود دارد تا مقادیر کرانی هر یک از این دو نرون به درستی تعیین شوند. تبادل این اطلاعات در شبکه عصبی راف معمول، قطعی است؛ به همین منظور، در این مقاله بهبودی با استفاده از تعریف یک ضریب به عنوان نرخ تبادل اطلاعات ارائه خواهد شد. استفاده از این ضریب نه تنها تضمین می‌کند که مقادیر کرانی هر نرون به درستی تعیین شوند، بلکه موجب می‌شود که هر نرون کرانی، نرون کرانه پایین و نرون کرانه بالا، بطور همزمان حاوی اطلاعاتی از هر دوی آنها باشد.

علاوه بر ساختار شبکه عصبی، موضوع مهم دیگر یادگیری شبکه عصبی است. با توجه به اینکه در مسائل شناسایی مقدار متغیر وابسته در دسترس است، برای آموزش شناساگر عصبی می‌توان از یادگیری با نظارت استفاده نمود. از این رو، فرآیند یادگیری در شبکه عصبی راف مانند یک شبکه پرسپترون پیشرو چندلایه می‌تواند با استفاده از الگوریتم پس انتشار خطا مبتنی بر گرادیان نزولی مرتبه اول پیاده‌سازی شود؛ بطوریکه مقدار خطای ایجاد شده در خروجی شبکه با استفاده از این الگوریتم کمینه می‌شود. هر چند که الگوریتم پس انتشار خطا مبتنی بر گرادیان نزولی در آموزش انواع شبکه‌های عصبی پیشرو روشی فراگیر و موفق بوده است با این حال دارای محدودیت‌هایی نظیر کمینه محلی است. در این مقاله رویکردی نوین به آموزش شبکه‌های عصبی براساس احتمال خطای موثر هر نرون ارائه می‌گردد، که در این مقاله آنرا یادگیری احتمالاتی می‌نامیم؛ این رویکرد را می‌توان یک روش یادگیری تقویتی و با نظارت تلقی کرد بطوریکه با کمینه کردن خطای موثر هر یک از پارامتر آموزش‌پذیر، خطای سراسری را کاهش می‌دهد.

در این مقاله برای بررسی کارایی شبکه‌های عصبی راف پیشنهادی به عنوان شناساگر سیستم‌های غیرخطی و همچین روش یادگیری پیشنهادی چندین شبیه‌سازی مختلف انجام شده است. در این آزمایشات از دو سری زمانی آشوبی مکی- گلاس و هنون، به عنوان سیستم‌های غیرخطی نمونه در شبیه‌سازی‌ها استفاده می‌شود. سپس کارایی شبکه عصبی راف پیشنهادی و همچنین روش یادگیری پیشنهادی براساس خطای شناسایی سیستم‌های نمونه، در برابر یک شبکه پرسپترون چند لایه و یک شبکه عصبی راف معمول که با الگوریتم یادگیری پس انتشار خطا مبتنی بر گرادیان نزولی مرتبه اول آموزش داده شده‌اند سنجیده می‌شوند.

سازماندهی این مقاله بدین ترتیب است که ابتدا در بخش دوم مفهوم نرون راف و شبکه عصبی راف معمول ساخته شده از این نرون‌ها به همراه الگوریتم پیشرو و الگوریتم یادگیری آن مطرح می‌گردد. در بخش سوم شبکه عصبی راف پیشنهادی به همراه الگوریتم پیشرو اصلاح شده‌ی آن ارائه می‌شود، و سپس در بخش چهارم الگوریتم یادگیری احتمالاتی، تعریف شده و کاربرد آن برای آموزش پارامترهای شبکه‌های عصبی راف پیشنهادی ارائه می‌شود. در نهایت نتایج حاصل از شبیه‌سازی‌ها در بخش ۵ بطور خلاصه ارائه شده است.

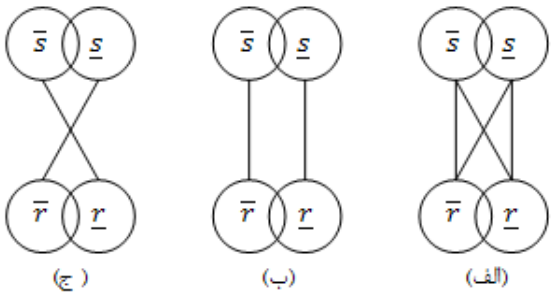
## ۲- نرون‌های راف

شبکه عصبی راف بر اساس نرون‌های راف طراحی می‌شود [۲]. نرون راف $r$ را می‌توان به صورت زوج $(\underline{r},\overline{r})$ از نرون‌ها نگریست، یکی برای کرانه پایین و دیگری برای کرانه بالا که به ترتیب با $\overline{r}$ و $\underline{r}$ نشان داده می‌شود. دو نرون راف از طریق اتصال دوگانه یا چهارگانه، به یکدیگر متصل می‌شوند. شکل (۱) سه نوع اتصال ممکن بین دو نرون راف را نشان می‌دهد؛ همپوشانی بین نرون‌های کرانه پایین و بالا نشان می‌دهد که آنها اطلاعات را با هم مبادله می‌کنند. شکل (۱) (الف) اتصال کامل بین دو نرون راف $r$ و نرون راف $s$ را نشان می‌دهد، که در آن $\overline{r}$ و $\underline{r}$ همزمان به $\overline{s}$ و $\underline{s}$ متصل شده‌اند. اگر نرون راف $r$ بطور کامل به نرون راف $s$ متصل شود، آنگاه چهار اتصال از $r$ به $s$ وجود دارد. در شکل (۱) (ب) و (۱) (ج) تنها دو اتصال از $r$ به $s$ وجود دارد. اگر اتصال بین دو نرون به صورت شکل (۱) (ب)، نرون راف $r$ فعالیت نرون راف $s$ را تحریک می‌کند و اگر اتصال بین دو نرون به صورت شکل (۱) (ج) نرون راف $r$ فعالیت نرون راف $s$ را مهار کند.

[1] Rough Neural Network
[2] Rough Neuron
[3] Interval-based Rough Neuron



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]




شکل ۱- سه نوع مختلف از اتصالات بین دو نرون راف، الف) اتصال چهارگانه ب) اتصال تحریک ج) اتصال مهار [۲]

فرض کنید که نرون‌ها راف در ساختار شبکه به طور کامل به یکدیگر متصل شوند یعنی شبکه شامل اتصالات تحریک و مهار باشد؛ مجموع وزندار شده ورودی‌ها به نرون راف $i$- ام از لایه $l$- ام در گام $t$- ام با استفاده از رابطه زیر محاسبه می‌شود:

$$net_i^l(t) = \sum_{i,j} w_{ij}^l(t) \times o_j^{l-1}(t)$$
$$i \in \{\underline{i}, \overline{i}\}, j \in \{\underline{j}, \overline{j}\}, l = 1, \dots, L \quad (۱)$$

که $i$ و $j$ اندیس نرون‌های کرانی هر یک از نرون‌های راف را نشان می‌دهند و $L$ تعداد لایه‌های فعال در شبکه است. خروجی نرون راف $i$- ام با استفاده از تابع انتقال $f$ به صورت زیر محاسبه می‌شود:

$$o_{\overline{i}}^l(t) = max\left(f\left(net_{\overline{i}}^l(t)\right), f\left(net_{\underline{i}}^l(t)\right)\right) \quad (۲)$$

$$o_{\underline{i}}^l(t) = min\left(f\left(net_{\overline{i}}^l(t)\right), f\left(net_{\underline{i}}^l(t)\right)\right) \quad (۳)$$

