



سیستم‌های ترکیبی فازی و عصبی

محمدعلی فائزی راد

پیش‌گفتار

شبکه‌های عصبی مصنوعی و سیستم‌های فازی از جمله مباحث پرکاربرد در هوش محاسباتی محسوب می‌شوند که با توجه به ماهیت میان‌رشته‌ای آنها، در علوم مختلف به کار می‌روند. پیرامون این دو حوزه، کتاب‌های متعددی در کشورمان تألیف و ترجمه شده و مجموعه‌های مفیدی برای یادگیری آنها در دسترس است. با این حال، فقدان یک مجموعه منسجم که تمرکز آن بر ترکیب مدل‌های فازی و عصبی باشد، احساس می‌شود. این در حالی است که سیستم‌های ترکیبی عصبی و فازی طی دو دهه اخیر مورد توجه پژوهشگران واقع شده و منابع گسترده‌ای درباره آنها منتشر شده است. اگرچه کتاب‌های محدودی به زبان فارسی وجود دارند که در یک یا چند فصل به گوشه‌ای از این مدل‌های ترکیبی اشاره کرده‌اند، اما جامعیت لازم و پوشش انواع مدل‌ها در کنار یکدیگر یافت نمی‌شود.

کتاب حاضر، مجموعه‌ای موجز پیرامون مدل‌های حاصل از ترکیب سیستم‌های فازی و شبکه‌های عصبی مصنوعی است که مبتنی بر ترجمه از منابع اصلی و گردآوری مباحث موضوعی از مراجع و پژوهش‌های مختلف شکل گرفته است. کتاب‌های مرجعی که بخشی از تمرکز خود را بر ترکیب سیستم‌های فازی و عصبی معطوف داشته‌اند، محدودیت‌هایی نظیر فقدان جامعیت، تشریح ناکامل ترکیب‌های ممکن و عدم اشاره کامل به کاربردهای واقعی یا مقالات پژوهشی دارند. به همین جهت در این کتاب تلاش شده است تا با استفاده از سایر منابع موجود از جمله مقالات معتبر به تکمیل مباحث پرداخته شده و به ترجمه یک منبع تنها اکتفا نشود.

فصل نخست این کتاب به ارائه مقدمه‌ای پیرامون ماهیت ترکیب مدل‌های فازی و عصبی می‌پردازد و انواع ترکیب‌های موجود را معرفی می‌کند. در فصل‌های دوم تا پنجم، ترکیب‌های رایج تشریح می‌شود

که بر پایه نوع ترکیب به فصل‌های مختلف تقسیم شده‌اند. در هر فصل، موردکاوی‌ها و مثال‌هایی نیز در خلال معرفی برخی مدل‌های متداول ارائه می‌شود. برخی مباحث تکمیلی ناظر بر تازه‌ترین نوآوری‌های پژوهشی طی سال‌های اخیر نیز در فصل ششم کتاب مورد بررسی قرار می‌گیرد تا نیم‌نگاهی به آینده سیستم‌های ترکیبی فازی و عصبی نیز به مخاطب ارائه شود. پیش‌نیاز این کتاب آشنایی با سیستم‌های فازی و شبکه‌های عصبی است، اما با توجه به منابع گسترده فارسی برای آشنایی با این دو حوزه، مباحث مقدماتی و آشنایی اولیه با آنها در این کتاب ارائه نشد تا اختصار در ارائه مطالب کتاب نیز حفظ شود.

یک مشکل جدی که در نگارش این کتاب وجود داشت، فقدان معادل فارسی برای برخی از واژه‌های تخصصی یا مواجهه با معادل‌های چندگانه و بعضاً نادرست در منابع مختلف بود. به‌منظور حل این مشکل، طی مطالعه جامعی که با استفاده از پایان‌نامه‌ها، مقالات و کتب فارسی صورت گرفت و همچنین مشورت‌هایی که در برخی از موارد با اساتید صاحب‌نظر در این حوزه انجام پذیرفت، تلاش کردیم که تا حد ممکن و معقول اصطلاحات و عبارات مطرح‌شده به فارسی روان ترجمه گردد. لذا بدون اینکه آسیبی به شیوایی و روان بودن متن وارد شود، اهتمامی بر فارسی‌سازی صورت گرفته است.

