

استفاده از شبکه‌های عصبی تابعی پله‌ای شعاعی مبتنی بر الگوریتم جستجوی گرانشی با بهینه عمومی راهنما به منظور دسته‌بندی دادگان سونار

سید محمدرضا موسوی^{۱*}، محمد خویشه^۲

۱- استاد، ۲- دانشجوی دکتری، دانشکده مهندسی برق، دانشگاه علم و صنعت ایران

(دریافت: ۹۴/۰۷/۲۵، پذیرش: ۹۵/۰۵/۱۱)

چکیده

با توجه به خصوصیات فیزیکی بسیار مشابه اکوی اهداف واقعی و کاذب سونار فعال، طبقه‌بندی و تمیز دادن آن‌ها از یکدیگر به یکی از زمینه‌های دشوار و پیچیده برای محققان و صنعتگران این حوزه تبدیل شده است. شبکه‌های عصبی تابعی پله‌ای شعاعی (RBF NN) یکی از پرکاربردترین شبکه‌های عصبی مصنوعی در دسته‌بندی اهداف دنیای واقعی هستند. آموزش از مهمترین بخش‌های توسعه این نوع شبکه است که در سال‌های اخیر بسیار مورد توجه قرار گرفته است. به منظور آموزش RBF NN از دیر باز استفاده از روش‌های بازگشتی و گرادیان نزولی مرسوم بوده است. با این وجود، دقت دسته‌بندی نامناسب، گیر افتادن در کمینه‌های محلی و سرعت همگرایی پایین از معایب روش‌های سنتی می‌باشد. در سال‌های اخیر استفاده از الگوریتم‌های ابتکاری و فرا ابتکاری به منظور غلبه بر این معایب بسیار مرسوم گردیده است. این مقاله برای آموزش RBF NN از الگوریتم جستجوی گرانشی با هدایت بهینه عمومی (LMGSA) به منظور غلبه بر نقص الگوریتم جستجوی گرانشی (GSA) در فاز بهره‌برداری، استفاده می‌کند. نتایج نشان می‌دهد که دسته‌بندی کننده طراحی شده در تمام زمینه‌ها نتایج بهتری نسبت به دسته‌بندی کننده‌های معیار ارائه می‌نماید. به منظور آزمودن دسته‌بندی کننده طراحی شده، این الگوریتم با الگوریتم‌های GSA، گرادیان نزولی (GD)، فیلتر کالمن (KF)، فیلتر کالمن تفکیک شده (DKF) و الگوریتم ژنتیک (GA) توسط سه مجموعه دادگان سنجیده می‌شود. معیارهای مورد سنجش عبارتند از: سرعت همگرایی، احتمال گیر افتادن در کمینه‌های محلی و دقت دسته‌بندی. در پایان نیز به عنوان یک کاربرد عملی دادگان سونار توسط این شبکه دسته‌بندی می‌شوند.

واژه‌های کلیدی: دسته‌بندی، دادگان سونار، تابعی پله‌ای شعاعی، الگوریتم جستجوی گرانشی تطبیقی با هدایت بهینه عمومی

۱- مقدمه

امروزه با پیشرفت علم، زمینه‌های جدیدی در حوزه جنگ الکترونیکی و سایبری مانند جاسوسی و جمع آوری اطلاعات در زیر سطح (با حداقل خطرپذیری کشف)، عبور و مرور و بهره برداری از زیر سطح به صورت مخفیانه و تولید اهداف کاذب برای گمراه کردن دشمن در صحنه نبرد، توسعه یافته‌اند. بدین منظور توانایی کشف و شناسایی اهداف سوناری و شناسایی اهداف واقعی از کاذب و تعیین نوع کلاس آن به زمینه‌های جدید پژوهشی تبدیل شده است.

اهداف آشکارسازی شده توسط سونار عبارت‌اند از: هدف واقعی، نویز، طنین و اکوی بدون هدف. نویز دارای انواع مختلفی مانند نویز حرارتی، محیطی، قاره‌ای، لنگرگاه و غیره می‌باشد که چون از جنس پینگ ارسالی توسط سونار نمی‌باشد، تمایز آن از هدف واقعی آسان است. اکوهای حاصل از برخورد پینگ ارسالی سونار با کف و سطح دریا را طنین گویند [۲]. با توجه به این‌که اکوهای حاصل از طنین دارای یک دامنه یکسان و همگن

مجموعه اقداماتی شامل، شناسایی راه‌های نفوذ، حمله و مقابله با آن‌ها به منظور افزایش ضریب امنیت، ایمنی، پایداری و حفاظت از زیرساخت‌های ارتباطی و اطلاعاتی را پدافند سایبری می‌گویند [۱]. اهمیت دریاها و وجود منابع و زیرساخت‌های بسیار حیاتی کشور در بستر، سطح و یا سواحل آن، توجه به بحث پدافند سایبری دریا پایه را بیش از پیش برجسته می‌سازد. کسب اشراف و توانایی اطلاعاتی و عملیاتی، یکی از پیش‌نیازهای پدافند سایبری در حوزه دریا می‌باشد. یکی از ابزارهای بسیار حیاتی در زمینه جنگ الکترونیکی و سایبری در دریاها سونار است. این وسیله توانایی کشف، مکان یابی، ره گیری و شناسایی اهداف زیرسطح و روی سطح آب را دارد.

* رایانامه نویسنده مسئول: m_mosavi@iust.ac.ir

شده‌اند. علاوه بر روش‌های مبتنی بر مشتق، روش‌های بدون مشتق گیری، از جمله الگوریتم ژنتیک^۸ [۱۳]، اتوماتای یادگیر^۹ [۱۴]، ترید تدریجی^{۱۰} [۱۵]، کلونی زنبور عسل مصنوعی^{۱۱} [۱۶]، بهینه ساز واکنش شیمیایی^{۱۲} [۱۷]، الگوریتم بهینه ساز مغناطیسی^{۱۳} [۱۸]، بهینه ساز نیروی مرکزی^{۱۴} [۱۹]، بهینه‌ساز دانه معکوس^{۱۵} [۲۰]، جستجوی سامانه متهم^{۱۶} [۲۱]، بهینه ساز عنکبوت اجتماعی^{۱۷} [۲۲]، الگوریتم آموزش و یادگیری^{۱۸} [۲۳]، بهینه‌ساز مبتنی بر جغرافیای زیستی^{۱۹} [۲۴]، بهینه‌ساز گرگ خاکستری^{۲۰} و بهینه‌یافته‌های آن [۲۵ و ۲۶]، الگوریتم ترکیبی PSO-GSA [۲۷] در آموزش شبکه‌های عصبی استفاده شده است.

هدف نهایی فرآیند یادگیری در شبکه‌های عصبی، پیدا کردن بهترین ترکیب از پارامترهای شبکه می‌باشد به طوری که در آموزش شبکه و نمونه‌های آزمون، کمترین مقدار خطا را داشته باشیم. روش‌های ابتکاری^{۲۱} و فراابتکاری^{۲۲} به مشتق گیری از توابع هدف نیاز ندارد. از سوی دیگر با توجه به پیدا کردن کمینه‌ی عمومی، خیلی بهتر از روش‌های مبتنی بر مشتق گیری (که در کمینه‌های محلی گیر می‌کنند) عمل می‌نمایند. همچنین آن‌ها می‌توانند کاربرد خود را در محدوده وسیعی از توابع هدف و ساختار شبکه‌های عصبی گسترش دهند.

صرف‌نظر از تفاوت‌های بین روش‌های فرا ابتکاری مختلف، یک ویژگی مشترک بین آن‌ها تقسیم فرآیند جستجو به دو مرحله شناسایی^{۲۳} و بهره‌برداری^{۲۴} است [۳۰-۲۸]. مرحله شناسایی به روند بررسی گسترده بخش‌های مطلوب فضای جستجو اشاره دارد. یک الگوریتم نیاز به اپراتورهای تصادفی دارد تا بتواند به صورت تصادفی و سرتاسری فضای جستجو را اکتشاف کند و مرحله شناسایی را پشتیبانی نماید. این در حالی است که مرحله بهره‌برداری اشاره به قابلیت جستجوی محلی در سراسر مناطق مطلوبی دارد که از مرحله شناسایی به دست آمده است.

می‌باشند، علی‌رغم این‌که از جنس پینگ ارسالی هستند، تمایز آن‌ها از هدف واقعی آسان است. هنگامی که جنس بستر دریا دارای تغییرات زیادی است و در ابعاد کوچک، بستر با جنس‌های مختلفی وجود دارد، اکوهای برگشتی از بستر دارای ویژگی‌های هدف گونه خواهند بود، به صورتی که حتی تابع چگالی احتمال هدف واقعی و بستر بسیار باهم شبیه خواهند بود. این گونه اهداف کاذب را اکوی بدون هدف گویند [۳]. طبقه‌بندی اکوی بدون هدف و اهداف واقعی به خاطر تشابه بسیار زیاد اکوهای برگشتی از آن‌ها، کار بسیار دشواری است. دسته‌بندی‌کننده‌های متعارف آماری (که اکثراً بر مبنای تئوری بیز استوار هستند)، به دلیل نیازمندی به محاسبه دانش قبلی و تابع چگالی احتمال (PDF)، وابسته به آزمایشات گسترده بوده و در نتیجه هزینه، زمان و تجهیزات بسیار زیادی احتیاج دارند [۴]. از طرف دیگر با توجه به اینکه مشخصات انتشار صوت در هر نقطه از دریا و در هر باز زمانی از شب و روز متفاوت است، دسته‌بندی‌کننده‌های باید انتخاب بشود که هم به آزمایشات کمتر وابسته و هم قابلیت تطبیق پذیری بالای داشته باشد. از اینرو شبکه‌های عصبی مصنوعی برای دسته‌بندی هدف واقعی و اکوی بدون هدف انتخاب می‌شوند. این گونه شبکه‌ها می‌توانند با یک مجموع دادگان محدود آموزش ببینند و در سناریوهای تست مختلف به کار برده شوند.

