Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهش‌های نظری و کاربردی هوش ماشینی

سال دوم، شماره دوم، پاییز و زمستان ۱۴۰۳، صفحات ۱۴۳ - ۱۲۵

# یادگیری شبکه عصبی با روش لونبرگ-مارکوارت بهبودیافته

رضا یظهری کرمانی[۱]، محمد ملائی امام‌زاده[۲*]، مجتبی برخورداری یزدی[۳]

[۱] دانشجوی کارشناسی ارشد، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان، کرمان، ایران

[۲] استادیار، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان، کرمان، ایران

[۳] دانشیار، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان، کرمان، ایران

## چکیده

دراین مقاله روشی جدید برای افزایش سرعت همگرایی و کارایی روش لونبرگ-مارکوارت ارائه می‌شود. روش لونبرگ-مارکوارت یک روش مبتنی بر نیوتن است که در بهینه‌سازی و تعیین ضرایب شبکه‌های عصبی دارای کارایی مناسبی در مقایسه با سایر روش‌ها ازجمله روش پس‌انتشارخطا است. بااین‌حال، عملکرد این روش به‌طور قابل‌توجهی به انتخاب ضریب دمپینگ مناسب بستگی دارد. ازجمله روش‌های مختلف تعیین ضریب دمپینگ، روش جستجو خطی مارکوارت و روش‌های مبتنی بر نرم خطا و مبتنی بر نرم ژاکوبین می‌باشند که در این مقاله با بررسی نقاط ضعف و قوت این روش‌ها، یک روش ترکیبی جهت افزایش سرعت همگرایی ارائه می‌شود. درروش پیشنهادی بازه جستجوی ضریب دمپینگ و درنتیجه مقدار پارامتر نرخ تنظیم کاهش‌یافته است. با انجام این اصلاحات دقت جستجوی ضریب دمپینگ افزایش‌یافته و سرعت همگرایی روش پیشنهادی افزایش می‌یابد. به‌منظور ارزیابی روش پیشنهادی، یادگیری یک شبکه‌عصبی برای شناسایی یک تابع غیرخطی پیچیده، یک مسأله رگرسیون و یک مسأله طبقه‌بندی مورد شبیه‌سازی قرارگرفته و نتایج به‌دست‌آمده نشان می‌دهد که روش پیشنهادی در مقایسه با سایر روش‌های مورد بررسی، دارای کارایی مناسبی بوده و توانسته است خطای یادگیری را به حد قابل قبولی کاهش داده و سرعت همگرایی بالاتری داشته باشد.

مقاله پژوهشی

تاریخ دریافت:
۱۴۰۳/۹/۱۹
تاریخ پذیرش:
۱۴۰۳/۱۲/۸

کلیدواژه‌ها:
ضریب دمپینگ، الگوریتم لونبرگ-مارکوارت، الگوریتم نیوتن، یادگیری شبکه‌های عصبی، بهینه‌سازی غیرخطی

نویسنده‌مسئول:
molaie@uk.ac.ir

: 10.22034/ABMIR.2025.22500.1082

E-ISSN: 2821-2037

## ۱- مقدمه

شبکه‌های عصبی به‌عنوان روشی با کارایی بالا در کاربردهای متنوعی همچون طبقه‌بندی، خوشه‌بندی، تشخیص الگو، پیش‌بینی و مدل‌سازی در بسیاری از حوزه‌های علمی و عملی مورداستفاده قرارگرفته‌اند. این شبکه‌ها زیرمجموعه‌ای از یادگیری ماشین هستند و ازنظر عملکرد و دقت، در مقایسه با مدل‌های رگرسیون و روش‌های آماری سنتی، برتری قابل‌توجهی از خود نشان می‌دهند [۱]. یکی از جدی‌ترین چالش‌هایی که شبکه‌های عصبی با آن‌ها مواجه هستند مسئله یادگیری و تنظیم پارامترها است. یادگیری در شبکه‌های عصبی را می‌توان یک مسئله بهینه‌سازی دانست که در آن، هدف به دست آوردن ضرایب بهینه به‌گونه‌ای است یک تابع هزینه خاص مرتبط با عملکرد مسأله بهینه (کمینه یا بیشینه) شود. روش‌های بهینه‌سازی، عمدتاً به دودسته کلی تقسیم می‌شوند: روش‌های مبتنی بر گرادیان [۲] و روش‌های هوش جمعی [۳]. روش‌های بهینه‌سازی مبتنی بر گرادیان از قدرت جهت‌یابی گرادیان به‌منظور یافتن راه‌حل‌های بهینه استفاده می‌کنند مانند روش پس انتشار خطا. در مقابل، روش‌های هوش جمعی از رفتار جمعی مشاهده‌شده در طبیعت، به‌ویژه از حرکات هماهنگ موجودات اجتماعی، الهام می‌گیرند مانند الگوریتم ازدحام ذرات. محدودیت اصلی روش‌های مبتنی بر گرادیان نیاز به داشتن مشتق تابع هزینه است که در صورت دسترسی به مشتق، به دلیل سادگی و حجم کم محاسبات و درعین‌حال وجود تضمین همگرایی، این روش‌ها دارای کارایی مناسبی می‌باشند.

به دلیل مشتق‌پذیری خروجی شبکه‌های عصبی، می‌توان از روش‌های مبتنی بر گرادیان در زمینه یادگیری این شبکه‌ها استفاده کرد و به دلیل استفاده این روش‌ها از قدرت جهات‌یابی گرادیان، می‌توان انتظار داشت که روش‌های مبتنی بر گرادیان در زمینه یادگیری این شبکه‌ها عملکرد بهتری نسبت به روش‌های تکاملی داشته باشند.

به‌طورکلی روش‌های بهینه‌سازی مبتنی بر گرادیان به دودسته اصلی روش‌های مرتبه اول و مرتبه دوم تقسیم می‌شوند.

روش‌های بهینه‌سازی مرتبه اول تنها بر مشتق اول تابع هدف تکیه‌دارند مانند الگوریتم گرادیان کاهشی [٤]. این روش‌ها نسبت به روش‌های مرتبه دوم دارای محاسبات ساده‌تر و کمتری می‌باشند که در حالت کلی باعث می‌شود ازنظر زمان، حافظه و پیچیدگی محاسبات، کم‌هزینه‌تر باشند، که برای طیف گسترده‌ای از مسائل بسیار کاربردی می‌باشند. اما در عین مزایایی که دارند، دارای معایبی مانند وابستگی به تعیین نرخ یادگیری و همگرایی کندتر نسبت به روش‌های مرتبه دوم هستند.

روش‌های بهینه‌سازی مرتبه دوم هم مشتق اول و هم مشتق دوم (ماتریس هسین) تابع هدف را لازم دارند مانند الگوریتم نیوتن [٥]. اگرچه این روش‌ها ازنظر محاسباتی هزینه‌برتر هستند و نیاز به زمان، حافظه و محاسبات بیشتری دارند، اما مهم‌ترین عیب آن‌ها این است که ممکن است در نقاط دور از نقطه بهینه همگرا نشوند، اما در نقاط نزدیک به جواب بهینه کارآمدتر باشند [٦].

مشکلات بعدی روش نیوتن حجم محاسباتی بالا و مسأله تکینگی است. برای رفع مشکل بار محاسباتی روش نیوتن، ابتدا روش گاوس- نیوتن و سپس روش لونبرگ-مارکوارت ارائه‌شده که روش لونبرگ-مارکوارت علاوه بر رفع مشکل حجم محاسبات، با افزودن پارامتری به نام ضریب دمپینگ مسأله تکینگی و همچنین مشکل واگرایی در نقاط دور از نقطه بهینه (که در صورت منفی معین شدن ماتریس هسین در نقاط ابتدایی جستجو ایجاد می‌شود) را حل کرد و با ارائه استراتژی تعیین ضریب دمپینگ متغیر، روش لونبرگ-مارکوارت را به ترکیبی از روش گرادیان و روش نیوتن تبدیل کرد. در نقاط ابتدایی جستجو با انتخاب ضریب دمپینگ بزرگ رفتار روش به گرادیان نزدیک‌تر و مشکل واگرایی حل می‌شود و در نقاط انتهایی جستجو با انتخاب ضریب دمپینگ کوچک رفتار روش به نیوتن شبیه‌تر شده و سرعت همگرایی روش در نزدیکی جواب بهینه افزایش می‌یابد [٦]. به همین دلیل کارایی روش لونبرگ-مارکوارت به‌شدت به استراتژی مورداستفاده در انتخاب ضریب دمپینگ وابسته است. درگذشته روش‌های متفاوتی برای تعیین ضریب دمپینگ ارائه‌شده است [۷].

در سال ۱۹٤٤ در [۸] دو روش برای تعیین ضریب دمپینگ معرفی شد: ۱) انتخاب مقداری ثابت و ۲) با استفاده از نرم دوم ماتریس حساسیت در هر تکرار. در سال ۱۹٦۳ در [۹] مارکوارت روشی

برای تخمین پارامترهای غیرخطی درمجموعه‌ای از معادلات معرفی کرده است. در این مقاله روش جستجوی خطی برای تعیین مقدار ضریب دمپینگ معرفی شد. در این روش در هر تکرار ضریب دمپینگ طی یک جستجوی خطی به‌گونه‌ای تنظیم می‌شود تا اطمینان حاصل شود که در مرحله بعد تخمین دقیق‌تری به وجود آمده باشد. در [۱۰] برای تعیین ضریب دمپینگ از نرم خطا استفاده‌شده است. بزرگ‌ترین مشکل استفاده از این روش، در نقاط دور از نقطه بهینه است. چرا که در این نقاط با افزایش خطا مقدار ضریب دمپینگ خیلی بزرگ است و باعث کاهش شدید طول گام و درنتیجه کند شدن الگوریتم می‌شود.

در [۱۰] و در [۱۱] به معرفی و بررسی روش‌هایی که از ماتریس حساسیت برای به دست آوردن ضریب دمپینگ استفاده می‌کنند پرداخته‌شده است. یکی از این روش‌ها، استفاده از ضرایب قطر اصلی این ماتریس است. در این حالت امکان از بین رفتن پارامتر وجود دارد و این روش را ناکارآمد می‌کند. روش دیگر استفاده از بزرگ‌ترین مقدار قطر اصلی است در این حالت نیز شانس زیادی برای از بین رفتن پارامتر وجود دارد. به این دلیل که حدس اولیه پارامترها ممکن است در مناطقی باشد که ضریب دمپینگ کافی تولید نمی‌کنند. نتایج این مقالات نشان می‌دهد که روش معرفی‌شده توسط مارکوارت می‌تواند تا حد زیادی سرعت الگوریتم را زمانی که در ناحیه یک دره است، افزایش دهد.

در این مقاله ابتدا با بررسی فرضیات و تئوری مورداستفاده در ارائه روش لونبرگ–مارکوارت نشان داده‌شده است که روش ضریب دمپینگ مبتنی بر خطا باید دارای کارایی بیشتری باشد و در ادامه هم نشان داده‌شده است که ضریب دمپینگ نه‌تنها باید مضربی از نرم خطا باشد بلکه باید در محاسبه آن از ضریبی متغیر برای نرم خطا (برحسب تکرار شبیه‌سازی) مورداستفاده قرار گیرد. سپس برای تعیین این ضریب، از روش جستجوی خطی پیشنهادشده است. ازآن جایی‌که اطلاحات انجام‌شده به‌منظور هم‌خوانی هر چه بیشتر با تئوری و فرضیات مسأله است انتظار می‌رود که این اصلاحات سرعت همگرایی روش و کارایی آن را بهبود ببخشد.