از آنجا که این کتاب یکی از نخستین تلاش‌ها برای گردآوری مجموعه‌ای به زبان فارسی پیرامون ترکیب منطق فازی و شبکه‌های عصبی مصنوعی است، یقیناً از داشتن ایراد و خطا مبرا نخواهد بود. به‌همین جهت نظرات و پیشنهادهای همه اساتید، دانشجویان و پژوهشگران گرامی درباره این اثر موجب امتنان خواهد شد.

محمدعلی فائزی راد

۱۴۰۲

فهرست مطالب

۱	مقدمه	۱
۶	۱-۱ ترکیب سیستم‌های عصبی و فازی	۶
۱۱	۲ سیستم‌های مشارکتی	۱۱
۱۲	۱-۲ سیستم‌های عصبی فازی مشارکتی	۱۲
۱۲	۱-۱-۲ ترکیب آبخاری	۱۲
۱۳	۲-۱-۲ نوع ابزار توسعه	۱۳
۱۸	۲-۲ سیستم‌های فازی عصبی مشارکتی	۱۸
۱۸	۱-۲-۲ شبکه عصبی به‌عنوان سازوکار اصلاحی	۱۸
۱۹	۲-۲-۲ شبکه عصبی در تعیین توابع عضویت (نوع ابزار توسعه)	۱۹
۲۶	۳-۲-۲ شبکه عصبی برای یادگیری قواعد فازی	۲۶
۴۱	۴-۲-۲ شبکه عصبی برای انتخاب ویژگی	۴۱
۴۲	۵-۲-۲ ترکیب آبخاری	۴۲
۴۲	۶-۲-۲ شبکه عصبی برای تعیین پارامتر	۴۲
۴۵	۳ سیستم‌های همزمان	۴۵

۴۹	سیستم‌های یکپارچه فازی عصبی
۵۰	۱-۴ شبکه‌های عصبی فازی با سیستم استنتاج فازی
۵۰	۱-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع ممدانی
۵۲	۲-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع تاکاگی-سوگنو
۵۵	۳-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع تسوکاموتو
۵۹	۲-۴ سیستم فازی مبتنی بر شبکه عصبی (شبکه Pi-Sigma)
۶۵	۳-۴ کنترل هوشمند مبتنی بر استدلال تقریبی
۶۵	۱-۳-۴ مدل ARIC
۷۰	۲-۳-۴ مدل GARIC
۷۸	۴-۴ شبکه‌های تابع پایه فازی (FBFN)
۸۲	۵-۴ کنترل‌کننده فازی عصبی (NEFCON)
۸۵	۱-۵-۴ طبقه‌بندی فازی عصبی (NEFCLASS)
۸۹	۲-۵-۴ تقریب تابع با رویکرد فازی عصبی (NEFPROX)
۹۰	۶-۴ سیستم فازی عصبی انطباقی
۹۱	۱-۶-۴ سیستم استنتاج فازی عصبی انطباقی (ANFIS)
۱۰۹	۲-۶-۴ سیستم استنتاج فازی عصبی هم‌کنش (CANFIS)
۱۱۳	۷-۴ سیستم‌های استنتاج فازی عصبی خودسازمانده
۱۱۳	۱-۷-۴ شبکه کنترل یادگیری انطباقی فازی (FALCON)
۱۱۹	۲-۷-۴ شبکه استنتاج فازی عصبی خودساخت (SONFIN)
۱۲۴	۳-۷-۴ سیستم استنتاج فازی عصبی تکاملی پویا (DENFIS)
۱۲۹	۸-۴ سایر سیستم‌ها
۱۲۹	۱-۸-۴ شبکه فازی (FUN)
۱۳۱	۲-۸-۴ نرم‌افزار استنتاج فازی با تنظیم‌گری (FINEST)