بنابراین خروجی نرون راف $i$- ام در لایه $l$- ام شبکه را می‌توان با مقدار بازه‌ای زیر نشان داد:

$$o_i^l(t) = \left[o_{\underline{i}}^l(t), o_{\overline{i}}^l(t)\right] \quad (۴)$$

استفاده از عملگرهای $min, max$ در روابط (۲) و (۳) تضمین می‌کند که همواره مقدار خروجی نرون کرانه بالا از خروجی نرون کرانه پایین بزرگتر باشد؛ تابع انتقال $f$ در این روابط می‌تواند یک تابع سیگموئیدی دو قطبی باشد که بصورت زیر تعریف می‌شود:

$$f(net) = \frac{1 - e^{-net}}{1 + e^{-net}} \quad (۵)$$

که در آن $net$ مجموع وزندار شده ورودی‌ها است. در صورت استفاده از نرون‌های راف در لایه آخر، خروجی نهایی شبکه را می‌توان با استفاده از رابطه زیر به راحتی محاسبه کرد:

$$o_i^L = \frac{o_{\overline{i}}^L + o_{\underline{i}}^L}{2} \quad (۶)$$

اگر دو نرون راف به طور جزیی متصل شده باشند، آنگاه مهار یا تحریک طبیعی اتصال به طور پویا با توجه به وزن‌های اتصال تعیین می‌شود. اگر یک اتصال جزیی از نرون راف $r$ به دیگر نرون راف $s$ تحریک فرض شود و $w_{\overline{rs}} < 0$ و $w_{\underline{rs}} < 0$ آنگاه اتصال بین نرون راف r و s با تخصیص $w_{\overline{rs}} = w_{\underline{rs}}$ و $w_{\underline{rs}} = w_{\underline{rs}}$ از تحریک به مهار تغییر می‌کند؛ پیوندهای $(\underline{r}, \underline{s})$ و $(\overline{r}, \overline{s})$ غیر فعال می‌شوند، در حالیکه پیوندهای $(\underline{r}, \overline{s})$ و $(\overline{r}, \underline{s})$ فعال می‌شوند. همچنین، اگر یک اتصال جزیی از نرون راف $r$ به دیگر نرون راف $s$ مهار فرض شود، و $w_{\overline{rs}} > 0$ و $w_{\underline{rs}} > 0$، آنگاه اتصال بین نرون راف $r$ و $s$ با اختصاص $w_{\overline{rs}} = w_{\overline{rs}}$ و $w_{\underline{rs}} = w_{\underline{r}\overline{s}}$ از مهار به تحریک تغییر می‌کند؛ پیوند $(\underline{r}, \overline{s})$ و $(\overline{r}, \underline{s})$ غیر فعال می‌شوند، در حالیکه پیوند $(\underline{r}, \underline{s})$ و $(\overline{r}, \overline{s})$ فعال می‌شوند.

از آنجا که مقادیر مطلوب برای نمونه‌ها معلوم هستند، وزن‌های اتصالات بین نرون‌ها در شبکه در یک پروسه یادگیری با نظارت با استفاده از الگوریتم پس انتشار خطا به عقب تنظیم می‌شوند. در جریان یادگیری، خطای خروجی برای تنظیم وزن‌های اتصالات از یک لایه به لایه پیشین به عقب منتشر می‌شود. در اینجا با استفاده از رابطه یادگیری قاعده دلتای تعمیم یافته در گام $t$- ام وزن اتصال بین نرون راف $i$- ام و نرون $j$- ام در لایه فعال $l$- ام شبکه به صورت زیر اصلاح می‌شود:

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \eta \times e(t) \times o_j^{l-1}(t) \times f'\left(net_i^l(t)\right) \quad (۷)$$

که $i \in \{\underline{i}, \overline{i}\}, j \in \{\underline{j}, \overline{j}\}$ اندیس‌هایی برای نشان دادن نرون‌های کرانی هستند؛ در این رابطه $f'$ مشتق تابع انتقال برحسب مجموع وزندار شده $net_i^l$ از ورودی‌های $o_j^{l-1}(t)$ است، و $\eta$ نرخ آموزش است که سرعت یادگیری را نشان می‌دهد.

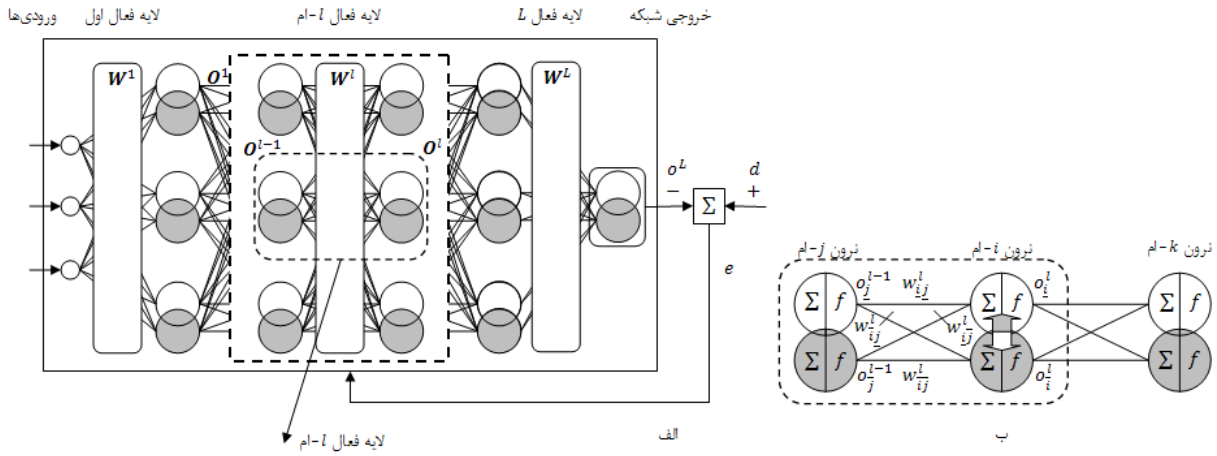
## ۳- شبکه عصبی راف پیشنهادی

در این بخش، درمقایسه با شبکه عصبی راف، طرح جدیدی از شبکه‌های عصبی راف پیشنهادی بر پایه مفاهیم نرون‌های راف پیشنهاد خواهد شد. تبادل اطلاعات بین نرون‌های کرانه پایین و کرانه بالای یک نرون راف در شبکه عصبی راف معمول از طریق عملگرهای $min$, $max$ صورت می‌گیرد، که تبادلی قطعی است؛ این موضوع موجب می‌شود که هر نرون کرانی برخی اطلاعات مفیدش را از دست دهد. به همین منظور در این مقاله با تعریف ضریبی به عنوان نرخ تبادل اطلاعات بین نرون‌های کرانی، بهبودی روی شبکه‌های عصبی راف ارائه خواهد شد. همچنین تفسیر ارائه شده از چگونگی اتصالات تحریک و مهار در بخش ۲ الزاماتی را برای پیاده‌سازی شبکه موجب می‌شود، که به همین منظور در این مقاله روش جدیدی برای این اتصالات پیشنهاد شده است.