شبکه‌های عصبی^۱ RBF یکی از پرکاربردترین ابزار برای محاسبات نرم می‌باشند. با استفاده از این شبکه‌ها می‌توان مسائل غیرخطی را حل نمود. به طور کلی شبکه‌های عصبی RBF به منظور دسته‌بندی الگو، پیش‌بینی داده و تقریب زدن توابع مورد استفاده قرار می‌گیرند [۵ و ۶]. صرف نظر از کاربردها، توانایی متمایز شبکه‌های RBF یادگیری می‌باشد. یادگیری به این معنی است که این شبکه‌ها همانند مغز انسان می‌توانند از یک تجربه یا آزمایش یاد بگیرند. این ویژگی (یادگیری) بخش ضروری همه شبکه‌های عصبی می‌باشد که ممکن است به دو نوع یادگیری با نظارت^۲ [۷] و یادگیری بدون نظارت^۳ [۸] تقسیم گردد. به طور کلی آموزش شبکه‌های عصبی یک چالش برای مسائل بهینه‌سازی غیرخطی می‌باشد. روش‌های زیادی بر پایه‌ی مشتق‌گیری برای تعلیم شبکه‌های عصبی RBF شامل گرادیان نزولی^۴ (GD) [۹]، فیلتر کالمن^۵ (KF) [۱۰]، فیلتر کالمن تفکیک‌شده^۶ (DKF) [۱۱] و پس انتشار^۷ (BP) [۱۲] استفاده

8 - Genetic Algorithm
9 - Learning Automata
10 - Simulated Annealing
11 - Artificial Bee Colony Algorithm
12 - Chemical Reaction Optimization
13 - Magnetic Optimization Algorithm
14 - Central Force Optimization
15 - Invasive Weed Optimization
16 - Charged System Search
17 - Social-Spider Optimization
18 - Teaching-Learning-based Optimization Algorithm
19 - Biogeography-Based optimizer
20 - Gray Wolf Optimizer
21 - Heuristic
22 - Meta-Heuristic
23 - Exploration
24 - Exploitation

1- RBF: Radial Basis Function
2- Supervised Learning
3- Unsupervised Learning
4- Gradient Decent
5- Kalman Filter
6- Decoupling the Kalman Filter
7-Well-Known Back-Propagation

فضای جستجو به صورت کامل توسط اجرام با وزن‌های مختلف جستجو می‌شود.

- نقطه ضعف الگوریتم GSA سرعت کم در فاز بهره‌برداری است. این ضعف توسط ترکیب با الگوریتم‌های دیگر مانند PSO-GSA [۲۷] حل شده است، ولی ترکیب دو الگوریتم از لحاظ محاسباتی بسیار پرهزینه می‌باشد. از طرف دیگر در بسیاری از کاربردهای سونار فعال (از قبیل کاربردهای نظامی) نیاز به پردازش‌های بلادرنگ است و پیچیدگی زمانی یک نقطه ضعف بزرگ به‌شمار می‌آید. این ضعف توسط gbest های هدایت شده در الگوریتم LMGSA جبران می‌شود.

سازماندهی مقاله بدین صورت می‌باشد که در بخش دوم شبکه‌های عصبی RBF توضیح داده خواهد شد. در بخش سوم الگوریتم آموزش شبکه RBF یعنی LMGSا توصیف گردیده است. در بخش چهارم شبکه‌های عصبی RBF توسط الگوریتم LMGSا آموزش داده می‌شوند. در بخش پنجم نتایج شبیه‌سازی تشریح گردیده و در نهایت در بخش ششم نتیجه‌گیری ارائه می‌گردد.

۲- شبکه عصبی تابع پله‌ای شعاعی

شبکه RBF یک نوع از شبکه‌های عصبی رو به جلو متشکل از سه لایه شامل لایه ورودی، لایه پنهان و لایه خروجی می‌باشد. هر کدام از این لایه‌ها دارای وظایف مختلفی هستند [۳۲]. بلوک دیگرام کلی مدل سازی با یک شبکه عصبی RBF در شکل (۱) نشان داده شده است. در شبکه‌های RBF، خروجی‌های لایه ورودی با محاسبه فاصله بین ورودی‌های شبکه و مراکز لایه پنهان تعیین می‌شود. لایه دوم لایه خطی پنهان است و خروجی‌های این لایه نمونه‌های وزن‌گذاری شده از خروجی‌های لایه ورودی هستند. هر نرون لایه پنهان از یک بردار پارامتر به نام مرکز برخوردار است. بنابراین یک توصیف کلی از شبکه را می‌توان به صورت رابطه (۱) نمایش داد [۳۱]:

$$\hat{y}_j = \sum_{i=1}^I W_{ij} \phi(\|x - C_i\|) + \beta_j \quad (1)$$

حالت استاندارد معمولاً با فاصله اقلیدسی و تابع پایه‌ای شعاعی نیز با تابع گاوسی در نظر گرفته شده و به صورت زیر تعریف می‌شود:

$$\phi(r) = \exp(-\alpha \|x - C_i\|^2) \quad (2)$$

در روابط (۱) و (۲) تعاریف زیر در نظر گرفته شده است:

پیدا کردن یک موازنه مناسب بین این دو مرحله با توجه به ماهیت تصادفی روش‌های فرا ابتکاری، مسئله‌ای چالش‌برانگیز می‌باشد.

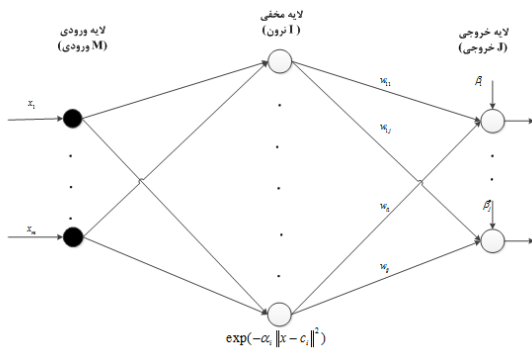
یکی از روش‌هایی که در این زمینه به خوبی عمل می‌کند، الگوریتم جستجوی گرانشی^۱ (GSA) [۳۱] می‌باشد. این الگوریتم از نیروهای گرانشی بین اجرام در طبیعت الهام گرفته است. الگوریتم جستجوی گرانشی عملکرد برتری را در میان دیگر الگوریتم‌های ابتکاری شناخته شده مانند بهینه‌سازی ازدحام ذرات و الگوریتم ژنتیک نشان داده است. با این حال، نرخ بهره‌برداری کند، ضعف اصلی آن است که باعث کاهش کارایی در هنگام مواجهه با مشکلات واقعی مهندسی با ابعاد بالا (از قبیل دادگان سونار) می‌شود. با توجه به اثر تجمعی تابع تناسب جرم در GSA، جرم‌ها در طول دوره‌های تکرار سنگین‌تر و سنگین‌تر می‌شوند. این کار باعث می‌گردد که جرم‌ها در مجاورت هم باقی بمانند و نیروهای گرانشی یکدیگر را در تکرارهای بعدی خنثی نمایند و از بهره‌برداری سریع و بهینه آن‌ها ممانعت کنند. برای غلبه بر این نقص بهترین جرم هر تکرار (gbest) به عنوان جرم راهنما^۲ جمع‌آوری و برای سرعت بخشیدن به مرحله بهره‌برداری و بهبود این ضعف به کار گرفته می‌شود. این تکنیک اساس الگوریتم جدید جستجوی گرانشی تطبیقی با جرم راهنما^۳ (LMGSA) می‌باشد.

در این مقاله شبکه عصبی RBF توسط الگوریتم LMGSا به منظور دسته‌بندی دادگان سونار (شامل هدف واقعی و کلاتر^۴) آموزش داده شده است. دلیل استفاده از شبکه RBF و الگوریتم آموزش LMGSا به شرح ذیل می‌باشد:

- شبکه RBF به دلیل استفاده از توابع پایه، توانایی منحصر به فردی در کار با دادگانی که به‌طور خطی قابلیت تفکیک ندارند را دارا می‌باشد. از طرفی دادگان سونار به‌صورت خطی قابل تفکیک نیستند و نیازمند دسته‌بندی‌کننده‌ای با ابعاد بالا می‌باشند.
- کلاتر و هدف واقعی دارای ویژگی‌های بسیار شبیه هم می‌باشند، در نتیجه باید الگوریتمی انتخاب شود که فضای جستجو را به‌طور کامل اکتشاف کند.
- نقطه قوت الگوریتم GSA نسبت به الگوریتم‌های فراابتکاری دیگر، قدرت اکتشاف فوق‌العاده آن می‌باشد، به‌طوری که

1 . GSA: Gravitational Search Algorithm
2 . Leader Mass
3 . LMGSا: Leader Mass Gravitational Search Algorithm
4 . Clutter

روش طراحی شبکه عصبی RBF باید شامل تعیین تعداد نرون‌ها در لایه پنهان باشد. به منظور به دست آوردن خروجی مورد نظر از شبکه عصبی RBF پارامترهای α ، C و β باید به درستی تنظیم شوند.



شکل (۲). ساختار یک شبکه عصبی RBF.