۹-۴ بحث و بررسی ۱۳۳

۵ شبکه‌های عصبی مبتنی بر منطق فازی

۱-۵ نورون‌های فازی ۱۴۲

۱-۱-۵ نورون فازی نوع یک ۱۴۳

۲-۱-۵ نورون فازی نوع دو ۱۴۸

۳-۱-۵ نورون فازی نوع سه ۱۵۰

۲-۵ فازی‌سازی مدل‌های شبکه عصبی ۱۵۳

۱-۲-۵ پرسپترون فازی ۱۵۳

۲-۲-۵ ART و ARTMAP فازی ۱۶۰

۳-۲-۵ شبکه خوشه‌بندی کوهونن فازی (FKCN) ۱۷۲

۳-۵ شبکه‌های عصبی با آموزش فازی ۱۷۶

۱-۳-۵ شبکه‌های عصبی با ورودی آموزشی فازی ۱۷۷

۲-۳-۵ شبکه‌های عصبی با پارامترهای فازی ۱۸۶

۳-۳-۵ شبکه پساتشار فازی (Fuzzy BP) ۱۹۴

۴-۵ شبکه‌های عصبی فازی تکاملی ۲۰۴

۶ مدل‌های نوین

۱-۶ ترکیب مجموعه‌های فازی نوع ۲ با شبکه‌های عصبی ۲۰۹

۲-۶ سیستم‌های عصبی فازی عمیق ۲۱۳

۲۱۷

منابع

فصل ۱

مقدمه

منطق فازی و شبکه‌های عصبی مصنوعی ظاهراً دو حوزه مجزا از هوش محاسباتی هستند که طی دهه‌های اخیر، بسیار مورد توجه واقع شده‌اند. سیستم‌های مبتنی بر منطق فازی تلاش می‌کنند که با استفاده از عبارات زبانی^۱ از استدلال شبه‌انسانی پیروی کنند درحالی‌که شبکه‌های عصبی کوشش می‌کنند تا از یادگیری مشابه مغز انسان تقلید کرده و اطلاعات را بر پایه‌ای کاملاً تجربی ذخیره سازند. روش‌های موجود در هر یک از این حوزه‌ها در بسیاری از فرآیندهای پیچیده و صنعتی با موفقیت پیاده‌سازی شده‌اند؛ با این حال هر دو کمبودهایی را تجربه کرده‌اند. شبکه‌های عصبی در تشخیص الگوها خوب عمل می‌کنند اما نمی‌توانند بیان کنند که چطور تصمیم‌گیری می‌کنند. از سوی دیگر، سیستم‌های فازی با اطلاعات محدود استنتاج کرده و به خوبی بیان می‌کنند که چطور به این تصمیم رسیده‌اند اما نمی‌توانند قواعد موردنیاز را به طور خودکار به دست‌بیاورند. سیستمی که بتواند با عدم قطعیت‌های ذهنی (مثل انسان) تعامل کند و منطق فازی را در شبکه‌های عصبی بگنجانند به یک سیستم ترکیبی منجر خواهد شد که شبکه فازی عصبی یا عصبی فازی نامیده می‌شود.

یکی از مهم‌ترین ابعاد در طراحی سیستم‌های فازی، ایجاد توابع عضویت (MF)^۲ و ساخت پایگاه

