

استفاده از هموارکننده L1/2 در یادگیری گرادینتی شبکه‌های عصبی خود سازمانده

زهرا مهرآوران*^(۱) سید جواد سید مهدوی^(۲) یحیی فرقانی^(۳)

(۱) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران.*

(۲) گروه مهندسی برق، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران.

(۳) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران.

تاریخ پذیرش: ۱۳۹۷/۳/۲۲

تاریخ دریافت: ۱۳۹۶/۱۲/۲۱

چکیده

یکی از مهمترین معیارها در آموزش شبکه‌های عصبی، سرعت همگرایی است. تحقق این معیار وابسته به آموزش مطلوب پارامترها و تعیین اندازه‌ی مناسب برای شبکه است اما مقالاتی که تاکنون بکار رفته‌اند بر روی یکی از این عوامل تمرکز داشته‌اند. به منظور تعیین اندازه‌ی مطلوب یک شبکه با توجه به پیچیدگی مساله از شبکه‌های عصبی خودسازمانده استفاده می‌کنیم. چالشی که در این شبکه‌ها دیده می‌شود سرعت همگرایی نسبتاً پایین آن‌ها است، در نتیجه برای بهبود سرعت همگرایی آموزش شبکه از الگوریتم Batch gradient (Bg) به همراه رگولاریزیشن L1/2 که توسط یک تابع هموارکننده، هموار شده‌است، استفاده می‌نماییم تا بدین ترتیب در کنار دو فرایند افزایشی و کاهش‌ی اندازه شبکه، پارامترها به خوبی آموزش ببینند و سرعت همگرایی بهبود یابد. نتایج حاصل از پیاده‌سازی و مقایسه‌ی روش حاضر با روش‌های پایه، از نظر معیارهای سرعت همگرایی و صحت کلاس‌بندی داده‌های تست، نشان از برتری روش پیشنهادی در بهبود صحت و بهبود سرعت همگرایی را می‌دهد.

واژه‌های کلیدی: الگوریتم Batch gradient، تابع هموارکننده، رگولاریزیشن L1/2، شبکه عصبی خودسازمانده

کار مشتق‌پذیر نمی‌باشد، باعث ایجاد نوساناتی در شیب تابع خطا می‌شود. برای حل این چالش می‌توان رگولاریزیشن $L1/2$ را به یک تابع هموارکننده بدهیم [۱۵،۱۶]. در نتیجه با در نظر گرفتن دو روند هرس کردن افزایشی و کاهش‌ی در کنار یکدیگر، بهبود سرعت همگرایی شبکه را نسبت به روش‌های ارائه شده تحقق می‌بخشیم. همچنین با توجه به اینکه روش پیشنهادی از الگوریتم Bg به همراه رگولاریزیشن $L1/2$ که توسط یک تابع هموار کننده هموار شده است، بهره می‌گیرد، همگرایی این روش با توجه به مقاله [۱۴] به اثبات می‌رسد. در راستای هرس کردن شبکه با استفاده از رگولاریزیشن، از رگولاریزیشن‌های مختلفی برای هرس کردن استفاده شده است که می‌توان به استفاده از رگولاریزیشن $L0$ برای هرس کردن شبکه اشاره کرد. این رگولاریزیشن دارای خاصیت پراکندگی بالایی است اما در برابر مسائل Np_hard عملکرد پایینی از خود نشان می‌دهد [۵]. برای رفع چالش رگولاریزیشن $L0$ این رگولاریزیشن با روش‌های گرادینانی آنالین به کار رفته است. اما این راهکار از سرعت همگرایی پایینی برخوردار است [۶]. رگولاریزیشن $L1/2$ رگولاریزیشنی است که در این مقاله مورد توجه قرار گرفته است. این رگولاریزیشن دارای خاصیت پراکندگی نسبتاً خوبی می‌باشد. همچنین در برابر مسائل Np_hard و مسائلی با ابعاد بالا عملکرد خوبی را از خود نشان می‌دهد [۳،۴]. چالش مطرح در رگولاریزیشن $L1/2$ این است که در ابتدای فرآیند آموزش مشتق‌پذیر نمی‌باشد. این امر باعث ایجاد نوساناتی در شیب تابع خطا می‌گردد. برای رفع این چالش رگولاریزیشن $L1/2$ به تابع هوارکننده، که همواره مشتق‌پذیر است داده شده است [۳،۴]. زمانی که تابع خطا محدب یا از نوع درجه دو می‌باشد، برای رسیدن به عملکرد بهتر، رگولاریزیشن $L1/2$ با الگوریتم Bp بکار رفته است [۷]. رگولاریزیشن HLR ترکیبی از رگولاریزیشن‌های $L2$ و $L1/2$ می‌باشد. این رگولاریزیشن هم از خاصیت پراکندگی بالای رگولاریزیشن $L1/2$ و هم