معیارهای محاسبه خطا مانند میانگین مربعات خطا (MSE) و یا مجموع مربعات خطا (SSE) را می‌توان برای ارزیابی عملکرد شبکه به کار برد. عبارت خطا برای شبکه عصبی RBF را می‌توان به شرح زیر تعریف کرد:

$$E^{SSE}(w, a, c, \beta) = \sum_{i=1}^I (y_i - \hat{y}_i)^2 \quad (3)$$

در اینجا y_i خروجی مطلوب و \hat{y}_i خروجی شبکه عصبی RBF را نشان می‌دهد. به حداقل رساندن تابع خطا، روش آموزش شبکه عصبی RBF است.

۳- الگوریتم‌های آموزش شبکه‌های عصبی

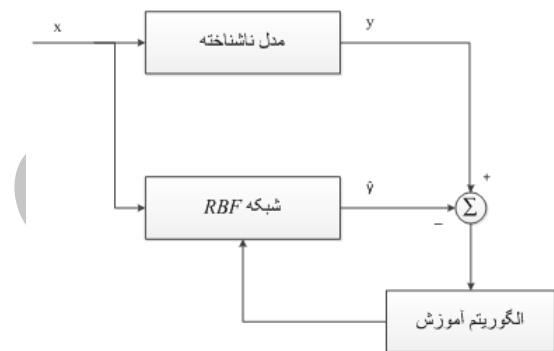
در این بخش الگوریتم LMGSA که توسط این مقاله برای آموزش شبکه عصبی RBF اعمال می‌گردد، به طور مفصل توضیح داده می‌شود. برای این کار در ابتدا الگوریتم کلاسیک GSA به طور مختصر شرح داده می‌شود.

۳-۱- الگوریتم جستجوی گرانشی

الگوریتم GSA از قانون گرانش نیوتن الهام گرفته شده است. الگوریتم GSA جستجو را با به کارگیری مجموعه‌ای از عوامل (راه‌حل انتخابی) انجام می‌دهد که دارای جرم‌های متناسب با مقدار تابع شایستگی هستند. در طول تکرار، جرم‌ها یکدیگر را توسط نیروهای گرانشی بینشان جذب می‌کنند. جرم سنگین‌تر، نیروی جاذبه بیشتری دارد. بنابراین سنگین‌ترین جرم که احتمالاً نزدیک به بهینه عمومی است، جرم‌های دیگر را به نسبت فاصله‌یشان جذب می‌نماید.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, N \quad (4)$$

- $i \in \{1, 2, 3, \dots, I\}$ به طوری که I تعداد نرون‌ها در لایه پنهان می‌باشد
- $j \in \{1, 2, 3, \dots, J\}$ به طوری که J تعداد نرون‌ها در لایه خروجی است
- w_{ij} وزن بین نرون i ام در لایه مخفی و خروجی j ام می‌باشد
- ϕ تابع پایه شعاعی
- a_i پارامتر انتشار^۱ نرون i ام
- X بردار داده ورودی
- C_i بردار مرکز نرون i ام
- β_j مقدار بایاس از خروجی نرون j ام
- \hat{y}_j خروجی شبکه از نرون j ام



شکل (۱). بلوک دیاگرام مدل سازی با یک شبکه عصبی RBF.

شکل (۲) معماری دقیقی از یک شبکه عصبی RBF را نشان می‌دهد. ورودی‌های M بعدی (x_1, \dots, x_m) در لایه ورودی قرار داده می‌شوند که ورودی‌ها را به لایه پنهان وصل می‌کنند. لایه پنهان شامل I نرون و هر نرون در این لایه بر حسب فاصله اقلیدسی بین مراکز و ورودی‌ها محاسبه می‌شود. نرون در لایه پنهان از یک تابع فعال‌سازی به نام تابع پایه^۲ برخوردار است. در مقالات، اغلب تابع گاوسی برای تابع پایه‌ای شعاعی انتخاب می‌شود و دارای یک پارامتر انتشار برای تنظیم شکل منحنی است $(\alpha_1, \dots, \alpha_i)$. خروجی‌های وزن‌دهی شده (w_{11}, \dots, w_{ij}) از لایه‌های پنهان به لایه خروجی منتقل می‌شوند. در اینجا، I نشان‌دهنده تعداد نرون‌های لایه‌های پنهان و J نشان‌دهنده بعد خروجی می‌باشد. لایه خروجی ترکیب خطی از خروجی‌های لایه پنهان و پارامترهای بایاس $(\beta_1, \dots, \beta_j)$ است. در نهایت، خروجی شبکه RBF به دست می‌آید $(\hat{y}_1, \dots, \hat{y}_j)$.

1- Spread Parameter

2- Basis Function

می‌کند. بنابراین شتاب همه عوامل به صورت رابطه (۹) محاسبه می‌شود:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (9)$$

که در آن d مبین بعد مسئله، t نشان‌دهنده یک زمان مشخص، M_{ii} مبین جرم اینرسی عامل i ام می‌باشد. سرعت و موقعیت عامل‌ها به ترتیب توسط روابط (۱۰) و (۱۱) محاسبه می‌شود:

$$V_i^d(t+1) = rand_i \times V_i^d(t) + a_i^d(t) \quad (10)$$

$$x_i^d(t+1) = x_i^d(t) + V_i^d(t+1) \quad (11)$$

همان‌طور که می‌توان از روابط (۱۰) و (۱۱) استنباط کرد، سرعت فعلی یک عامل به عنوان بخشی از آخرین سرعت که به شتابش افزوده شده تعریف می‌گردد ($0 \leq rand_i \leq 1$). تنها بخشی از سرعت اولیه به منظور جلوگیری از فرارجهش عوامل جستجو در مرزهای آن استفاده شده است. هم‌چنین موقعیت فعلی یک عامل توسط آخرین موقعیت خودش که به سرعت فعلی آن اضافه شده است، تنظیم می‌شود. برای اعمال درست قوانین حرکت نیوتن، سرعت به‌کار گرفته شده باید متوسط سرعت در زمان t و $t+1$ باشد. با این حال، نسخه اصلی GSA فقط از سرعت نهایی استفاده کرده است.

از آن‌جا که عوامل جرم‌ها توسط توابع شایستگی تعریف می‌شوند، عاملی با سنگین‌ترین جرم، مناسب‌ترین عامل است. با توجه به معادلات بالا، سنگین‌ترین عامل دارای بالاترین نیروی جاذبه و کم‌ترین جنبش است. از آن‌جایی که یک رابطه مستقیم بین جرم و تابع شایستگی وجود دارد، یک روش نرمال شده برای مقیاس‌بندی جرم‌ها به شرح زیر اتخاذ شده است:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (12)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (13)$$

که در آن، $fit_i(t)$ مبین مقدار تناسب عامل i ام در زمان t ام می‌باشد. در GSA ابتدا همه عوامل با مقادیر تصادفی راه‌اندازی می‌گردند. در طول تکرار، سرعت‌ها و موقعیت‌ها با روابط (۱۰) و (۱۱) تعریف می‌شوند. در همین حال، دیگر پارامترها مانند ثابت گرانش و جرم‌ها توسط روابط (۵) و (۱۳) محاسبه می‌شوند. در نهایت GSA توسط برآوردن معیار پایانی خاتمه می‌یابد.

که در آن N مبین تعداد جرم‌ها، n نشان‌دهنده بعد مسئله و \mathcal{X}_i^d مبین موقعیت i امین عامل در d امین بعد الگوریتم می‌باشد. الگوریتم با قرار دادن تصادفی همه عوامل در فضای جستجو شروع می‌شود. در طول تمام بازه‌ها، نیروی گرانشی از عامل i ام تا عامل j ام در زمان مشخص t به شرح زیر تعریف می‌شود:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \quad (5)$$

که در آن، $M_{aj}(t)$ مبین جرم گرانشی فعال مربوط به عامل j ام و $M_{pi}(t)$ مبین جرم گرانشی منفعل مربوط به عامل i ام می‌باشد. $G(t)$ مبین ثابت گرانش در زمان t ، ϵ نشان‌دهنده ثابتی کوچک و $R_{ij}(t)$ مبین فاصله اقلیدسی بین دو عامل i ام و j ام است.

به منظور ارتقا اکتشاف در اولین تکرارها و بهره‌برداری از تکرارهای آخر، G با مقدار تطبیقی طراحی شده است. بنابراین در طول تکرارها افزایش می‌یابد. به عبارت دیگر، G عوامل جستجو را به حرکت با گام‌های بزرگ در تکرارهای اولیه تشویق می‌کند، اما آن‌ها به حرکت آرام در تکرارهای نهایی محدود می‌شوند. عامل گرانش (G) و فاصله اقلیدسی بین دو عامل i ام و j ام به شرح زیر محاسبه می‌گردد:

$$G(t) = G_0 \times \exp\left(-\alpha \times \frac{iter}{max\ iter}\right) \quad (6)$$

$$R_{ij}(t) = |X_i(t) - X_j(t)| \quad (7)$$

که در آن، a مبین یک ضریب کاهشی، G_0 نشان‌دهنده ثابت گرانش اولیه و $iter$ مبین تعداد تکرار فعلی است. هم‌چنین $max\ iter$ حداکثر تعداد تکرارها را نشان می‌دهد. در یک فضای مسئله با ابعاد یکسان برای d نیروی کل که بر عامل i ام اعمال می‌شود با رابطه زیر محاسبه می‌شود:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (8)$$

که در آن، $rand_j$ یک عدد تصادفی در بازه $[0, 1]$ است. برای برخورداری از گام حرکت تصادفی در امتداد نیروی جاذبه از هر عامل و نیروی حاصل نهایی، جزء تصادفی در این رابطه قرار داده شده است. این مورد به برخورداری از رفتارهای متنوع‌تر در حرکت به سمت عوامل جستجو کمک می‌کند. قانون حرکت نیوتن نیز در این الگوریتم، به‌کار گرفته شده است که شتاب یک جرم را متناسب با نیروی اعمال شده و معکوس با جرم آن بیان

نزدیک می‌شوند، اما آن‌ها الزاماً قادر به سرعت بخشیدن برای حرکت به سمت بهینه نیستند. لازم به ذکر است که GSA هیچ حافظه‌ای برای ذخیره‌سازی بهترین راه‌حلی که تاکنون به دست آمده ندارد. بنابراین ممکن است بهترین راه‌حل از دست برود، به طوری که بهترین جرم با دیگر جرم‌های با شایستگی کم‌تر جذب شود.

