

توسعه مبانی شبکه عصبی بر پایه اصل گسترش "زاده" با داده‌های فازی

شریف ملکوتی^۱، رضا کارگر^۲، زهره طائب^۳، هادی باقرزاده ولمی^۴

^(۱و۳و۴) گروه ریاضی کاربردی، دانشکده علوم پایه، واحد یادگار امام (ع)، دانشگاه آزاد اسلامی، تهران، ایران

^(۲) گروه ریاضی کاربردی، دانشکده علوم پایه، واحد قم، دانشگاه آزاد اسلامی، قم، ایران

تاریخ ارسال مقاله: ۱۴۰۰/۰۳/۲۹ تاریخ پذیرش مقاله: ۱۴۰۰/۰۷/۲۴

چکیده

شبکه‌های عصبی فازی به دلیل عدم استفاده از اصل گسترش "زاده" قادر به اعمال مفهوم فازی در ساختار شبکه نیستند. در این مقاله سعی شده است که اصل گسترش زاده را به صورت گسسته بیان کنیم سپس این اصل در الگوریتم پسا انتشار است. در این مقاله ابتدا مختصری از شبکه عصبی پسا انتشار بیان می‌شود. در ادامه مقاله، مطالب زیر دنبال می‌شود: بخش دوم تعاریف مختصر از شبکه عصبی و پس انتشار الگوریتم و مفاهیم اولیه فازی و اصل گسترش بیان می‌گردد.

بخش سوم معایب روش‌هایی که مبتنی بر اصل گسترش نمی‌باشند، بیان می‌گردد سپس با جایگزینی چهار عملگر اصلی که طبق اصل گسترش زاده باز تعریف شده‌اند روشی تازه ارائه می‌گردد. بخش چهارم یک مثال مطرح می‌شود که حل آن توسط روش ارائه شده در بخش سوم خواهد بود در بخش پنجم به جمع‌بندی نتایج روش ارائه شده پرداخته خواهد شد.

واژه‌های کلیدی: شبکه عصبی مصنوعی، منطق فازی، الگوریتم پسا انتشار، اصل گسترش "زاده".

۱- مقدمه

در طول دهه گذشته یا بیشتر، پیشرفت‌های قابل توجهی در دو حوزه تکنولوژیکی متمایز صورت گرفته است: منطق فازی و شبکه‌های محاسباتی. نظریه منطق فازی یک چارچوب ریاضی برای درک عدم قطعیت‌های مرتبط با فرآیندهای شناختی انسان، مانند تفکر و استدلال فراهم می‌کند. همچنین، ساختار ریاضی را برای تقلید از ویژگی‌های ادراکی و زبانی خاص مرتبط با شناخت انسان فراهم می‌کند. از سوی دیگر، الگوهای شبکه عصبی محاسباتی در فرآیند درک یادگیری باور نکردنی و ویژگی‌های تطبیقی مکانیزم‌های عصبی ذاتی در گونه‌های بیولوژیکی خاص، تکامل یافته‌اند. شبکه‌های عصبی محاسباتی، در مقیاس کوچک، برخی از عملیات محاسباتی مشاهده‌شده در یادگیری و سازگاری زیستی را تکرار می‌کنند. یکپارچه‌سازی این دو زمینه، منطق فازی و شبکه‌های عصبی؛ یک زمینه تکنولوژیکی نوظهور - شبکه‌های عصبی فازی - را به وجود آورده است. شبکه‌های عصبی فازی این پتانسیل را دارند که مزایای دو زمینه جذاب، یعنی منطق فازی و شبکه‌های عصبی را در یک کپسول واحد ارائه دهند. رویکردهای مختلفی برای ترکیب منطق فازی و شبکه‌های عصبی ارائه شده است. این رویکردها می‌توانند در چندین دسته طبقه‌بندی شوند. مثال زیر یک نمونه از چنین طبقه‌بندی شبکه‌های عصبی فازی است.

دسته ۱: سیستم‌های مبتنی بر قاعده فازی با توانایی یادگیری.

دسته ۲: سیستم‌های فازی نشان داده شده توسط معماری شبکه.

دسته ۳: شبکه‌های عصبی برای استدلال فازی.

دسته ۴: شبکه‌های عصبی فازی.

دسته ۵: رویکردهای دیگر [1].

نمونه‌های معمولی دسته اول سیستم‌های فازی قابل آموزش توسط واتانابه و ایشیوچی و نومورا^۲ و همکاران هستند. سیستم‌های فازی انطباقی توسط وانگ را نیز می‌توان در این دسته طبقه‌بندی کرد [2]، [3]. قوانین فازی در سیستم‌های فازی از داده‌های ورودی-خروجی با روش کاهشی آموزش داده می‌شود که مشابه الگوریتم پس انتشار برای شبکه‌های عصبی است [4].

در میان این روش‌ها، فقط شبکه عصبی فازی (FNN) توانایی ترکیب یادگیری شبکه‌های عصبی و تفسیرپذیری مبتنی بر قوانین پایه‌ای سیستم‌های فازی را دارد [5و6].

با این وجود در کاربردهای عملی دو مشکل مهم و اصلی برای طراحی شبکه عصبی فازی مطرح می‌شود:

نحوه مشخص کردن اندازه شبکه**شیوه تنظیم و سازگار کردن پارامترها**

برای تعیین اندازه FNN، چندین FNN خودسازمانده پیشنهاد شده است. برای مثال وانگ و همکارانش یک طرح خودسازمانده آنلاین، دقیق و سریع را برای FNN، (FAOS-PFNN) در مدل‌سازی سیستم‌های غیرخطی معرفی کردند. FAOS-PFNN قادر به تنظیم ساختار خود با تکیه بر خطای مدل‌سازی در فرایند آموزش می‌باشد. نتایج نشان می‌دهد که FAOS-PFNN می‌تواند یک ساختار فشرده و مدل‌سازی بسیار دقیقی را به دست آورد. روش‌های ذکر شده، تنها می‌توانند عصب‌های اضافی را در فرآیند یادگیری حذف کنند. این روش‌ها، از بار محاسباتی سنگینی برخوردارند، زیرا اکثر زمان آموزش، صرف فرآیند آموزش می‌شود که بیش از حد نیاز است. در اغلب FNN های خودسازمانده، طراحی ساختار نسبت به شناسایی پارامترها از اهمیت بالاتری برخوردار می‌باشند.

به منظور تنظیم پارامترهای FNN، الگوریتم پسا انتشار^۳ به عنوان یکی از پرتکرارترین تکنیک‌ها مورد استفاده قرار گرفته است. اما آنچنان که در مقالات به وفور دیده می‌شود، الگوریتم BP از زمان آموزش بسیار بالا و کمترین موضعی‌سازی رنج می‌برد. برای بهبود عملکرد آموزش، از یک الگوریتم آموزش هیبریدی تسریع‌یافته با ترکیب تکنیک‌های خوشه‌بندی در یک ورژن سازگار از الگوریتم BP برای FNN های آموزشی استفاده می‌شود. ژائو و همکارانش یک روش بهینه‌سازی لونیگ-مارکارد (LM) را با استفاده از یک رویکرد آموزشی گرادیان یکپارچه توسعه دادند تا تقریب دقیق‌تری را به دست آورند. ایرادی که بر الگوریتم LM و نوع‌های دیگر آن وارد است، افزایش الزامات حافظه، ناشی از محاسبه ماتریس ژاکوبین تابع خطا است. مساله دیگر اینکه این الگوریتم هنوز هم یک روش بهینه‌سازی موضعی است.

ماشینچی و همکارانش یک روش آموزش مبتنی بر الگوریتم ژنتیک دوفازی (GA) را پیشنهاد دادند که در آن FNN مبتنی بر GA، می‌تواند وزن‌های فاز بهینه را تخمین زده و برآورد خوبی از شکل تابع عضویت ارائه دهد. همچنین مائو و همکارانش یک شبکه عصب فازی را بر اساس روش بهینه‌سازی ازدحام ذرات (IOAP-FNN) ابداع کردند. این روش می‌تواند رابطه بین سیگنال‌های شناسایی فرکانس رادیو و موقعیت یک سبد خرید را تعیین کند و نتایج مدل‌سازی، اثربخشی روش را ثابت می‌کند. الگوریتم‌های تکاملی جدیدتری جهت بهینه‌سازی پارامترهای FNN معرفی شد که از زمان اجزایی بالایی برخوردار می‌باشند. [7]

ایشیوچی و همکاران، معماری شبکه عصبی فازی با اعداد مثلثی فازی متقارن را برای اوزان، بایاس و ورودی‌های فازی پیشنهاد کرد، که با حد پایین، وسط و بالا اعداد مثلث فازی مشخص می‌شود. [8]

$$a = (m_a, \lambda_a, p_a)$$

بر اساس کار ایشیوچی که در آن از وزن فازی مثلثی استفاده می‌شود، یک الگوریتم یادگیری ایجاد کرد که در آن از الگوریتم برای محاسبه اوزان استفاده می‌شود. مقدار وزن فازی جدید به عنوان پارامترهای جدید محسوب می‌شود. در این مقاله ما یک شبکه فازی ارائه می‌دهیم که رابط‌های ورودی و خروجی بر اساس اصل گسترش‌زاده باز تعریف شده است.