همانند شبکه‌های عصبی راف معمول، شبکه عصبی راف پیشنهادی نیز از نرون‌های راف مقدار کرانی ساخته می‌شود، و بنابراین ساختاری مشابه با آنها دارد؛ شکل (۲) (الف) ساختار یک شبکه عصبی راف را به همراه لایه‌های پنهان آن نشان می‌دهد. همانطور که در شکل (۲) نشان داده شده است در مساله شناسایی سیستم‌های غیرخطی به وسیله شبکه



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]




شکل ۲- الف) ساختار شبکه عصبی راف با $L$ لایه فعال ب) برشی از لایه فعال $l$- ام

عصبی راف ورودی‌ها و مقدار مطلوب متناظر آنها به جای الگوهای راف مقادیر حقیقی هستند؛ هر لایه پنهان شامل تعدادی نرون راف است که توسط اتصالات چهارگانه به نرون‌های راف در لایه پسین و پیشین خود متصل شده است؛ لایه خروجی نیز شامل یک نرون راف است که خروجی آن توسط رابطه (۶) محاسبه می‌شود.

## ۳-۱- الگوریتم پیشرو برای شبکه عصبی راف پیشنهادی

مطابق شکل (۲) (ب) فرض کنید که نرون راف $j$- ام از لایه $l-1$- ام از طریق مجموعه وزن‌های چهارگانه $w_{ij}^l =$ $\left\{w_{\underline{ij}}^l, w_{\underline{ij}}^{\overline{l}}, w_{\overline{ij}}^{\underline{l}}, w_{\overline{ij}}^{\overline{l}}\right\}$ به نرون راف $i$- ام از لایه $l$- ام متصل شده است. در اینجا بر خلاف تفسیر پیشین الگوریتم پیشرو، مجموع ورودی‌های وزندار شده نرون‌های کرانی توسط اتصالات مهار از مجموع ورودی‌های وزندار شده نرون‌های کرانی توسط اتصالات تحریک کم می‌گردد. بنابراین با تعمیم رابطه (۱) مجموع ورودی‌های وزندار شده نرون راف $i$- ام لز لایه $l$- ام در گام $t$- ام را می‌توان به صورت زیر محاسبه کرد:

$$net_{\underline{i}}^l = \sum_{j=1}^{n_{l-1}} w_{\underline{ij}}^l o_{\underline{j}}^{l-1} - \sum_{j=1}^{n_{l-1}} w_{\overline{ij}}^l o_{\overline{j}}^{l-1}, \quad for\ i = 1, \dots, n_l \qquad (۸)$$

و

$$net_{\overline{i}}^l = \sum_{j=1}^{n_{l-1}} w_{\overline{ij}}^l o_{\overline{j}}^{l-1} - \sum_{j=1}^{n_{l-1}} w_{\underline{ij}}^l o_{\underline{j}}^{l-1}, \quad for\ i = 1, \dots, n_l \qquad (۹)$$

که در آنها $n_l$ تعداد نرون‌ها راف در لایه $l$- ام و $o_j^{l-1} = \left[o_{\underline{j}}^{l-1}(t), o_{\overline{j}}^{l-1}(t)\right]$ ورودی نرون راف است. خروجی هر یک از نرون‌های کرانی نرون راف را می‌توان به صورت زیر محاسبه:

$$o_{\underline{i}}^l = (1-\gamma) f\left(net_{\underline{i}}^l\right) + \gamma f\left(net_{\overline{i}}^l\right), \quad for\ i = 1, \dots, n_l \qquad (۱۰)$$

و

$$o_{\overline{i}}^l = (1-\gamma) f\left(net_{\overline{i}}^l\right) + \gamma f\left(net_{\underline{i}}^l\right), \quad for\ i = 1, \dots, n_l \qquad (۱۱)$$

که در دو رابطه اخیر $f$ تابع انتقال تعریف شده در رابطه (۵) بوده و $\gamma$ نرخ تبادل اطلاعات بین نرون‌های کرانی نرون راف است که برای تابع انتقال سیگموئیدی به صورت زیر بدست می‌آید:

$$\gamma = \begin{cases} 1 - \dfrac{\left|o_{\overline{i}}^l - o_{\underline{i}}^l\right|}{2} & if \quad o_{\overline{i}}^l < o_{\underline{i}}^l \\ \emptyset & if \quad o_{\overline{i}}^l \geq o_{\underline{i}}^l \end{cases} \qquad (۱۲)$$

نرخ تبادل اطلاعات $\emptyset \leq \gamma \leq 1$ نه تنها تضمین می‌کند که خروجی نرون کرانه بالا همواره بزرگتر از خروجی نرون کرانه پایین باشد، بلکه تضمین می‌کند که هر نرون کرانی برخی اطلاعات مفید خود را حفظ نماید؛ توجه داشته باشید که نرخ تبادل اطلاعات برای تابع خطی در لایه خروجی چنانچه $o_{\overline{i}}^l < o_{\underline{i}}^l$ باشد برابر با $\gamma \geq 0.5$ و در غیر اینصورت برابر با صفر در نظر گرفته می‌شود. در نهایت خروجی شبکه از رابطه (۶) بدست می‌آید.

## ۴- یادگیری احتمالاتی

در این بخش روش یادگیری جدیدی در آموزش شبکه‌های عصبی موسوم به یادگیری احتمالاتی ارائه می‌شود. در منظر اول این روش ممکن است تا حدودی شبیه به یادگیری بولتزمن در شبیه‌سازی ذوب فلز [۸] و ماشین بولتزمن [۹]، و همچنین الگوریتم‌های یادگیری آلوپکس [۱۰-۱۲] به نظر رسد؛ اما فلسفه این روش از اساس با روش‌های یاد شده تفاوت دارد و علاوه بر اینکه درک آن راحت است، نیازی به پارامتر دما ندارد. فرض کنید $e$ خطای خروجی شناساگر برحسب مقدار مطلوب باشد که از رابطه زیر محاسبه می‌شود:



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]

$$error = desired - output \quad (۱۳)$$

که در آن $desired$ و $output$ به ترتیب مقادیر مطلوب متناظر ورودی‌های شناساگر و خروجی شناساگر هستند؛ بدیهی است زمانی خطای خروجی صفر است که شناساگر مقدار مطلوب را به درستی شناسایی کرده باشد، در غیر اینصورت شناساگر دارای خطای غیر صفری است که از پارامترهای آن ناشی می‌شود؛ با توجه به رابطه (۸) رابطه زیر را می‌توان بدست آورد:

$$\begin{cases} error < \emptyset & if \;\; desired < output \\ error > \emptyset & if \;\; desired > output \\ error = \emptyset & if \;\; desired = output \end{cases} \quad (۱۴)$$

از رابطه (۱۴) به وضوح می‌توان دید که خطا، زمانی منفی است که خروجی شناساگر از مقدار مطلوب بزرگتر است بنابراین انتظار این است که با کاهش عمومی مقادیر پارامترهای شناساگر، خروجی آن کاهش یابد؛ و بر عکس خطای شناساگر زمانی مثبت است که خروجی شناساگر از مقدار مطلوب کوچکتر باشد بنابراین انتظار این است که با افزایش عمومی مقادیر پارامترهای شناساگر، خروجی آن افزایش یابد؛ از این رو با توجه به علامت خطا سه تفسیر زیر را می‌توان به عنوان قانون یادگیری عمومی در شناساگرها برای کاهش خطا در نظر گرفت:

۱- چنانچه علامت خطای خروجی شناساگر برحسب مقدار مطلوب منفی است، برای کاهش خروجی بایستی پارامترهای شناساگر بطور عمومی تضعیف گردند.
۲- چنانچه علامت خطای خروجی شناساگر برحسب مقدار مطلوب مثبت است، برای افزایش خروجی بایستی پارامترهای شناساگر بطور عمومی تقویت گردند.
۳- در غیر اینصورت ، یعنی مقدار خطای صفر، مقادیر پارامترهای شناساگر تغییر نمی‌کنند.