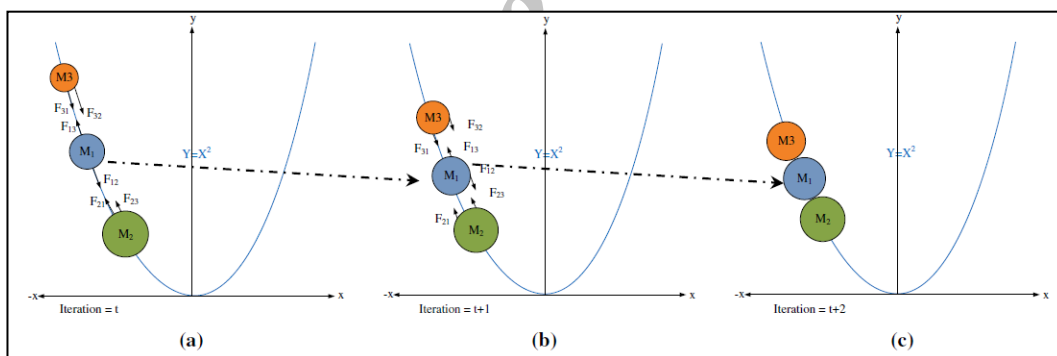
ایده اصلی روش LMGSA ذخیره‌سازی و استفاده از مکان بهترین جرم برای سرعت بخشیدن به مرحله بهره‌برداری است. شکل (۴) به کارگیری بهترین راه‌حل برای سرعت بخشیدن به حرکت جرم‌ها به سمت بهینه عمومی را نشان می‌دهد. همان‌طور که در این شکل نشان داده شده است، عنصر gbest یک جزء سرعت اضافی را به سمت آخرین مکان شناخته شده برای بهترین جرم اعمال می‌کند. بدین وسیله "نیروی خارجی gbest" از گیرکردن جرم‌ها در بهینه‌های محلی جلوگیری می‌کند. در این روش دو مزیت وجود دارد: اول اینکه شتاب بخشیدن به حرکت ذرات به سمت مکان بهترین جرم، به آن‌ها برای پیشی گرفتن و رسیدن به بهترین جرم در تکرار بعدی (همان‌طور که در شکل (۴) نشان داده شده) کمک می‌کند، ثانیاً بهترین راه‌حل به دست آمده تاکنون برای استفاده در تکرارهای بعدی ذخیره می‌شود.

۲-۳- الگوریتم جستجوی گرانشی تطبیقی با هدایت

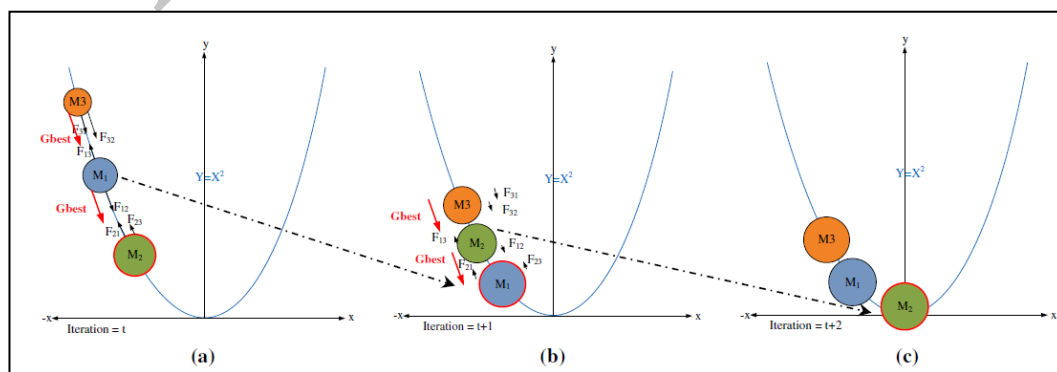
بهینه عمومی

در GSA ثابت گرانش (G) سرعت جرم‌ها را تنظیم می‌کند به طوری که با آن، راه‌حل‌ها مکان خود را در فضای راه‌حل تغییر می‌دهند. با توجه به رابطه (۵)، G بزرگ منجر به شتاب‌های بالا در نتیجه جنبش سریع جرم‌ها در تکرارهای اولیه می‌شود. با این حال، G به‌طور تکراری با توجه به معادله (۶) کاهش می‌یابد و به GSA در طول دوره بهره‌برداری کمک می‌کند [۳۳ و ۳۴]. بنابراین مرحله بهره‌برداری با نیروی جاذبه ضعیف‌تر و جرم‌های سنگین‌تر مقارن می‌شود. متأسفانه جرم‌های سنگین با جنبش کند و نیروی جاذبه ضعیف‌تر، سرعت هم‌گرایی را به‌طور قابل توجهی کاهش می‌دهند. بنابراین به نظر می‌رسد که GSA در مرحله بهره‌برداری از سرعت جستجوی کند ناشی از این عوامل رنج می‌برد. شکل (۳) مسأله تک بعدی ساده‌ای را نشان می‌دهد که در آن تابع شایستگی برابر $y = x^2$ است.

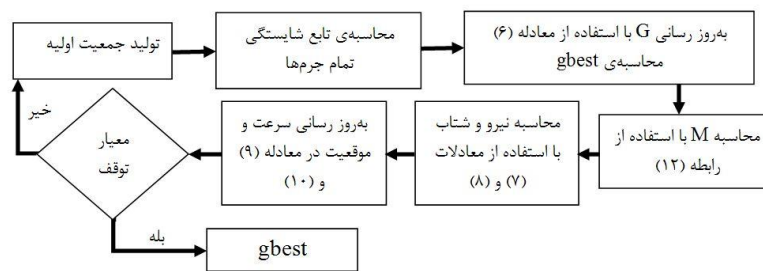
همان‌طور که در شکل (۳) مشاهده می‌شود، جرم‌های M_1 و M_2 توسط M_3 در تکرارهای $t+1$ و $t+2$ جذب می‌شوند. با این حال، این جرم‌ها نیز M_2 را جذب و به آرامی آن را به دوری از مقدار بهینه حرکت می‌دهند. در هر حال ذرات به مقدار بهینه



شکل (۳). رفتار جنبشی جرم‌ها در GSA.

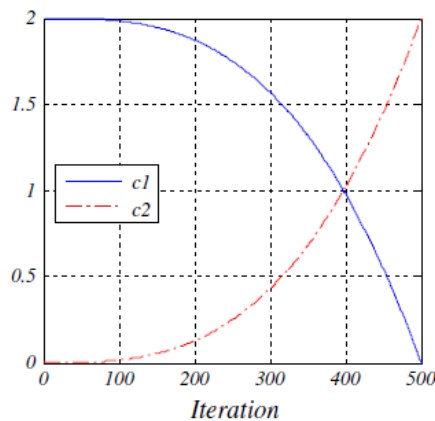


شکل (۴). حرکت جرم‌ها در روش LMGSA.



شکل (۵). بلوک دیاگرام کلی روش LMGSA.

تدریجی بین این دو مرحله است. علاوه بر این، این رویکرد تطبیقی بر اکتشاف در اولین تکرارها و بهره‌برداری در تکرارهای نهایی تمرکز می‌کند.

شکل (۶). منحنی ضرایب c_1' و c_2' .

۴- آموزش شبکه‌های عصبی RBF با استفاده از

الگوریتم LMGSA

به‌طور کلی، سه روش استفاده از الگوریتم‌های تکاملی برای آموزش شبکه‌های عصبی RBF وجود دارد. اولین روش عبارت است از به‌کارگیری شبکه‌های تکاملی به‌منظور یافتن ترکیب وزن یال‌ها، بایاس گره‌های خروجی، پارامترهای انتشار تابع اساسی و بردارهای مراکز لایه پنهان برای داشتن کمترین مقدار خطا در یک شبکه عصبی RBF. دومین شیوه عبارت است از به‌کارگیری شبکه‌های تکاملی به‌منظور یافتن ساختار مناسب شبکه عصبی RBF در یک مسئله خاص و آخرین شیوه شامل به‌کارگیری شبکه‌های تکاملی به‌منظور یافتن پارامترهای الگوریتم یادگیری مبتنی بر گرادینان از قبیل نرخ یادگیری و اندازه حرکت است. در این مقاله، الگوریتم LMGSA با استفاده از شیوه نخست به یک شبکه عصبی RBF اعمال می‌گردد. به‌منظور طراحی یک الگوریتم آموزش دهنده برای شبکه‌های عصبی RBF لازم است در الگوریتم LMGSA، وزن یال‌ها، بایاس گره‌های خروجی،

روش ارائه شده توسط رابطه (۱۴) مدل‌سازی می‌شود:

$$V_i(t+1) = \text{rand} \times V_i(t) + C_1' \times ac_i(t) + C_2' \times (gbest - X_i(t)) \quad (14)$$