با توجه به نوع ورودی‌ها و اوزان سه نوع شبکه عصبی فازی معرفی خواهیم کرد:

الف) وزن قطعی و ورودی فازی

ب) وزن فازی و ورودی قطعی^۴

ج) ورودی و اوزان هر دو فازی هستند.

در این مقاله در خصوص حالت (ج) که همه موارد قبلی را هم مشمول می‌شود و (کلیت) دارد بررسی و مطالعه انجام خواهد شد.

تحقیق و مطالعه در این موضوع دو خاصیت و ویژگی دارد یکی در موضوع هوش مصنوعی که قابلیت عملکرد مستقل آن با استفاده از آموزش و یادگیری از طریق آموزش که خود امکان سازگاری با شرایط جدیدی را در پی خواهد داشت و دیگری منطق فازی به عنوان ابزاری برای توصیف دانش فازی (حلقه واصل جهان طبیعت با علم ریاضی) است. که در این تحقیق سعی بر استفاده از ترکیب هر دو موارد و استفاده از مزایای آن شده است قبلاً هم در این خصوص اقداماتی صورت گرفته که در اینجا سعی بر تکمیل این ویژگی و استفاده از اصل گسترش‌زاده در همه

^۳ BackPropagation

^۴ Crisp

جوانب شده است. بالاخص در محاسبات حاصلضرب اوزان بر ورودی‌های فاز و مجموع آن با بایاس است که موجب محاسبات دقیق‌تر و نهایتاً تصمیم‌گیری صحیح‌تر خواهد شد.

۲- مفاهیم اولیه

۲-۱ شبکه عصبی مصنوعی:

شبکه عصبی مصنوعی^۵ یک الگوریتم محاسباتی توزیع شده بر اساس ساختار سیستم عصبی انسان است. معماری یک شبکه عصبی با اتصال چندین پردازنده ابتدایی شکل می‌گیرد، این یک سیستم تطبیقی است که دارای الگوریتمی برای تنظیم اوزن و پارامترهای آزاد برای دستیابی به کارایی مورد نیاز مسئله بر اساس نمونه داده نماینده است [9]. استفاده از شبکه‌های عصبی در بسیاری از زمینه‌های تحقیقاتی گسترش و رشد یافته است به ویژه روش بسیار مهمی برای پیش‌بینی سری‌های زمانی و تشخیص الگو می‌باشد.

یکی از انواع شبکه‌های عصبی نظارتی که در مطالعه ما بیشتر مورد توجه قرار خواهد گرفت، شبکه مبتنی بر پسا انتشار است. این نوع شبکه بیشتر در زمینه‌های ذکر شده در بالا مورد استفاده قرار می‌گیرد.

این شبکه عصبی مبتنی بر یادگیری نظارت شده است، که شبکه با داشتن مقادیر صحیح ورودی و خروجی کار می‌کند و شبکه وزنه‌های یال‌های خود را تنظیم و اصلاح می‌کند تا خطای خروجی محاسبه شده را به حداقل برساند. اوزان شبکه عصبی در مرحله یادگیری از اهمیت ویژه برخوردار است، و بر عملکرد فرایند یادگیری شبکه عصبی تأثیر بسزایی دارد. مهمترین ویژگی شبکه‌های عصبی مصنوعی توانایی آنها در یادگیری از مجموعه‌ای از الگوهای آموزشی است، یعنی قادر به یافتن مدلی متناسب با داده‌های جدید است [9].

نورون مصنوعی از چندین قسمت تشکیل شده است (شکل ۱ را ببینید). در یک طرف ورودی‌ها، اوزن، مجموع و در آخر تابع فعال وجود دارد. مقادیر ورودی در اوزان ضرب می‌شوند و در نهایت با هم جمع خواهند شد.

$$\sum_{i=1}^n x_i w_{ij} \quad (2-1)$$

این تابع با اضافه کردن مقدار آستانه b_i تکمیل می‌شود که مقدار آستانه (بایاس) در شبکه عصبی برابر با مقدار +۱ می‌باشد.

$$\sum_{i=0}^n x_i w_{ij}, \quad x_0 = b_i, w_0 = +1 \quad (2-2)$$

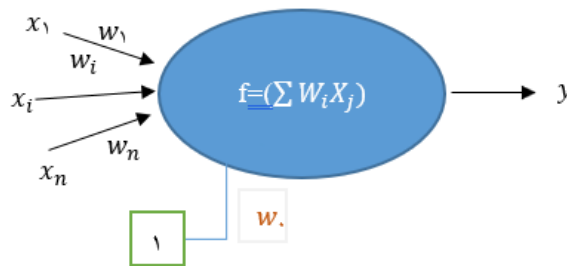
بعد از اضافه کردن مقدار آستانه در ادامه تابع f را به مجموع بالا اعمال می‌کنیم، نتیجه مقدار نهایی خروجی است که با y_i نمایش داده می‌شود.

$$y_i = f \left(\sum_{i=0}^n x_i w_{ij} \right) \quad (2-3)$$

با اعمال تابع به مجموع رابطه (۱) در نهایت مقدار خروجی به دست خواهد آمد که تابع f می‌تواند یک تابع غیرخطی با ارزش باینری +۱ یا -۱ باشد یا یک تابع خطی $f(z) = z$ و یا یک تابع لوجستیک سیگموئید به فرمول زیر می‌باشد. [10]

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

^۵ Artificial neural network(ANN)



شکل ۱ (نرون مصنوعی): قسمت‌ها مختلف

۱-۱-۲ الگوریتم پسا انتشار^۶

کارهای زیادی در خصوص تنظیم یا کنترل اوزان وجود دارد اما در این تحقیق مهمترین آن که الگوریتم پسا انتشار است مورد بررسی قرار خواهد گرفت [11], [12].

در این روش، در هر دور (یعنی در هر تکرار) دو مرحله خواهیم داشت. مرحله‌ی اول حرکت رو به جلو (پیشرو) ^۷ است، با ضرب داده‌های ورودی در وزن‌ها و سپس جمع آن با مقدار آستانه^۸ انجام می‌شود. سرانجام در همان مرحله اول به یک خروجی می‌رسیم که احتمالاً با خروجی واقعی تفاوت دارد. اینجاست که توسط تابع خطا (زیان) مشخص می‌کنیم که مرحله پیشرو چه مقدار خطایی داشته است. حال که فهمیدیم الگوریتم با توجه به وزن‌ها و انحراف‌ها چه مقدار خطایی دارد، به مرحله دوم می‌رویم. در این مرحله می‌توانیم به عقب بازگشته و وزن‌ها و انحراف‌ها را هنگام‌سازی کنیم، یعنی وزن‌ها و انحراف‌ها را به شکلی تغییر دهیم تا در تکرار بعدی نتیجه‌ای نزدیک‌تر به خروجی واقعی و با خطای کم‌تر را تولید کنند. این تکرار پیشرو و پسا انتشار آنقدر انجام می‌شود تا خروجی شبکه برای تمامی داده‌های آموزشی به نزدیک‌ترین مقدار واقعی خود برسد. به این ترتیب الگوریتم یاد گرفته و از این به بعد می‌تواند با مشاهده‌ی ویژگی‌های یک داده که تا حالا ندیده است، تصمیم بگیرد.

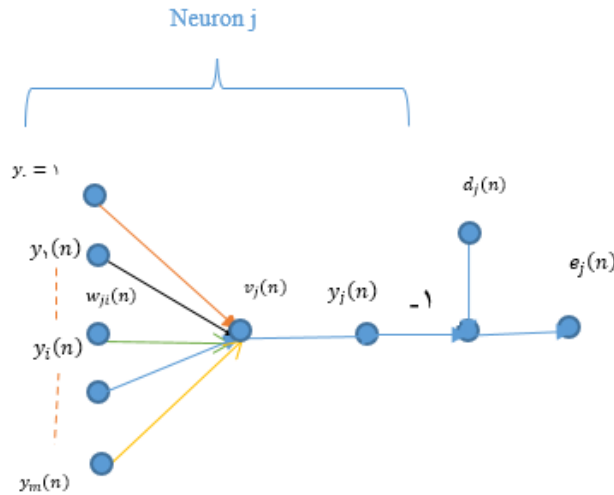
نرخ یادگیری انطباقی، با اجازه دادن به تغییرات در میزان یادگیری در فرایند آموزش (افزایش یا کاهش) می‌توان بر بهبود عملکرد الگوریتم تاثیر بسزای گذاشت [13].