1. Linguistic

2. Membership functions

قواعد^۱ است که به فرآیندی فرسایشی تبدیل شده است. اگرچه عملاً انتخاب توابع عضویت نقش حساسی را در موفقیت بازی می‌کند اما روش خودکاری برای این کار وجود ندارد. این انتخاب اغلب به صورت سعی و خطا بر اساس نظرات خبرگان صورت می‌پذیرد. ایجاد قوانین فازی در نقش گلوگاه هر نوع سیستم دارای پایگاه قواعد است. در بیشتر کاربردهای موجود، قوانین فازی توسط یک خبره ایجاد می‌شود که این مورد فقط به سیستم‌هایی با تعداد ورودی کم اختصاص دارد. با افزایش ورودی‌ها، خروجی‌ها و متغیرهای زبانی، تعداد قوانین ممکن سیستم به صورت نمایی افزایش می‌یابد که تعریف مجموعه کاملی از قوانین و توابع عضویت منطبق با عملکرد منطقی سیستم را برای خبرگان دشوار می‌سازد. ساخت توابع عضویت، ایجاد پایگاه قواعد و تنظیم پارامترهای مقیاس‌گذاری با استفاده از الگوریتم‌های فرگشتی (تکاملی)^۲ در کتاب‌ها و مقالات متعددی مورد اشاره واقع شده است. زمانی که هیچ اطلاعات پیشینی درباره توابع عضویت و پایگاه قواعد در دسترس نیست، استفاده از الگوریتم‌های فرگشتی مناسب است. کاربردهای موفقیت‌آمیز متعددی از سیستم‌های فازی فرگشتی در ادبیات این موضوع گزارش شده و در مقالات متعددی مرور این روش‌ها و کاربردهای آنها یافت می‌شود. الگوریتم‌های فرگشتی فرآیند آهسته‌ای دارند و عملکرد آنها به طور ذاتی به اندازه جمعیت و تعداد نسل‌های مورد نیاز برای استوار شدن مسأله بستگی دارد. این موضوع ممکن است برای برخی کاربران خوشایند نباشد. زمانی که دانش تخصصی برای ایجاد توابع عضویت وجود نداشته باشد، طراح مدل باید از داده‌های محیطی که ممکن است موجود باشند یا نباشند، به ساختن پردازنده موضوع دیگری که در سیستم‌های فازی وجود دارد، بحث پردازش پایگاه قواعد با تعداد $R = n_1 \times n_2 \times \dots \times n_N$ قاعده است که در آن n_i ($i = 1, 2, \dots, N$) تعداد توابع عضویت (مجموعه‌های فازی اولیه) را برای هر یک از N ورودی نشان می‌دهد. پردازش چنین پایگاه قواعد بزرگی زمان‌بر است؛ در نتیجه استفاده از روش‌های شناخته‌شده فازی‌زدایی^۳ (نظیر روش مرکز ثقل) در محاسبه خروجی‌ها، زمان قابل توجهی خواهد گرفت و حتی در برخی موارد ممکن است پاسخ سیستم را خراب کند (سیدیک^۴، ۲۰۰۲). مشکل این روش‌های فازی‌زدایی با استفاده از سیستم‌های فازی از نوع

1. Rule-Base

2. Evolutionary algorithms

3. Defuzzification

4. Siddique, N.

سوغنو^۱ یا تسوکاموتو^۲ رفع شده است. در این سیستم‌ها هر تابع عضویت تالی با تابع چندجمله‌ای یا یکنوا با دو تا سه پارامتر (سیستم‌های غیرخطی مرتبه بالاتر به پارامترهای بیشتری نیاز دارند) جایگزین می‌شود که موجب تخمین مجموعه بیشتری از پارامترهای تالی می‌شود. این مرحله از توسعه سیستم‌های فازی به دنبال ایجاد تسهیل در ساخت توابع عضویت و پایگاه قواعد است؛ همچنین تلاش می‌کند که با استفاده از یک روش ساده فازی‌زدایی و پردازش قواعد، زمان پردازش تولید خروجی‌ها را حداقل کند. بنابراین به یک روش خودکار برای طراحی و تنظیم سیستم‌های فازی نیاز است. یکی از اهداف این کتاب آن است که با بهره‌گیری از یک رویکرد مناسب در شبکه‌های عصبی و استفاده از اطلاعاتی نظیر داده‌های تجربی، داده‌های عددی جزئی و عبارات یا داده‌های زبانی جزئی، بتوانیم ایجاد و یادگیری یا آموزش توابع عضویت فازی و پایگاه قواعد را به همراه تنظیم پارامترها ممکن سازیم.