از جمله معیاری که در بحث شبکه‌های عصبی مطرح می‌باشد سرعت همگرایی است. تحقق این معیار وابسته به الگوریتم‌های یادگیری شبکه و معماری شبکه است. زیرا زمانی یک شبکه عصبی از سرعت همگرایی خوبی برخوردار است که پارامترهای آن به خوبی آموزش داده شوند و اندازه شبکه با پیچیدگی مساله هماهنگی لازم را داشته باشد. الگوریتم‌هایی که در بحث آموزش شبکه عصبی به کار برده می‌شوند یا بر روی آموزش پارامترهای شبکه یا بر روی اندازه شبکه متمرکز می‌باشند [۱]. شبکه عصبی پیش‌خور از جمله شبکه‌های عصبی معروف است که دارای مکانیزمی مشابه با مکانیزم سیستم عصبی انسان می‌باشد. این شبکه در طی فرآیند آموزش دارای ساختار ثابت می‌باشد، در نتیجه اگر پیچیدگی مساله از پیچیدگی ساختار شبکه بیشتر باشد، خطای شبکه در ابتدای آموزش به مقدار چشم‌گیری افزایش می‌یابد و طی گام‌های بیشتری به صفر همگرا می‌شود. برای رسیدن به ساختار شبکه‌ای مناسب از شبکه عصبی خودسازمانده استفاده می‌کنیم، چراکه این شبکه دارای این قابلیت است، تا بدون توجه به ساختار اولیه در صورت نیاز با روند افزایشی ساختار شبکه را با پیچیدگی مساله منطبق سازد [۲]. چالش موجود در شبکه عصبی خودسازمانده این است که شبکه دارای سرعت همگرایی نسبتاً پایینی می‌باشد [۱۹]، از این رو برای بهبود سرعت همگرایی آن می‌توان از الگوریتم Bg به همراه ترم رگولاریزیشن $L1/2$ که توسط یک تابع هموارکننده، هموار شده است، استفاده نماییم، چرا که رگولاریزیشن $L1/2$ دارای پراکندگی بالا و عملکرد خوب در برابر داده‌هایی با ابعاد بالا است [۳]، در نتیجه می‌توانیم وزن‌های کم اهمیت را حذف کنیم. در این صورت اگر ساختار شبکه برای مساله بزرگ در نظر گرفته شده بود، بتوانیم با استفاده از رگولاریزیشن به ساختار کوچک‌تر دست یابیم. رگولاریزیشن $L1/2$ به علت آنکه در ابتدای

از خاصیت همبستگی بالای متغیرها بهره می‌گیرد. اما در عمل پیش‌بینی عملکرد ضعیفی دارد [۸]. رگولاریزیشن $L1$ نیز در تحلیل داده‌ها عملکرد خوبی از خود نشان داده است همچنین باعث بهبود قابلیت تعمیم‌پذیری الگوریتم Bp شده‌است [۹]. رگولاریزیشن بیزین نیز دارای قابلیت تعمیم‌پذیری بالا و اما خاصیت پراکندگی پایین می‌باشد [۱۰]. شبکه‌های عصبی خودسازمانده این قابلیت را دارند تا بدون توجه به ساختار اولیه، به ساختار متناسب با داده‌های ورودی دست یابند. در راستای کاربرد شبکه‌های عصبی خودسازمانده مقالاتی ارائه شده است، که می‌توان اشاره داشت به:

استفاده از شبکه عصبی خودسازمانده در کلاس‌بندی سیستم‌های غیر خطی باعث عملکرد خوب در برابر داده‌های فازی و کاهش خطای تست شده است اما در این رویکرد سرعت همگرایی پایین است [۲]. در بحث خوشه‌بندی زمانی که داده‌ها از هم‌پوشانی زیادی برخوردارند، استفاده از شبکه خودسازمانده ESOINN که دارای ساختار ساده بوده و از یادگیری بدون نظارت بهره می‌گیرد، نتایج خوبی را به دنبال داشته است [۱۱]. شبکه خودسازمانده کوهنن نیز به علت سادگی در ساختار و استفاده از شیوه یادگیری رقابتی عملکرد خوبی در عمل خوشه‌بندی و مسائل بهینه‌سازی از خود نشان می‌دهد [۱۵]. در یادگیری‌های با نظارت برای سرعت بخشیدن به فرآیند آموزش و رسیدن به صحت بالا در کلاس‌بندی داده غیرثابت از شبکه خودسازمانده ARTMAP استفاده شده است اما این شبکه در یادگیری بدون نظارت عملکرد خوبی را از خود نشان نمی‌دهد [۱۲]. در یادگیری بدون نظارت برای داشتن عملکرد بهتر از رویکرد شعاعی در کنار شبکه خودسازمانده استفاده شده‌است [۱۶]. درمساله فروشنده دوره‌گرد برای بهبود سرعت همگرایی و دقت از ترکیب شبکه کوهنن و هاپفیلد استفاده شده است [۱۷]. از جمله کاربردهای دیگر شبکه عصبی خودسازمانده می‌توان اشاره داشت به زمینه‌های پیش‌بینی سهام [۱۸]، بررسی

رخساره‌های زمین [۱۹] و ارزیابی سیستم‌های توزیع برق اشاره کرد [۲۰].