حال فرض کنید شناساگر یک شبکه عصبی با پارمترهای آموزش پذیر $\theta$ باشد؛ احتمال خطای موثر پارامتر $\theta$ را می‌توان از رابطه زیر محاسبه نمود:

$$p(\theta) = \frac{1}{1 + e^{-g.e_\theta}} \quad (۱۵)$$

که در آن $g$ بهره تابع احتمال است، و $e_\theta$ خطای موثری پارامتر $\theta$ است. چنانچه $e_\theta = \emptyset$ باشد، $p(\theta) = 0.5$ است و در نتیجه تغییر پارامتر $\Delta\theta = \emptyset$ می‌باشد؛ در غیر اینصورت با توجه به مقدار $p(\theta)$ جهت تغییرات مشخص می‌گردد. اگر $e_\theta < \emptyset$ باشد، $\emptyset \le p(\theta) < 0.5$ بوده و پارامتر $\theta$ بایستی به احتمال $p(\theta) < u^-$ به اندازه $-\Delta\theta$ کاهش یابد که $u^-$ عددی تصادفی با توزیع یکنواخت در بازه $[0,0.5]$ است؛ همچنین اگر $e_\theta > \emptyset$ باشد، $0.5 < p(\theta) \le 1$ بوده و پارامتر $\theta$ بایستی به احتمال $p(\theta) > u^+$ به اندازه $+\Delta\theta$ افزایش یابد که $u^+$ عددی تصادفی با توزیع یکنواخت در بازه $[0.5,1]$ است؛ در غیر اینصورت پارامتر $\theta$ بدون تغییر باقی می‌ماند. فرآیند بالا را می‌توان به عنوان آموزش محلی پارامتر $\theta$ تعبیر نمود؛ بطور خلاصه قانون یادگیری احتمالاتی برای پارامتر آموزش‌پذیر $\theta$ با رابطه زیر بیان می‌شود:

$$\begin{cases} \theta(new) = \theta(old) - \Delta\theta \;\; if \;\; p(\theta) < u^- \\ \theta(new) = \theta(old) + \Delta\theta \;\; if \;\; p(\theta) > u^+ \\ \theta(new) = \theta(old) \qquad \text{درغیراینصورت} \end{cases} \quad (۱۶)$$

که در آن $\Delta\theta$ مقدار تغییرات پارامتر $\theta$ با گامی مناسب است.

## ۴-۱- الگوریتم یادگیری برای شبکه عصبی راف پیشنهادی

شبکه عصبی راف بهبود یافته براساس الگوریتم یادگیری احتمالاتی که در بخش ۳ معرفی گردید، آموزش داده می‌شود؛ برای این منظور بایستی تابع انرژی زیر با استفاده از اصلاح وزنهای شبکه به عنوان پارامترهای آموزش‌پذیر، کمینه شود:

$$E(t) = \frac{1}{2}\big(e(t)\big)^2 = \frac{1}{2}\big(d(t) - o^L(t)\big)^2 \quad (۱۷)$$

که در آن $e(t)$ خطای خروجی شبکه در گام $t$- ام است، و $d(t)$، $o^L(t)$ به ترتیب مقدار مطلوب در گام $t$- ام و خروجی شبکه هستند که $L$ اندیس لایه خروجی را نشان می‌دهد. خطای موثری که در این مقاله استفاده می‌شود، خطاهای محلی نرون راف است، یعنی:

$$e_{w_{ij}^l}(t) = \delta_i^l(t), \qquad i \in \{\underline{i}, \bar{i}\}, j \in \{\underline{j}, \bar{j}\} \quad (۱۸)$$

بنابراین احتمال خطای موثر $e_{w_{ij}^l}$ برای وزن $w_{ij}^l$ را می‌توان به صورت زیر بازنویسی کرد:

$$p\left(w_{ij}^l(t)\right) = \frac{1}{1 + e^{-g.\delta_i^l(t)}}, i \in \{\underline{i}, \bar{i}\}, j \in \left\{\underline{j}, \bar{j}\right\} \quad (۱۹)$$

با بازنویسی قانون یادگیری که در رابطه (۱۶) بیان گردید، وزن‌های شبکه را می‌توان به صورت زیر به روزرسانی کرد:

$$\begin{cases} w_{ij}^l(t+1) = w_{ij}^l(t) - \Delta w_{ij}^l(t) \; if \; p\left(w_{ij}^l(t)\right) < u^- \\ w_{ij}^l(t+1) = w_{ij}^l(t) + \Delta w_{ij}^l(t) \; if \; p\left(w_{ij}^l(t)\right) > u^+ \\ w_{ij}^l(t+1) = w_{ij}^l(t) \qquad \text{درغیراینصورت} \end{cases} \quad (۲۰)$$

برای $i \in \{\underline{i}, \bar{i}\}, j \in \{\underline{j}, \bar{j}\}$ ، که در رابطه اخیر $\Delta w_{ij}^l(t)$ تغییرات وزن در گام $t$- ام است، که به صورت زیر محاسبه می‌شود:

$$\Delta w_{ij}^l(t) = \beta o_j^{l-1}, \qquad i \in \{\underline{i}, \bar{i}\}, j \in \{\underline{j}, \bar{j}\} \quad (۲۱)$$



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]

که $o_j^{l-1} = \left[o_{\underline{j}}^{l-1}, o_{\bar{j}}^{l-1}\right]$ ورودی نرون راف $i$- ام است، و $0 < \beta \leq 1$ نرخ آموزش می‌باشد. در رابطه (۱۸) $\delta_i^l(t),\ i \in \{\underline{i}, \bar{i}\}$ خطاهای محلی برای نرون راف $i$- ام از لایه $l$- ام در گام $t$- ام است و از رابطه زیر محاسبه می‌شود:

$$\delta_i^l(t) = \partial E(t)/\partial net_i^l(t), \qquad i \in \{\underline{i}, \bar{i}\} \tag{۲۲}$$

با محاسبه مشتقات جزیی در رابطه اخیر خطاهای محلی برای هر یک از نرون‌های کرانی ( نرون کرانه پایین و نرون کرانه بالا) به صورت زیر بدست می‌آیند:

$$\begin{aligned}\delta_{\underline{i}}^l(t) = &\left(\delta_{\underline{k}}^{l+1}(t)\left((1-\gamma)w_{\underline{ki}}^{l+1} - \gamma w_{\underline{ki}}^{l+1}\right)\right.\\ &\left.+\delta_{\bar{k}}^{l+1}(t)\left((1-\gamma)w_{\bar{k}\underline{i}}^{l+1} - \gamma w_{\underline{k}\underline{i}}^{l+1}\right)\right) f'\left(net_{\underline{i}}^l(t)\right)\end{aligned} \tag{۲۳}$$

$$\begin{aligned}\delta_{\bar{i}}^l(t) = &\left(\delta_{\bar{k}}^{l+1}(t)\left((1-\gamma)w_{\bar{i}\bar{k}}^{l+1} - \gamma w_{\underline{i}\bar{k}}^{l+1}\right)\right.\\ &\left.+\delta_{\underline{k}}^{l+1}(t)\left((1-\gamma)w_{\underline{ki}}^{l+1} - \gamma w_{\bar{k}\underline{i}}^{l+1}\right)\right) f'\left(net_{\bar{i}}^l(t)\right)\end{aligned} \tag{۲۴}$$