## ۲- طرح مسئله و مدل‌سازی ریاضی

در این بخش به مدل‌سازی ریاضی مسئله مورد بررسی و معرفی روابط پرداخته می‌شود.

### ۲-۱ الگوریتم گرادیان‌کاهشی

الگوریتم گرادیان کاهشی متداول‌ترین و پرکاربردترین روش بهینه‌سازی مرتبه اول است که در آن، برای رسیدن به جواب بهینه (کمینه کردن تابع هزینه $F$) متغیرهای بهینه‌سازی ($w$) را در خلاف جهت گرادیان حرکت داده (به‌روزرسانی کرده) که این مطلب در رابطه زیر بیان‌شده است:

$$w_{k+1} = w_k - \alpha_k \times \nabla F(w_k) \qquad (۱)$$

که در آن $w_k$ و $w_{k+1}$ به ترتیب مقدار فعلی و مقدار به‌روزرسانی شده پارامترهای بهینه‌سازی، $\alpha_k$ طول گام و $\nabla F$ مشتق تابع هزینه نسبت به پارامترهای $w_k$ است. البته اگر هدف از بهینه‌سازی یادگیری شبکه عصبی باشد به $\alpha_k$ به‌جای طول گام، نرخ یادگیری هم گفته می‌شود. برای به دست آوردن $\nabla F$ باید از تابع هزینه نسبت به‌تمامی پارامترها طبق رابطه (۲) مشتق گرفته شود:

$$\nabla F(w_k) = \left[\frac{\partial F}{\partial w_1} \frac{\partial F}{\partial w_2} \frac{\partial F}{\partial w_3} \cdots \frac{\partial F}{\partial w_n}\right] \qquad (۲)$$

که در اینجا n تعداد متغیرها و $w_i$ متغیر i- ام مسئله بهینه‌سازی است.

$$w = [w_1 \; w_2 \; w_3 \cdots w_n] \qquad (۳)$$

روش‌های مرتبه اول به دلیل سادگی و کارایی، برای حل طیف وسیعی از مسائل مناسب هستند. بااین‌حال، وابستگی به نرخ یادگیری و همگرایی کندتر، از معایب این روش‌ها به شمار می‌رود.

### ۲-۲ الگوریتم نیوتن

درروش نیوتن مشتق مرتبه دوم تابع هزینه (ماتریس هسین) برای تعیین جهت بهینه‌سازی و طول گام استفاده می‌شود. درروش گرادیان (مرتبه اول) مقدار به‌روزرسانی متناسب با $\nabla F$ است و با نزدیک شدن به جواب بهینه سرعت همگرایی کاهش می‌یابد که این مساله درروش نیوتن (مبتنی بر گرادیان مرتبه دوم) حل‌شده است و رابطه زیر برای به‌روزرسانی متغیرها استفاده می‌شود:

$$w_{k+1} = w_k - H(w_k)^{-1} \times \nabla F(w_k) \qquad (٤)$$

که در آن $H$ ماتریس هسین است و به‌صورت زیر محاسبه می‌شود:

$$H \equiv \nabla^2 F = \frac{\partial \nabla F}{\partial w} = \frac{\partial^2 F}{\partial w^2} \quad (۵)$$

با استفاده از $H$ در قانون به‌روزرسانی روش نیوتن، سرعت همگرایی روش دررسیدن به جواب بهینه (نسبت به روش گرادیان) بیشتر شده است، اما برای محاسبه ماتریس $H$، محاسبات پیچیده‌تری لازم است. این الگوریتم با در نظر گرفتن اطلاعات مرتبه دوم، جهت حرکت و اندازه حرکت را به‌طور مناسب‌تری تعیین می‌کند که در سرعت همگرایی و نزدیک شدن به جواب بهینه مفید است. از معایب الگوریتم نیوتن می‌توان به روابط پیچیده لازم برای محاسبه مشتقات مرتبه دوم موجود در ماتریس هسین اشاره کرد و همچنین نیاز به محاسبه معکوس آن و همین‌طور حساسیت روش نسبت به انتخاب اولیه مقدار متغیرها را ذکر کرد. همین‌طور در این الگوریتم در صورت غیر محدب بودن مسئله بهینه‌سازی ممکن است به نقاط بهینه محلی یا نقاط زینی همگرا شود. ازآنجایی‌که الگوریتم نیوتن از معکوس ماتریس هسین برای محاسبه جهت حرکت در هر تکرار استفاده می‌کند، اگر ماتریس هسین در نقطه خاصی دارای مقدار ویژه صفر یا مقدار ویژه بسیار کوچک باشد، این روش ممکن است با تکینگی مواجه شود. برای رفع این مشکل تحقیقات و پژوهش‌هایی انجام‌شده و روش‌هایی همانند روش گاوس–نیوتن، لونبرگ–ماکوارت و ... معرفی‌شده‌اند. که بدون محاسبه مستقیم ماتریس هسین و با تقریب آن از حجم محاسبات کاسته و هم چنین مشکل تکینگی را حل کرده‌اند.

## ۳-۲ الگوریتم گاوس–نیوتن

الگوریتم گاوس–نیوتن به دنبال تخمین ماتریس هسین بدون محاسبه صریح آن است، که فرآیند بهینه‌سازی را کارآمدتر می‌کند و درعین‌حال مزایای اطلاعات مرتبه دوم را نیز حفظ می‌کند [۱۲]. برای محاسبه ماتریس هسین باید $\nabla^2 F$ محاسبه گردد. F تابع هزینه در مسئله بهینه‌سازی است و در این مقاله تابع هزینه به‌صورت زیر تعریف می‌شود:

$$F \equiv F(w_k) = \frac{1}{2} E(w_k)^T E(w_k) \quad (۶)$$

که در آن E بردار خطا است و با توجه به مسئله تعریف می‌شود.

$$E_k \equiv E(w_k) \quad (۷)$$

در ادامه $\nabla F$ به‌صورت زیر به دست می‌آید:

$$\nabla F \equiv \nabla F(w_k) = \frac{\partial E^2}{\partial w} = 2 \times E \times \frac{\partial E}{\partial w} \quad (۸)$$

که $\frac{\partial E}{\partial w}$ مشتق خطاها نسبت به تمام پارامترها است که همان ماتریس ژاکوبین است. پس می‌توان رابطه (۸) را به‌صورت زیر نوشت:

$$J \equiv J(w_k) \square \frac{\partial E}{\partial w_k} \Rightarrow \nabla F(w_k) = \frac{\partial E^2}{\partial w} = 2 \times E \times J \quad (۹)$$

در ادامه می‌توان ماتریس $H$ را به‌صورت زیر به دست آورد:

$$H \equiv \nabla^2 F = \frac{\partial^2 E^2}{\partial w^2} = \frac{\partial (E \times J)}{\partial E} \square \frac{\partial E}{\partial w} + \frac{\partial (E \times J)}{\partial J} \square \frac{\partial J}{\partial w} \quad (۱۰)$$

$$H \equiv \nabla^2 F = \frac{\partial^2 E^2}{\partial w^2} = JJ^T + E \square \frac{\partial^2 E}{\partial w^2} \quad (۱۱)$$

با محاسبه مشتقات ذکرشده در رابطه (۱۰) می‌توان آن را به دو قسمت (اولی فاقد مشتق دوم و دومی شامل مشتق دوم) تقسیم کرد. با توجه به این‌که کارایی روش نیوتن برای نواحی نزدیک به مقدار بهینه است، در این نواحی مقدار خطا نزدیک به صفر می‌شود. به دلیل این‌که در جمله $E \square \frac{\partial^2 E}{\partial w^2}$ (که شامل مشتق دوم است) مقدار خطا مستقیماً وجود دارد و در نزدیکی جواب بهینه تقریباً صفر و قابل صرف‌نظر است لذا با چشم‌پوشی ازجمله دوم (مشتق دوم) در ماتریس هسین می‌توان بدون نیاز به انجام محاسبات پیچیده مشتق مرتبه دوم $E \square \frac{\partial^2 E}{\partial w^2}$ و فقط با استفاده از مشتق مرتبه اول $J$ ماتریس هسین را با دقت مناسبی به‌صورت زیر تقریب زد:

$$H_{GN} = \lim_{E \to \circ} JJ^T + E \square \frac{\partial^2 E}{\partial w^2} \Rightarrow H_{GN} = JJ^T \quad (۱۲)$$

$$w_{k+1} = w_k - H_{GN}^{-1} \nabla F = w_k - \left(JJ^T\right)^{-1} \times J^T E_k \quad (۱۳)$$

که این روش به نام روش گاوس–نیوتن نامیده شده است [۱۳].

### ۴-۲ الگوریتم لونبرگ-مارکوارت

در الگوریتم گاوس-نیوتن هم مانند الگوریتم نیوتن ممکن است تکینگی رخ دهد. برای رفع این مشکل و ایجاد تعادل بین نقاط قوت روش‌های نیوتن و گرادیان کاهشی الگوریتم لونبرگ-مارکوارت یک ضریب دمپینگ به نام ($\lambda$) معرفی می‌کند که پایداری فرآیند بهینه‌سازی را بهبود می‌بخشد [۱۴]. ماتریس تقریب زده‌شده درروش لونبرگ-مارکوارت پس از اعمال ضریب $\lambda$ را می‌توان با رابطه (۱۴) نشان داد.

$$H_{LM} = JJ^T + \lambda I \qquad (۱۴)$$

درروش لونبرگ-مارکوارت با تنظیم مقدار $\lambda$ می‌توان بین الگوریتم گرادیان کاهشی و نیوتن جابه‌جا شد به این صورت که اگر مقدار $\lambda$ بزرگ انتخاب شود نحوه به‌روزرسانی مانند روش گرادیان کاهشی می‌شود:

$$if\ \lambda \square \|J\|^2 \Rightarrow H_{LM} \square \lambda I\ , \quad w_{K+1} = w_K - \lambda^{-1} JE \qquad (۱۵)$$

اگر مقدار کوچک انتخاب شود نحوه به‌روزرسانی مانند روش گاوس-نیوتن می‌شود:

$$if\ \lambda \square \|J\|^2 \Rightarrow H_{LM} \square JJ^T = H_{GN} \qquad (۱۶)$$

## ۳- روش‌های تعیین ضریب دمپینگ

همان‌طور که قبلاً اشاره شد، با انتخاب ضریب دمپینگ، روش لونبرگ-مارکوارت می‌تواند بین دو روش نیوتن و روش گرادیان تغییر وضعیت دهد. در این راستا بهترین استراتژی در انتخاب مقدار ضریب دمپینگ به این صورت است که در گام‌های اولیه که ممکن است روش نیوتن به دلیل امکان عدم مثبت معین بودن ماتریس H و همچنین به دلیل بزرگ بودن طول گام، واگرا شود، مقدار ضریب دمپینگ را بزرگ انتخاب شود تا رفتار روش لونبرگ-مارکوارت شبیه روش گرادیان شده و با رابطه (۱۵) طول گام آن‌هم در این حالت که متناظر با معکوس ضریب دمپینگ است کاهش پیدا کند. درنتیجه روش لونبرگ-مارکوارت در گام‌های اول با انتخاب ضریب دمپینگ بزرگ مشابه گرادیان با طول گام کوچک بوده و همگرا خواهد بود. به‌تدریج با نزدیک شدن به جواب بهینه، دیگر مشکل عدم مثبت معین بودن H درروش روش نیوتن وجود ندارد و از طرفی سرعت همگرایی روش گرادیان کم می‌شود و این حالت با کاهش دادن مقدار ضریب دمپینگ مشابه رابطه (۱۶) روش لونبرگ-مارکوارت به روش گاوس-نیوتن شبیه شده و با سرعت مناسبی به سمت جواب بهینه همگرا می‌شود.