در خصوص الگوریتم پسا انتشار لازم به توضیح است که با توجه به همگرایی گرادین کاهشی و تحت تاثیر یودن الگوریتم از آن این الگوریتم نیز همگرا خواهد بود و اوزان در جهت کاهشی تغییر خواهد کرد. در ادامه به بیان ریاضی الگوریتم پسا انتشار در ساختار شبکه عصبی می‌پردازیم [14]. شکل ۲ را در نظر بگیرید:

^۶ back propagation of error

^۷ feed forward

^۸ Bias



شکل ۲ (الگوریتم پسا انتشار): با بیان ریاضی

همانطور که در شکل ۲ مشخص شده نورون j توسط مجموعه سیگنال‌های فعالی تغذیه می‌شود که توسط یک لایه نورون در سمت چپ آن تولید شده است. مقدار $v_j(n)$ حاصل مجموع ورودی‌های تابع فعال مرتبط با نورون j می‌باشد، پس می‌توان نوشت:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (2-4)$$

که در آن m تعداد کل ورودی‌ها (به استثنای بایاس) اعمال شده به نورون j است. وزن سیناپسی w_{j0} (مربوط به ورودی ثابت $y_0 = +1$) برابر بایاس b_j اعمال شده بر نورون j است. از این رو، سیگنال فعال $y_i(n)$ در خروجی نورون j در تکرار n به صورت زیر ظاهر می‌شود:

$$y_i(n) = \varphi_j(v_j(n)) \quad (2-5)$$

به روش مشابه الگوریتم LMS، الگوریتم پسا انتشار یک $\Delta w_{ji}(n)$ تصحیح شده را برای وزن سیناپسی $w_{ji}(n)$ اعمال می‌کند، که متناسب با مشتق جزئی $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$ می‌باشد. بر طبق قانون زنجیره‌ای حساب دیفرانسیل گرادینان را به صورت زیر بیان می‌کنیم:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \quad (6-2)$$

مشتق جزئی $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$ یک عامل حساس و تعیین کننده در جهت‌دهی جستجوی فضای وزن سیناپس $w_{ji}(n)$ است، لذا پس از ساده کردن خواهیم داشت:

$$(\varphi_j) \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi_j'(v_j(n)) y_j(n) \quad (2-7)$$

$\Delta w_{ji}(n)$ تصحیح شده بر $w_{ji}(n)$ که تعریف شده بوسیله قانون دلتا است به عبارتی داریم:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad (2-8)$$

η پارامتر نرخ یادگیری الگوریتم پسا انتشار می‌باشد.

علامت منهای ظاهر شده در معادله (۲-۸) بر اساس الگوریتم گرادیان کاهشی تعریف شده است (تغییر جهت وزن به کاهشی شدن تضمین کننده کاهش مقدار $\mathcal{E}(n)$ است) و با عنایت به روابط به (۲-۸) و (۲-۷) داریم :

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (2-9)$$

که گرادیان موضعی $\delta_j(n)$ به صورت زیر تعریف می‌گردد:

$$\begin{aligned} \delta_j(n) &= \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} \\ &= \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n) \phi_j'(v_j(n)) \quad (2-10) \end{aligned}$$

گرادیان موضعی به تغییرات مورد نیاز در وزن سیناپسی اشاره می‌کند.

طبق رابط (۲-۱۰) گرادیان موضعی خروجی نرون j ($\delta_j(n)$) برابر با حاصلضرب خطای سیگنال نرون j ($e_j(n)$) در مشتق جرئی تابع فعالساز مربوط به آن ($\phi_j'(v_j(n))$) است.

از معادلات (۲-۹) و (۲-۱۰) متوجه می‌شویم یک عامل کلیدی در محاسبه و تنظیم وزن ($\Delta w_{ji}(n)$) خطای سیگنال ($e_j(n)$) خروجی نرون j می‌باشد.

در این صورت با دو حالت مواجه هستیم که بستگی به جایی که نرون j در شبکه واقع شده است دارد. در مورد اول، نرون j یک گره خروجی است. محاسبه این مورد ساده است زیرا هر گره خروجی شبکه در واکنش مربوطه به خود، تولید می‌شود که از آن می‌توان مستقیماً میزان خطای سیگنال مربوطه را محاسبه نمود، اما در مورد دوم نرون j قابل مشاهده نمی‌باشد، در گره‌های پنهان گرچه مستقیماً قابل مشاهده و دسترس نیستند ولی در خطای تولیدی در خروجی شبکه سهیم هستند. با این وجود باید مقدار پاداش و جریمه گره‌های پنهان و میزان تأثیر آن در ایجاد خطا به نحوی محاسبه و لحاظ گردد.

مورد اول : نرون j گره خروجی

نرون j در لایه خروجی شبکه قرار داشته باشد که آن واکنش دلخواه خود را برای پاسخ دارد. می‌توانیم از معادله

$$e_j(n) = d_j(n) - y_j(n)$$

برای محاسبه خطای سیگنال مربوطه به این نرون $e_j(n)$ با تعیین $e_j(n)$ ، محاسبه گرادیان موضعی $\delta_j(n)$ با معادله (۲-۱۰) امری ساده است.

مورد دوم : نرون j گره پنهان

وقتی نرون j در یک لایه مخفی (میانی) از شبکه قرار دارد، واکنش دلخواهی نخواهد داشت. بر این اساس، تشخیص خطای سیگنال برای یک نرون پنهان به صورت بازگشتی تعیین می‌شود و از نظر خطای سیگنال به تمام نرون‌هایی

که آن نرون پنهان مستقیماً با آنها ارتباط دارد عقب گرد خواهد داشت که موجب پیچیده شدن توسعه الگوریتم پسا انتشار می‌شود.

نرون j را به عنوان یک گره پنهان در شبکه در نظر می‌گیریم.

مطابق با معادله (۲-۱۰) باید گرادیان موضعی $\delta_j(n)$ برای نرون پنهان j به ترتیب زیر دوباره تعریف گردد:

$$\begin{aligned}\delta_j(n) &= -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \phi_j'(v_j(n))\end{aligned}$$

(۲-۱۱) نرون j پنهان است

در خط دوم معادله فوق از رابطه (۶،۱) استفاده شده است. برای محاسبه مشتق جزئی $\frac{\partial \mathcal{E}(n)}{\partial y_j(n)}$ و در ادامه فرآیند خواهیم داشت:

(۲-۱۲) نرون k یک گروه خروجی است

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

که از جایگزین اندیس k به جای j در معادله

$$\mathcal{E}(n) = \sum_{j \in C} \mathcal{E}_j(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

بدست می‌آید. این تغییر را چون نرون j پنهان است به اجبار انجام می‌دهیم.

در ادامه با مشتق معادله (۲-۱۲) نسبت به سیگنال فعال $y_j(n)$ خواهیم داشت :

$$(۲-۱۳) \quad \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)}$$

سپس از قانون زنجیره ای برای مشتق جزئی $\frac{\partial e_k(n)}{\partial y_j(n)}$ استفاده می‌کنیم و معادله (۲-۱۳) را بازنویسی می‌کنیم.