اغلب کاربردهای موفق شبکه‌های عصبی در بازشناسی الگو، پردازش سیگنال، مدل‌سازی و کنترل سیستم‌های پیچیده جاری می‌شود که به سبب ساختارهای انعطاف‌پذیر و وجود الگوریتم‌های یادگیری در آنها است. روند این یادگیری دانش‌محور نیست بلکه داده‌محور است. اگر مجموعه‌ای از داده‌های تجربی موجود باشد، می‌توان سیستم عصبی را با دقت کافی به منظور ایجاد نگاشتی بین ورودی‌ها و خروجی‌ها آموزش داد. نگاشت ورودی/خروجی در قالب یک ساختار عصبی لایه‌ای ساختاردهی می‌شود که به آن شبکه عصبی می‌گویند. تنوع زیادی در ساختارهای شبکه‌های عصبی و الگوریتم‌های یادگیری وجود دارد که متناسب کاربردهای گوناگون آن‌لاین یا آفلاین است. الگوریتم‌های یادگیری و ساختارهای مختلفی در کتاب‌های آموزش شبکه‌های عصبی مصنوعی تشریح شده است. روش‌های یادگیری و ساختارهای مختلف به داده‌های تجربی متفاوتی نیز نیاز دارند. به عنوان مثال، یک الگوریتم یادگیری با نظارت در شبکه‌های عصبی پیش‌خور^۳ به مجموعه زوج داده‌ها شامل ورودی‌ها و هدف‌ها (خروجی‌های مطلوب) نیاز دارد؛ در حالی که الگوریتم‌های آموزش بدون نظارت فقط به داده‌های ورودی احتیاج دارد. یک شبکه عصبی معمولاً نیازمند زمان زیادی برای یادگیری است که به ساختار آن بستگی دارد، اما باید در نظر داشت که هر فراخوانی فقط شامل یک عبور یکتا است. شبکه‌های عصبی‌ای نظیر شبکه عصبی احتمالی

1. Sugeno
2. Tsukamoto
3. Feedforward

(PNN)^۱، شبکه عصبی تابع پایه شعاعی (RBFNN)^۲ و شبکه عصبی رگرسیون تعمیم‌یافته (GRNN)^۳ دارای زمان آموزش کمتری هستند (معمولاً به یک عبور یکتا نیاز دارند) و برای کاربردهای آنلاین مناسب‌ترند.

قاعده ساده و روشنی برای تعیین اندازه شبکه وجود ندارد. تعداد نورون‌ها در لایه ورودی و خروجی بر اساس مسأله و داده‌ها مشخص می‌شود. به‌طور کلی، اندازه شبکه بر اساس لایه‌های پنهان^۴ و تعداد نورون‌ها در هر لایه گزارش می‌شود. برخی از ساختارهای شبکه‌های عصبی نظیر هاپفیلد^۵، ART-1، SOM و LVQ دارای تعداد لایه از پیش تعیین‌شده هستند و شبکه‌های پس‌انتشار کلاسیک نیز سه لایه دارند. اندازه دقیق لایه پنهان در شبکه عصبی سه لایه، معمولاً یک پارامتر بحرانی نبوده و زمان آموزش نیز در لایه‌های پنهان با اندازه مشابه چندان متفاوت نیست. افزایش اندازه لایه پنهان آشکارسازهای ویژگی^۶ را ایجاد می‌کند. اگر تعداد نورون‌ها در لایه پنهان خیلی کم باشد، نگاهت درست ویژگی‌ها با شکست مواجه خواهد شد و اگر تعداد نورون‌ها در لایه پنهان خیلی زیاد باشد، تعمیم‌پذیری را دچار مشکل می‌کند. با استفاده از تعداد بیشتر لایه‌های پنهان که موجب افزایش قدرت پردازش شبکه عصبی می‌شوند، می‌توان غیرخطی بودن یک سیستم را بیان کرد. شبکه‌های عصبی بزرگ، معمولاً به حجم زیاد نمونه‌های آموزشی، زمان آموزش طولانی‌تر و زمان محاسبات بیشتر نیازمند هستند. بسیار اهمیت دارد که نمونه‌های آموزشی بتوانند نماینده مناسبی برای همه فضای ورودی/خروجی باشند، زیرا در غیر این صورت شبکه عصبی به خوبی نمی‌تواند تعمیم دهد. شبکه‌های عصبی می‌توانند با مجموعه ناقصی از داده‌ها کار کنند اما داده‌هایی که موجود نیستند، می‌توانند مشکل‌ساز شوند. مجموعه‌های داده زائد یا نقاط داده بی‌فایده، موجب تحمیل زمان آموزش و هزینه‌های محاسبات غیرضروری خواهند شد. پس این موضوع اهمیت دارد که چگونه باید داده‌ها را به شبکه ارائه کرد و به زبان مناسب تبدیل نمود. در اغلب موارد، تبدیل داده‌های مشاهده یا اندازه‌گیری شده به قالب دیگری که برای شبکه‌های عصبی معنادار باشد، سودمند است. به‌عنوان مثال، می‌توان دما یا درجه حرارت را در قالب مقادیر واقعی اندازه‌گیری شده ارائه کرد و یا اینکه آن