۲. مفاهیم پایه

۱-۲. رگولاریزیشن $L1/2$

از جمله روش‌های هرس کردن شبکه روش‌های رگولاریزیشن می‌باشند. با کمک روش‌های رگولاریزیشن وزن‌های کم اهمیت حذف می‌شوند و وزن‌هایی که اهمیت بیشتری دارند نقش پرنگ‌تری ایفا می‌کنند. در بین ترم‌های رگولاریزیشن در این مقاله از رگولاریزیشن $L1/2$ استفاده می‌کنیم. چرا که رگولاریزیشن $L1/2$ بعد از $L0$ دارای خاصیت پراکندگی بالا بوده و در برابر داده‌هایی با ابعاد بالا و مسائل Np_hard عملکرد خوبی را از خود نشان می‌دهد [۳،۴]. چالش که در رگولاریزیشن $L1/2$ وجود دارد این است که این رگولاریزیشن در ابتدا مشتق‌پذیر نمی‌باشد. این امر باعث ایجاد نوساناتی در شیب تابع خطا می‌گردد. رگولاریزیشن $L1/2$ به صورت فرمول ۱ می‌باشد [۱۳].

(۱)

$$k = \lambda \sum_{i=1}^p \frac{1}{\sqrt{|w_i|}} |w_i|$$

در فرمول فوق λ ضریب رگولاریزیشن است و w_i بردار وزن می‌باشد.

۲-۲. تابع هموار کننده

مشتق‌پذیر نبودن رگولاریزیشن $L1/2$ باعث ایجاد نوساناتی در شیب تابع خطا می‌گردد. برای حل این چالش، رگولاریزیشن $L1/2$ به یک تابع هموار کننده داده می‌شود. مزیت‌هایی که تابع هموارکننده به همراه خواهد داشت عبارتند از: از بین بردن نوسانات شیب تابع خطا، هرس کردن بهتر شبکه، اثبات همگرایی شبکه و افزایش قابلیت تعمیم‌پذیری شبکه می‌باشد. تابع هموار کننده به صورت فرمول ۲ می‌باشد [۳،۴].

$$F(X) = \begin{cases} |x| \rightarrow \text{if } (|x|) \geq a \\ -\frac{1}{8a^4} + \frac{3}{4a}x^2 + \frac{3}{8}a \rightarrow \text{if } (|x|) < a \end{cases} \quad (2)$$

در فرمول فوق ON خروجی هر نورون لایه نرمالایز می‌باشد. سپس برای نورون اضافه شده باید پارامترهای وزن، شعاع و مرکز به ترتیب به صورت فرمول‌های زیر محاسبه گردند [۲].

(۴)

$$w_{\text{add}} = \frac{y - o_0}{OR_{\text{new}}}$$

در فرمول فوق y خروجی واقعی و o_0 خروجی شبکه و OR بردار خروجی لایه ی Rbf می‌باشد.

$$\sigma_i^{\text{add}} = \sigma_i \quad (5)$$

با توجه به فرمول فوق شعاع نورون اضافه شده برابر است با انحراف معیار نورونی که SI آن از حد آستانه بیشتر بود، می‌باشد.

$$C_i^{\text{add}} = \frac{c_i + x}{2} \quad (6)$$

در فرمول فوق c_i مرکز نورونی است که مقدار SI آن از آستانه بیشتر شده است و x نیز بردار ورودی می‌باشد.

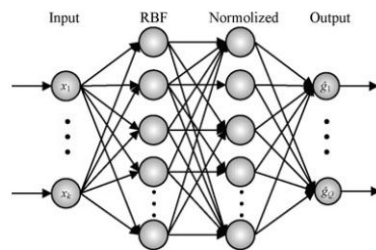
۳. روش پیشنهادی

یکی از الگوریتم‌های رایج برای آموزش شبکه‌های عصبی الگوریتم Bg می‌باشد. برای بهبود عملکرد الگوریتم Bg و هرس کردن بهتر شبکه، یک ترم از رگولاریزیشن $L1/2$ که توسط یک تابع هموارکننده، هموار شده است به تابع خطای الگوریتم Bg اضافه می‌شود. بدین ترتیب با حذف وزن‌های غیرضروری سرعت همگرایی شبکه بهبود پیدا می‌کند و هرس کردن شبکه بهتر انجام می‌گردد [۳، ۴]. چالشی که در آموزش شبکه‌های عصبی وجود دارد، این است که اگر پیچیدگی مساله از پیچیدگی ساختار شبکه بیشتر باشد، خطا در ابتدا به مقدار چشم‌گیری افزایش می‌یابد و با گام‌های بیشتری به صفر همگرا می‌شود. برای حل این چالش روش پیشنهادی را ارائه می‌کنیم تا در کنار عمل هرس کردن وزن‌های شبکه، معیار اندازه‌ی شبکه در

در فرمول فوق a برابر $0,005$ در نظر گرفته شده است [۴] و x بردار داده‌های ورودی می‌باشد.

۲-۳. شبکه عصبی خود سازمانده

یک دسته از شبکه‌های عصبی، شبکه‌های عصبی خودسازمانده می‌باشند. این شبکه‌ها این قابلیت را دارند تا بدون توجه به ساختار اولیه، با توجه به الگوی ورودی طی یک فرآیند افزایشی به ساختار مطلوب خود دست یابند. شبکه‌های عصبی خودسازمانده انواع مختلفی دارند که در این بین شبکه عصبی خودسازمانده‌ای که مورد استفاده قرار می‌گیرد به صورت شکل ۱ می‌باشد.