### ۳-۱ تعیین ضریب دمپینگ با استفاده از روش جستجوی خطی مارکوارت[1]

در ارائه روش لونبرگ-مارکوارت برای اولین بار در [۸] ضریب دمپینگ ثابت فرض شده و در [۹] برای تنظیم مقدار بهینه ضریب دمپینگ در حین حل مسئله درروش لونبرگ-مارکوارت از روش جستجوی خطی مشابه الگوریتم ۱ استفاده شد. در این روش در هر تکرار ضریب دمپینگ طی یک جستجوی خطی به‌گونه‌ای تنظیم می‌شود تا اطمینان حاصل شود که در مرحله بعد تخمین دقیق‌تری به دست آید. در هر تکرار مقدار تابع هزینه به ازای مقدار $\lambda$ و $\frac{\lambda}{\beta}$ محاسبه می‌شود که درروش مارکوارت $\beta = 10$ در نظر گرفته‌شده است.

در صورت کاهش تابع هزینه مقدار جدید $\lambda$ در تکرار بعد استفاده می‌شود اما اگر تابع هزینه افزایش یابد مقدار $\lambda$ در $\beta$ ضرب می‌شود. این عمل تا زمان ادامه پیدا می‌کند که تابع هدف کاهشی شود. در این صورت مقدار جدید $\lambda$ در تکرار بعد استفاده می‌شود.

الگوریتم (۱): الگوریتم تعیین ضریب دمپینگ به روش مارکوارت

Let β=10.
Let $\lambda^{(r-1)}$ denote the value of λ that was calculated from the previous iteration.
Let $\lambda^{(0)} = 10^{(2)}$
Compute $\varphi^r\left(\lambda^{(r-1)}\right)$ And $\varphi^r\left(\frac{\lambda^{(r-1)}}{\beta}\right)$
If $\varphi^r\left(\frac{\lambda^{(r-1)}}{\beta}\right) \le \varphi^r$, Let $\lambda^{(r)} = \frac{\lambda^{(r-1)}}{\beta}$

[1] Damping Factor using Line Search: DFLS

If $\varphi^r\left(\frac{\lambda^{(r-1)}}{\beta}\right) > \varphi^r$ , And $\varphi^r\left(\lambda^{(r-1)}\right) \le \varphi^r$ Let $\lambda^{(r)} = \lambda^{(r-1)}$

Else if $\varphi^r\left(\frac{\lambda^{(r-1)}}{\beta}\right) > \varphi^r$ , And $\varphi^r\left(\lambda^{(r-1)}\right) > \varphi^r$ , increase $\lambda$ by successive multiplication by β until for smallest k, $\varphi^r\left(\lambda^{(r-1)}\beta^k\right) \le \varphi^r$. Let $\lambda^{(r)} = \lambda^{(r-1)}\beta^k$

Return $\lambda^{(r)}$

## ۳-۲ تعیین ضریب دمپینگ با استفاده از نرم خطا[1]

در مرجع [۱۱] برای تعیین ضریب دمپینگ از نرم خطا استفاده‌شده است. درروش تعیین ضریب دمپینگ مبتنی بر نرم خطای جستجو از نرم خطای جستجو برای انتخاب مقدار ضریب دمپینگ در هر مرحله استفاده می‌شود. تقریب ماتریس هسین با استفاده از این روش طبق رابطه (۱۷) صورت می‌گیرد:

$$H_E = JJ^T + \|E\| I \qquad (۱۷)$$

در مرجع [۱۵] و [۱۰] با بررسی روش نرم خطا روشی با استفاده از ضریب دمپینگ ۰,۱ نرم خطا معرفی‌شده و باعث بهبود عملکرد این روش شده است.

$$H_E = JJ^T + 0/1\|E\| I \qquad (۱۸)$$

## ۳-۳ تعیین ضریب دمپینگ با استفاده از نرم ژاکوبین[2]

درروش تعیین ضریب دمپینگ مبتنی بر نرم ماتریس ژاکوبین از نرم ماتریس ژاکوبین برای انتخاب مقدار ضریب دمپینگ در هر مرحله استفاده می‌شود. تقریب ماتریس هسین با استفاده از این روش طبق رابطه (۱۹) صورت می‌گیرد:

$$H_J = JJ^T + \|J\| I \qquad (۱۹)$$

مزیت دو روش معرفی‌شده ازنظر حجم محاسبات و زمان اجرا به پیچیدگی مسئله اضافه نمی‌کنند و روش‌هایی ساده برای محاسبه مقدار ضریب دمپینگ هستند [۱۰].

## ٤- بررسی نتایج روش‌های قبلی

در این مقاله نتایج شبیه‌سازی الگوریتم لونبرگ-مارکوارت برای یک شبکه عصبی پرسپترون با یک‌لایه میانی و ۱۰ نورون در لایه میانی در ۳۰۰ تکرار برای تابع سینوس نمایش داده می‌شود.

یادگیری شبکه عصبی به‌منظور کمینه کردن تابع هزینه زیر است:

$$F = E^T E = \sum (y_r[k] - y_n[k])^2 \qquad (۲۰)$$

که در آن $y_n$ خروجی شبکه عصبی و $y_r$ خروجی سیستم واقعی است که توسط شبکه عصبی، شناسایی و مدل‌سازی می‌شود. رابطه بین خروجی $y$ و ورودی $x$ در سیستم واقعی به‌صورت زیر به‌عنوان یک رابطه غیرخطی نمایش داده‌شده است.

$$y = \sin(x) \ , \ x \in [0 \ , \ 2\pi] \qquad (۲۱)$$

در این شبکه نشان داده‌شده در شکل ۱، وزن‌ها و بایاس‌های لایه میانی و وزن‌ها و بایاس لایه خروجی به‌عنوان پارامترهای بهینه‌سازی تعریف‌شده است.

$$W_1 = \begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,10} \\ b_{1,1} \\ \vdots \\ b_{1,10} \end{bmatrix} \quad W_2 = \begin{bmatrix} w_{2,1} \\ \vdots \\ w_{2,10} \\ b_{2,1} \end{bmatrix} \quad W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \qquad (۲۲)$$

که در آن $w_{1,j}$ وزن لینک واسط بین ورودی و نرون j-ام لایه میانی است. همچنین $b_{1,j}$ بایاس مورداستفاده در تابع فعال‌سازی نرون j-ام است که این تابع تانژانت هایپربولیک در نظر گرفته‌شده است و به‌صورت زیر تعریف‌شده است.

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \qquad (۲۳)$$

همچنین $w_{2,j}$ وزن لینک واسط بین خروجی j-ام نرون لایه میانی تا نرون لایه خروجی است و $b_{2,1}$ بایاس مورداستفاده در رابطه نرون خروجی است.

با استفاده از سه روش معرفی‌شده مارکوارت، نرم خطا و نرم ژاکوبین، مسأله یادگیری شبکه‌عصبی برای شناسایی تابع غیرخطی (۲۱) مطابق شکل ۱ انجام‌شده است. در شکل ۲ نتایج بهینه‌سازی برحسب تکرار و در شکل ۳ نتایج برحسب زمان اجرای روش آمده است.

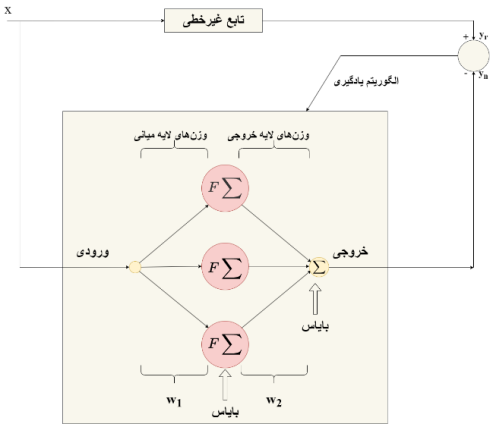
همان‌طور در شکل ۲ مشاهده می‌شود، تعیین ضریب دمپینگ با روش مبتنی بر نرم ژاکوبین (DFNJ) سرعت همگرایی و خطای نهایی مناسبی ندارد. روش مارکوارت (جستجوی خطی با DFLS:β=۱۰) دارای نتایج بهتری است اما روش مبتنی بر نرم

[1] Damping Factor using Norm of Error: DFNE

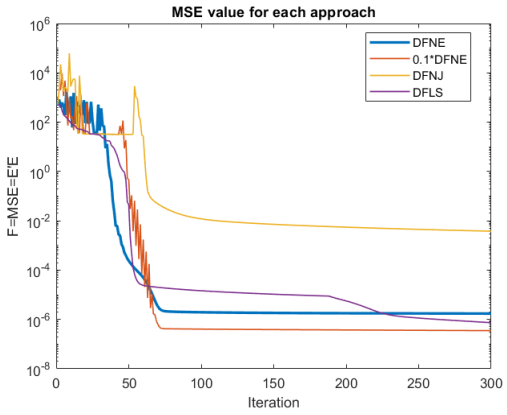
[2] Damping Factor using Norm of Jacobian: DFNJ

خطا (DFNE) دارای همگرایی سریع‌تر و خطای نهایی کمتری نسبت به سایر روش‌ها است که البته در نتایج آن گاهی نوسان هم دیده می‌شود.

در شکل ۳ مقدار تابع هدف برای هرکدام از روش‌ها برحسب زمان اجرا در هر تکرار نشان داده‌شده است. می‌توان مشاهده کرد که روش نرم خطا علاوه بر خطای نهایی کمتر به زمان کمتری نسبت به روش مارکوارت نیاز دارد.

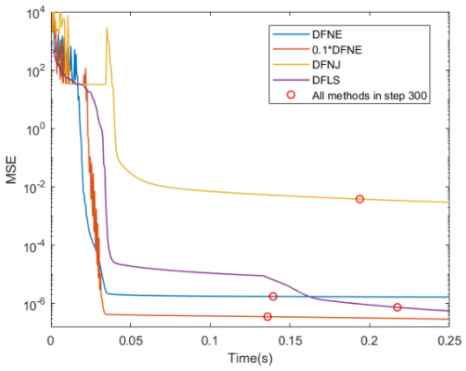



**شکل (۱): یادگیری شبکه عصبی به‌منظور یادگیری تابع غیرخطی**




**شکل (۲): خطای یادگیری برحسب تکرار برای تابع غیرخطی (۲۱)**

در این مقاله سعی شده است تا با ترکیب روش تعیین ضریب دمپینگ مبتنی بر نرم خطا و روش جستجوی خطی مورداستفاده درروش مارکوارت مشکل نوسانی بودن روش مبتنی بر نرم خطا را حل کرده و سرعت همگرایی را هم نسبت به سایر روش‌ها بهبود بخشید.