که در این روابط $f'\left(net_i^l(t)\right)$، مشتق تابع انتقال برای نرون راف $i \in \{\underline{i}, \bar{i}\}$ می‌باشد. خطاهای محلی را می‌توان به طور تکراری محاسبه نمود؛ همچین خطاهای محلی برای نرون راف $i$- ام در لایه آخر را می‌توان به صورت زیر بدست آورد:

$$\delta_{\underline{i}}^L(t) = \frac{1}{2}e(t)\left((1-\gamma)f'\left(net_{\underline{i}}^L\right) + \gamma f'\left(net_{\bar{i}}^L\right)\right) \tag{۲۳}$$

$$\delta_{\bar{i}}^L(t) = \frac{1}{2}e(t)\left((1-\gamma)f'\left(net_{\underline{i}}^L\right) + \gamma f'\left(net_{\bar{i}}^L\right)\right) \tag{۲۴}$$

که در آن $L$ اندیس لایه خروجی است.

## ۵- نتایج شبیه‌سازی

در این بخش کارایی شبکه عصبی راف بهبود یافته و همچنین الگوریتم یادگیری احتمالاتی جهت شناسایی سری‌های زمانی آشوبی بخش ۵-۱ و ۵-۲ به عنوان سیستم‌های غیرخطی نمونه، براساس خطای شناسایی مورد بررسی قرار می‌گیرد. برای این منظور شبکه‌های عصبی زیر پیاده‌سازی شده‌اند:

- شبکه‌های پرسپترون چند لایه پیشرو $(MLP)$
- شبکه‌های عصبی راف مدل لینگراس $(RMLP)$
- شبکه عصبی راف پیشنهادی $(PRMLP)$

که هر یک از این شبکه‌ها با استفاده از روش‌های یادگیری زیر آموزش داده می‌شوند:

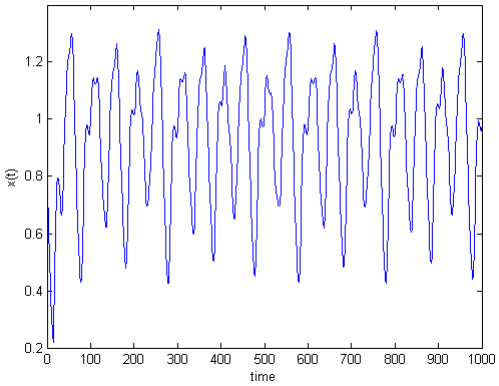
- الگوریتم پس انتشار خطا مبتنی بر گرادیان نزولی مرتبه اول $(BP)$
- الگوریتم یادگیری احتمالاتی مبتنی بر خطای محلی $(PB)$

### ۵-۱- سری زمانی آشوبی مکی- گلاس

سری زمانی مکی- گلاس که در شکل (۳) نشان داده شده است، اولین بار توسط مکی و گلاس برای پیش بینی تولید گلوبل سفید خون ارائه شده است. این سری زمانی یک رابطه دیفرانسیل تاخیر زمانی است که به صورت زیر تعریف می‌شود:

$$\dot{x}(t) = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \tag{۲۷}$$

که در آن به ازای $a = 0.2, b = 0.1, c = 10$ سری آشوبی است و $\tau = 17$ می‌باشد.




شکل ۳- سری زمانی آشوبی مکی- گلاس

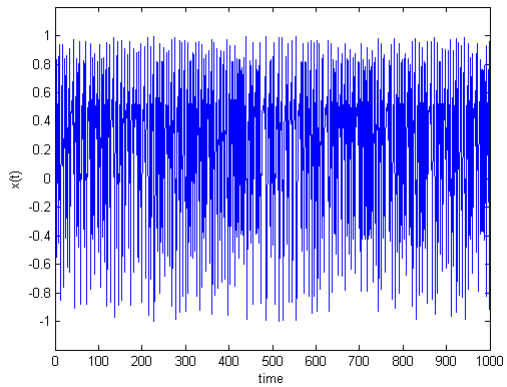
### ۵-۲- نگاشت هنون

نگاشت هنون که در شکل (۴) نشان داده شده است، یک سیستم پویای زمان گسسته ۲- بعدی است. این نگاشت توسط هنون به عنوان مدل ساده شده‌ای از بخش پوانکره مدل لورنز معرفی شده است. بطور ریاضی، نگاشت هنون با معادلات زیر تعریف می‌شود:

$$x_{t+1} = y_t + 1 - ax_t^2 \tag{۲۸}$$

$$y_{t+1} = bx_t^2 \tag{۲۹}$$

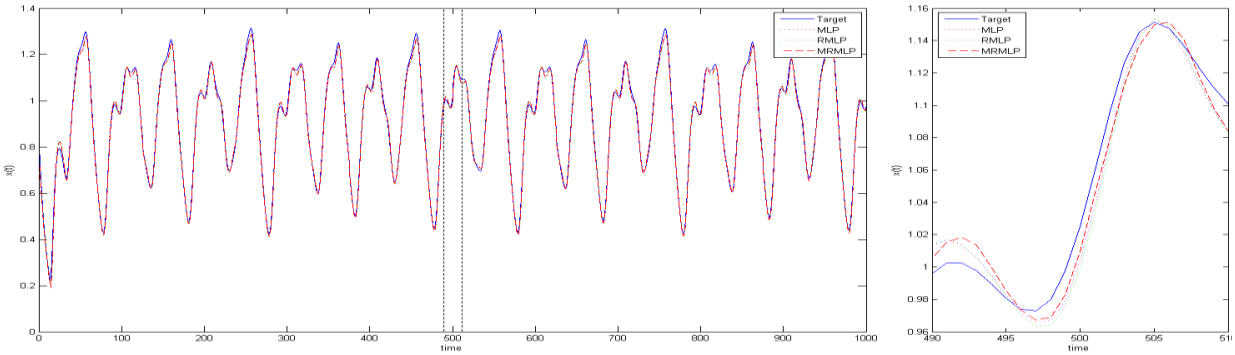
نگاشت وابسته به دو پارامتر $a, b$ است که به ازای مقادیر متعارف $a = 1.4$ و $b = 0.3$ سری آشوبی است.