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (14-2)$$

و خواهیم داشت:

(۲-۱۵) نرون k یک گره خروجی است

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \phi_k v_k(n)$$

از این رو:

$$(۲-۱۷) \quad \frac{\partial e_k(n)}{\partial v_k(n)} = -\phi_k'(v_k(n))$$

برای نرون k داریم:

$$(۲-۱۷) v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n)$$

با توجه به مشتق رابطه (۱۷-۲) نسبت به $y_j(n)$ خواهیم داشت:

$$(۲-۱۸) \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

با عنایت به رابطه (۱۸-۲) و (۱۶-۲) و اعمال آن در رابطه (۱۴-۲) خواهیم داشت:

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = - \sum_k e_k(n) \varphi'_k(v_k(n)) w_{kj}(n) = - \sum_k \delta_k(n) w_{kj}(n) \quad (۱۹-۲)$$

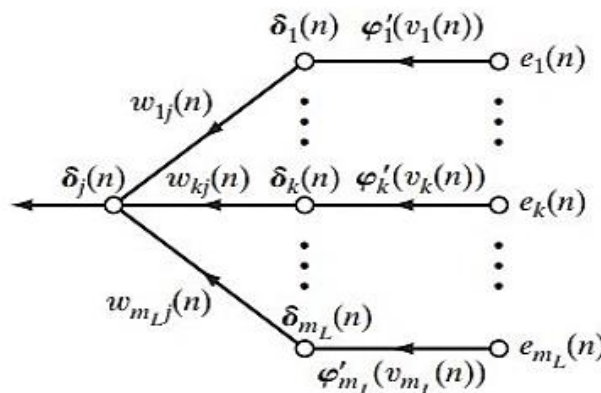
که در آن از تعریف گرادیان موضعی δ_k استفاده شده در رابطه (۲-۱۰) بکار رفته است، سرانجام رابطه (۲۰-۲) را در رابطه (۱۱-۲) جایگزین کرده و داریم:

(۲-۲۰) نرون j پنهان است

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

رابطه فوق به فرمول پسا انتشار با گرادیان موضعی $\delta_j(n)$ معروف می‌باشد.

شکل ۳ نمایش نمودار سیگنال جریان رابطه (۲۱-۲) را نشان می‌دهد. با فرض اینکه لایه خروجی از سلول‌های عصبی m_I تشکیل شده است.



شکل ۳ (جریان سیگنال نمودار): قسمتی از الحاقی سیستم مربوط به پسا انتشار سیگنال های خطا.

فاکتور $\varphi_j'(v_j(n))$ در رابطه (۲۰-۲) که در محاسبه گرادیان موضعی $\delta_j(n)$ نقش دارد تنها بستگی به تابع فعالساز مربوط به نرون پنهان j دارد.

روابط بدست آمده برای الگوریتم پسا انتشار را می‌توان به صورت ذیل خلاصه نمود:

اول: $\square w_{ji}(n)$ تصحیح شده و اعمال شده بر وزن سیناپسی متصل به نورون i به نورون j که طبق قانون دلتا تعریف شده است:

(وزن تصحیح شده $\square w_{ji}(n)$) = (نرخ یادگیری η) $\times$ (گرادیان موضعی $\delta_j(n)$) $\times$ (سیگنال ورودی j ، $y_j(n)$)
رابطه (۲۱-۲)

دوم: گرادیان موضعی $\delta_j(n)$ که وابسته به گره خروجی یا پنهان بودن نرون j می‌باشد.

الف- در صورتی که نرون j گره خروجی باشد $\delta_j(n)$ مساوی است با حاصلضرب $\varphi_j'(v_j(n))$ و سیگنا خطای $e_j(n)$ ، که هر دو مربوط به نرون j می‌باشد. (رابطه (۱۱،۱) ملاحظه شود)

ب- اگر نرون j پنهان باشد $\delta_j(n)$ برابر است با حاصلضرب $\varphi_j'(v_j(n))$ مربوطه و مجموع اوزان δS محاسبه شده برای نرون‌های جانبی یا لایه خروجی متصل به نرون j می‌باشد (رابطه (۲۱-۲) ملاحظه شود)

۳- اصل گسترش زاده:

قبل از آن که به بیان اصل گسترش بپردازیم لازم است که به دو مفهوم اشاره شود:

۳-۱ حاصل ضرب دکارتی مجموعه‌های فازی:

فرض می‌کنیم که A و B دو مجموعه نامتناهی فازی باشند آنگاه داریم:

$$A = \left\{ \frac{x}{\mu(x)}, x \in X, \mu(x) \in [0,1] \right\}$$

$$B = \left\{ \frac{y}{\mu(y)}, y \in Y, \mu(y) \in [0,1] \right\}$$

$$A \times B = \left\{ \frac{(x,y)}{\mu(x,y)}, \mu(x,y) = \min \{ \mu(x), \mu(y) \}, (x,y) \in X \times Y \right\}$$

۳-۲ گسسته‌سازی اعداد فازی

در این روش که جایگزینی برای محاسبات فازی با استفاده از آلفا برش‌ها است به جهت افزایش میزان دقت محاسبات فازی بسته به توانایی سیستم پردازش متغیر است. بدین منظور که هر چقدر توان پردازش بالاتر باشد محاسبات فازی دقیق‌تر و با همان نسبت مقدار شبکه عصبی فازی به عدد بهینه نزدیک‌تر خواهد بود در ادامه روش گسسته‌سازی فازی را تعریف و طرز استفاده آن را توضیح خواهیم داد:

گسسته‌سازی در عملیات ماشین‌های محاسباتی لازم است که نخست فرض کنید که $\tilde{A} = (0,1,2)$ یک فازی مثلثی باشد. (شکل ۴ را ملاحظه کنید)

$$\mu(x) = \begin{cases} x & 0 \leq x \leq 1 \\ 2-x & 1 \leq x \leq 2 \\ 0 & \text{etc} \end{cases}$$

وقتی این عدد گسسته می‌شود به شکل زیر است تعداد نقاط ۲۱ عدد می‌باشد.

$$\tilde{A}_d = \left\{ \frac{0}{0}, \frac{.1}{.1}, \frac{.2}{.2}, \dots, \frac{1.8}{.2}, \frac{1.9}{.1}, \frac{2}{0} \right\}$$

فرض کنیم تابعی چون f از مجموعه قطعی X به مجموعه قطعی Y موجود باشد به طوری که:
 $f: X \rightarrow Y$
 حال فرض کنیم که مجموعه X فازی شده باشد حال می‌خواهیم برای $f(x)$ درجه عضویت را محاسبه کنیم
 یعنی:

$$\mu(f(x))$$

برای این منظور نخست تصویر معکوس $f(x)$ را محاسبه کنیم یعنی

$$f^{-1}(\{f(x)\}) = \{t \in X : f(t) = f(x)\}$$

دقت کنید که عبارت مذکور الزاماً مجموعه تک عضوی $\{x\}$ نخواهد بود مگر اینکه f یک به یک باشد. حاصل اصل گسترش "زاده" چنین تعریف می‌کنیم:

$$\mu(f(x)) = \sup \{ \mu(t) : t \in f^{-1}(\{f(x)\}) \}$$

هنگامی که از گسسته‌فازی استفاده کنیم می‌توانیم از مفهوم max برای sup استفاده کنیم.

مثال: فرض کنید که اعداد حقیقی به شرح ذیل فازی شده است:

$$\mu(x) = \begin{cases} \frac{(x+2)^2}{4} & -2 \leq x \leq 0 \\ 1 - \frac{x}{2} & 0 \leq x \leq 2 \\ 0 & \text{etc} \end{cases}$$

بنا داریم که برای تابع $f(x) = x^2$ با دامنه $\mathbb{R}$ درجه عضویت محاسبه کنیم. بنابراین برد تابع f اعداد حقیقی نامنفی است حال این برد را به دو قسمت افراز می‌کنیم:

$$R_f = [0, 4] \cup (4, \infty)$$

بنابراین اگر $b \in (4, \infty)$ آنگاه:

$$f^{-1}(\{b\}) = \mathbb{R} - [-2, 2] \rightarrow \mu(b) = 0$$

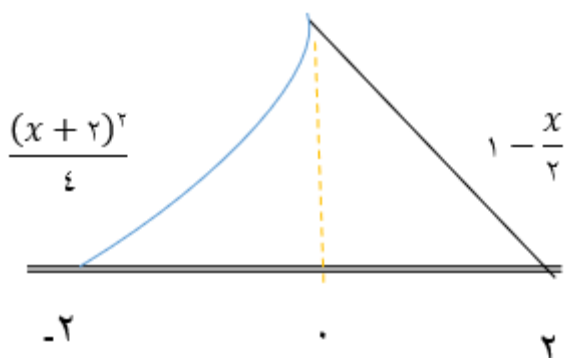
و اگر $b \in [0, 4]$

$$f^{-1}(\{b\}) = \{\sqrt{b}, -\sqrt{b}\}$$

$$\mu(b) = \max \left\{ \frac{(-\sqrt{b} + 2)^2}{4}, \left(1 - \frac{\sqrt{b}}{2}\right) \right\} = \left(1 - \frac{\sqrt{b}}{2}\right)$$

بنابراین :

$$\mu_f(Y) = \begin{cases} 1 - \frac{\sqrt{b}}{2} & , Y \in [0, 4] \\ 0 & etc \end{cases}$$



شکل ۴ (عدد فازی گسسته)

این محاسبه ظاهراً ساده بوده ولی در توابع پیچیده به این سادگی نخواهد بود بنابراین لازم است که در عملیات محاسباتی از گسسته‌سازی استفاده شود.