**شکل (۳): خطای یادگیری برحسب زمان اجرا برای تابع غیرخطی (۲۱)**

## ۵- روش پیشنهادی برای تعیین ضریب دمپینگ[1]

با مقایسه رابطه (۱۱) با رابطه (۱٤) می‌توان چنین نتیجه گرفت که درروش لونبرگ-مارکوارت از تقریب زیر برای ساده‌سازی استفاده‌شده است:

$$\begin{cases} H \equiv JJ^T + E\left[\frac{\partial^2 E}{\partial w^2}\right] \Rightarrow \lambda I \approx E\left[\frac{\partial^2 E}{\partial w^2}\right] \\ H_{LM} = JJ^T + \lambda I \end{cases} \qquad (۲٤)$$

از رابطه (۲٤) می‌توان چنین نتیجه گرفت که درروش مارکوارت، ماتریس قطری $\lambda I$ باید جایگزین ماتریس $E\left[\frac{\partial^2 E}{\partial w^2}\right]$ شود. ازآنجایی‌که ماتریس قطری $\lambda I$ نمی‌تواند با ماتریس $E\left[\frac{\partial^2 E}{\partial w^2}\right]$ مساوی باشد، درروش پیشنهادی فرض می‌شود که حداقل نرم آن‌ها برابر هم باشد:

$$\left\|\lambda I^*\right\| = \left\|E\left[\frac{\partial^2 E}{\partial w^2}\right]\right\| \Rightarrow \lambda^* = \|E\|\left[\left\|\frac{\partial^2 E}{\partial w^2}\right\|\right] \qquad (۲۵)$$

اگر از مقدار $\lambda^*$ رابطه فوق استفاده شود خطای بین تقریب $H_{LM}$ و ماتریس هسین اصلی کمترین مقدار خود می‌شود و انتظار می‌رود که تقریب $H_{LM}$ به ازای $\lambda^*$، بهترین نتایج را به همراه داشته باشد. البته درروش مارکوارت سعی شده تا به‌جای رابطه فوق، مقدار ضریب دمپینگ را با انجام عملیات جستجوی محدود به دست آورد ولی لزوماً این ضریب دمپینگ، مقدار بهینه نخواهد بود.

[1] Damping Factor using the Proposed Method: DFPM

از طرف دیگر درروش تعیین ضریب دمپینگ مبتنی بر نرم خطا، در ماتریس $H_E$ می‌توان چنین نتیجه گرفت که در این روش از تقریب زیر استفاده‌شده است:

$$\begin{cases} H \equiv JJ^T + E\dfrac{\partial^2 E}{\partial w^2} \\ H_E = JJ^T + 0.1\|E\| \end{cases} \Rightarrow \begin{cases} \left\|\dfrac{\partial^2 E}{\partial w^2}\right\| \approx 0.1 \\ \lambda_E = 0.1\|E\| \end{cases} \qquad (۲۶)$$

که در این روش سعی شده است تا ضریب دمپینگ متناظر با خطا تغییر کند و هیچ تنظیم دیگری ندارد و اجرای آن بسیار ساده است و همچنین در شبیه‌سازی‌های شکل ۲، مشاهده می‌شود این روش نسبت به روش مارکوارت (که برای تعیین ضریب دمپینگ از جستجوی خطی استفاده می‌کند) دارای سرعت همگرایی بیشتری است و این نتایج بهتر به دلیل همخوانی بهتر مقدار ضریب دمپینگ به‌دست‌آمده از این روش (۲۶) با ضریب دمپینگ بهینه به‌دست‌آمده در رابطه (۲۵) است. البته در نتایج شکل ۲ و ۳ مشاهده می‌شود که نوسان‌هایی وجود دارد. دلیل هم آن است که در این تقریب (۲۶)، $\dfrac{\partial^2 E}{\partial w^2}$ برابر با مقدار ثابتی فرض شده است و در طول اجرای روش از ضریب ثابت ۰،۱ بجای آن استفاده می‌شود.

در این مقاله با در نظر گرفتن موارد فوق و این‌که ضریب دمپینگ بهینه باید تا حد امکان مقدار رابطه (۲۵) را نمایندگی کند، درروش پیشنهادی ابتدا پارامتر $\eta_K$ به‌صورت زیر تعریف‌شده است:

$$\eta_K \approx \left\|\frac{\partial^2 E}{\partial w^2}\right\| \Rightarrow E\frac{\partial^2 E}{\partial w^2} \cong \eta_K \|E\| I \qquad (۲۷)$$

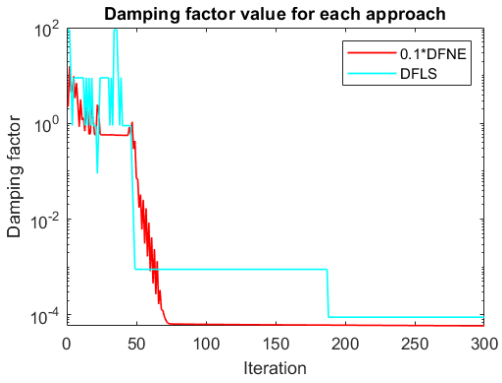
ازآنجایی‌که محاسبه $\dfrac{\partial^2 E}{\partial w^2}$ نیاز به مشتق مرتبه دوم دارد و محاسبات آن بسیار پیچیده است درروش پیشنهادی، به‌جای محاسبه مقدار $\eta_k$ از رابطه (۲۷)، پیشنهادشده است که مقدار $\eta_k$ را در هر گام توسط یک مسأله جستجوی خطی به دست آید و روش پیشنهادی به‌صورت زیر ارائه می‌شود:

$$H_{new} = JJ^T + \eta_k \|E\| I \qquad (۲۸)$$

که ماتریس $H_{new}$ تخمین زده‌شده به‌اندازه کافی به ماتریس هسین واقعی نزدیک است و از طرفی نیاز به انجام محاسبات پیچیده $\dfrac{\partial^2 E}{\partial w^2}$ برطرف شده است.

درروش جستجوی خطی مورداستفاده درروش مارکوارت از نرخ تنظیم β=۱۰ استفاده‌شده است که $\lambda$ به‌صورت ۱۰ برابر در هر جستجو افزایش یا کاهش می‌یابد. درروش پیشنهادی چون بجای $\lambda$ درروش مارکوارت از $\eta_k \|E\|$ استفاده‌شده است، و این دو پارامتر ( $\lambda$ درروش مارکوارت و $\eta$ درروش پیشنهادی) دارای بازه متفاوتی می‌باشند، لذا بهتر است از نرخ تنظیم جدیدی (متناظر با این روش) استفاده شود.

در شکل ٤ مقدار ضریب دمپینگ در هر تکرار برای دو روش نرم خطا و روش مارکوارت نمایش داده‌شده است. مشاهده می‌شود که مقدار $\lambda_E = 0.1\|E\|$ تقریباً به‌صورت پیوسته (باکمی نوسان) تغییر می‌کند ولی درروش مارکوارت مقدار $\lambda$ (از روش جستجوی خطی محدود) فقط چهار مقدار گسسته دارد و تغییر آن ناگهانی است لذا نمی‌تواند بهترین $\lambda$ ممکن را محاسبه کند. علت این تغییرات ناگهانی انتخاب مقدار بزرگ β=۱۰ است که باید بتواند کل بازه $10^{-5} < \lambda < 10^{2}$ را جستجو کند. ولی درروش پیشنهادی این مقاله به‌جای $\lambda$ از مقدار $\eta\|E\|$ استفاده‌شده است که تغییرات $\|E\|$ تا حدی از تغییرات خیلی زیاد در بازه $\eta$ جلوگیری می‌کند و سبب می‌شود که ناحیه مورد جستجو برای $\eta$ خیلی کمتر از ناحیه جستجوی $\lambda$ شود. همین مسأله سبب شده است که درروش پیشنهادی جستجو $\eta$ دقیق‌تر انجام شود و انتظار می‌رود نتایج بهتری حاصل شود.




**شکل (٤): مقدار ضریب دمپینگ در هر تکرار در تعیین ضریب دمپینگ به روش نرم خطا و مارکوارت**

با توجه به شکل ٤ درروش مارکوارت بازه تغییرات ضریب دمپینگ به‌صورت زیر است:

$$\left.\begin{aligned}\lambda_{200}&=0.9\times10^{-5}\\ \lambda_{0}&=0.9\times10^{2}\end{aligned}\right\}\Rightarrow\left|\frac{\lambda_0}{\lambda_{200}}\right|=10^{7} \quad (۲۹)$$

درروش پیشنهادی داریم:

$$\left.\begin{aligned}\lambda_{200}&=\eta_{200}\left\|E\right\|_{200}=0.9\times10^{-5}\\ \lambda_{0}&=\eta_{0}\left\|E\right\|_{0}=0.9\times10^{2}\end{aligned}\right\}\Rightarrow \quad (۳۰)$$

$$\frac{\eta_{200}\lambda_0}{\eta_0\lambda_{200}}=\frac{\lambda_{200}}{\lambda_0}\times\frac{\left\|E\right\|_0}{\left\|E\right\|_{200}}$$

حال اگر بازه تغییرات نرم خطا را مشابه شکل ۳ و ٤ در نظر بگیریم:

$$\left.\begin{aligned}\left\|E\right\|_{200}&\square\ 0.6\times10^{-5}\\ \left\|E\right\|_{0}&\square\ 2\times10^{0}\end{aligned}\right\}\Rightarrow\frac{\left\|E\right\|_0}{\left\|E\right\|_{200}}=3.3\times10^{5} \quad (۳۱)$$

بنابراین از رابطه (۳۰) می‌توان نتیجه گرفت:

$$\left|\frac{\eta_0}{\eta_{200}}\right|=\frac{\lambda_0}{\lambda_{200}}\square\frac{\left\|E\right\|_{200}}{\left\|E\right\|_0}=30 \quad (۳۲)$$

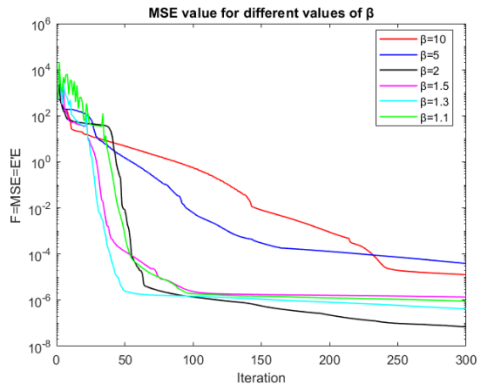
لذا درروش پیشنهادی بازه تغییرات $\eta$ ( $\frac{\eta_{\max}}{\eta_{\min}}\square\ 30$ ) بسیار کم‌تر از بازه تغییرات $\lambda$ درروش مارکوارت ( $\frac{\lambda_{\max}}{\lambda_{\min}}\square\ 10^{7}$ ) است و درنتیجه دیگر لازم نیست درروش پیشنهادی از نرخ تنظیم بزرگی (مانند β=۱۰ درروش جستجوی مارکوارت) استفاده شود. درنتیجه با انتخاب β کوچک‌تر، ضمن این‌که کل بازه $\eta$ مورد جستجو قرار می‌گیرد، اندازه پرش‌های $\eta$ بسیار کمتر شده و درنتیجه جستجو دقیق‌تر انجام می‌شود.

در شکل ۵، نتایج روش پیشنهادی به ازای مقادیر متفاوت نرخ تنظیم ضریب دمپینگ (β) آورده شده است. همچنین در شکل ٦ مقدار ضریب دمپینگ به ازای مقادیر مختلف نرخ تنظیم ضریب دمپینگ (β) رسم شده است.