1. Probabilistic Neural Network
2. Radial-Basis Function Neural Network
3. Generalized Regression Neural Network
4. Hidden layers
5. Hopfield
6. Feature detectors

را به عبارات زبانی نظیر «سرد»، «خنک»، «معتدل»، «گرم» و «داغ» تبدیل نمود. شبکه‌های عصبی به اندازه و بزرگی مقادیر بسیار حساس هستند؛ مثلاً هنگامی که فاصله مقدار حداقل و حداکثر خیلی زیاد باشد. زمانی که دامنه طبیعی مقادیر داده‌ها با دامنه عملیاتی یا مرجع شبکه عصبی تفاوت دارد، نرمال کردن یا مقیاس گذاری مجدد داده‌های پیوسته در یک دامنه متناسب ضروری و مفید خواهد بود.

پژوهش‌های متعددی به منظور ترکیب ویژگی‌های منطق فازی و شبکه‌های عصبی انجام شده است. در ابتدا نیاز است که ویژگی‌های سیستم‌های فازی و شبکه‌های عصبی را تحلیل کنیم تا ویژگی‌هایی را بیابیم که امکان پشتیبانی و همکاری را به هر کدام می‌دهند. به عبارت دیگر، ویژگی‌هایی در هر دو سیستم وجود دارد که می‌توانند در جهت ایجاد یک سیستم چابک ترکیب شوند. سیستم‌های فازی و شبکه‌های عصبی، سیستم‌های ریاضی مستقل از مدل هستند که هر یک مزایا و معایب خویش را دارد. هدف اصلی سیستم‌های فازی عصبی، ترکیب کردن آنها به منظور ایجاد سیستمی است که ویژگی‌های مطلوب آنها را حداکثر کرده و و معایبشان را کاهش دهد. معمولاً پدیده‌های ذهنی (نظیر ادراک و استدلال) خارج از دامنه مرسوم نظریه شبکه‌های عصبی است. سیستم‌های فازی با بهره‌گیری از مجموعه‌های فازی و استدلال فازی، دارای قدرت بالایی در ارائه دانش زبانی و ساختاریافته برای مدل‌سازی عدم قطعیت‌های همراه با شناخت، تفکر و ادراک انسانی هستند. محدودیت سیستم‌های فازی در این است که معمولاً به منظور فراگیری دانش به خبرگان حوزه مورد نظر متکی هستند. از سوی دیگر، شبکه‌های عصبی در ارائه نگاشت‌های غیرخطی در ساختار خویش قدرتمند بوده و معمولاً دانش را با آموزش از طریق داده‌های تجربی حاصل می‌کنند. بنابراین یکپارچه‌سازی سیستم‌های فازی و شبکه‌های عصبی می‌تواند بدون اطلاعات پیشینی قابل توجهی سیستم‌ها را مدل‌سازی کند و به وسیله یادگیری از داده‌های تجربی با عدم قطعیت همراه شود. پارادایم‌هایی که بر اساس این یکپارچه‌سازی بنا شده‌اند، دارای پتانسیل قابل توجهی در سیستم‌های کنترل، سیستم‌های انطباقی^۱ و سیستم‌های خودگردان^۲ هستند. مزایا و مقایسه ویژگی‌های هر دو فناوری (عصبی و فازی) در جدول ۱-۱ برجسته شده است.