در این رابطه، $V_i(t)$ مبین سرعت عامل i ام در تکرار t ام می‌باشد. C_1' و C_2' مبین ضرایب شتاب هستند. rand یک عدد تصادفی است بین صفر و یک، $ac_i(t)$ مبین شتاب عامل i ام در تکرار t ام و $gbest$ موقعیت بهترین راه‌حل به‌دست‌آمده تاکنون است. در معادله (۱۴)، اولین جز (یعنی $\text{rand} \times V_i(t) + C_1' \times ac_i(t)$) مشابه $V_i(t+1) = \text{rand} \times V_i(t) + C_1' \times ac_i(t) + C_2' \times (gbest - X_i(t))$ در آن بر فاز اکتشاف تأکید می‌شود. دومین جز (یعنی $C_2' \times (gbest - X_i(t))$) مسئول جذب جرم‌ها به سمت بهترین جرمی است که تا زمان t به‌دست می‌آید. فاصله هر جرم از بهترین جرم توسط $gbest - X_i(t)$ محاسبه می‌شود. نیروی نهایی به سمت بهترین جرم بخش تصادفی فاصله‌ای می‌باشد که توسط C_2' تعریف شده که یک نیروی نهایی خارجی خواهد بود که بر روی هر جرم اعمال شده است. در هر تکرار، موقعیت جرم‌ها به‌صورت رابطه (۱۵) به‌روزرسانی می‌شود. مراحل کلی روش LMGSA در شکل (۵) نشان داده می‌شود.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (15)$$

عنصر دائمی اضافه شده در این روش برای به‌روزرسانی سرعت در فاز اکتشاف اثر منفی خواهد گذاشت. به‌منظور جلوگیری از کاهش توانایی اکتشاف در روش به‌روزرسانی سرعت جدید، از مقادیر تطبیقی برای C_1' و C_2' (همان‌طور که در شکل (۶) نشان داده شده است)، استفاده می‌کنیم. هنگامی که الگوریتم به مرحله بهره‌برداری می‌رسد، به صورت تطبیقی C_1' را کاهش و C_2' را افزایش می‌دهیم، به‌گونه‌ای که جرم‌ها تمایل به سرعت بخشیدن به سمت بهترین راه‌حل را دارند. هیچ مرز مشخصی بین مراحل اکتشاف و بهره‌برداری در الگوریتم‌های تکاملی وجود ندارد. روش تطبیقی بهترین گزینه برای اجازه دادن به انتقال

بیش آموزش شبکه، افزایش پیچیدگی ساختاری و زمان اجرای الگوریتم می‌شود. با توجه به مرجع [۲۷] و بررسی‌های به عمل آمده در آن تعداد نرون‌های لایه مخفی ۴ عدد انتخاب می‌شود. با افزایش بیشتر تعداد نرون‌های لایه مخفی، در حالی که پیچیدگی مکانی و زمانی شبکه را افزایش می‌دهیم، عملکرد شبکه به مقدار قابل توجهی افزوده نمی‌شود. جرم‌های الگوریتم LMGSA شامل پارامترهای وزن (\bar{w})، انتشار ($\bar{\alpha}$)، بردارهای مرکز (\bar{c}) و بردارهای بایاس ($\bar{\beta}$) می‌باشد. یک جرم در الگوریتم LMGSA می‌تواند به صورت رابطه (۱۶) بیان شود:

$$P_i = [\bar{w} \bar{\alpha} \bar{c} \bar{\beta}] \quad (16)$$

همانطور که ذکر شد هدف نهایی روش‌های یادگیری، آموزش شبکه‌های عصبی مصنوعی است. مهم‌ترین بخش در یادگیری، فرآیند آموزش است. هر نمونه‌ی آموزش باید شامل محاسبه‌ی شاخص مناسب بودن همه‌ی جرم‌ها باشد. در این مقاله تابع شاخص مناسب بودن^۲ جرم (برای همه‌ی نمونه‌های آموزش) به روش مجموع مربعات خطا^۳ و به صورت رابطه (۱۶) محاسبه می‌گردد:

$$f = E^{SSE} \quad (17)$$

بلوک دیاگرام کلی آموزش شبکه RBF توسط الگوریتم LMGSA در شکل (۷) نمایش داده شده است.

۵- تنظیم پارامترها و انجام آزمایش

در این قسمت به منظور آزمایش کارایی الگوریتم LMGSA در آموزش شبکه عصبی RBF، این شبکه علاوه بر الگوریتم LMGSA توسط الگوریتم‌های معیار GD، KF، DKF، GA و GSA آموزش داده می‌شود. پارامترها و مقادیر اولیه این الگوریتم‌ها در جدول (۱) نمایش داده شده است. در ابتدا دسته‌بندی‌کننده‌های طراحی شده بر روی دادگان Iris و Lenses (توصیف شده در جدول (۲)) اعمال شده و عملکرد دسته‌بندی‌کننده‌ها از نظر نرخ دسته‌بندی، اجتناب از گیرکردن در کمینه محلی و سرعت همگرایی آزموده می‌شود. هر الگوریتم ۱۰ بار اجرا شده و نرخ دسته‌بندی، میانگین و انحراف معیار حداقل خطا و مقدار P-value در جداول (۳) و (۴) به ترتیب برای مجموعه دادگان Iris و Lenses نمایش داده شده است. نرخ

پارامترهای انتشار تابع اساسی لایه پنهان و بردارهای مراکز لایه پنهان به صورت مناسب نمایش داده شوند.

۴-۱- نمایش مسئله آموزش شبکه عصبی RBF

توسط الگوریتم LMGSA

به طور کلی، سه روش برای نمایش ترکیب پارامترهای مجهول وجود دارد: بردار، ماتریس و حالت باینری [۲۷]. در نمایش برداری، هر عنصر با یک بردار نمایش داده می‌شود. برای آموزش یک شبکه‌ی عصبی RBF، باید همه‌ی وزن‌ها، بایاس‌ها، پارامترهای انتشار و مراکز معلوم باشند. در نمایش ماتریسی، هر عنصر به صورت یک ماتریس نمایش داده می‌شود. برای نمایش باینری، هر عنصر به صورت رشته‌ای از بیت‌های باینری نمایش داده می‌شود. هر کدام از این شیوه‌های نمایش، مزایا و معایب خاص خود را دارد که می‌تواند در یک مسئله‌ی خاص مفید واقع گردد [۳].

در روش نخست، تبدیل عناصر به بردار، ماتریس و یا رشته‌ای از بیت‌های باینری آسان می‌باشد، اما فرآیند بازیابی آنها پیچیده خواهد بود. به همین دلیل اغلب این روش در شبکه‌های عصبی ساده مورد استفاده قرار می‌گیرد. در روش دوم، برای شبکه‌های با ساختار پیچیده، بازیابی آسان‌تر از کد کردن عناصر است. این روش برای الگوریتم‌های یادگیری در شبکه‌های عصبی عمومی بسیار مناسب می‌باشد. در روش سوم، نیاز است که متغیرها به صورت باینری نمایش داده شوند. در این حالت وقتی که ساختار شبکه پیچیده گردد، طول هر عنصر نیز افزایش می‌یابد. بنابراین فرآیند کد کردن و دیکد کردن خیلی پیچیده خواهد شد.

در این مقاله، چون با شبکه‌های عصبی RBF پیچیده سروکار نداریم از روش برداری استفاده شده است. به منظور کاهش زمان اجرای برنامه شبکه‌های عصبی RBF، از جعبه ابزارهای عمومی Matlab استفاده نخواهد شد. همان‌طور که قبلاً اشاره شد، آموزش شبکه عصبی RBF را می‌توان با انتخاب مقادیر بهینه برای پارامترهای زیر به دست آورد:

- وزن‌های بین لایه‌های پنهان و لایه خروجی (W)
- پارامترهای انتشار تابع اساسی لایه پنهان (α)
- بردارهای مرکز لایه پنهان (c)
- پارامترهای بایاس نرون‌های لایه خروجی (β)

تعداد نرون‌های لایه پنهان در شبکه‌های عصبی بسیار مهم است. بکارگیری نرون‌ها بیشتر از آن چه مورد نیاز است منجر به

1- Overlearning

2 - Fitness Function

3 - Sum Squared Error (SSE)

شکل‌های (۸ و ۹) نمایش داده شده است. سپس در قسمت بعد دادگان سونار به‌طور مفصل توضیح داده شده است و دسته‌بندی‌کننده‌های طراحی شده را بر روی این نوع دادگان نیز آزمایش می‌کنیم.

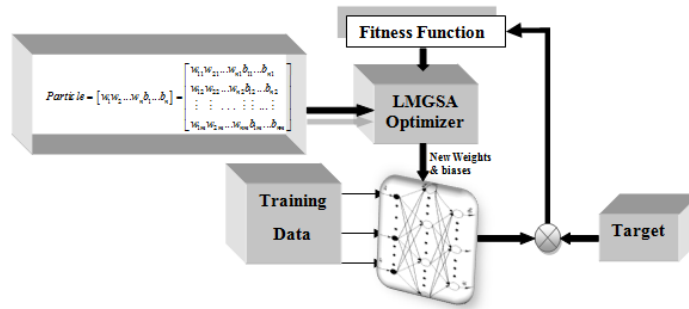
دسته‌بندی، دقت دسته‌بندی‌کننده طراحی شده را نشان می‌دهد و مقادیر میانگین و انحراف معیار حداقل خطا و P-value نشان دهنده قدرت الگوریتم در اجتناب از بهینه محلی می‌باشد (برای توضیحات بیشتر در این زمینه به [۳] مراجعه شود). نمونه‌ای از نتایج این مقایسه به ترتیب برای دادگان Iris و Lenses در