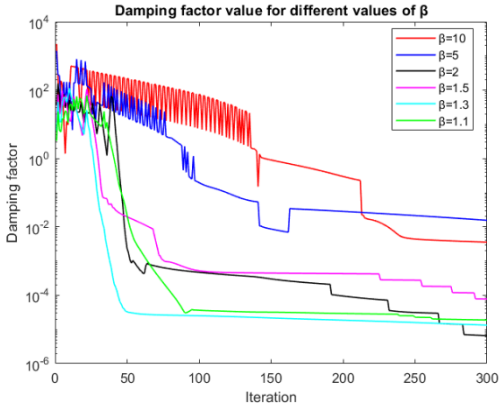
همان‌طور که در شکل ٦ مشاهده می‌شود بهترین نتایج حاصل به ازای مقدار β=۲، روش پیشنهادی کمترین خطای حالت دائم را نشان می‌دهد. با افزایش مقدار β بیشتر از ۲ همگرایی خطای یادگیری کمتر می‌شود که به دلیل عدم دقت در جستجوی خطی به ازای βهای بزرگ است که این مسأله در شکل ٦ هم به‌صورت نوسانات زیاد در $\eta$ به‌دست‌آمده به ازای مقادیر بزرگ β=۵ و β=۱۰ دیده می‌شود و این نشان می‌دهد که مسأله جستجوی خطی به ازای مقادیر بزرگ نرخ تنظیم β>۲ به‌درستی انجام نمی‌شود.

در شکل ۷ نتایج حاصل از تعیین ضریب دمپینگ به روش جستجوی خطی مارکوارت، روش مبتنی بر نرم خطا، روش مبتنی بر نرم ژاکوبین و روش پیشنهادی (β=۲)، رسم شده است. همان‌طور که مشاهده می‌شود درروش پیشنهادی دیگر نوسانات روش نرم خطا وجود ندارد و همچنین سرعت همگرایی افزایش‌یافته و خطای نهایی کمتر شده است که نشان‌دهنده عملکرد خوب روش پیشنهادی نسبت به سایر روش‌ها است.

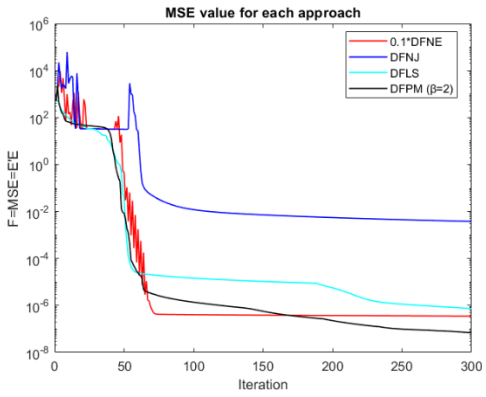
در شکل ۸ مقدار ضریب دمپینگ به ازای روش‌های متفاوت نمایش داده‌شده است و نشان می‌دهد که ضریب دمپینگ به‌دست‌آمده درروش پیشنهادی به‌خوبی توانسته است کل بازه موردنظر را در حین مسأله بهینه‌سازی پوشش دهد و همین مسأله سبب نتایج مناسب در شکل ۷ شده است.



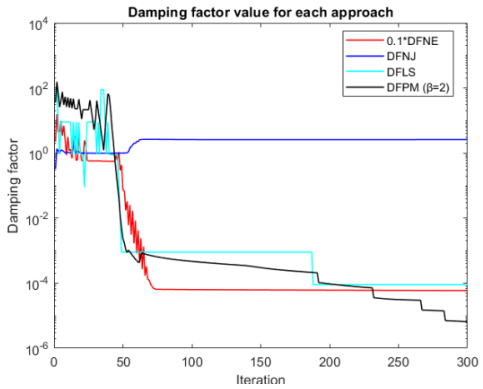

**شکل (۵): نتایج به‌دست‌آمده به ازای مقادیر متفاوت نرخ تنظیم ضریب دمپینگ درروش پیشنهادی برای تابع غیرخطی (۲۱)**




**شکل (٦): مقدار ضریب دمپینگ در هر تکرار به ازای مقادیر متفاوت نرخ تنظیم ضریب دمپینگ برای تابع غیرخطی (۲۱)**




شکل (۷): مقایسه نتیجه روش پیشنهادی با روش‌های پیشین برای تابع غیرخطی (۲۱)



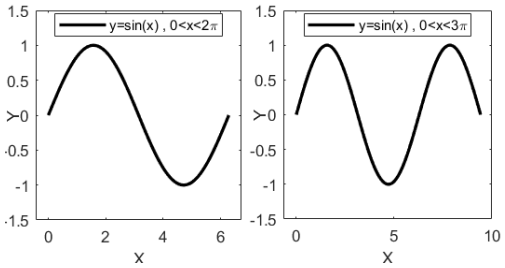

شکل (۸): مقایسه مقدار ضریب دمپینگ درروش پیشنهادی و روش‌های پیشین برای تابع غیرخطی (۲۱)

در شکل ۹ تابع $y=\sin(x)$ برای $0\le x\le 2\pi$ که یک تابع غیرخطی با پیچیدگی متوسط است، رسم شده است و در کنار آن تابع $y=\sin(x)$ برای $0\le x\le 3\pi$ نیز رسم شده است. مشاهده می‌شود با افزایش دامنه x، پیچیدگی این تابع غیرخطی نیز افزایش‌یافته است. نتایج قبلی برای شناسایی تابع $y=\sin(x)$ برای $0\le x\le 2\pi$ انجام‌شده است. در ادامه برای بررسی اثر پیچیدگی مسأله بر مقدار بهینه β (درروش پیشنهادی) و همچنین بررسی عملکرد روش پیشنهادی در مقایسه با سایر روش‌ها، شبیه‌سازی‌ها برای شناسایی تابع (۳۳) انجام‌شده است.

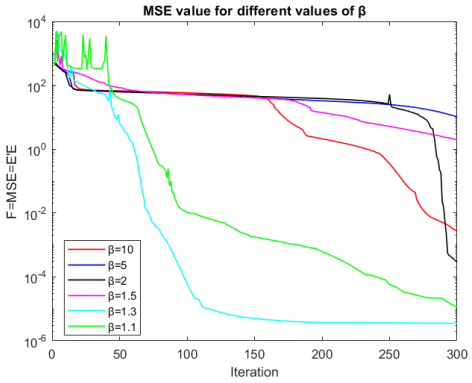
$$y=\sin(x)\quad ,\ x\in[0\ ,\ 3\pi] \qquad (۳۳)$$

در شکل ۱۰، نتایج روش پیشنهادی به ازای مقادیر مختلف نرخ تنظیم برای تابع (۳۳) نمایش داده‌شده است.

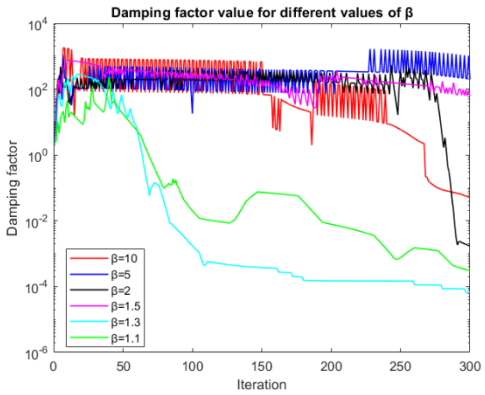
با پیچیده‌تر شدن مسئله مقدار $\frac{\partial^2 E}{\partial w^2}$ هم غیرخطی‌تر شده و درنتیجه پیدا کردن $\eta_K$ به‌نحوی‌که مطابق رابطه (۲۷) بتواند معادل $\left\|\frac{\partial^2 E}{\partial w^2}\right\|$ باشد سخت می‌شود. لذا برای این مسأله انتظار می‌رود که نرخ جستجوی β کوچک‌تر باشد تا جستجو دقیق‌تر انجام شود. در شکل ۱۰ مشاهده می‌شود که نتایج بهینه‌سازی برای β<۱،۳ بهتر از نتایج β>۱،۵ است. علت این مسأله می‌تواند در شکل ۱۱ دیده شود. مشاهده می‌شود که ضریب دمپینگ به ازای نرخ جستجوی بالا (β>۱،۵ ) بیشتر حالت نوسانی و جهش‌های بالا دارد و نمی‌تواند به‌درستی به دست آید. برای β<۱،۳ مقدار به‌دست‌آمده برای ضریب دمپینگ لامبدا حالت نوسانی ندارد و به همین دلیل در شکل ۱۰ مشاهده می‌شود که به ازای β=۱،۱ و β=۱،۳ نتایج بهینه‌سازی نسبت به β>۱،۵ بهتر است.




شکل (۹): توابع مورد بررسی معرفی‌شده در روابط (۲۱) و (۳۳)




شکل (۱۰): نتایج به‌دست‌آمده به ازای مقادیر مختلف نرخ تنظیم ضریب دمپینگ برای تابع غیرخطی (۳۳)

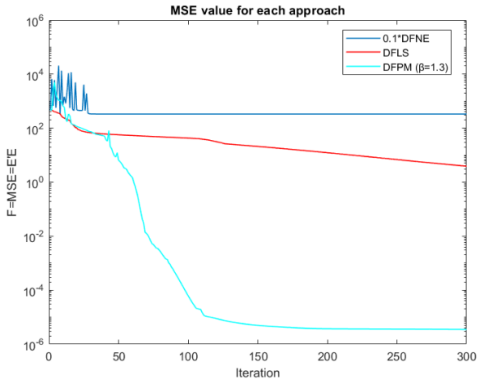



**شکل (۱۱): مقدار ضریب دمپینگ در هر تکرار به ازای مقادیر مختلف نرخ تنظیم ضریب دمپینگ برای تابع غیرخطی (۳۳)**

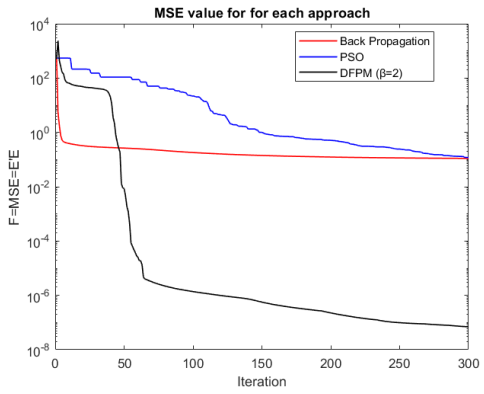
شکل ۱۲ نتیجه شبیه‌سازی روش‌های مختلف برای رابطه (۳۳) را نشان می‌دهد. همان‌طور که مشاهده شد روش پیشنهادی برای تابع (۲۱) عملکرد بهتری نسبت به سایر روش‌ها داشت، با پیچیده‌تر شدن تابع مورد بررسی روش نرم خطا عملکرد مناسبی ندارد و همان‌طور که در شکل ۱۲ مشاهده می‌شود کارایی خود را در این حالت از دست می‌دهد. از طرفی روش مارکوارت به‌کندی به سمت جواب بهینه حرکت می‌کند، اما روش پیشنهادی با سرعت مناسبی به سمت جواب بهینه همگرا شده است و عملکرد مناسبی را نشان می‌دهد.

در انتها نتایج ناشی از روش پیشنهادی و روش‌های رایج پس‌انتشارخطا و ازدحام ذرات برای یادگیری شبکه‌عصبی به‌منظور مدل‌سازی تابع (۲۱) و (۳۳) به ترتیب در شکل‌های ۱۳ و ۱٤ نمایش داده‌شده است. که عملکرد بسیار خوب روش پیشنهادی ازنظر سرعت همگرایی و خطای نهایی را در مقایسه با روش‌های رایج نشان می‌دهد.