روش اصولی آن به شرح زیر است:

الف- ابتدا اعداد فازی را گسسته کنیم.

ب- $f(x)$ را محاسبه می‌کنیم

ج- شبیه‌یابی می‌کنیم:

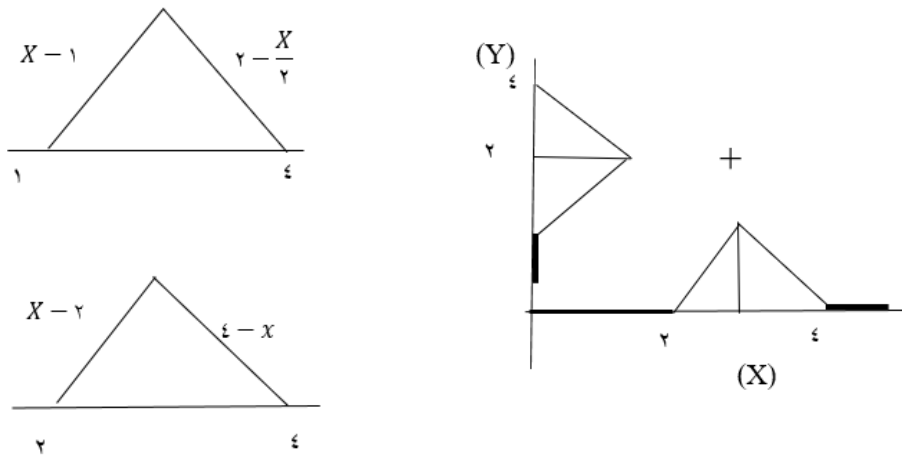
$$x_i \square x_j \Rightarrow |f(x_i) - f(x_j)| < \varepsilon$$

$$\min\{\mu(f(x_j))\} \text{ برای } f(x_i) \text{ در نظر می‌گیریم}$$

۳-۳ مثال دوم جمع دو عدد فازی:

اینجا می‌خواهیم جمع دو عدد فازی را حساب کنیم عمل جمع به شرح زیر است (شکل ۵ ملاحظه شود)

$$f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \quad f(x, y) = x + y$$



شکل ۵ (جمع دو عدد فازی)

پس

$$[2,4] + [1,4] = [3,8]$$

تمرکز بر بازه‌ی $[1,4]$ و $[2,4]$ می‌باشد به عنوان مثال اگر بخواهیم $f(x, y) = 5$ بینهایت زوج موجود است که

$$x + y = 5$$

$$\mu(x, y) = \min\{\mu(x), \mu(y)\} = 0$$

$$\mu(f(x + y)) = 5$$

$$= \sup\{\mu(x + y) : x + y = 5\}$$

بنابراین اگر یکی از این دو خارج از بازه باشد در $\sup$ یا $\max$ حذف می‌شوند. حال برای $3 \leq z \leq 8$ ، $\mu(z)$ را محاسبه می‌کنیم.

$$\mu(z) = \sup\{\min(\mu(x), \mu(y)) : x + y = z\}$$

اگر $z = 5$ آنگاه

$$x = 2 \quad \mu_x(2) = 1 \quad 2 \in X$$

$$y = 3 \quad \mu_y(3) = 1 \quad 3 \in Y$$

$$x + y = 5 \Rightarrow \mu(x, y) = \min\left\{\mu_x(2), \mu_y(3)\right\} = 1$$

$$\mu_z(5) = \sup_{x+y=5} \left\{ \min\left\{\mu_x(x), \mu_y(y)\right\} \right\} = 1$$

زیرا یک جفت یافت شد که $\mu(x, y) = 1$ از سویی دیگر همواره کمتر یا مساوی یک است پس $\mu_z(5) = 1$ حال برای $c \leq 5$ است به محاسبات طولانی می‌رسیم.

$$x + y = c \Leftrightarrow \begin{cases} 2 \leq x \leq 3 & 1 \leq y \leq 2 \\ 2 \leq x \leq 3 & 2 \leq y \leq 4 \\ 3 \leq x \leq 4 & 1 \leq y \leq 2 \\ 3 \leq x \leq 4 & 2 \leq y \leq 4 \end{cases}$$

حالت چهارم تنها در $x = 3$ و $y = 2$ رخ می‌دهد

$$5 \leq x + y \leq 8$$

که $c = 5$ که قبلاً بحث شد حال سه حالت دیگر مورد بررسی قرار می‌گیرد:
حالت اول:

$$(2 \leq x \leq 3 \quad 1 \leq y \leq 2)$$

$$\mu_x(x) = x - 2$$

$$\begin{aligned} \mu_y(y) &= y - \min\left\{\mu_x(x), \mu_y(y)\right\} \\ &= \frac{x - 2 + (y - 1) - |x - 2 - (y - 1)|}{2} \\ &= \frac{x + y - 3 - |x - y - 1|}{2} \end{aligned}$$

$$, (3 \leq c \leq 5)$$

حالت دوم: $(2 \leq x \leq 3 \quad 2 \leq y \leq 4)$

$$\mu_x(x) = x - 2, \mu_y(y) = \frac{4 - y}{2}$$

$$\min\left\{\mu_x(x), \mu_y(y)\right\} = \frac{x - 2 + \left(\frac{4 - y}{2}\right) - \left|x - 2 - \left(\frac{4 - y}{2}\right)\right|}{2} = \frac{2x - y - |2x + y - 8|}{4}$$

اما از سویی $2 \leq x \leq 3 \quad x + y \leq 5$

بنابراین

$$\begin{cases} 2x + y \leq 8 \\ 2x + y - 8 \leq 0 \end{cases}$$

پس $4 \leq C \leq 5$,

$$\frac{2x - y - |2x + y - 8|}{4} = \frac{2x - y + 2x + y - 8}{4} = x - 2$$

حالت سوم را بررسی می‌کنیم

$$3 \leq x \leq 4$$

$$1 \leq x \leq 2$$

$$\min\left\{\mu_x(x), \mu_y(y)\right\} = \frac{4-x+y-1-|4-x-(y-1)|}{2}$$

$$= \frac{y-x+3-|5-(x+y)|}{2}$$

اما $4 \leq C \leq 5$ پس: $x+y \leq 5$

$$\frac{y-x+3+x+y-5}{2} = y-1$$

در حالت اول (A)

$$\frac{x+y-3-|x-y-1|}{2}$$

ماکزیمم این حالت برای $x+y=c$ زمانی رخ می‌دهد که

$$|x-y-1|=0$$

یعنی

$$x-y=1$$

پس

$$x+y=c$$

$$x-y=1$$

$$x = \frac{c+1}{2}$$

چون $3 \leq c \leq 5$ بنابراین

$$2 \leq x \leq 3$$

پس دستگاه جواب دارد بنابراین ماکزیمم در ناحیه اول یعنی:

$$2 \leq x \leq 3$$

$$1 \leq y \leq 2 \quad \mu_z(c) = \frac{c-3}{2}$$

در حالت دوم ماکزیمم زمانی رخ می‌دهد که:

$$\mu_z(c) = x-2$$

$$x+y=c, \quad 2 \leq y$$

که y همواره 2 باشد زیرا افزایش y کاهش x را دربردارد و کاهش x و کاهش y پس $x=c-2$

$$\mu_z(c) = c-4$$

در حالت سوم نظیر حالت دوم

$$\mu_z(c) = y-1$$

$$x+y=c, \quad 3 \leq x < 4$$

ماکزیمم زمانی رخ می‌دهد که $x=3$ زیرا افزایش x کاهش y و متعاقباً کاهش μ_z پس $y=c-3$

$$\mu_z(c) = c-3-1 = c-4$$

حال مطابق اصل گسترش زاده

$$\mu_z(c) = \max \left\{ \frac{c-3}{2}, c-4, c-4 \right\}$$

اگر $4 \leq c \leq 5$ و اگر $3 \leq C \leq 4$ که مشخصات

$$c-3=2$$

$$\frac{c-3}{2} \geq c-4$$

$$c-3 \geq 2c-8$$

$$C \leq 5$$

بنابراین $3 \leq c \leq 5$

$$\mu_z(c) = \frac{c-3}{2}$$

به همین شیوه استدلال می‌توان نشان داد که

$$5\mu_z(c) = \frac{8-c}{3}, \quad 5 \leq z \leq 8$$

در واقع یک فرآیند خاص محاسباتی نشان دادیم که جمع این دو عدد فازی بنابر اصل گسترش فازی مثلثی است. اثبات در حالت کلی سیار طولانی و خسته‌کننده است. در حالت ضرب و تقسیم به راحتی مثال نقص وجود دارد که ضرب دو عدد فازی مثلثی لزوماً فازی مثلثی نیست بلکه همواره فازی مثلثی نیست.