جدول (۱). پارامترها و مقادیر اولیه الگوریتم‌های آموزش

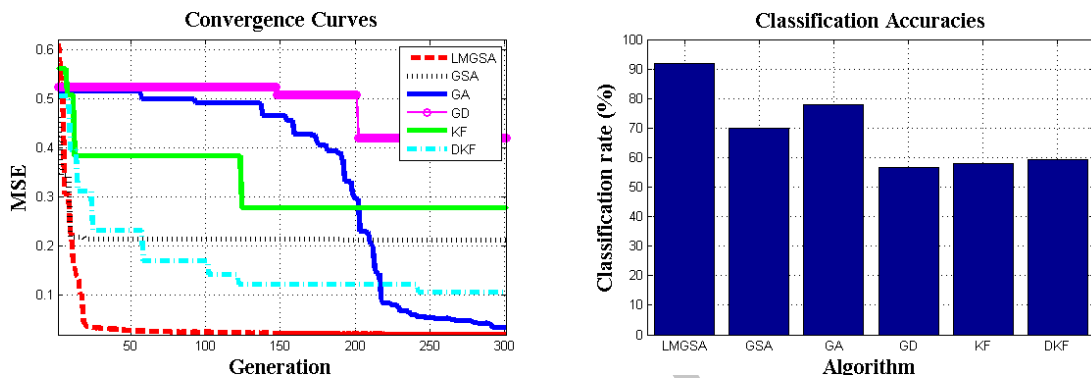
مقدار	پارامتر	الگوریتم
۴۰۰۰	تعداد نسل‌ها	GA
چرخ رولت	نوع انتخاب	
تک نقطه‌ای	نوع ادغام	
۰/۸	نرخ ادغام	
یکنواخت	نوع جهش	GD
۰,۰۵	نرخ جهش	
$\eta = 0.01$	پارامتر یادگیری	KF
$R=40I$	ماتریس کواریانس خطای تخمین حالت	
$Q=40I$	ماتریس کواریانس نویزهای مصنوعی ω_k	
$P0=40I$	ماتریس کواریانس نویزهای مصنوعی V_k	GSA
۶	تعداد جرم‌ها	
۲۰	G_0	
۵۰۰	α	
۶۰	حداکثر تکرار	LMGSA
۱	تعداد جرم‌ها	
۲۰	G_0	
۵۰۰	α	
$(-2t^3 / T^3) + 2$	حداکثر تکرار	
$2t^3 / T^3$	c_1	
	c_2	

جدول (۲). دادگان مورد استفاده در این مقاله

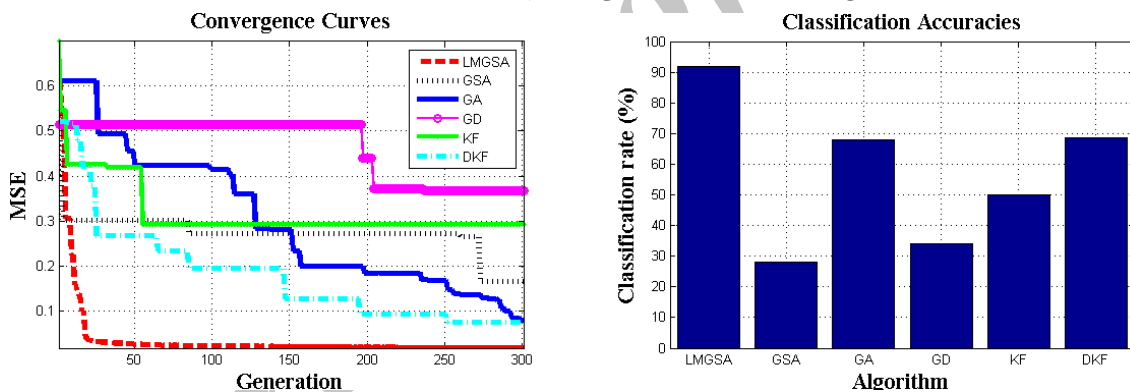
سال	تعداد مثال	تعداد ویژگی	ویژگی مشخصه	وظیفه پیش‌فرض	نام
۱۹۸۸	۱۵۰	۴	چند متغیری	دسته‌بندی	Iris
۱۹۹۰	۲۴	۴	چند متغیری	دسته‌بندی	Lenses



شکل (۷). بلوک دیاگرام کلی دسته‌بندی کننده RBFLMGSA



شکل (۸). نرخ دسته‌بندی و سرعت همگرایی الگوریتم‌های مختلف اعمال شده بر روی دادگان Iris



شکل (۹). نرخ دسته‌بندی و سرعت همگرایی الگوریتم‌های مختلف اعمال شده بر روی دادگان Lenses

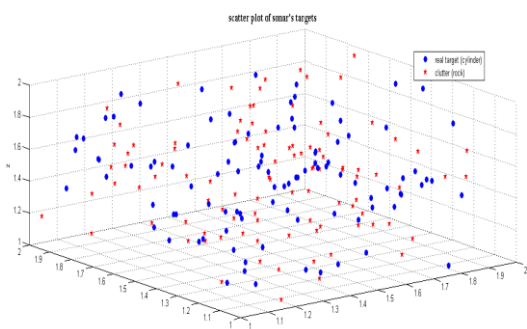
جدول (۳). نتایج حاصل از اعمال دسته‌بندی کننده‌های مختلف بر روی دادگان Iris

MSE (AVE±STD)	P-Values	نرخ دسته‌بندی	الگوریتم
0.1003±8.2720e-04	N/A	90.6667%	RBFLMGSA
0.1512±0.4469	0.0011	85.3522%	RBFSGA
0.1311±0.1176	0.4449e-04	87.8222%	RBFSGA
0.1729±0.2333	7.2111e-04	83.6633%	RBFSGD
0.1577±0.2224	0.4449e-04	85.3333%	RBFSGKF
0.1488±0.7723	1.0008e-11	86.1212%	RBFSGDKF

جدول (۴). نتایج حاصل از اعمال دسته‌بندی کننده‌های مختلف بر روی دادگان Lenses

MSE (AVE±STD)	P-Values	نرخ دسته‌بندی	الگوریتم
0.1283±8.2720e-04	N/A	89.3337%	RBFLMGSA
0.1519±0.0269	0.4449e-04	84.2222%	RBFSGA
0.2011±0.2076	0.8887e-03	82.6666%	RBFSGA
0.3149±0.2965	1.2149e-04	72.2233%	RBFSGD
0.2527±0.1744	1.2228e-08	73.6666%	RBFSGKF
0.1919±0.3123	0.2998e-17	86.1111%	RBFSGDKF

کلاتر یا هدف کاذب را ایفا می‌کند) می‌باشند. همان‌گونه که در شکل (۱۰) مشاهده می‌شود، ویژگی هدف واقعی و کاذب بسیار شبیه هم می‌باشد و با یک دسته‌بندی کننده خطی یا غیرخطی مرتبه پایین نمی‌توان آن‌ها را از یکدیگر تفکیک کرد (برای قابل مشاهده بودن، دادگان به سه بعد کاهش داده شده‌اند).



شکل (۱۰). ترسیم پراکندگی دادگان سونار.

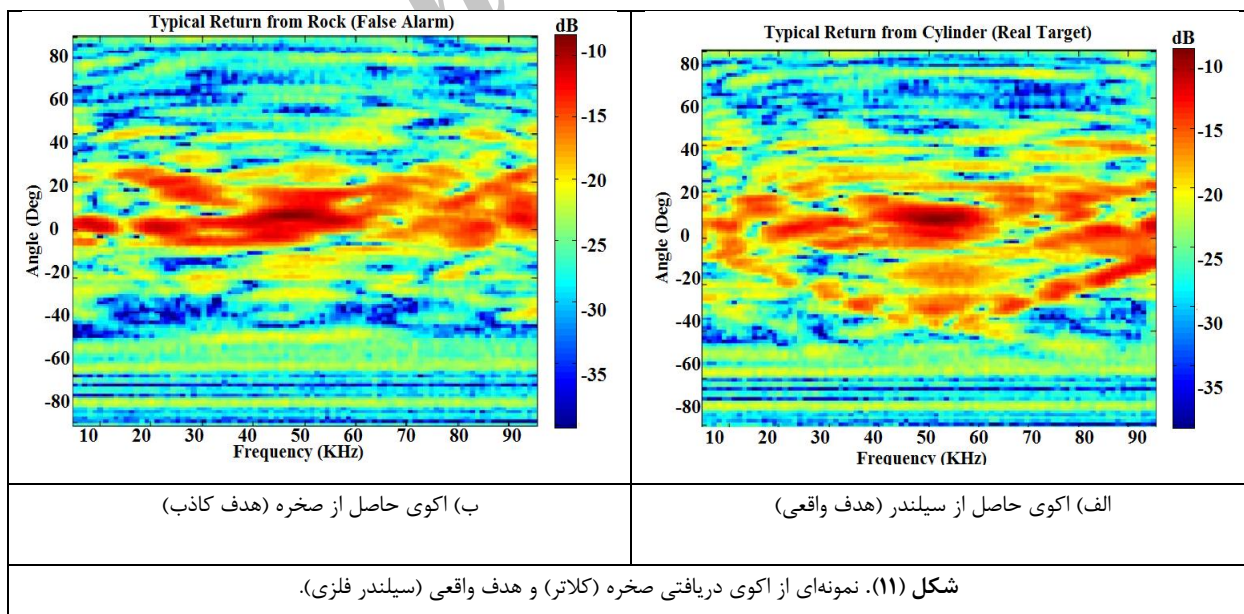
در این آزمایش یک سیلندر فلزی به طول ۵ فوت و یک صخره هم‌اندازه با آن در بستر شنی دریا قرار داده شده‌اند و یک پالس چیرپ FM خطی پهن باند^۲ ($ka=55/6$) به سمت آن‌ها فرستاده شده است. اکوهای برگشتی در فاصله ۱۰ متری از آن‌ها جمع‌آوری شده است. بر اساس SNR اکوی دریافتی از ۱۲۰۰ اکو جمع‌آوری شده ۲۰۸ اکو که SNR آن‌ها بین 4dB تا 15dB است، انتخاب شده‌اند. از این ۲۰۸ اکو، ۱۱۱ عدد مربوط به سیلندر فلزی و ۹۷ عدد مربوط به صخره هستند. شکل (۱۱) نمونه‌هایی از اکوهای دریافتی از صخره و سیلندر فلزی را نشان می‌دهد.