شکل ۴- سری زمانی آشوبی هنون



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]




شکل ۵- شناسایی سری زمانی آشوبی مکی- گلاس توسط شبکه‌های عصبی راف با ۳ نرون در لایه پنهان

## ۵-۳- فرضیات شبیه‌سازی‌ها

در مسائل شناسایی غیرخطی، سری‌های زمانی آشوبی را می‌توان به صورت تابع $x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-k})$ فرض کرد، بنابراین بردار $[x_{t-1}, x_{t-2}, \dots, x_{t-k}]$ ورودی شبکه عصبی را شکل می‌دهد در حالیکه $x_t$ مقدار مطلوب متناظر آن است. در شبیه‌سازی‌ها $k$ برای هر دو سری زمانی آشوبی هنون و مکی- گلاس برابر با ۴ در نظر گرفته شده است و هر سری در بازه زمانی صفر تا ۱۰۰۰ مورد بررسی قرار می‌گیرد؛ بنابراین مجموعه داده‌ها برای هر یک از این سری‌ها شامل ۱۰۰۰ بردار ۴- بعدی است که ۷۵٪ آن برای آموزش و ۲۵٪ دیگر آن برای آزمایش شبکه‌های عصبی استفاده می‌شوند. هر یک از شبکه عصبی دارای دو لایه فعال است که شرایط اولیه تمامی آنها از قبیل مقادیر اولیه وزن‌ها، یکسان در نظر گرفته می‌شوند؛ مقادیر اولیه وزن‌ها برای تمامی این شبکه‌ها بطور تصادفی در بازه $[-1,1]$ تولید شده است. همچنین تمامی نرون‌ها در لایه پنهان از توابع انتقال سیگموئید دو قطبی استفاده می‌کنند؛ تعداد دوره‌های آموزشی برای هر شبکه ۱۰۰ دوره است، و نرخ‌های آموزش در یادگیری پس انتشار و یادگیری احتمالاتی به ترتیب برابر با $\eta = 0.1$ و $\beta = 0.01$ هستند؛ و مقدار بهره در الگوریتم یادگیری احتمالاتی $g = 10$ می‌باشد. جداول، شامل اطلاعاتی از قبیل نام اختصاری شبکه‌ها، ساختار آنها (تعداد لایه‌ها، تعداد نرون‌ها در لایه پنهان، تعداد اتصالات موجود در شبکه)، و مقدار خطای شناسایی است. در این آزمایشات دو رویکرد مد نظر بوده است: در رویکرد اول تعداد نرون‌ها در تمامی شبکه‌ها برابر در نظر گرفته می‌شود بطوریکه لایه پنهان هر شبکه شامل ۳ نرون (نرون راف) باشد؛ در رویکرد دوم تعداد اتصالات در تمامی شبکه‌ها برابر در نظر گرفته می‌شود بطوریکه لایه پنهان در شبکه‌های عصبی- راف شامل ۲ نرون راف و برای شبکه پرسپترون شامل ۵ نرون می‌باشد.

## ۵-۴- شبیه‌سازی اول: مقایسه شبکه عصبی راف پیشنهادی با دیگر شبکه‌های عصبی

در این شبیه‌سازی تمامی شبکه‌های عصبی با استفاده از الگوریتم پس انتشار مبتنی بر گرادیان نزولی مرتبه اول آموزش داده می‌شوند. نتایج حاصل از انجام این آزمایش برای شناسایی سری‌های زمانی آشوبی مکی گلاس و هنون به ترتیب در جداول ۱ و ۲ بر حسب خطای شناسایی خلاصه شده است.

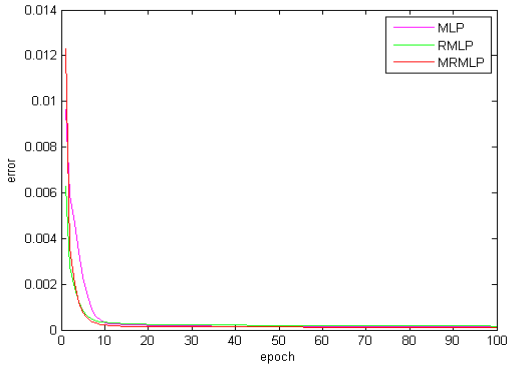
نتایج مندرج در جداول ۱ و ۲ نشان می‌دهد، شبکه عصبی راف بهبود یافته عملکرد مناسبتری نسبت به شبکه عصبی راف معمول در شناسایی سری‌های زمانی آشوبی دارد؛ هرچند شبکه پرسپترون قادر است سری زمانی آشوبی مکی- گلاس را به خوبی شناسایی کند با این حال بر خلاف شبکه‌های عصبی راف این شبکه قادر به شناسایی مناسبی از سری زمانی آشوبی هنون نیست. نمودارهای شناسایی و همگرایی برای شبکه‌های عصبی- راف و پرسپترون که شامل ۳ نرون در لایه پنهان هستند، برای سری زمانی مکی- گلاس به ترتیب در شکل‌های ۵ و ۶، و برای سری زمانی هنون به ترتیب در شکل‌های ۷ و ۸ نشان داده شده‌اند. از نمودارهای همگرایی که در شکل‌های ۶ و ۸ نشان داده شده است، به وضوح می‌توان دید که روند همگرایی شبکه‌های عصبی راف نسبت به شبکه پرسپترون تا حدودی سریعتر است.

جدول ۱- نتایج شبیه‌سازی اول برای سری زمانی مکی- گلاس براساس الگوریتم پس انتشار

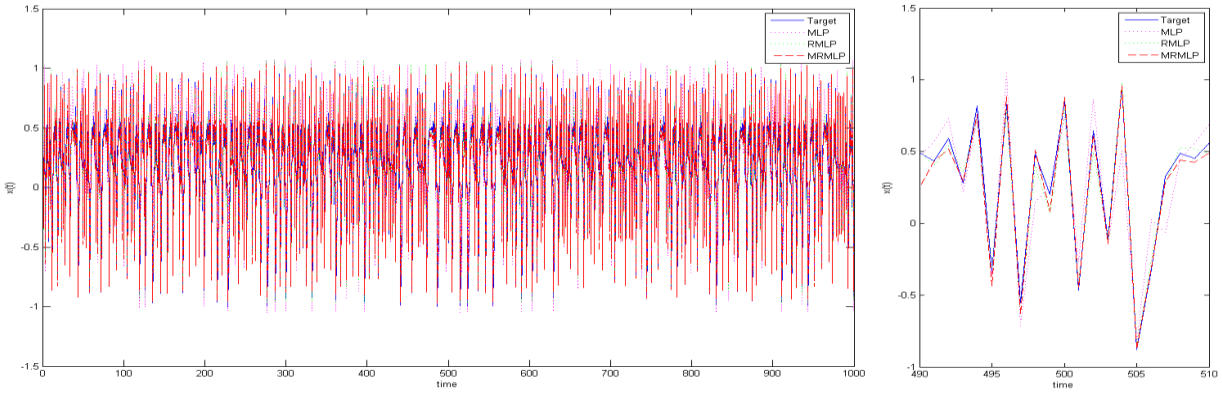
| | تعداد نرون‌ها | تعداد اتصالات | خطای شناسایی | |
|---|---|---|---|---|
| | | | آموزش | آزمایش |
| *MLP* | ۳ | ۱۵ | $0.2223\times10^{-3}$ | $0.2352\times10^{-3}$ |
| | ۵ | ۲۵ | $0.2675\times10^{-3}$ | $0.2797\times10^{-3}$ |
| *RMLP* | ۳ | ۳۶ | $0.2472\times10^{-3}$ | $0.2488\times10^{-3}$ |
| | ۲ | ۲۴ | $0.2896\times10^{-3}$ | $0.2874\times10^{-3}$ |
| *Prposed RMLP* | ۳ | ۳۶ | $0.1815\times10^{-3}$ | $0.1911\times10^{-3}$ |
| | ۲ | ۲۴ | $0.2755\times10^{-3}$ | $0.2821\times10^{-3}$ |



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]