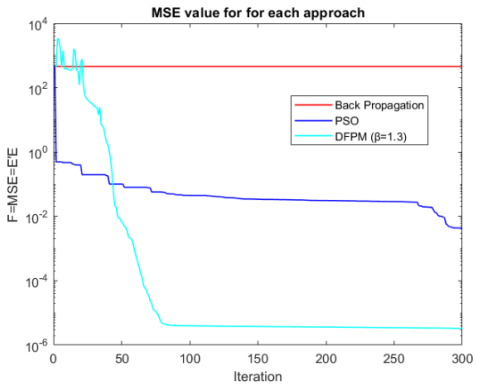
در انتها لازم است تاکید شود که β بهینه با توجه به پیچیدگی مسأله مقداری بین ۱,۱ تا ۲ پیشنهاد می‌شود که طراح باید با توجه به پیچیدگی مسأله موردنظر β را از این بازه انتخاب کند.




**شکل (۱۲): مقایسه روش‌های معرفی‌شده برای تابع غیرخطی (۳۳)**




**شکل (۱۳): مقایسه روش پیشنهادی با روش پس انتشار خطا و روش ازدحام ذرات برای تابع غیرخطی (۲۱)**




**شکل (۱٤): مقایسه روش پیشنهادی با روش پس انتشار خطا و روش ازدحام ذرات برای تابع غیرخطی (۳۳)**

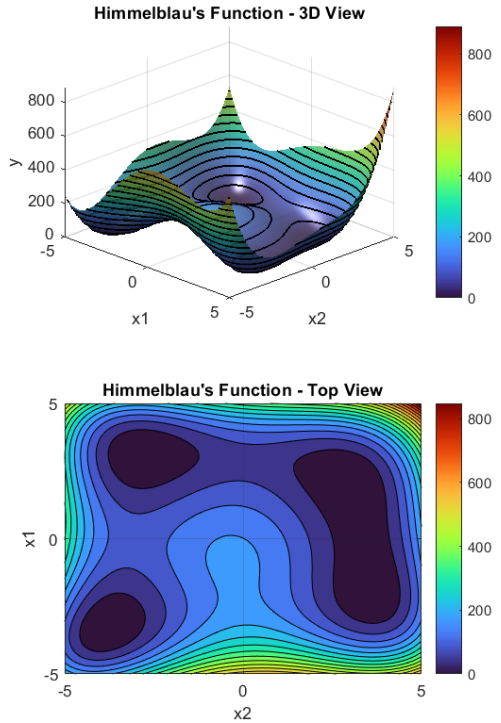
## ٦- کاربرد روش پیشنهادی برای توابع پیچیده‌تر

در این بخش به‌منظور ارزیابی و بررسی بیشتر کارایی روش پیشنهادی در مواجهه با مسائل پیچیده‌تر، شبیه‌سازی‌ها روی سه مسأله دیگر انجام‌شده و در ادامه نتایج مورد بررسی قرار می‌گیرد.

### ٦-١ مدل‌سازی تابع غیرخطی هیمل‌بلاو[1]

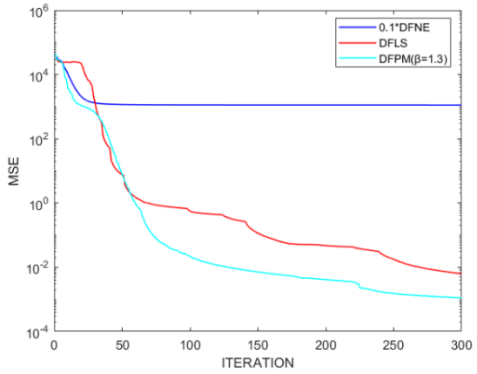
برای شبیه‌سازی اول، تابع هیمل بلاو انتخاب و شبیه‌سازی‌شده است. همان‌طور که در شکل ١٥ مشاهده می‌شود این تابع دارای ٤ نقطه زینی است [١٦]:

$$y = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 + -7)^2 \qquad (٣٤)$$

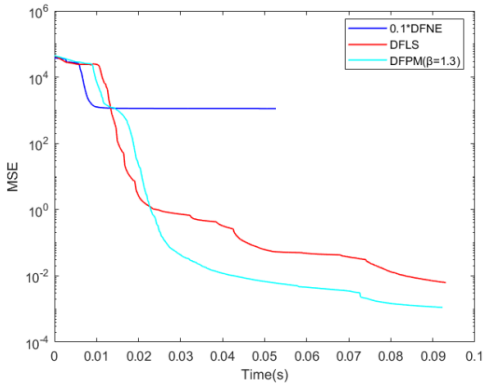



**شکل (١٥): نمایش سه‌بعدی تابع رابطه (٣٤)**

برای شناسایی این تابع مشابه بخش‌های قبل، از یک شبکه عصبی با یک‌لایه میانی شامل ١٠ نورون استفاده‌شده است و لایه ورودی شامل ٢ نورون است که درمجموع شبکه عصبی مورداستفاده شامل ٤١ وزن است. روش پیشنهادی و سایر روش‌های مورد بررسی برای یادگیری این شبکه عصبی استفاده‌شده است که نتایج شبیه‌سازی روش پیشنهادی در مقایسه با سایر روش‌ها برای این تابع در شکل ١٦ و ١٧ قابل‌مشاهده است. که می‌توان عملکرد مناسب روش پیشنهادی (DFPM) هم برحسب تکرار و هم برحسب زمان اجرا در مقایسه با دو روش نرم خطا (DFNE) و مارکوارت (DFLS) را مشاهده کرد. در شکل ١٨ می‌توان مشاهده کرد که روش پیشنهادی بازه بزرگ‌تر و پیوسته‌ای از مقادیر ضریب دمپینگ را نسبت به روش نرم خطا پوشش می‌دهد و همچنین دارای نوسانات و پرش‌های کمتری در ضریب دمپینگ نسبت به روش مارکوارت است و بازه تغییرات را به‌طور پیوسته کاملاً پوشش می‌دهد که باعث جستجوی بهتر فضا و به دست آمدن نتایج بهتر با استفاده از روش پیشنهادی شده است.

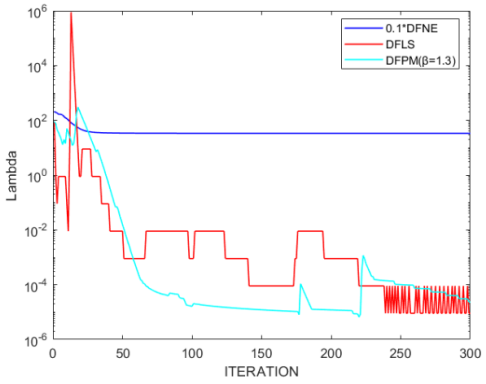



**شکل (١٦): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای تابع غیرخطی (٣٤) برحسب تعداد تکرار**




**شکل (١٧): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای تابع غیرخطی (٣٤) برحسب زمان اجرا**

[1] HimmelBlau's Function



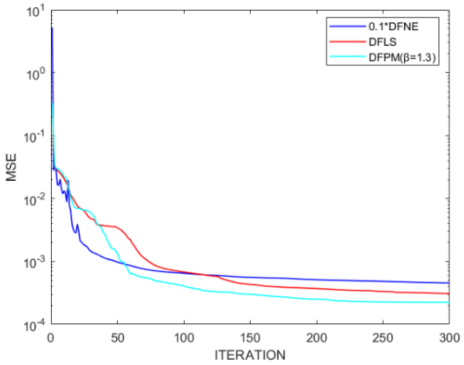

**شکل (۱۸): مقایسه مقدار ضریب دمپینگ روش پیشنهادی با روش نرم خطا و مارکوارت برای تابع غیرخطی (۳٤)**

## ٦-۲ مسئله رگرسیون بر روی مجموعه داده‌های واقعی

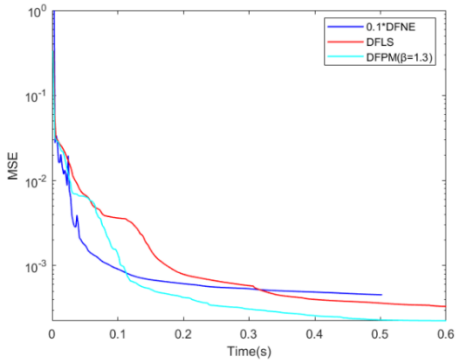
برای شبیه‌سازی دوم از مجموعه داده بار برودتی ساختمان موجود در مخازن یادگیری ماشین UCI[1] استفاده‌شده است. ورودی مسأله شامل هشت ویژگی (۱-وسعت نسبی ساختمان نسبت به ساختمان‌های دیگر، ۲-زیربنا، ۳-مساحت دیوارها، ٤-مساحت سقف، ٥-ارتفاع ساختمان، ٦-جهت ساختمان، ۷-وسعت سطح نورگیر (شیشه‌ای) و ۸-نسبت سطوح شیشه‌ای به‌کل سطوح) و یک داده خروجی (بار برودتی ساختمان) است که کل داده‌های موجود ۷٦۸ داده است [۱۷].

به‌منظور یادگیری از یک شبکه عصبی با یک‌لایه میانی شامل ۲۰ نورون استفاده‌شده است و لایه ورودی شامل ۸ نورون است که درمجموع شبکه عصبی مورداستفاده شامل ۲۰۱ وزن است. از ۵۰۰ داده اول برای یادگیری این شبکه عصبی استفاده‌شده است که در شکل ۱۹ و ۲۰ خطای یادگیری به ازای روش‌های مختلف قابل‌مشاهده است. همان‌طور که در شکل ۱۹ و ۲۰ مشاهده می‌شود روش پیشنهادی (DFPM) دارای نتایج بهتری در آموزش شبکه‌عصبی برای این مسأله رگرسیون است.

در شکل ۲۱ مشاهده می‌شود که روش مارکوارت و روش پیشنهادی، در مقایسه با روش نرم خطا، تغییرات ضریب دمپینگ را با دامنه وسیع‌تری پوشش می‌دهند، که نشان‌دهنده پویایی بیشتر این دو روش است و انتظار می‌رود نتایج بهتری حاصل شود. در مقایسه بین روش مارکوارت و روش پیشنهادی، مشاهده شد که روش پیشنهادی دارای نوسانات و پرش‌های کمتری در ضریب دمپینگ است و بازه تغییرات را به‌طور پیوسته پوشش می‌دهد در حالی که درروش مارکوارت (DFLS) ضریب دمپینگ دارای پرش‌های بزرگی بوده است. این مسأله، منجر به عملکرد بهتر روش پیشنهادی در تخمین ضریب دمپینگ شده و با ایجاد تعادل بین پویایی و پیوستگی، عملکرد بهتری را ارائه می‌دهد.

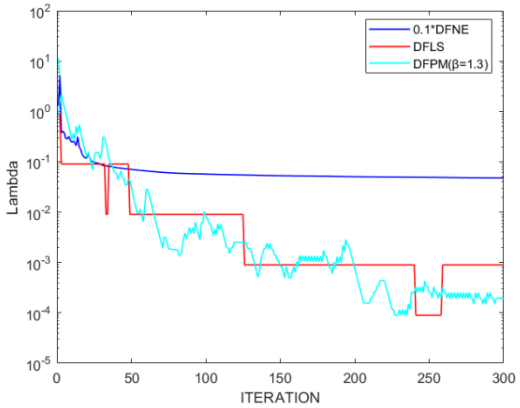



**شکل (۱۹): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله رگرسیون برحسب تعداد تکرار**




**شکل (۲۰): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله رگرسیون برحسب زمان اجرا**

[1] UCI Machine Learning Repository




**شکل (۲۱): مقایسه مقدار ضریب دمپینگ روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله رگرسیون**

پیشنهادی با ایجاد تعادل بین پویایی و پیوستگی، عملکرد بهتری را ارائه می‌دهد.