در شکل‌های (۹ - ۸) نمونه‌ای از ۱۰ بار اجرای آزمایش نمایش داده شده است. همان‌گونه که در این شکل‌ها مشاهده می‌شود، دسته‌بندی کننده طراحی شده با الگوریتم LMGSA از نظر نرخ دسته‌بندی و سرعت همگرایی نسبت به الگوریتم‌های دیگر عملکرد بهتری دارد. در مورد دادگان Iris، الگوریتم LMGSA دادگان را با دقت ۹۰٪ دسته‌بندی می‌کند. این در حالی است که بهترین الگوریتم از میان الگوریتم‌های مقایسه (در این مورد GA) دقتی در حدود ۸۷٪ (میانگین ۱۰ بار اجرا) را ارائه می‌نماید. این بهبود عملکرد به خاطر ترکیب توانایی اکتشاف الگوریتم GSA و قدرت بهره‌برداری gbest های هدایت‌شده می‌باشد.

در مورد دادگان Lenses، الگوریتم LMGSA دادگان را با دقت ۸۹٪ دسته‌بندی می‌کند، این در حالی است که بهترین الگوریتم از میان الگوریتم‌های مقایسه (در این مورد DKF) دقتی در حدود ۸۶٪ را ارائه می‌نماید. این مثال به خوبی نشان می‌دهد که الگوریتم LMGSA در مورد مجموعه دادگان با مثال‌های کم (در اینجا ۲۴ عدد) نیز نتایج بسیار خوبی را نشان می‌دهد و دادگان را با نرخ قابل قبول دسته‌بندی می‌کند.

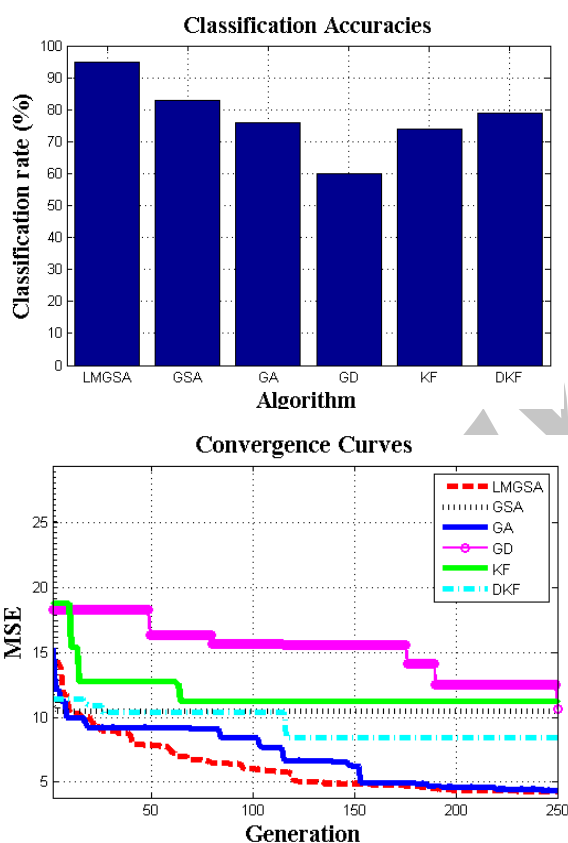
۵-۱- دادگان سونار

دادگان سونار مورد استفاده در این مقاله از آزمایش Gorman و Sejnowski موجود در مراجع [۳۵ و ۳۶] استخراج شده است. در این آزمایش دو نوع اکو^۱ (سیگنال بازگشتی) وجود دارد. اولی مربوط به یک سیلندر فلزی (نقش هدف واقعی را برعهده دارد) و دومی مربوط به یک صخره‌ی هم‌اندازه‌ی سیلندر (نقش



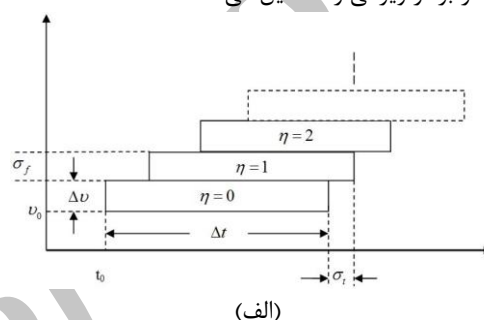
که توسط الگوریتم‌های مختلف آموزش داده شده است، اعمال می‌شود. نتایج در شکل (۱۳) و جدول (۵) نمایش داده شده است.

همان‌گونه که در شکل (۱۳) نمایش داده شده است، الگوریتم LMGSA با ۸۸٪ بهترین عملکرد و الگوریتم GD با ۶۹٪ ضعیف‌ترین عملکرد را به خود اختصاص داده‌اند. با توجه به ماهیت نوسانی بودن و داشتن بیشینه و کمینه‌های محلی زیاد، احتمال گیر کردن در بهینه‌های محلی برای الگوریتمی مانند GD بسیار زیاد می‌باشد. عملکرد ضعیف الگوریتم‌های GD و KF تأیید کننده این مطلب می‌باشد. این در حالی است که الگوریتم‌های LMGSA، GSA و GA با ماهیت تصادفی و عدم استفاده از مشتق دارای عملکرد بهتری نسبت به الگوریتم‌های دیگر هستند.

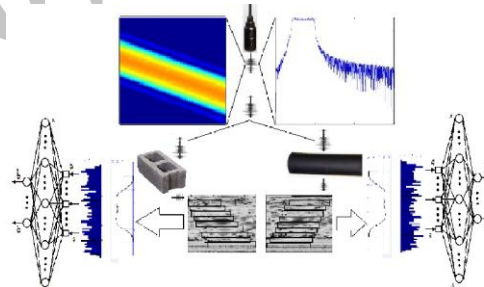


شکل (۱۳). نرخ دسته‌بندی و سرعت همگرایی الگوریتم‌های مختلف اعمال شده بر روی دادگان سونار.

پیش‌پردازش استفاده شده برای به دست آوردن پوش طیفی در شکل (۱۲) نشان داده شده است. در شکل (۱۲-الف) مجموعه‌ای از دریاچه‌های نمونه‌برداری مشاهده می‌شود؛ در شکل (۱۲-ب) مجموعه‌ی دریاچه‌های نمونه‌برداری بر روی طیف‌نگاره دو بعدی تابع فوری‌ی اکوی سونار قرار گرفته‌اند. پوش طیفی توسط جمع‌آوری اثرات هر دریاچه به دست می‌آید. در این آزمایش پوش طیفی از ۶۰ نمونه طیفی که بین ۰ و ۱ نرمالیزه شده‌اند، تشکیل شده است. هر کدام از این اعداد نماینده مجموع انرژی موجود در دریاچه نمونه‌برداری مربوطه می‌باشد؛ به‌طورمثال انرژی موجود در دریاچه اول ($\eta=0$) بعد از نرمالیزه شدن، عدد اول از ۶۰ عدد موجود در بردار ویژگی را تشکیل می‌دهد.



(الف)



(ب)

شکل (۱۲). پیش‌پردازش استفاده شده برای به دست آوردن پوش طیفی.

۵-۲- دسته‌بندی اهداف سوناری

پس از پیش‌پردازش انجام شده بر روی اکوهای بازگشتی سونار و به دست آوردن مجموعه داده نرمالیزه شده بین صفر و یک، در این قسمت از مقاله مجموعه داده با ابعاد ۶۰×۲۰۸ (۲۰۸ نمونه که هر کدام دارای ۶۰ ویژگی می‌باشند) را به شبکه RBF

جدول (۵). نتایج حاصل از اعمال دسته‌بندی‌کننده‌های مختلف بر روی دادگان سونار

MSE (AVE±STD)	P-Values	نرخ دسته‌بندی	الگوریتم
0.1283±8.2720e-04	N/A	88.6667%	RBFLMGSA
0.1519±0.0269	0.0039	85.3522%	RBFGSA
0.2011±0.2076	0.2239e-03	80.8222%	RBFGA
0.3149±0.2965	6.2149e-04	69.6633%	RBFGD
0.2527±0.1744	7.2798e-12	75.3333%	RBFKF
0.1919±0.3123	7.2118e-18	81.1212%	RBFDKF