همان طور که گفته شد اصل گسترش نیاز به گسسته سازی دارد و تنها روش ممکن نیز همین است. این اصل ابزاری است برای گسترش و تعمیم مفاهیم ریاضی غیر فازی به گونه ای که بصورت کمیت‌های فازی درآیند. این اصل به ویژه در تعمیم عملگرهای جبری بین اعداد و تعریف این عملگرها برای اعداد فازی مفید است.

۳-۴ تعریف اعمال اصلی بر اساس اصل گسترش زاده:

فرض کنید $M, N \in F(R)$ با توابع عضویت پیوسته باشند و $IR \times IR \rightarrow IR$ یک عملگر دوتایی بر اعداد حقیقی باشد. اگر تعمیم $*$ را برای اعداد فازی؛ $\otimes$ نشان دهیم.

با استفاده از اصل گسترش حاصل $M \otimes N$ به صورت یک مجموعه فازی یا تابع عضویت زیر تعریف می‌شود.

$$(M \otimes N)(z) = \sup \min [M(x), N(y)]$$

$$Z = x * y$$

۳-۵ تعمیم چهار عمل اصلی برای اعداد فازی:

در حالت خاص برای چهار عمل اصلی، تعریف فوق به صورت‌های زیر در می‌آید:

$$z = x + y \dots\dots\dots (M \oplus N)(z) = \sup \min [M(x), N(y)]$$

$$z = x - y \dots\dots\dots (M \ominus N)(z) = \sup \min [M(x), N(y)]$$

$$z = x * y \dots\dots\dots (M \square N)(z) = \sup \min[M(x), N(y)]$$

$$z = \frac{x}{y} \dots\dots\dots (M \div N)(z) = \sup \min[M(x), N(y)]$$

۴- شبکه عصبی فازی

با نگاه اجمالی به روابط و عملگرهای فازی و گسترش فرمولها و روابط قانون‌های یادگیری در محاسبات با اعداد فازی اوزان یال‌ها در این نوع محاسبات به شکل ذیل انجام خواهد شد:

$$w_{ji}(t+1) = w_{ji}(t) \oplus w_{ji}(t)$$

شدت تغییرات اوزان به نرخ یادگیری و میزان خطای شبکه بستگی دارد به طوری که اندازه تغییرات اوزان به خطا بستگی دارد اگر تغییرات اوزان همگرا به صفر باشد تغییرات خطا نیز به صفر همگرا است. بنابراین اگر خطای شبکه به صفر برسد همگرایی واقعی بدست می‌آید و اگر خطا و وزن‌ها کوچک شوند شبکه نیز تقریباً همگرا می‌شود.

۴-۱ مزایای این روش نسبت به الگوریتم استاندارد پسا انتشار:

شبکه می‌تواند از اعداد فازی استفاده نماید.

یادگیری در یک شبکه فازی سرعت بیشتری از یادگیری پسا انتشار در شبکه با حالت استاندارد دارد.

کار بر روی اعداد فازی هدف اصلی این کار است.

الگوریتم فازی با استفاده از آموزش، توانایی ایجاد کنترل‌های جدید برای نظارت در عملکرد شبکه را به صورت غیردستی و خودکار بویژه در محاسبه مقادیر بزرگتر را دارد.

قبل از بررسی و توصیف شبکه‌های عصبی فازی به طور خلاصه شبکه‌های عصبی پیش‌رو استاندارد (غیرفازی) را معرفی می‌کنیم. یک شبکه عصبی ۳ لایه پیش‌رو با n_1 ورودی و n_H بخش میانی و n_O بخش خروجی را در نظر می‌گیریم.

یک بردار n_1 بعدی اعمال شده به شبکه عصبی به صورت ذیل تعریف می‌کنیم:

$$X_p = (X_{p1} \dots\dots X_{pn_1})$$

رابطه ورودی-خروجی هر کدام از بخش را می‌توان به صورت زیر نوشت:

بخش ورودی:

$$O_{pi} = x_{pi} \quad i = 1, 2, \dots, n_1 \quad (3-1)$$

لایه میانی:

$$O_{pj} = f(net_{pj}) \quad j = 1, 2, \dots, n_H \quad (3-2)$$

$$net_{pj} = \sum_{i=1}^{n_1} w_{ji} O_{pi} + \theta_j \quad j = 1, 2, \dots, n_H \quad (3-3)$$

لایه خروجی:

$$O_{pk} = f(net_{pk}) \quad k = 1, 2, \dots, n_O \quad (3-4)$$

$$(3-5) \quad net_{pk} = \sum_{j=1}^{n_H} w_{kj} O_{pj} + b_k \quad k = 1, 2, \dots, n_O$$

این مجموع بواسطه تابع سیگموئید در بازه $[0, 1]$ تصویر می‌شود.

w_{kj} و w_{ji} اوزان متصل و b_k و b_j بایاس هستند و f تابع فعال که به صورت زیر تعریف می‌شود:

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

بردار هدف (خروجی) مربوط به بردار ورودی X_p به طور زیر مشخص می‌کنیم:

$$\tau_p = (\tau_{p1}, \tau_{p2}, \dots, \tau_{pn_o})$$

تابع هزینه جهت به حداقل رساندن آموزش شبکه عصبی را می‌توان به صورت زیر تعریف نمود:

$$e_p = \sum_{k=1}^{n_o} \frac{(\tau_{pk} - O_{pk})^2}{2} \quad (6-3)$$

خروجی واقعی از لایه خروجی k -ام که توسط روابط ورودی خروجی شبکه عصبی (۳-۱) الی (۳-۵) محاسبه می‌گردد.

در الگوریتم $BP [n]$ اوزان w_{ji} و w_{kj} و بایاس b_j و b_k بروزرسانی شده و تابع هزینه e_p در (۳-۶) کاهش می‌یابد. به طور مثال: وزن w_{ji} بر اساس روابط و قوانین ذیل تغییر می‌یابد:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \quad (7-3)$$

$$\Delta w_{ji}(t) = -\eta \frac{\partial e_p}{\partial w_{ji}} \quad (3-8)$$

η یک عدد مثبت که به عنوان نرخ یادگیری از آن یاد می‌کنیم در این مقاله مقدار آن را $\eta = 0.1$ در نظر می‌گیریم.

t اندیس، عدد اصلاحی می‌باشد و مقدار اصلاحی $\Delta w_{ji}(t)$ به صورت زیر تعریف خواهد شد:

$$(9-3) \Delta w_{ji}(t) = -\eta \frac{\partial e_p}{\partial w_{ji}} + \alpha \Delta w_{ji}(t-1)$$

و همین ترتیب w_{kj} و بایاس‌ها b_k و θ_j توسط روابط (۳-۷) الی (۳-۹) اصلاح می‌شوند.

۴-۱ شبکه عصبی پیش‌رو فازی:

ورودی‌ها و اوزان و بایاس‌های شبکه عصبی پیش‌رو استاندارد که بوسیله روابط ۱ الی ۵ تعریف شده را به اعداد فازی توسعه می‌دهیم، لازم به ذکر است در فازی سازی شبکه عصبی تغییری در ساختار و معماری شبکه عصبی استاندارد اتفاق نخواهد افتاد.

اعداد فازی را با نماد ($\sim$) از اعداد حقیقی متمایز می‌کنیم. رابطه ورودی و خروجی شبکه عصبی فازی شده را به صورت زیر بیان خواهیم کرد:

$$X_p = (X_{p1}, \dots, X_{pn_1})$$

X_p بردار ورودی فازی است.