شکل ۱- شبکه عصبی خودسازمانده استفاده شده در روش

پیشنهادی [۲]

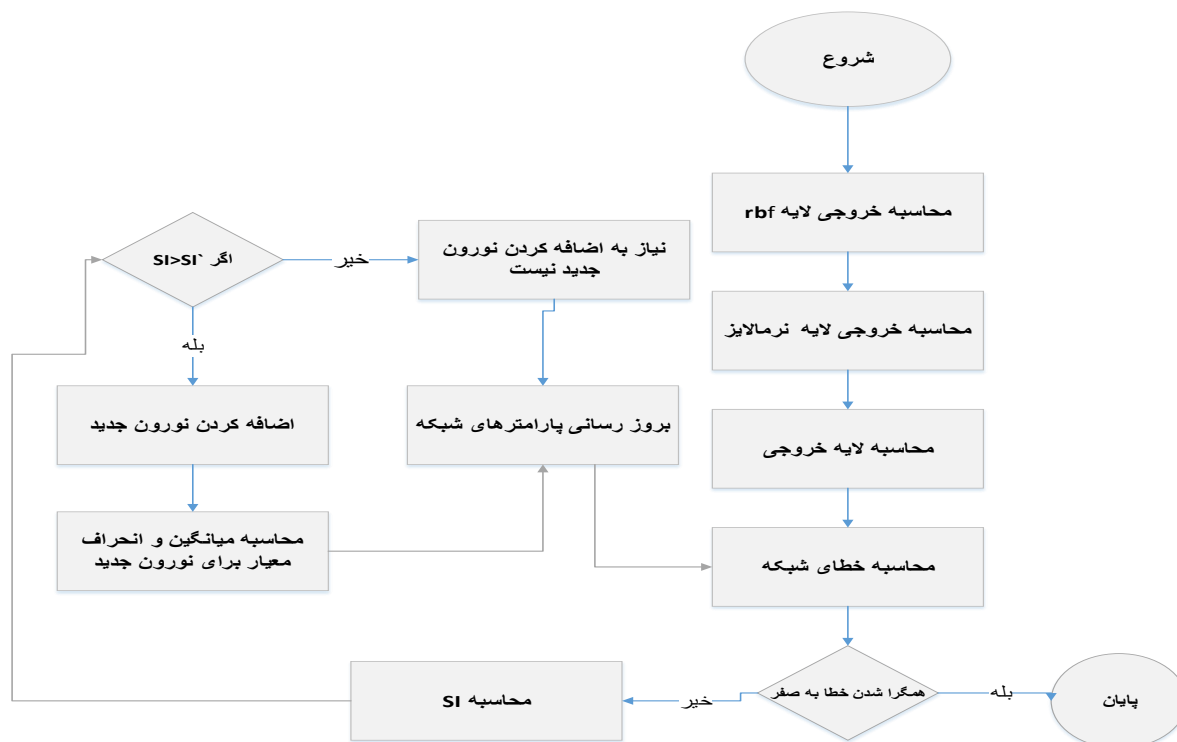
با توجه به شکل ۱، شبکه عصبی خودسازمانده از دو لایه مخفی نرمالایز و Rbf تشکیل شده است. تعداد نورون‌های دو لایه مخفی برابر بوده و هر نورون دارای دو ویژگی مرکز و شعاع می‌باشد. بردار وزن تنها بین لایه نرمالایز و خروجی قرار دارد [۲]. عملکرد این شبکه بدین صورت است که بعد از محاسبه خطای شبکه چنانچه شرط همگرایی تحقق نیافته بود، به ازای تمام نورون‌های لایه نرمالایز پارامتر SI محاسبه می‌گردد. چنانچه مقدار این پارامتر از یک حد آستانه بیشتر شد لازم است طی یک فرآیند افزایشی نورونی به لایه‌های مخفی اضافه گردد.

(۳)

$$SI(ON_j) = \log(1 + 10^{ON_j})$$

صورتی که سایز شبکه متناسب با پیچیدگی مساله نباشد نیز مورد توجه قرارگیرد. در روش پیشنهادی از یک شبکه عصبی خودسازمانده استفاده می‌کنیم و برای آموزش این شبکه‌ی عصبی خود سازمانده از الگوریتم Bg به همراه رگولاریزیشن $L1/2$ که توسط تابع هموارکننده هموار شده است، مورد استفاده قرار می‌گیرد. شبکه عصبی خودسازمانده این قابلیت را دارد تا تعداد نورون‌های لایه مخفی را با توجه به نیاز و پیچیدگی مساله افزایش دهد. در این راستا عملکرد رگولاریزیشن $L1/2$ سبب حذف وزن‌های غیر مرتبط و دادن ارزش بیشتر به وزن‌های پراهمیت می‌شود. در نتیجه وجود شبکه عصبی

خودسازمانده در کنار بهره‌گیری از الگوریتم Bg به همراه ترم رگولاریزیشن هموار شده باعث بهبود سرعت همگرایی نسبت به روش‌های ارائه شده می‌گردد. شبکه عصبی خودسازمانده دو قابلیت: ۱-تنظیم تعداد لایه‌های شبکه، ۲-تنظیم تعداد نورون‌های لایه مخفی را دارا می‌باشد [۲]. در روش پیشنهادی فرض بر آن است که شبکه عصبی تنها قابلیت تنظیم تعداد نورون‌های لایه مخفی را دارد. بدین منظور شبکه‌ی عصبی خودسازمانده دارای ساختاری مشابه ساختار شکل ۱ می‌باشد [۲]. فلوجارت روش پیشنهادی به صورت شکل ۲ می‌باشد.