1. Adaptive systems

2. Autonomous

شبکه‌های عصبی	سیستم‌های فازی
<ul style="list-style-type: none"> • نیازی به مدل ریاضی نیست • حصول دانش معمولاً بر اساس نمونه‌هاست و این دانش در دل ساختار شبکه است. • الگوریتم‌های یادگیری با نظارت و بدون نظارت در دسترس هستند. • قوانین نمی‌توانند استخراج شوند. • قادر به یادگیری از داده‌های تجربی 	<ul style="list-style-type: none"> • نیازی به مدل ریاضی نیست • حصول دانش از خبرگان حوزه موردنظر است و این دانش به وسیله پایگاه قواعد ارائه می‌شود. • هیچ الگوریتم یادگیری وجود ندارد اما پیاده‌سازی ساده امکان پذیر است. • قوانین باید در دسترس باشند. • قادر به کار بدون اطلاعات پیشینی فراوان

جدول ۱-۱: مقایسه سیستم‌های عصبی و فازی

۱-۱ ترکیب سیستم‌های عصبی و فازی

سیستم فازی عصبی^۱، پارامترهای یک سیستم فازی را با استفاده از روش‌های یادگیری حاصل از شبکه‌های عصبی تعیین می‌کند. مهم‌ترین دلیل ترکیب شبکه‌های عصبی با سیستم‌های فازی، قابلیت یادگیری آنها است. چنین ترکیبی باید بتواند قواعد زبانی و/یا توابع عضویت را بیاموزد یا آنها را تنظیم کند. یادگیری در اینجا به معنای زیر تلقی می‌شود:

۱. ساخت یک پایگاه قواعد

۲. تعدیل و تنظیم توابع عضویت از نقطه ابتدا

۳. تعیین و محاسبه سایر پارامترهای سیستم

از سوی دیگر، پیوند بین این دو فناوری می‌تواند به صورت سیستم عصبی فازی^۲ با هدف پشتیبانی از یک شبکه عصبی باشد. سیستم عصبی فازی، یک شبکه عصبی است که از رویکرد فازی برای افزایش قابلیت یادگیری و بهبود عملکرد خود استفاده می‌کند. یک سیستم عصبی فازی با هدف حداقل کردن تلاش‌ها

1. Neuro-Fuzzy System

2. Fuzzy-Neural System

برای آموزش و کاهش هزینه محاسبات، به کنترل ورودی‌های شبکه عصبی، پارامترهای یادگیری و پارامترهای ساختاری مختلف شبکه عصبی می‌پردازد (هو^۱ و هرتز^۲، ۱۹۹۴؛ ایشیبوچی^۳ و همکاران، ۱۹۹۳، ۱۹۹۵). کنترل در اینجا به معنای گام‌های کنترل سرعت یادگیری، مقیاس‌گذاری داده‌های ورودی و اصلاح و تعدیل سایر پارامترهای ساختاری است.

اگرچه ماهیت سیستم‌های فازی عصبی با سیستم‌های عصبی فازی تفاوت دارد، عبارت «سیستم عصبی فازی» در ادبیات مرتبط با این موضوع رایج نیست. بنابراین در بسیاری از موارد، وقتی از «سیستم‌های فازی عصبی» یاد می‌شود، منظور هر دو مدل سیستم‌های فازی عصبی و عصبی فازی است. در این کتاب نیز ممکن است در شرایطی که تأکیدی بر محوریت شبکه عصبی یا سیستم فازی وجود ندارد و صرفاً ترکیب آنها اهمیت دارد، از عبارت سیستم‌های فازی عصبی استفاده شده باشد. با این وجود، در موارد متعدد تلاش شده است که این جایگزینی انجام نشود و هریک به صورت مستقل به کار گرفته شود.

با توجه به بررسی ادبیات (سیدیک^۴ و عادل^۵، ۲۰۱۳)، به طور کلی سه نوع ترکیب برجسته بین

شبکه‌های عصبی و سیستم‌های فازی وجود دارد:

- سیستم‌های مشارکتی^۶
- سیستم‌های همزمان^۷
- سیستم‌های هیبریدی^۸

سیستم مشارکتی به عنوان یک پیش‌پردازنده تلقی می‌شود که معمولاً یک سیستم به عنوان محور تلقی شده و سیستم دیگر به بهبود آن کمک می‌کند. در سیستم فازی عصبی سازوکار یادگیری یک شبکه عصبی با استفاده از داده‌های آموزشی در تعیین توابع عضویت سیستم فازی و قواعد فازی به کار می‌رود؛ سپس شبکه عصبی به پس‌زمینه بازگشته و سیستم فازی را رها می‌کند تا به طور مستقل کار کند. از سوی دیگر، در