- [7] P. Auer, H. Burgsteiner, and W. Maass, "A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons," *Neural Networks*, vol. 21, no. 5, pp. 786-795, June 2008.
- [8] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, vol. 1, no. 2, pp. 281-294, 1989.
- [9] N. Karayiannis, "Reformulated Radial Basis Neural Networks Trained by Gradient Descent," *IEEE Transactions on Neural Networks*, vol. 10, no.3, pp. 657-671, 1999.
- [10] C. Liu, H. Wang, and P. Yao, "On Terrain-Aided Navigation for Unmanned Aerial Vehicle using B-spline Neural Network and Extended Kalman Filter," *IEEE Conference on Guidance, Navigation and Control (CGNCC)*, pp. 2258- 2263, 2014.
- [11] D. Simon, "Training Radial Basis Neural Networks with the Extended Kalman Filter," *Neurocomputing*, vol. 48, no. 1-4, pp. 455-475, 2002.
- [12] Q. Zhang and B. Li, "A Low-Cost GPS/INS Integration Based on UKF and BP Neural Network," *IEEE Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 100-107, 2014.
- [13] X. Li, T. Zhang, Z. Deng, and J. Wang, "A Recognition Method of Plate Shape Defect Based on RBF-BP Neural Network Optimized by Genetic Algorithm," *IEEE Conference on Control and Decision*, pp. 3992-3996, 2014.
- [14] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction," Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, vol. 220, no. 4598, pp. 671-680, 1983.
- [16] C. Ozturk and D. Karaboga, "Hybrid Artificial Bee Colony Algorithm for Neural Network Training," *IEEE Congress on Evolutionary Computation (CEC 2011)*, pp. 84-88, 2011.
- [17] J. J. Yu, A.Y. Lam, and V. O. Li, "Evolutionary Artificial Neural Network based on Chemical Reaction Optimization," *IEEE Congress on Evolutionary Computation (CEC 2011)*, pp. 2083-2090, 2011.
- [18] S. Mirjalili and A. S. Sadiq, "Magnetic Optimization Algorithm for Training Multi-Layer Perceptron," *IEEE Conference on Communication Software and Networks (ICCSN 2011)*, pp. 42-46, 2011.
- [19] R. C. Green, L. Wang, and M. Alam, "Training Neural Networks Using Central Force Optimization and Particle Swarm Optimization: Insights and Comparisons," *Expert System with Application*, vol. 39, no. 1, pp. 555-563, 2012.
- [20] P. Moallem and N. Razmjoooy, "A Multi-Layer Perceptron Neural Network Trained by Invasive Weed Optimization for Potato Color Image Segmentation," *Trends in Applied Sciences Research*, vol. 7, no. 6, pp. 445-455, 2012.
- [21] L. A. Pereira, L. C. Afonso, J. P. Papa, Z. A. Vale, C. C. Ramos, D. S. Gastaldello, and A. N. Souza, "Multilayer Perceptron Neural Networks Training Through Charged System Search and Its Application for Non-Technical Losses Detection on Innovative Smart Grid Technologies," *IEEE PES Conference on Latin America (ISGT LA 2013)*, pp. 1-6, 2013.
- [22] L. Pereira, D. Rodrigues, P. Ribeiro, J. Papa, and S. A. Weber, "Social-Spider Optimization-Based Artificial Neural Networks Training and its Applications for Parkinson's Disease Identification," *IEEE Symposium on Computer-based Medical Systems (CBMS 2014)*, pp. 14-17, 2014.
- [23] E. Uzlu, M. Kankal, A. Akpınar, and T. Dede, "Estimates of Energy Consumption in Turkey using Neural Networks with the Teaching-Learning-based Optimization Algorithm,"

از طرفی دیگر می‌توان مشاهده نمود در این آزمایش الگوریتم GSA و LMGSa (که از مشتقات GSA می‌باشد)، به دلیل دارا بودن قدرت اکتشاف زیاد، در این نوع دادگان عملکرد بهتری دارند. همانگونه که بیان شد دادگان سوناری به دلیل پوشش دادن کل فضای جستجو برای دسته‌بندی نیازمند الگوریتمی می‌باشند که در مرحله اکتشاف قدرتمند و الگوریتم‌های خانواده GSA نسبت به الگوریتم‌های فراابتکاری دیگر در این زمینه بسیار بهتر هستند.

۶- نتیجه‌گیری

در این مقاله از یک روش فرا ابتکاری جدید به نام الگوریتم جستجوی گرانشی تطبیقی با هدایت بهینه عمومی به منظور آموزش شبکه‌های عصبی تابع شعاعی پایه استفاده شد. برای سنجش عملکرد دسته‌بندی‌کننده طراحی شده از مجموعه دادگان Iris، Lenses و Sonar استفاده گردید و نتایج حاصل با الگوریتم‌های معیار GD، KF، DKF، GA و GSA مقایسه شد. نتایج نشان داد که الگوریتم LMGSa قادر به ارائه نتایجی بسیار بهتر از نظر سرعت همگرایی و دقت دسته‌بندی در مقایسه با الگوریتم‌های معیار است. نتایج حاصله نشان دهنده آن است که با استفاده از دسته‌بندی‌کننده طراحی شده با LMGSa (RBFLMGSa) نرخ دسته‌بندی برای مجموعه دادگان Iris، Lenses و Sonar نسبت به الگوریتم GSA کلاسیک به ترتیب ۴/۹۶۶۷، ۵/۱۲۵ و ۳/۲۵۰۳ درصد افزایش پیدا کرد.

۷- مراجع

- [1] E. G. Carayannis, F. J. David, and M. P. Eftymiopoulos, "Cyber-Development, Cyber-Democracy and Cyber-Defense Challenges, Opportunities and Implications for Theory, Policy and Practice," *Springer Science Business and Media*, pp. 5-22, 2014.
- [2] M. R. Mosavi, M. Khishe and M. Aghababaie, "Modeling and Mitigation of Active Sonar Clutter", *Noshahr University of Marine Science and Technology*, 2015. (In Persian)
- [3] M. R. Mosavi, M. Khishe, A. Ghamgosar and M. J. Ghalandari, "Classification of Sonar Data Set using the Gray Wolf Optimizer Algorithm", *Journal of Electronics Industries*, Vol.7, No.1, pp.27-41, 1395. (In Persian)
- [4] M. R. Mosavi, M. Khishe and E. Ebrahimi, "Classification of Sonar Targets using OMKC, Genetic Algorithm and Statistical Moments," *Journal of Advances in Computer Research*, vol. 7, no. 1, pp. 143-156, 2016.
- [5] V. Abedifar, M. Eshghi, S. Mirjalili, and S. M. Mirjalili, "An Optimized Virtual Network Mapping using PSO in Cloud Computing," *21st Iranian Conference on Electrical Engineering*, pp. 1-6, 2013.
- [6] L. S. Nguyen, D. Frauendorfer, M. S. Mast, and D. Gatica-Perez, "Hire Me: Computational Inference of Hirability in Employment Interviews based on Nonverbal Behavior," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 1018-1031, 2014.

- Energy, vol. 75, pp. 295-303, 2014.
- [24] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a Biogeography-based Optimizer Train Your Multi-Layer Perceptron," *Journal of Information Sciences*, vol. 269, pp. 188-209, June 2014.
- [25] N. Muangkote, K. Sunat, and S. Chiewchanwattana, "An Improved Grey Wolf Optimizer for Training q -Gaussian Radial Basis Functional-link Nets," 2014 International Computer Science and Engineering Conference (ICSEC), pp. 209-214, 2014.
- [26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [27] M. R. Mosavi, M. Khishe, and A. Moridi, "Sonar Dataset Classification using Hybrid PSO-GSA Method," *Marine Technology*, vol. 3, no. 1, pp. 1-14, 2016.
- [28] O. Olorunda and A. P. Engelbrecht, "Measuring Exploration/Exploitation in Particle Swarms using Swarm Diversity," *IEEE World Congress on Computational Intelligence*, pp. 1128-1134, 2008.
- [29] L. Lin and M. Gen, "Auto-Tuning Strategy for Evolutionary Algorithms: Balancing between Exploration and Exploitation," *Soft Computing*, vol. 13, no. 2, pp. 157-168, 2009.
- [30] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training Feedforward Neural Networks using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125-11137, 2012.
- [31] R. E. Precup, R. C. David, E. M. Petriu, and M. B. Radac, "Adaptive GSA-Based Optimal Tuning of PI Controlled Servo Systems with Reduced Process Parametric Sensitivity, Robust Stability and Controller Robustness," *IEEE Transactions on Cybernetics*, vol. 44, no. 11, pp. 1997-2009, 2014.
- [32] B. Yu and X. He, "Training Radial Basis Function Networks with Differential Evolution," *IEEE Conference on Granular Computing*, pp. 369-372, 2006.
- [33] M. Gauci, T. J. Dodd, and R. Groß, "Why 'GSA: A Gravitational Search Algorithm' is not Genuinely based on the Law of Gravity," *Natural Computing*, vol. 11, no. 4, pp. 719-720, 2012.
- [34] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary Gravitational Search Algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727-745, 2010.
- [35] <http://archive.ics.uci.edu/ml/datasets>.
- [36] R. P. Gorman and T. J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," *Neural Networks*, vol. 1, pp. 75-89, 1988.

The Use of Radial Basis Function Networks Based on Leader Mass Gravitational Search Algorithm for Sonar Dataset Classification

M. R. Mosavi*, M. Khishe

*Iran University of Science and Technology

(Received: 17/10/2015, Accepted: 01/08/2016)

ABSTRACT

Due to the very similar physical characteristics of active sonar echoes, which are related to true and false targets, the task of classification and distinguishing them from each other has become difficult and complex issues for the researchers and industrialists in this field. Radial Basis Function Neural Networks (RBF NN) is one of the most used artificial NNs in the classification of the real-world objects. Training is an important part of the development of this type of network that it has been highly regarded in recent years. For RBF NN training, the use of recursive and gradient descent methods has traditionally been common. However, poor classification accuracy, trapped in local minima, and low convergence speed are among the disadvantages of these methods. In recent years, the use of heuristic and meta-heuristic methods has been very common to overcome these disadvantages. To overcome the GSA's weakness in the exploitation phase, this paper introduces and uses Leader Mass Gravitational Search algorithm (LMGSA) in the training of RBF NNs. The results show that the designed classifier provides better results than benchmark classifiers in all areas. To a comprehensive comparison, the designed classifier is compared to GSA, Gradient Descent (GD), Genetic Algorithm (GA), Kalman Filter (KF), and Extended Kalman Filter (EKF) algorithms through three benchmark datasets. The evaluation criteria are convergence rate, the probability of being caught in local minima, and classification accuracy. Finally, as a practical application, sonar dataset is classified by the designed classifier.

Keywords: Classification, Sonar dataset, Radial basis function, Leader Mass Gravitational Search algorithm

* Corresponding Author Email: m_mosavi@iust.ac.ir