شکل ۲- فلوجارت روش پیشنهادی

می‌گردند. سپس با توجه به خروجی لایه نرمالایز و بردار وزن بین لایه نرمالایز و لایه خروجی، خروجی شبکه با توجه به فرمول ۹ محاسبه می‌شود [۲].

$$OR_j = \prod_{i=1}^{NI} \exp\left(-\frac{(x_i - c_{ij})}{2\sigma_{ij}^2}\right) \quad (7)$$

در فرمول فوق پارامتر c مرکز و پارامتر σ شعاع هر

روش پیشنهادی شامل ۵ گام اصلی می‌باشد که عبارتند از:

گام ۱: محاسبه خروجی شبکه

در ابتدا لازم است تا خروجی شبکه خودسازمانده محاسبه شود. بدین منظور ابتدا خروجی لایه rbf و سپس خروجی لایه نرمالایز، با توجه به فرمول‌های ۸ و ۷ محاسبه

نورون لایه rbf می‌باشد.

$$ON_i = \frac{1}{NR} \sum_{i=1}^{NR} OR_i ON_i \quad (8)$$

با توجه به فرمول ۸ خروجی هر نورون لایه نرمالایز از تقسیم خروجی نورون متناظر آن در لایه rbf بر مجموع خروجی نورون‌های لایه rbf محاسبه می‌شود.

$$OO = \sum_{i=1}^{NR} W_i ON_i \quad (9)$$

در فرمول ۹ پارامتر W بردار وزن از لایه نرمالایز به لایه خروجی می‌باشد و ON بردار خروجی نورون‌های لایه نرمالایز است.

گام ۲: محاسبه‌ی خطای شبکه

گام دوم، گام نوآوری روش پیشنهادی می‌باشد. بعد از محاسبه خروجی شبکه در گام اول، نوبت محاسبه خطای شبکه می‌رسد. در این راستا از فرمول ۱۰ در جهت محاسبه خطای شبکه استفاده می‌کنیم [۳،۴].

(۱۰)

$$E(W) = E(W) + \lambda \sum_{i=1}^q \sum_{k=0}^p f(w_{ik})^{1/2}$$

فرمول ۱۰ خطای الگوریتم Bg که به آن ترم رگولاریزیشن $L1/2$ اضافه شده است، می‌باشد. در فرمول فوق جمله اول خطای شبکه می‌باشد که ترمی از تفاضل خروجی شبکه از خروجی واقعی است. جمله دوم رگولاریزیشن $L1/2$ می‌باشد که در فرمول ۱ بیان شده است [۱۵].

گام ۳: اضافه شدن نورون به لایه مخفی

بعد از آنکه مقدار خطا در گام دوم محاسبه شد، چنانچه مقدار تابع خطا به صفر همگرا نشد باید بررسی شود که آیا لازم است نورونی اضافه شود یا خیر. برای بررسی این مساله از معیار SI استفاده می‌کنیم. این معیار در فرمول ۳ تعریف شده است [۲]. بدین ترتیب برای هر نورون لایه نرمالایز پارامتر SI را محاسبه می‌شود و چنانچه مقدار پارامتر SI از یک حد آستانه (در این تحقیق یک

نظر گرفته شده است) بیشتر باشد، باید یک نورون به لایه‌های مخفی اضافه گردد [۲]. هر نورون در ساختار شبکه دارای یک شعاع و مرکز می‌باشد، بنابراین باید مقادیر وزن و شعاع و مرکز مربوط به آن نورونی که در صورت نیاز اضافه می‌گردد را بدست آوریم. برای این منظور به ترتیب از فرمول‌های ۴، ۵ و ۶ استفاده می‌کنیم.

گام ۴: بروزرسانی پارامترهای شبکه

به طور کلی پارامترهایی که در روش پیشنهادی در صورت عدم گرایش خطا به صفر باید بروز رسانی شوند برابر است با:

۱- وزن‌های شبکه

۲- شعاع نرون‌ها

۳- مرکز نرون‌ها

برای بروزرسانی پارامترهای نام برده شده باید از تابع خطا نسبت به هر یک از آن‌ها مشتق گرفته شود بدین ترتیب فرمول‌های بروزرسانی به ترتیب نام برده شده به صورت زیر خواهند بود.