در مرحله ارزیابی تعداد ۲۶۸ داده باقیمانده برای تست مورداستفاده قرارگرفته که نتایج به‌دست‌آمده در شکل ۲۲ قابل‌مشاهده است. برای نمایش دقیق‌تر و امکان بررسی بیشتر داده‌ها، در شکل ۲۳ فقط نتایج به‌دست‌آمده برای داده‌های ۱۳۵ تا ۱۴۸ رسم شده است که به‌وضوح در این شکل دقت بیشتر روش پیشنهادی مشاهده می‌شود که به‌خوبی داده‌های تست را دنبال کرده و تخمین دقیق‌تری را نسبت به سایر روش‌ها ارائه کرده است.

یکی از معیارهای مورداستفاده در ارزیابی مسائل رگرسیون، معیار $R^2$score است که در رابطه ۳۵ تعریف‌شده است. هرچه این معیار به ۱ نزدیک‌تر باشد نشان‌دهنده کارایی بیشتر روش است. برای هر روش از رابطه ۳۵ محاسبه‌شده و در جدول ۱ قابل‌مشاهده است.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \mathrm{Real\ data} - \mathrm{Prediction}}{\sum_{i=1}^{N} \mathrm{Real\ data} - \mathrm{Mean\ of\ real\ data}} \qquad (۳۵)$$

که نتایج جدول ۱ عملکرد مناسب‌تر روش پیشنهادی در مقایسه با سایر روش‌ها برای مسائل رگرسیون را نشان می‌دهد.

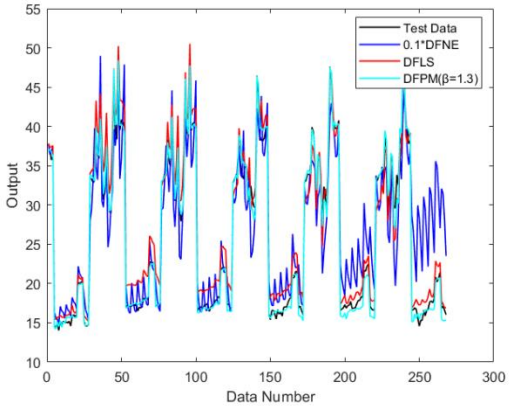
## ۶-۳ مسئله طبقه‌بندی بر روی مجموعه داده‌های واقعی

برای شبیه‌سازی سوم، یک مسأله طبقه‌بندی انتخاب‌شده است که شامل مجموعه داده‌های سرطان سینه جمع‌آوری‌شده توسط دکتر ویلیام از بیماران دانشگاه پزشکی ویسکانسین است. در این مجموعه، داده‌های ورودی از تصاویر اشعه ایکس به‌دست‌آمده‌اند و داده خروجی می‌تواند دو مقدار ۰ (عدم وجود احتمال سرطان) و ۱ (وجود احتمال سرطان) است که این مجموعه شامل ۵۷۰ داده است [۱۸].

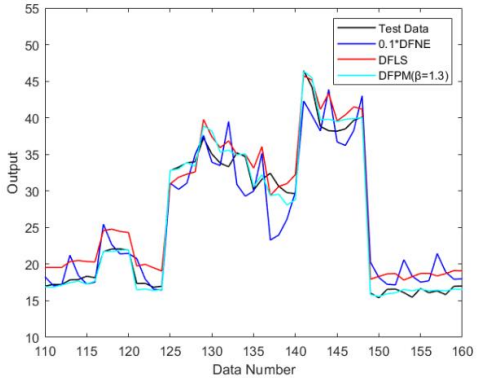
به‌منظور یادگیری از یک شبکه عصبی با یک‌لایه میانی شامل ۱۰ نورون استفاده‌شده است و لایه ورودی شامل ۵ نورون است که درمجموع این شبکه عصبی شامل ۷۱ وزن است. همچنین از ۴۰۰ داده اول برای یادگیری این شبکه عصبی در مرحله آموزش

**جدول (۱): مقایسه شاخص R2 برای روش‌های مورد بررسی**

| Method | $R^2$score |
|---|---|
| DFPM | ۰/۹۹۳ |
| DFNE | ۰/۷۶۷ |
| DFLS | ۰/۹۵۱ |



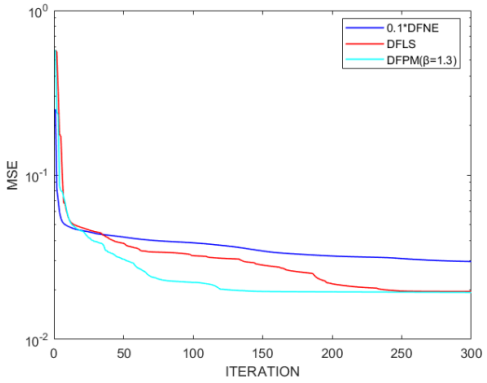

**شکل (۲۲): مقایسه تخمین بر روی داده‌های تست با روش پیشنهادی، روش نرم خطا و مارکوارت برای مسأله رگرسیون**



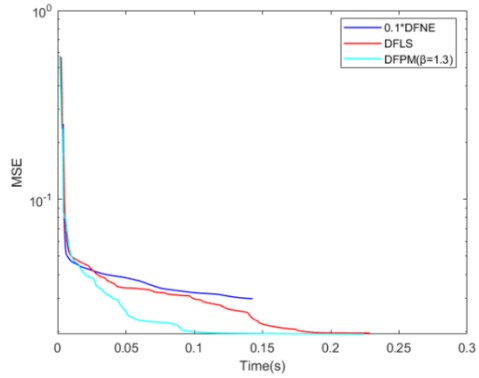

**شکل (۲۳): نمایش داده‌های ۱۱۰ تا ۱۶۰ از شکل ۲۲**

استفاده‌شده است که در شکل ۲۴ و ۲۵ میانگین خطای مرحله یادگیری در ۵۰ بار اجرای شبیه‌سازی به ازای روش‌های مختلف قابل‌مشاهده است. همان‌طور که در دو شکل ۲۴ و ۲۵ مشاهده می‌شود روش پیشنهادی دارای عملکرد مناسبی است و زودتر خطای یادگیری را کاهش داده است.

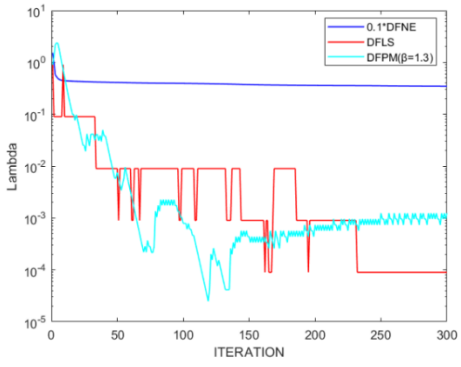
در شکل ۲۶ مقدار ضریب دمپینگ برای هر سه روش مورد بررسی، نمایش داده‌شده است. همان‌طور که مشاهده می‌شود، در




**شکل (۲۴): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله طبقه‌بندی برحسب تعداد تکرار**




**شکل (۲۵): مقایسه روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله طبقه‌بندی برحسب زمان اجرا**




**شکل (۲۶): مقایسه مقدار ضریب دمپینگ روش پیشنهادی با روش نرم خطا و مارکوارت برای مسأله طبقه‌بندی**

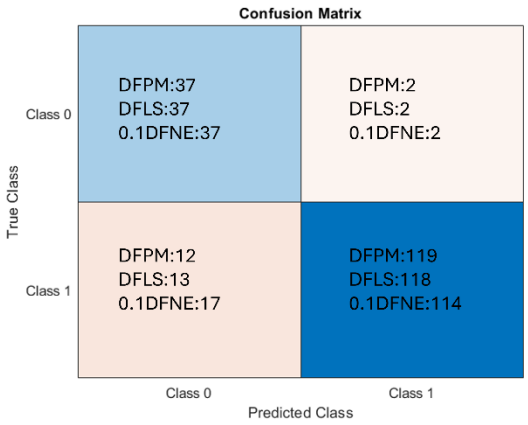
روش مارکوارت و روش پیشنهادی بازه تغییرات ضریب دمپینگ به نسبت روش نرم خطا بیشتر است. که انتظار می‌رود نتایج این دو روش بهتر از روش نرم خطا شود. همچنین در بین دو روش مارکوارت و روش پیشنهادی مشاهده می‌شود که درروش پیشنهادی ضریب دمپینگ به‌دست‌آمده دارای دامنه نوسانات و پرش‌های کمتری است و کل بازه تغییرات را به‌طور پیوسته پوشش داده است و به همین دلیل این روش داری نتایج بهتری است.

در مرحله ارزیابی، تعداد ۱۷۰ داده ورودی و خروجی مورداستفاده قرارگرفته که نتایج به‌دست‌آمده در شکل ۲۷ برای هر سه روش قابل‌مشاهده است. در ستون اول سطر اول، تعداد تشخیص درست عدم سرطان و ستون دوم سطر اول تعداد تشخیص اشتباه عدم سرطان است. ستون اول سطر دوم تشخیص اشتباه برای وجود سرطان است و در ستون دوم سطر دوم، تعداد تشخیص درست سرطان است که برای ارزیابی از معیار پوشش و دقت که به ترتیب در روابط (۳۶) و (۳۷) آمده، استفاده‌شده است.

$$\text{Re}call = \frac{True\ Positive}{True\ Positive + False\ negative} \qquad (۳۶)$$

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \qquad (۳۷)$$

در جدول ۲ نتایج به‌دست‌آمده از شبیه‌سازی ذکرشده است. در سطر دوم این جدول خطای یادگیری روش‌ها، در سطر سوم مقدار معیار پوشش و در سطر چهارم میزان دقت هر روش نشان داده‌شده است که روش پیشنهادی بهترین نتایج را داشته است. مشاهده می‌شود که روش پیشنهادی دارای نتایج بهتری است و دارای دقت و عملکرد مناسبی در مسائل طبقه‌بندی است.




**شکل (۲۷): نمایش ماتریس درهم‌ریختگی برای مسأله طبقه‌بندی**

جدول (۲): مقایسه معیارهای عملکرد روش‌های مورد بررسی در مسأله طبقه‌بندی

| Method | DFPM | DFNE | DFLS |
|---|---|---|---|
| MSE | ۰/۰۱۹ | ۰/۰۳ | ۰/۰۱۹۴ |
| Recall | ۹۰/۸٪ | ۸۷/۰۲٪ | ۹۰/۵٪ |
| Accuracy | ۹۱/۷٪ | ۸۸/۸۲٪ | ۹۱/۱٪ |

## ۷- نتیجه‌گیری

در این مقاله، روش‌های مختلف تعیین ضریب دمپینگ در الگوریتم لونبرگ-مارکوارت، شامل روش‌های مارکوارت، روش‌های مبتنی بر نرم خطای جستجو و مبتنی بر نرم ماتریس ژاکوبین، با یکدیگر مقایسه شده‌اند. نتایج شبیه‌سازی‌ها نشان می‌دهد که تعیین ضریب دمپینگ به روش مارکوارت دارای کارایی خوبی است. در سایر روش‌ها، روش مبتنی بر نرم خطا دارای نتایج مناسب‌تری است که حتی نسبت به روش مارکوارت در مواردی نتایج بهتر می‌شود. البته درروش ضریب دمپینگ مبتنی بر نرم خطا، گاهی نوسان وجود دارد. در این مقاله سعی شده است تا با ترکیب روش تعیین ضریب دمپینگ مبتنی بر نرم خطا و روش جستجوی خطی مورد تطبیق یافته با شرایط جدید مشکل نوسانی بودن را حل کرده و سرعت همگرایی را هم بهبود بخشید. در ادامه با بیان دلیل، بهترین ضریب برای نرخ تنظیم ضریب دمپینگ ($\beta$) درروش پیشنهادی تعیین شد. روش پیشنهادی و اصلاحات اعمال‌شده در آن با تئوری و فرضیات موجود درروش لونبرگ-مارکوارت هماهنگی بیشتری نسبت به سایر روش‌ها دارد و درنتیجه نتایج بهتری به‌دست می‌آید. روش پیشنهادی نتایج قابل‌توجهی را در مقایسه با روش‌های قبلی ارائه می‌دهد و علاوه بر این‌که به خطای نهایی کمتری می‌رسد نوسانات روش نرم خطا را به‌طورکلی برطرف و همگرایی در هر تکرار را هم تضمین می‌کند. در ادامه روش‌های مذکور برای آموزش شبکه‌عصبی به‌منظور شناسایی یک تابع غیرخطی پیچیده، مسأله رگرسیون و مسأله طبقه‌بندی با داده‌های واقعی شبیه‌سازی شدند. در نتایج به‌دست‌آمده مشاهده شد که روش پیشنهادی در مقایسه با دو روش دیگر عملکرد بهتری داشته و روش پیشنهادی در این مقاله می‌تواند به‌عنوان یک ابزار مفید برای حل مسائل بهینه‌سازی در شبکه‌های عصبی مورداستفاده قرار گیرد.