در ادامه نیز داریم:

لایه ورودی فازی :

$$O_{pi} = x_{pi} \quad i = 1, 2, \dots, n_1 \quad (3-10)$$

لایه میانی فازی (پنهانی):

$$O_{pj} = \tilde{f}(net_{pj})$$

$$j = 1, 2, \dots, n_H \quad (3-11)$$

$$net_{pj} = \sum_{i=1}^{n_1} w_{ji} \square O_{pi} \oplus b_j \quad j = 1, 2, \dots, n_H \quad (3-12)$$

لایه خروجی :

$$O_{pk} = \tilde{f}(net_{pk}), k = 1, 2, \dots, n_O \quad (3-13)$$

$$net_{pk} = \sum_{j=1}^{n_H} w_{kj} \square O_{pj} \oplus b_k \quad (3-14)$$

توجه داشته باشید که w_{ji} و w_{kj} اوزان فازی و b_j و b_k بایاس فازی هستند.
از طرفی داریم

$$\tilde{f}(x) = 1 \div (1 \oplus \tilde{g}(x))$$

$$\tilde{g}(x) = e^{-\tilde{x}}$$

۲-۴ محاسبات روابط ورودی و خروجی:

روابط فرمول‌های (۳-۱۰) الی (۳-۱۴) بوسیله اصل گسترش زاده بازنویسی شده و اعمال جبری تعریف شده در روابط فازی شده شبکه عصبی مورد استفاده قرار خواهند گرفت.

در فرمول‌ها و روابط (۳-۱۰) الی (۳-۱۴) ضرب و جمع و نگاشت‌های غیرخطی اعداد فازی در شبکه عصبی فازی شده به صورت زیر تعریف می‌شود:

$$\mu_{(\tilde{M} \oplus \tilde{N})}(z) = \sup_{z=x+y} \{ \min[\mu_{\tilde{M}}(x), \mu_{\tilde{N}}(y)] \}$$

$$\mu_{(\tilde{M} \square \tilde{N})}(z) = \sup_{z=x*y} \{ \min[\mu_{\tilde{M}}(x), \mu_{\tilde{N}}(y)] \}$$

$$\mu_{f(Net)}(z) = \max_{z=f(x)} \{ \mu_{f(Net)}(x) \}$$

$\tilde{M}$ و $\tilde{N}$ هر دو عدد فازی هستند و μ تابع عضویت مربوط به هر کدام از اعداد فازی می‌باشد.

در ادامه برای فازی‌سازی شبکه عصبی با الگوریتم پسا انتشار که هم قابلیت محاسبه و کار با اعداد واقعی و هم کار با اعداد فازی را دارد. به عبارتی در شبکه عصبی علاوه بر با اعداد حقیقی با مجموعه اعداد فازی مثلی که نوعی از اعداد فازی هم هست قابلیت کار دارد.

گسترش و توسعه شبکه عصبی به شبکه عصبی فازی در دو جهت اتفاق می‌افتد:

الف - فازی‌سازی شبکه

ب - فازی‌سازی الگوریتم یادگیری

الف) فازی سازی شبکه:

در شبکه عصبی با دو شیء مختلف به نام‌های نورون و یال مواجه هستیم که هم باید وزن‌های متصل به آنها فازی و هم نورون‌ها هم فازی باشد.

پس لاجرم در اینجا هم با دو قسمت بخش مواجه خواهیم شد.

(۱) یال‌ها: اوزان یال‌ها دیگر اعداد ممیز نیستند، اعداد فازی هستند ورودی‌ها از چپ و خروجی‌ها از راست همگی اعداد فازی هستند.

(۲) نورون‌ها: فازی سازی قسمت نورون‌ها با چند روش بدیهی قابل گسترش می‌باشد.

تابع خروجی در یک نورون حقیق تابع سیگموئید در نظر گرفته شده است بنابراین در روند فازی سازی این تابع نیز باید فازی شود و برای این کار با استفاده از اصل گسترش زاده آن را تخمین خواهیم زد

در ادامه با فرض اینکه $a = (m_a, \lambda_a, p_a)$ یک عدد فازی مثلثی باشد خواهیم داشت.

$$\tilde{S}(a) = (s(m_a), s(\lambda_a), s(p_a))$$

در محاسبه الگوریتم BP نیاز به مشتق تابع سیگموئید هم هست که داریم:

$$\tilde{S}'(a) = (\tilde{s}'(m_a), \tilde{s}'(\lambda_a), \tilde{s}'(p_a))$$

در ادامه جهت یادآوری :

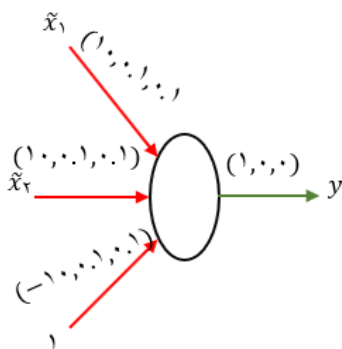
$$\lambda_a = m_a - l_a$$

λ_a پنهای چپ است.

$$p_a = m_a - r_a$$

p_a پنهای راست است.

۵- در این بخش یک مثال برای شبکه عصبی فازی ارائه می‌شود:



$$y = \tilde{x}_1 \square \tilde{\omega}_1 \oplus \tilde{x}_2 \square \tilde{\omega}_2 \oplus 1 \square \tilde{\omega}_0$$

فرض کنید مطابق شکل روبرو یک نرون با ۳ ورودی و یک یال خروجی که یکی از ورودی‌ها همان بایاس با ارزش مقدار ۱ می‌باشد و مابقی پارامترهای ورودی با مقدار فازی است.

$$\tilde{x}_1 = (0.1, 0.001, 0.001) \rightarrow$$

$$\tilde{x}_1 = \begin{cases} 1 - \frac{0.1 - x}{0.001} & 0.099 \leq x \leq 0.1 \\ 1 - \frac{x - 0.1}{0.001} & 0.1 \leq x \leq 0.101 \\ 0 & \text{etc} \end{cases}.$$

$$\tilde{x}_2 = (0.1, 0.001, 0.001) \rightarrow$$

$$\tilde{x}_2 = \begin{cases} 1 - \frac{0.1 - x}{0.001} & 0.099 \leq x \leq 0.1 \\ 1 - \frac{x - 0.1}{0.001} & 0.1 \leq x \leq 0.101 \\ 0 & \text{etc} \end{cases}$$

$$\tilde{w}_1 = \begin{cases} 1 - \frac{10 - x}{1} & 9.9 \leq x \leq 10 \\ 1 - \frac{x - 10}{1} & 10 \leq x \leq 10.1 \\ 0 & \text{etc} \end{cases}$$

$$\tilde{w}_2 = (10, 0.1, 0.1) \rightarrow$$

$$\tilde{w}_2 = \begin{cases} 1 - \frac{10 - x}{1} & 9.9 \leq x \leq 10 \\ 1 - \frac{x - 10}{1} & 10 \leq x \leq 10.1 \\ 0 & \text{etc} \end{cases}$$

$$\tilde{w}_0 = (-10, 0.1, 0.1) \rightarrow$$

$$\tilde{w}_0 = \begin{cases} 1 - \frac{(-10) - x}{/1} & -10.1 \leq x \leq -10 \\ 1 - \frac{x + 10}{/1} & -10 \leq x \leq -9.9 \\ 0 & etc \end{cases}$$

بنا به تعریف اصل گسترش برای ضرب و جمع داریم:

$$\tilde{x}_1 \oplus \tilde{x}_2 = \sup_{\tilde{x}_1 + \tilde{x}_2} \min \{M(\tilde{x}_1), M(\tilde{x}_2)\}$$

$$\tilde{x}_1 \square \tilde{x}_2 = \sup_{\tilde{x}_1 \times \tilde{x}_2} \min \{M(\tilde{x}_1), M(\tilde{x}_2)\}$$

با توجه به اینکه مقدار دو عدد فازی $\tilde{x}_1$ و $\tilde{w}_1$ و $\tilde{x}_2$ و $\tilde{w}_2$ با هم برابر است حاصلضرب $\tilde{x}_2 * \tilde{w}_2$ و مقدار تابع عضویت آن با جدول ذیل یکی خواهد شد.

و مقدار حاصلضرب $1 * \tilde{w}_0$ (بایاس) برابر خواهد بود با جدول (۱):

در این جدول به صورت نمونه مقادیر $(\tilde{x}_1, \mu(\tilde{x}_1))$ و $(\tilde{x}_1 * \tilde{w}_1, \mu(\tilde{x}_1 * \tilde{w}_1))$ محاسبه و اشکال مربوطه ارائه شده است.

در این بخش، ما یک الگوریتم یادگیری کلی از شبکه عصبی فازی را استخراج کرده، الگوریتم یادگیری بدست آمده را می‌توان برای تنظیم اوزان فازی در حالت‌های مختلف استفاده کرد.