(۱۱)

$$W = W + \eta (2eON + \lambda \frac{df(w)}{2f w^{1/2}})$$

فرمول ۱۱ مربوط به بروزرسانی پارامترهای وزن شبکه می‌باشد. پارامتر e، اختلاف خروجی مطلوب و خروجی شبکه است. و f تابع هموارکننده‌ای است که برای هموار کردن رگولاریزیشن $L1/2$ به کار می‌رود. پارامتر λ ضریب ترم رگولاریزیشن است و پارامتر η نیز نرخ یادگیری می‌باشد.

(۱۲)

$$\sigma_{ij} = \sigma_{ij} + \eta \left(\sum_{j=1}^k \frac{e w_i (ON_i - ON_i^2) (x - c_{ij})^2}{\sigma_{ij}^{3/2}} \right)$$

فرمول ۱۲ مربوط به بروزرسانی پارامتر شعاع نورون‌های لایه مخفی شبکه است که در آن پارامتر e خطای شبکه است و پارامتر w، بردار وزن شبکه و ON خروجی لایه نرمالایز، x بردار ورودی، c مرکز و σ شعاع

فعلی نوروں می باشد.

(۱۳)

$$c_{ij} = c_{ij} + \frac{\eta_{ew_i}(ON_i - ON_i^2)(x - c_{ij})}{\sigma_{ij}}$$

فرمول ۱۳ مربوط به بروزسانی پارامتر مرکز نوروں های لایه مخفی می باشد که از مشتق تابع خطا نسبت به پارامتر مرکز، فرمول فوق محاسبه می گردد.

گام ۵: بررسی مجدد خطای شبکه

بعد از بروزسانی پارامترهای شبکه در گام ۴، دوباره با توجه به پارامترهای جدید شبکه، خروجی شبکه محاسبه می شود و مجدداً خطای شبکه محاسبه می گردد و بررسی می شود که آیا شرط همگرایی تحقق یافته است یا خیر، اگر شرط همگرایی تحقق یافته بود، فرآیند آموزش به پایان می رسد، در غیر این صورت گام های ۱-۴ آنقدر ادامه می یابند تا خطا به صفر همگرا شود.

۴. نتایج

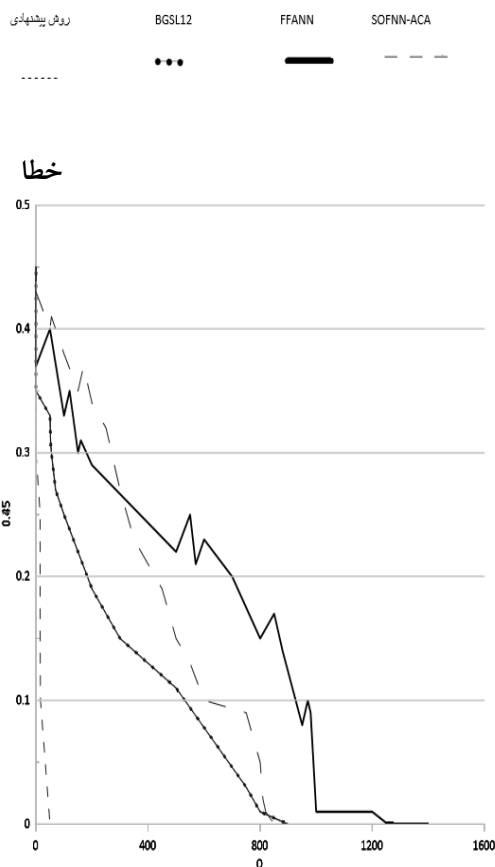
روش پیشنهادی به همراه روش های FFANN [۱۵] و BGSL1/2 [۳] و SOFNN-ACA [۲] بر روی چهار پایگاه داده از مجموعه UCI به نام های Iris، Pima، Wine، Glass و Hepatitis مورد مقایسه قرار گرفته اند. مشخصات این پایگاه های داده در جدول زیر آمده اند:

جدول ۱- اطلاعات مجموعه داده ها

نام پایگاه داده	تعداد داده ها	تعداد ویژگی ها	تعداد کلاس ها
Iris	۱۵۰	۴	۳
Wine	۱۷۸	۱۳	۳
Glass	۲۱۴	۹	۶
Hepatitis	۸۰	۱۹	۲
Pima	۷۶۸	۸	۲

به منظور ارزیابی روش پیشنهادی و مقایسه بهبود

کارایی نسبت به سایر روش ها، از معیار بهبود سرعت همگرایی و صحت داده های تست و انحراف معیار داده های تست استفاده می کنیم. آزمایشات با استفاده از تکنیک 10_Fold تقویت شده اند تا نحوه همگرایی الگوریتم ها تاثیر کمتری از مقداردهی اولیه تصادفی داده ها داشته باشند. تعداد تکرارها در هر الگوریتم ۱۴۰۰ در نظر گرفته شده است و ۱۰٪ از داده های هر پایگاه داده به عنوان داده های تست در نظر گرفته است. پیاده سازی روش پیشنهادی و ۳ روش مورد مقایسه از طریق متلب R2014a انجام شده است. نتایج حاصل از همگرایی روش پیشنهادی و سه روش FFANN، BGSL12 و SOFNN-ACA در شکل زیر مشاهده می شوند.