## References

[1] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” Heliyon, vol. 4, no. 11, p. e00938, Nov. 2018, doi: 10.1016/j.heliyon.2018.e00938.

[2] M. Sh. Daoud, M. Shehab, H. M. Al-Mimi, L. Abualigah, R. A. Zitar, and M. K. Y. Shambour, “Gradient-Based Optimizer (GBO): A Review, Theory, Variants, and Applications,” Archives of Computational Methods in Engineering, vol. 30, no. 4, pp. 2431–2449, May 2023, doi: 10.1007/s11831-022-09872-y.

[3] X. Wei and H. Huang, “A survey on several new popular swarm intelligence optimization algorithms,” 2023, doi: 10.21203/rs.3.rs-2450545/v1.

[4] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016.

[5] B. T. Polyak, “Newton’s method and its use in optimization,” Eur J Oper Res, vol. 181, no. 3, pp. 1086–1096, Sep. 2007, doi: 10.1016/j.ejor.2005.06.076.

[6] M. S. Osigbemeh, C. Osuji, M. O. Onyesolu, and U. P. Onochie, “Comparison of the Artificial Neural Network’s Approximations for the Levenberg-Marquardt Algorithm and the Gradient Descent Optimization on Datasets,” Artificial Intelligence Evolution, pp. 24–38, Mar. 2024, doi: 10.37256/aie.5120243781.

[7] O. Umar, I. M. Sulaiman, M. Mamat, M. Y. Waziri, and N. Zamri, “On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems,” J Phys Conf Ser, vol. 1734, no. 1, p. 012018, Jan. 2021, doi: 10.1088/1742-6596/1734/1/012018.

[8] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” Q Appl Math, vol. 2, no. 2, pp. 164–168, 1944, doi: 10.1090/qam/10666.

[9] D. W. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” Journal of the Society for Industrial and Applied Mathematics, vol. 11, no. 2, pp. 431–441, Jun. 1963, doi: 10.1137/0111030.

[10] M. K. Transtrum and J. P. Sethna, “Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization,” Jan. 2012.

[11] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory,” 1978, pp. 105–116. doi: 10.1007/BFb0067700.
[12] Y. Wang, “Gauss–Newton method,” WIREs Computational Statistics, vol. 4, no. 4, pp. 415–420, Jul. 2012, doi: 10.1002/wics.1202.
[13] C. Chen, S. Reiz, C. Yu, H.-J. Bungartz, and G. Biros, “Fast Approximation of the Gauss-Newton Hessian Matrix for the Multilayer Perceptron,” Oct. 2019.
[14] M. R. Osborne, “Nonlinear least squares — the Levenberg algorithm revisited,” The Journal of the Australian Mathematical Society. Series B. Applied Mathematics, vol. 19, no. 3, pp. 343–357, Jun. 1976, doi: 10.1017/S033427000000120X.
[15] A. Suratgar, M. B. Tavakoli, and A. Hoseinabadi, “Modified Levenberg-Marquardt method for neural networks training,” in Proceedings - Wec 05: Fourth World Enformatika Conference, 2005. doi: 10.5281/zenodo.1333881.
[16] Y. H. Irawan and P. T. Lin, “Parametric optimization technique for continuous and combinational problems based on simulated annealing algorithm,” Journal of Energy, Mechanical, Material, and Manufacturing Engineering, vol. 8, no. 2, pp. 75–82, 2023, doi: 10.22219/jemmme.v8i2.29556.
[17] Tsanas and A. Xifara, “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools,” Energy and Buildings, vol. 49, pp. 560–567, 2012, doi: 10.1016/j.enbuild.2012.03.003.
[18] S. S. Mohammadi, M. Salehirad, M. M. Emamzadeh et al., “Improved equilibrium optimizer for accurate training of feedforward neural networks,” Optical Memory and Neural Networks, vol. 33, pp. 133–143, 2024, doi: 10.3103/S1060992X24700048.

## پیوست ۱

درروش لونبرگ–مارکوارت به‌روزرسانی متغیرهای بهینه‌سازی (وزن‌های شبکه عصبی) در جهت عکس گرادیان ($\left[JJ^T+\lambda I\right]^{-1}\times J^T E_k$) انجام می‌شود. ازآنجایی‌که ماتریس $H = JJ^T+\lambda I$ یک ماتریس مثبت معین است لذا معکوس آن‌هم یعنی $H^{-1} = \left[JJ^T+\lambda I\right]^{-1}$ مثبت معین است. از طرفی ماتریس مثبت معین دارای خاصیت زیر است:

$$\forall x \neq 0 \Rightarrow x^T H x > 0 \qquad (۳۸)$$

فرض کنید تابع هزینه زیر برای روش لونبرگ–مارکوارت طبق رابطه (٦) تعریف‌شده باشد:

$$F = \frac{1}{2} E^T E \quad , \quad E = Y_{ref}(X) - Y(X,W) \qquad (۳۹)$$

در رابطه فوق Yref داده‌های خروجی سیستم اصلی هستند که مستقل از w (وزن‌های شبکه عصبی) هستند و فقط Y که خروجی شبکه عصبی است به w ربط دارد.

همچنین اگر فرض شود که تغییرات $\Delta w_k$ (فرضاً w اسکالر است) در هر گام کوچک باشد. در این صورت اگر بر اثر اعمال تغییرات $\Delta w_k$، تابع هزینه F هم دارای تغییرات $\Delta F$ شود، داریم:

$$\frac{\Delta F}{\Delta w} \approx \frac{\partial F}{\partial w} \Rightarrow \Delta F \approx \frac{\partial F}{\partial w}\cdot \Delta w \qquad (٤٠)$$

همچنین در حالت چند متغیره (W یک بردار است) به‌طور مشابه رابطه زیر برقرار است:

$$\begin{cases} \Delta F = \dfrac{\partial F}{\partial w}^T .\Delta W \\ \dfrac{\partial F}{\partial w} = E^T \cdot \dfrac{\partial E}{\partial w} \end{cases} \Rightarrow \Delta F = \left[E^T \cdot \frac{\partial E}{\partial w}\right]^T .\Delta W \qquad (٤١)$$

که با توجه به رابطه (۳۹)، $\frac{\partial E}{\partial w}$ به‌صورت زیر تعریف‌شده است.

$$\frac{\partial E}{\partial w} = -\frac{\partial Y}{\partial w} = -J \quad , \quad J \approx \frac{\partial Y}{\partial w} \qquad (٤٢)$$

و $\Delta F$ به‌صورت زیر است:

$$\Delta F = -E^2 J^T \Delta w \qquad (٤٣)$$

که در این مقاله $\Delta w$ به‌صورت زیر تعریف‌شده است:

$$\Delta w = -H^{-1}\nabla F = -H^{-1} E J^T \qquad (٤٤)$$

با جایگذاری رابطه (٤٤) در رابطه (٤٣) داریم:

$$\Delta F = -E^2 J^T H^{-1} J < 0 \qquad (٤٥)$$

حال با توجه به این‌که $H^{-1}$ ماتریس مثبت معین است و J یک بردار است، $J^T H^{-1} J$ مقداری مثبت است درنتیجه $\Delta F$ منفی می‌شود یعنی این‌که تابع هزینه در اثر تغییرات $\Delta w$ آمده درروش لونبرگ–مارکوارت کاهشی است. این کاهش ادامه دارد تا زمانی که $\Delta F$ صفر شود. $\Delta F = -E^2 J^T H^{-1} J$ یعنی تا زمانی که یا E صفر شود یا J صفر شود. چون $J = \frac{\partial Y}{\partial w}$ لذا صفر شدن J همان نقطه و جواب بهینه است. لذا روش حتماً به جواب بهنیه E=0 یا $\frac{\partial Y}{\partial w} = 0$ می‌رسد. قبلاً از شرط اینکه $\Delta w$ به اندازه کافی کوچک باشد در روابط، استفاده کرده بودیم و به این شرط نشان داده شد که درروش مارکوارت به ازای $\Delta w$ ارائه‌شده حتماً $\Delta F$ منفی (F کاهشی) است. درروش جستجوی خطی در هر تکرار λ (η درروش پیشنهادی) چندین بار تغییر می‌کند تا مطمئن شویم که رابطه $F_k > F_{k+1}$ اولاً کاهشی و دوماً دارای بیشترین کاهش است، لذا روش جستجوی خطی خودش هم (مستقل از اثبات فوق) تضمین کننده کاهشی بودن تابع هزینه است.

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 2, No. 2, Autumn & Winter 2024, pp. 125 - 143

# Neural Network Learning Using an Improved Levenberg-Marquardt Algorithm

Reza Yazhari Kermani [1], Mohammad Mollaie Emamzadeh [2*], Mojtaba Barkhordari Yazdi[3]

[1]MSc. Student, Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

[2]Assistant Professor, Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

[3]Associate Professor, Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

## Article Information

**Original Research Paper**

**Received:**
2024 December 9

**Accepted:**
2025 February 26

**Keywords:**
Damping Factor, Levenberg-Marquardt Algorithm, Newton Algorithm, Neural Network Learning, Non-linear Optimization

**Corresponding Author*:**
molaie@uk.ac.ir

## Abstract

This paper presents a new method to improve the convergence speed and efficiency of the Levenberg-Marquardt algorithm. The Levenberg-Marquardt algorithm is a Newton-based method that is efficient in optimization and determining the weights of neural networks compared to other methods, including the backpropagation method. However, the performance of this method depends significantly on the selection of an appropriate damping factor. Among the various methods for determining the damping factor, the Marquardt line search method and methods based on error norm and based on Jacobian norm are mentioned, which in this paper, by examining the strengths and weaknesses of these methods, a combined method is presented to increase the convergence speed. In the proposed method, the search range of the damping factor and, as a result, the value of the adjustment rate parameter is reduced. By making these corrections, the accuracy of the damping factor search is increased and the convergence speed of the proposed method is increased. In order to evaluate the proposed method, learning a neural network to identify several problems including a nonlinear complex function, a regression and a classification task has been simulated and the results show that the proposed method has good efficiency compared to other methods studied and has been able to reduce the learning error to an acceptable level and has a higher convergence speed.

: 10.22034/ABMIR.2025.22500.1082

E-ISSN: 2821-2037 