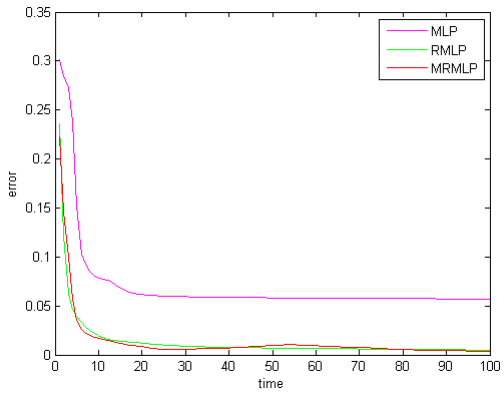

شکل ۶- نمودار همگرایی شبکه‌های عصبی راف در مقایسه با شبکه پرسپترون برای سری زمانی آشوبی مکی- گلاس



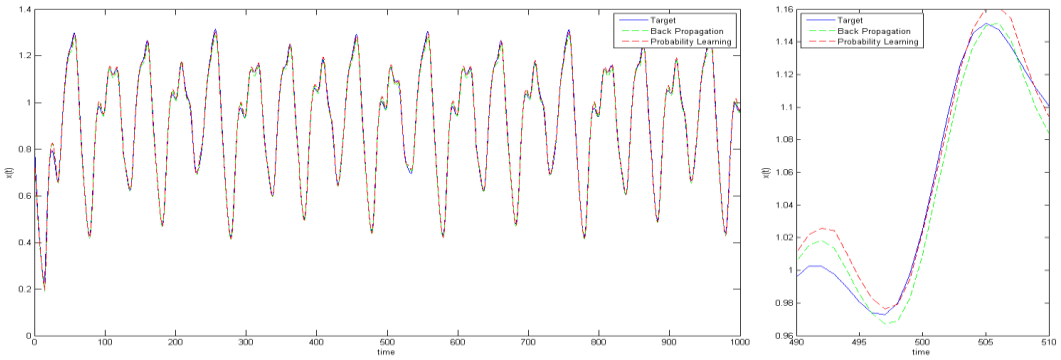

شکل ۷- شناسایی سری زمانی آشوبی هنون توسط شبکه‌های عصبی راف با ۳ نرون در لایه پنهان

جدول ۲- نتایج شبیه‌سازی اول برای سری زمانی آشوبی هنون براساس الگوریتم پس انتشار

| | تعداد نرون‌ها | تعداد اتصالات | خطای شناسایی | |
|---|---|---|---|---|
| | | | آموزش | آزمایش |
| *MLP* | ۳ | ۱۵ | 0.0584 | 0.0591 |
| | ۵ | ۲۵ | 0.1043 | 0.1092 |
| *RMLP* | ۳ | ۳۶ | 0.0050 | 0.0058 |
| | ۲ | ۲۴ | 0.0084 | 0.0093 |
| *Prposed RMLP* | ۳ | ۳۶ | 0.0038 | 0.0046 |
| | ۲ | ۲۴ | 0.0062 | 0.0067 |




شکل ۸- نمودار همگرایی شبکه‌های عصبی راف در مقایسه با شبکه پرسپترون برای سری زمانی آشوبی هنون



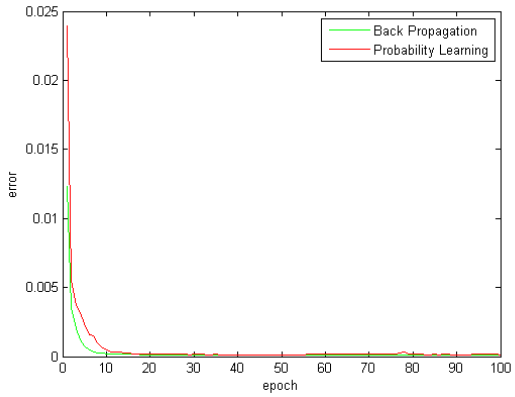

شکل ۹- نمودار شناسایی سری زمانی آشوبی مکی- گلاس با شبکه عصبی راف بهبود یافته با ۳ نرون در لایه پنهان با الگوریتم پس انتشار خطا و یادگیری احتمالاتی



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]

جدول ۳- نتایج شبیه‌سازی دوم برای سری زمانی مکی- گلاس براساس الگوریتم یادگیری احتمالاتی

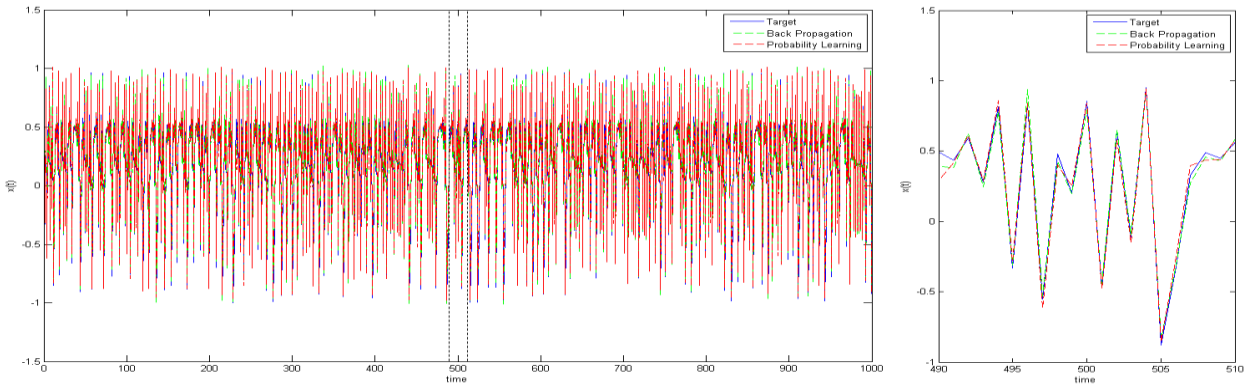
| | تعداد نرون‌ها | تعداد اتصالات | خطای شناسایی | |
|---|---|---|---|---|
| | | | آموزش | آزمایش |
| *MLP* | ۳ | ۱۵ | $0.2064\times10^{-3}$ | $0.2213\times10^{-3}$ |
| | ۵ | ۲۵ | $0.1332\times10^{-3}$ | $0.1441\times10^{-3}$ |
| *RMLP* | ۳ | ۳۶ | $0.1489\times10^{-3}$ | $0.1656\times10^{-3}$ |
| | ۲ | ۲۴ | $0.3022\times10^{-3}$ | $0.3093\times10^{-3}$ |
| *Prposed RMLP* | ۳ | ۳۶ | $0.1270\times10^{-3}$ | $0.1352\times10^{-3}$ |
| | ۲ | ۲۴ | $0.2211\times10^{-3}$ | $0.2210\times10^{-3}$ |



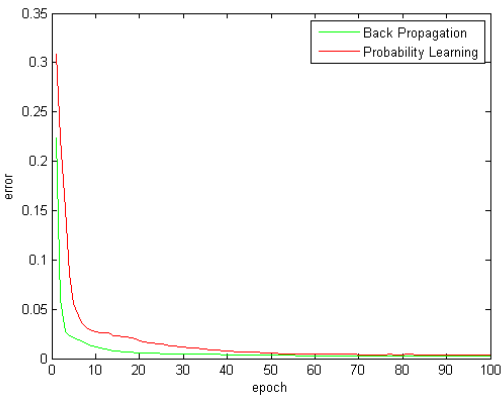

شکل ۱۰- نمودار همگرایی برای یادگیری احتمالاتی در مقایسه با الگوریتم پس انتشار برای سری زمانی آشوبی مکی- گلاس