شکل ۳: نحوه همگرایی الگوریتم پیشنهادی و FFANN و BGSL12 و SOFNN-ACA بر روی پایگاه داده Pima

همان طور که در شکل ۳ مشاهده می کنیم، روش

پیشنهادی نسبت به ۳ روش مورد مقایسه دارای سرعت همگرایی بالاتر بوده و نیز از نوسانات کم‌تر شیب تابع خطا بهره‌مند است. همچنین صحت حاصل از تخمین الگوریتم های فوق بر روی داده های تست در جدول زیر نشان داده شده است.

جدول ۲-صحت داده‌های تست

مجموعه داده	روش پیشنهادی	FFANN	BGSL1/2	SOFNNACA
Iris	۰/۷۸۷۰	۰/۵۸۰۰	۰/۷۳۳۳	۰/۵۸۵۷
Pima	۰/۴۶۸۳	۰/۲۱۱۴	۰/۱۳۸	۰/۱۹۱۸
Hepatitis	۰/۶۶۲۵	۰/۴۵۰۰	۰/۵۱۲۵	۰/۵۳۲۳
Wine	۰/۳۷۲۳	۰/۵۳۱	۰/۳۳۵۳	۰/۳۵۵۷
Glass	۰/۳۴۹۴	۰/۱۴۷	۰/۳۳۳۳	۰/۲۳۱

جدول فوق صحت داده‌های تست می‌باشد. صحت داده‌ها از تقسیم تعداد داده‌هایی که برچسب آنها درست تخمین زده شده‌اند بر تعداد کل داده‌ها به دست می‌آید. همچنین انحراف معیار داده‌های تست با توجه به میانگین خطاهای حاصل از داده‌های تست و مجموع تفاضل‌های خطای هر داده تست از این میانگین محاسبه می‌شود. مقادیر جدول زیر از میانگین انحراف معیارهای ۳۰ بار اجرای برنامه به دست آمده‌اند.

جدول ۳-انحراف معیار داده‌های تست

مجموعه داده	روش پیشنهادی	FFANN	BGSL1/2	SOFNNACA
Iris	۳/۵۵۴۶	۵/۹۸۵۷	۴/۷۰۵۷	۴/۸۸۱۳
Pima	۱/۱۵۸۳	۳/۰۲۸۳	۴/۸۲۸۳	۳/۵۳۱۸
Hepatitis	۰/۰۵۶	۲/۵۵۶۵	۰/۹۱۲۵	۰/۳۹۸۱
Wine	۰/۰۲۷	۰/۰۱۷۱	۳/۰۴۶۱	۱/۵۳۳۲
Glass	۰/۰۶۳۶	۱/۲۱۴۷	۱/۰۰۶۳	۱/۷۲۸۱

۵. بحث و جمع‌بندی

با توجه به نتایج بدست آمده در بخش ۴ مشاهده می‌کنیم که روش ارائه شده، با توجه به معیارهای صحت و انحراف معیار عملکرد نسبتاً خوبی را از خود نشان می‌دهد چرا که روش ارائه شده از قابلیت شبکه عصبی خود سازمانده که می‌تواند بدون توجه به ساختار اولیه به

ساختار مطلوب برسد بهره می‌گیرد همچنین در کنار این قابلیت از خاصیت هرس کردن رگولاریشن L1/2 که قابلیت پراکندگی بالا و عملکرد خوب در برابر داده‌هایی با ابعاد بالا را دارد بهره گرفته است. اما چالشی که در روش پیشنهادی وجود دارد این است که روش مورد بررسی در مواردی که داده‌ها از هم‌پوشانی بالایی برخوردارند، عملکرد ضعیف‌تری را از خود نشان می‌دهد از این رو به ازای مجموع داده‌ی Wine عملکرد ضعیف‌تری را از خود نشان داده است.

۶. نتیجه‌گیری

یکی از مسائلی که در زمینه آموزش شبکه‌های عصبی مطرح می‌باشد، معیار سرعت همگرایی است. عواملی که در سرعت همگرایی و صحت داده‌های تست تاثیرگذار می‌باشند، می‌توان به آموزش مناسب پارامترهای شبکه و اندازه مطلوب شبکه اشاره کرد. شبکه عصبی خودسازمانده، این قابلیت را دارد تا چنانچه پیچیدگی مساله از پیچیدگی ساختار شبکه بیشتر بود با عمل افزودن نورون به لایه مخفی، اندازه مطلوب برای شبکه را تعیین کند. برای آموزش شبکه عصبی خود سازمانده از الگوریتم Bg که یک الگوریتم رایج در زمینه‌ی آموزش شبکه‌های عصبی است استفاده می‌کنیم، چرا که این الگوریتم این قابلیت را دارد تا در بهینه‌ی محلی گیر نیفتد. برای هرس کردن وزن‌های شبکه در کنار عمل افزودن نورون توسط شبکه عصبی خودسازمانده، از رگولاریشن L1/2 استفاده می‌کنیم، در نتیجه به کمک این ترم از رگولاریشن که به وسیله یک تابع هموارکننده هموار می‌شود، این قابلیت به وجود می‌آید که چنانچه ساختار شبکه بزرگتر از سایز مطلوب برای مساله باشد، با هرس کردن وزن‌های غیر ضروری و کم اهمیت، به سرعت فرآیند یادگیری بیفزاییم. در نتیجه با در نظر گرفتن ساختار مناسب و آموزش مناسب پارامترها در روش پیشنهادی، این امکان فراهم می‌گردد تا سرعت همگرایی شبکه افزایش یابد و همچنین

معیارهای صحت و انحراف معیار داده‌های تست نیز بهبود
 یابد. در کنار روش پیشنهادی از الگوریتم PSO که یک الگوریتم
 بهینه سازی است، برای تعیین مناسب‌ترین مقادیر
 پارامترهای تابع هموارکننده استفاده نماییم.