فرض کنید بردار ورودی فرض کنید که زوج ورودی-خروجی (X_p, T_p) برای یادگیری شبکه عصبی فازی داده شده است.

$$X_p = (x_{p1}, \dots, x_{pn_1})$$

$$T_p = (t_{p1}, \dots, t_{pn_1})$$

هدف از یادگیری توسط شبکه عصبی فازی کاهش اختلاف بین T_p و X_p است. یعنی مطلوب است که برابری زیر تقریباً برقرار باشد:

$$T_{pk} \cong k O_{pk}, \quad k = 1, 2, \dots, n_O$$

در ادامه تابع هزینه که طی آموزش توسط شبکه عصبی فازی حداقل می‌رسد شبکه عصبی فازی باید اختلاف بین بردار هدف فازی T_{pk} و بردار خروجی فازی O_{pk} را اندازه‌گیری کند. ابتدا یک تابع هزینه e_{pk} برای مجموعه‌های O_{pk} و T_{pk} به شرح زیر تعریف می‌کنیم:

$$e_{pk} = T_{pk} \ominus O_{pk}$$

۶- نتیجه‌گیری :

رویه ارائه شده در این مقاله دارای چندین مزیت نسبت به حالت استاندارد الگوریتم پسا و جلو انتشار دارد :

(الف) شبکه عصبی علاوه بر اعداد غیر فازی توانایی کار با اعداد فازی هم خواهد داشت.

(ب) سرعت آموزش در این حالت سریع‌تر خواهد بود.

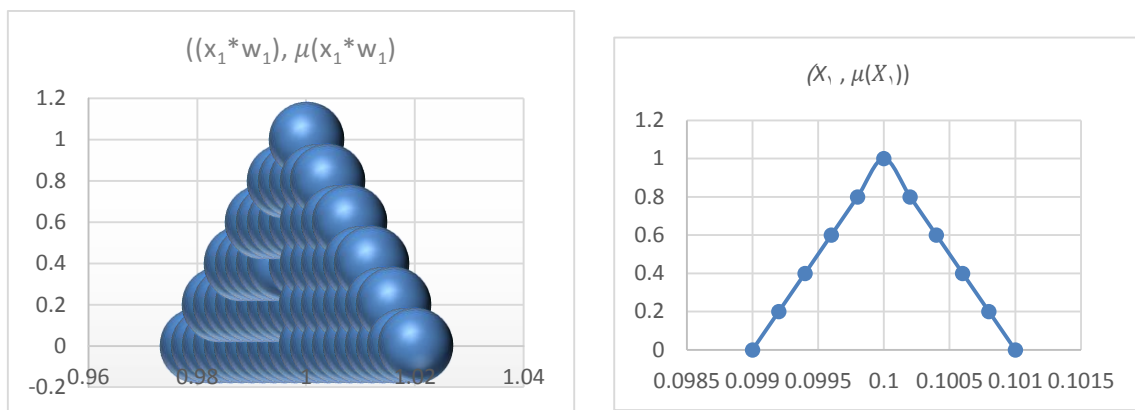
(ج) از طرفی چون ذاتاً شبکه‌ها برای مسائل طبیعی استفاده می‌شود پس می‌توان گفت که کاربرد فازی خود یک مزیتی فزونی نسبت به حالت کلاسیک است.

د) در این مقاله معماری شبکه عصبی فازی معرفی شده توانایی محاسبات اعداد فازی مثلثی و دوزنقه‌ای چه متقارن و چه غیرمتقارن را خواهد داشت.

در آخر در این مقاله رویه نوینی به نام گسسته‌سازی فازی ارائه شده که به نسبت اعداد فازی تابع عضویت مربوطه را محاسبه که بسته به توان سیستم رایانه‌ای دقت و سرعت بیشتری در محاسبه ارائه می‌دهد که این خود نهایتاً منجر به بهبود عملکرد و تصمیم‌گیری آموزش شبکه عصبی با اصلاح اوزان در شبکه عصبی خواهد شد.

جدول ۱ (محاسبه مقادیر $(\tilde{X}_1, \mu(\tilde{X}_1))$ و $(\tilde{X}_1 * \tilde{W}_1, \mu(\tilde{X}_1 * \tilde{W}_1))$)

ردیف	X_1	$\mu(X_1)$	ردیف	$X_1 * W_1$	$\mu(X_1 * W_1)$	ردیف	$X_1 * W_1$	$\mu(X_1 * W_1)$
1	0.099	0	1	0.9801	0	31	1.00596	0
2	0.0992	0.2	2	0.98208	0	32	0.996004	0.8
3	0.0994	0.4	3	0.98406	0	33	0.998	0.8
4	0.0996	0.6	4	0.98604	0	34	0.999996	0.8
5	0.0998	0.8	5	0.98802	0	35	1.001992	0.6
6	0.1	1	6	0.99	0	36	1.003988	0.4
7	0.1002	0.8	7	0.99198	0	37	1.005984	0.2
8	0.1004	0.6	8	0.99396	0	38	1.00798	0
9	0.1006	0.4	9	0.99594	0	39	1	1
10	0.1008	0.2	10	0.99792	0	40	1.002	0.8
11	0.101	0	11	0.9999	0	41	1.004	0.6
			12	0.984064	0.2	42	1.006	0.4
			13	0.986048	0.2	43	1.008	0.2
			14	0.988032	0.2	44	1.01	0
			15	0.990016	0.2	45	1.004004	0.8
			16	0.992	0.2	46	1.006008	0.6
			17	0.993984	0.2	47	1.008012	0.4
			18	0.995968	0.2	48	1.010016	0.2
			19	0.997952	0.2	49	1.01202	0
			20	0.999936	0.2	50	1.008016	0.6
			21	1.00192	0	51	1.010024	0.4
			22	0.988036	0.4	52	1.012032	0.2
			23	0.990024	0.4	53	1.01404	0
			24	0.992012	0.4	54	1.012036	0.4
			25	0.994	0.4	55	1.014048	0.2
			26	0.995988	0.4	56	1.01606	0
			27	0.997976	0.6	57	1.016064	0.2
			28	0.999964	0.4	58	1.01808	0
			29	1.001952	0.2	59	1.0201	0
			30	1.00394	0			



شکل ۶ اشکال مربوط به جدول شماره ۱

مراجع:

- [1] Simon Haykin]McMaster University Hamilton, Ontario, Canada Neural Networks and Learning Machines 2009.
- [2] [Honkela], T., 2007. "Philosphical aspects of neural, probabilistic and fuzzy modeling of language use and translation," Proceedings of IJCNN, International Joint Conference on Neural Networks, pp. 2881–2886, Orlando, FL, August.
- [3] [Hush], D., P. Kelly, C. Scovel, and I. Steinwart, 2006. "QP algorithms with guaranteed accuracy and run time for support vector machines," J. Machine Learning Research, vol. 7, pp. 733–769.
- [4] [Kruse] Kruse/Gebhardt/Klawonn, Fuzzy-Systeme, Teubner Stuttgart 1993[Hayashi] Y. Hayashi, J. Buckley, E. Czogola, Direct Fuzzification of Neural Networks and Fuzzified Delta Rule, Iizuka, Japan 1992, Proc. 2nd Intern. Conf. on Fuzzy Logic and Neural Networks, pp. 73-76.
- [5] [Lin] C.-T. Lin, C. Lee, Neural-Network-Based Fuzzy Logic Control and Decision System, IEEE Transactions on Computers, Vol. 40, Nr. 12, S 1320- 1336, 1991
- [6] [Okada] Hiroyuki Okada et. al., Knowledge Implementation Multilayer Neural Networks with Fuzzy Logic, Proceedings of the International Conference on Fuzzy Logic & Neural Networks, Iizuka, Japan 1990, pp. 99-102
- [7] [Brause] Ruediger Brause, Neuronale Netze, Teubner Stuttgart 1991
- [8] [Ro jas] Raul Ro jas, Theorie der neuronalen Netze, Springer-Verlag Berlin Heidelberg New York 1993.
- [9] Krishnamraju, P. V., Buckley, J. J., Rcilly, K. D., and Hayashi, Y., Genetic learning algorithms for fuzzy neural nets, Proceedings of FUZZ-IEEE'94, Orlando, 1969-1974, 1994.
- [10] Zadeh, L. A., The concept of a linguistic variable and its application to approximate reasoning. I, II, III, Inform. Sci. 8, 199-249, 301-357: 9, 43 80, 1975.
- [11] Web of Neural Network And Deep Learning
- [12] Neural Networks and Learning Machines , Simon Haykin ,2008
- [13] رساله دکتری رضا کارگر استاد راهنما فرهاد حسین زاده و... دانشگاه آزاد اسلامی واحد علوم تحقیقات دانشکده علوم پایه زمستان ۹۲
- [14] محمد جواد اصغر پور ، تصمیم گیری های چند معیاره انتشارات دانشگاه تهران چاپ ششم ۱۳۸۷ .