جدول ۴- نتایج شبیه‌سازی دوم برای سری زمانی آشوبی هنون براساس الگوریتم یادگیری احتمالاتی

| | تعداد نرون‌ها | تعداد اتصالات | خطای شناسایی | |
|---|---|---|---|---|
| | | | آموزش | آزمایش |
| *MLP* | ۳ | ۱۵ | 0.0661 | 0.0704 |
| | ۵ | ۲۵ | 0.1013 | 0.0981 |
| *RMLP* | ۳ | ۳۶ | 0.0022 | 0.0025 |
| | ۲ | ۲۴ | 0.0083 | 0.0123 |
| *Prposed RMLP* | ۳ | ۳۶ | 0.0077 | 0.0091 |
| | ۲ | ۲۴ | 0.0035 | 0.0037 |




شکل ۱۱- نمودار شناسایی سری زمانی آشوبی هنون توسط شبکه عصبی راف بهبود یافته با ۳ نرون با یادگیری احتمالاتی




شکل ۱۲- نمودار همگرایی برای یادگیری احتمالاتی در مقایسه با الگوریتم پس انتشار برای سری زمانی آشوبی هنون



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]

### ۵-۵- شبیه‌سازی دوم: بررسی کارایی الگوریتم یادگیری احتمالاتی

با رعایت همان شرایط اولیه در شبیه‌سازی اول، در این آزمایش تمامی شبکه‌های عصبی با استفاده از الگوریتم یادگیری احتمالاتی که در این مقاله معرفی شد، آموزش داده می‌شوند. نتایج حاصل از انجام این آزمایش برای شناسایی سری‌های زمانی آشوبی مکی گلاس و هنون به ترتیب در جداول ۳ و ۴ بر حسب خطای شناسایی خلاصه شده است.

نتایج مندرج در جداول ۳ و ۴ به ترتیب در مقایسه با نتایج مندرج در جداول ۱ و۲، نشان می‌دهد که الگوریتم یادگیری احتمالاتی ارائه شده در این مقاله در بیشتر موارد نتایج بهتری را نسبت به الگوریتم پس انتشار بر حسب خطای شناسایی فراهم می‌کند. همانند شبیه‌سازی قبل، در این شبیه‌سازی شبکه عصبی راف پیشنهادی عملکرد بهتری را نسبت به شبکه پرسپترون در شناسایی سری زمانی آشوبی هنون داشته است. نمودارهای شناسایی و همگرایی برای شبکه عصبی راف بهبود یافته با ۳ نرون در لایه پنهان، که با استفاده از الگوریتم‌های پس انتشار و یادگیری احتمالاتی آموزش داده شده‌اند، برای سری زمانی آشوبی مکی- گلاس به ترتیب در شکل‌های ۹ و ۱۰، و برای سری زمانی آشوبی هنون به ترتیب در شکل‌های ۱۱ و ۱۲ نشان داده شده‌اند. همانطور که در شکل‌های ۱۰ و ۱۲ دیده می‌شود، همگرایی الگوریتم یادگیری احتمالاتی نسبت به الگوریتم پس انتشار کندتر است.

## ۶- نتیجه‌گیری

در این مقاله بهبودی برای شبکه عصبی راف با استفاده از تعریف نرخ تبادل اطلاعات پیشنهاد گردید. همچنین روش یادگیری جدیدی موسوم به یادگیری احتمالاتی ارائه شد که بر مبنای احتمال خطای موثر پارامترهای آموزش‌پذیر شبکه عمل می‌کند. سپس شبکه‌های عصبی راف به عنوان شناساگر سیستم‌های غیرخطی مطرح شدند، برای این منظور شبکه‌های عصبی راف، برای شناسایی سری‌های زمانی آشوبی به کار گرفته شدند. نتایج نشان می‌دهد که عموما خطای شناسایی شبکه عصبی راف پیشنهادی نسب به شبکه‌های دیگر در شرایط یکسان، کمتر و نرخ همگرایی آن نیز تا حدودی سریعتر می‌باشد. هر چند شبکه پرسپترون قادر است سری زمانی آشوبی مکی- گلاس را به خوبی شناسایی کند، اما برخلاف شبکه عصبی راف پیشنهادی قادر به شناسایی مناسب سری زمانی آشوبی هنون نیست، ازاینرو هرچند شبکه عصبی راف، با پیچیدگی بیشتری همراه است، اما استفاده از آن به عنوان شناساگر سیستم غیرخطی توجیه‌پذیر است. نتایج نشان می‌دهد که الگوریتم یادگیری احتمالاتی، در شرایط یکسان نسبت به الگوریتم پس‌انتشار موثرتر عمل می‌کند.

## مراجع

[1] O. Nelles, "Nonlinear System Identification: from Classical Approach to Neural Network and Fuzzy Model", Springer-Verlag, New York, 2001.

[2] P. Lingras, "Rough Neural Networks", in: Proceeding of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU96), Granada, Spain, pp.1445-1450, 1996.

[3] S. K. Pal, J. F. Peters; L. Polkowski, A. Skowron, "Rough-Neuro Computing: An Introduction", in: (eds.) Pal, S. K.; Polkowski, L.; Skowron, A.; Rough-Neural computing: Techniques for Computing with Words, Springer, pp. 15–42, 2004.

[4] J. F. Peters, M. S. Szczuka, "Rough Neurocomputing: A Survey of Basic Models of Neurocomputation", in: Proceeding of Rough Sets and Current Trends in Computing (RSCTC2002), Malvern, USA, pp. 308–315, 2002.

[5] S. R. Abdallah, Y. F. Hassan, "Rough Neural Intelligent Approach for Image Classification", International Journal of Hybrid Intelligent Systems, Vol. 3, No. 4, pp. 205–218, 2006.

[6] D. Yamaguchi, F. Katayama, M. Takahashi, M. Arai, K. J. Mackin, "The Medical Diagnostic Support System using Extended Rough Neural Network and Multiagent", in: Proceeding of International Conference on 13th Artificial Life and Robotics, Vol.13 ,No. 1, pp. 184-187, 2007.

[7] A. A. Bakar, F. Febriyani, "Rough Neural Network Model for Tuberculosis Patient Categorization", in: Proceeding of the International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, pp. 765–768, 2007.

[8] S. Krikpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671–680, 1983.

[9] D. H. Ackley, G. E. Hinton, T. J. Sejnowski, "A Learning Algorithm for Boltzmann Machines", Cognitive Science, Vol. 9, No. 1, pp. 147–169, 1985.

[10] K. P. Unnikrishnan, K. P. Venugopal, "Alopex: A Correlation Based Learning Algorithm for Feed-Forward and Recurrent Neural Networks", International Journal of Neural Computation, Vol. 6, pp. 469–490, 1994.

[11] A. Bia, "Alopex-B: A new, simple, but Yet Faster Version of the Alopex Training Algorithm", International Journal of Neural Systems, Vol. 11, No. 6, pp. 497–507, 2001.

[12] Haykin, S. Becker, "Theory Mont Carlo Sampling -Based Alopex Algorithm for Neural Networks", in: *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP2004), Vol. 5, pp. 501–504, 2004.



[ Downloaded from joc.kntu.ac.ir on 2026-10-30 ]
[ DOR: 20.1001.1.20088345.1391.6.1.5.7 ]