۷. کارهای آینده

برای بهبود بیشتر صحت می‌توان به عنوان کار آینده

۸. مراجع

1. J. Zhang, N. Ji, J. Liu, J. Pan, and D. Meng, "Enhancing performance of the backpropagation algorithm via sparse response regularization," *Neurocomputing*, vol. 153, pp. 20–40, 2015.
2. H. Han, X. L. Wu, and J. F. Qiao, "Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 554–564, 2016.
3. W. Wu, Q. Fan, J. M. Zurada, J. Wang, D. Yang, and Y. Liu, "Batch gradient method with smoothing L1 / 2 regularization for training of feedforward neural networks," *Neural Networks*, vol. 50, pp. 72–78, 2014.
4. K. S. Mohamed, Y. S. Mohammed, A. A. Elzain, K. M. Makin, and E. A. A. Noh, "Batch Gradient Method with Smoothing L 1 / 2 Regularization and Momentum for Pi-sigma Networks," no. 2012, pp. 2002–2012, 2014.
5. H. S. Zhang, Y. L. Tang, and X. D. Liu, "Batch gradient training method with smoothing regularization for l(0) feedforward neural networks," *Neural Comput. Appl.*, vol. 26, no. 2, pp. 383–390, 2015.
6. Online gradient method with smoothing ℓ_0 regularization for feedforward neural networks," *Neurocomputing*, vol. 224, pp. 1–8, Feb. 2017.
7. Q. Fan, W. Wu, and J. M. Zurada, "Convergence of batch gradient learning with smoothing regularization and adaptive momentum for neural networks," *Springerplus*, vol. 5, no. 1, p. 295, 2016.
8. H. H. Huang, X. Y. Liu, and Y. Liang, "Feature selection and cancer classification via sparse logistic regression with the hybrid L1/2 +2 regularization," *PLoS One*, vol. 11, no. 5, 2016.
9. F. Li, J. M. Zurada, L. Fellow, Y. A. N. Liu, and W. E. I. Wu, "Input Layer Regularization of Multilayer Feedforward Neural Networks," vol. 5, pp. 10979–10985, 2017.
10. L. M. Saini, "Peak load forecasting using Bayesian regularization, Resilient and adaptive backpropagation learning based artificial neural networks," *Electr. Power Syst. Res.*, vol. 78, no. 7, pp. 1302–1310, 2008.
11. S. Furao, T. Ogura, and O. Hasegawa, "An enhanced self-organizing incremental neural network for online unsupervised learning," *Neural Networks*, vol. 20, no. 8, pp. 893–903, 2007.
12. G. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: supervised real-time learning and
13. Xu, Z. B., Zhang, H., Wang, Y., & Chang, X. Y. (2009). L1/2 regularizer. *Science in China Series F: Information Sciences*, 52(1), 1–9.
14. Fan, Qinwei, Jacek M. Zurada, and Wei Wu. 2014. "Convergence of Online Gradient Method for Feedforward Neural Networks with Smoothing L1/2 regularization Penalty." *Neurocomputing* 131:208–16.

۱۵. آزاده سلطانی "بهبود شبکه خود سازمانده کوهونن" دهمین کنفرانس سیستم های فازی ایران، دانشگاه شهید بهشتی، تیر ۸۹

۱۶. مهدی سعادت مند طرزجان "یک شبکه عصبی جدید با ساختاری سازنده" دانشگاه فردوسی مشهد، نشریه دانشکده فنی، جلد ۳۹، شماره ۴، آبان ماه ۱۳۸۴، از صفحه ۴۶۹ تا ۴۶۹.

۱۷. پیام حنفی زاده "مدل ترکیبی شبکه عصبی مصنوعی پیشخور و خودسازمانده کوهونن برای پیش بینی سهام" پژوهش

مطالعات مدیریت صنعتی سال هشتم زمستان ۹۱.

۱۸. شب‌نم شهبازی " طبقه بندی و شناسایی رخساره‌های زمین‌شناسی " دانشگاه صنعتی امیرکبیر، نشریه دانشکده فنی، دوره ۴۳،

شماره ۳، شهریورماه ۱۳۸۸، از صفحه ۲۸۳ تا ۲۹۶.

۱۹. عارف جلیل ایرانی "ارزیابی قابلیت اطمینان در سیستم‌های توزیع شده با استفاده از شبکه‌های عصبی خود سازمانده"،

سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران، ۲۱ اسفند ۹۰.