1. Hu, Q.
2. Hertz, D. B.
3. Ishibuchi, H.
4. Siddique, N.
5. Adeli, H.
6. Cooperative Systems
7. Concurrent Systems
8. Hybrid Systems

سیستم عصبی فازی می‌توان اطلاعات پیش‌نیاز برای شبکه عصبی را توسط سیستم فازی فراهم آورد یا نرخ یادگیری الگوریتم شبکه عصبی را با استفاده از سیستم فازی کنترل کرد. در سیستم‌های همزمان، شبکه عصبی و سیستم فازی به همدیگر کمک می‌کنند تا بتوانند پارامترهای مورد نیاز را به‌طور پیوسته تعیین کنند. در سیستم‌های هیبریدی (یکپارچه یا آمیخته)، سیستم فازی در قالب ساختاری شبیه به شبکه عصبی ظاهر می‌شود به طوری که بتوان یک الگوریتم یادگیری را در سیستم فازی اجرا نمود. از سوی دیگر، در سیستم‌های هیبریدی می‌توانیم شبکه عصبی داشته باشیم که فازی شده است یا از آموزش فازی بهره‌مند می‌شود. جزئیات این سه گونه هم‌افزایی بین شبکه عصبی و سیستم‌های فازی در فصل‌های بعدی تشریح می‌شود.

اگرچه طبقه‌بندی فوق، یک دسته‌بندی متداول در ادبیات مربوط به سیستم‌های ترکیبی فازی و عصبی است، اما رویکردهای دیگری نیز برای دسته‌بندی این سیستم‌ها گزارش شده است. دو رویکرد دیگر در این زمینه، دسته‌بندی براساس یادگیری و نیز براساس ساختار است (شهاب‌الدین^۱ و پیلای^۲، ۲۰۱۸). در طبقه‌بندی براساس یادگیری، سیستم‌های فازی و عصبی براساس شیوه آموزش آنها به پنج گروه تقسیم می‌شوند که شامل گرادیان، یادگیری ترکیبی، یادگیری مبتنی بر جمعیت، یادگیری مبتنی بر ELM^۳ و یادگیری مبتنی بر SVM^۴ می‌شود. از سوی دیگر، اگر ساختار سیستم‌های ترکیبی فازی و عصبی را در نظر بگیریم، می‌توانیم آنها را به دو دسته کلی ایستا^۵ و خودسازمان‌ده^۶ تقسیم کنیم. در ساختار ایستا، با سیستم‌هایی مواجه هستیم که ساختار آنها در خلال یادگیری ثابت باقی می‌ماند؛ به عبارت دیگر، تعداد قواعد فازی، تعداد پارامترهای مقدم و تالی، تعداد ورودی‌ها و گره‌های قواعد در خلال عملیات فازی-عصبی ثابت است. اکثر سیستم‌های مبتنی بر گرادیان، جمعیت، SVM و ELM از ساختار ایستا استفاده می‌کنند. به‌طور متقابل، در ساختار خودسازمان‌ده، پارامترها و ساختار طی فرآیند یادگیری تغییر کرده و متناسب می‌شوند. در مسیر اجرای الگوریتم تنظیم پارامترها، الگوریتم یادگیری ساختاری نیز برای ساختاردهی خودکار به سیستم مورد استفاده قرار می‌گیرند. همچنان‌که رویه یادگیری ادامه دارد و داده‌های

1. Shihabudheen, K. V.
2. Pillai, G. N.
3. Extreme Learning Machine
4. Support Vector Machine
5. Static
6. Self-organizing

ورودی را به صورت آنالین دریافت می کند، گره های قواعد به طور پویا ایجاد می شوند. در خلال یادگیری، گره های قواعد فازی و گره های عبارات ورودی/خروجی براساس الگوریتم های یادگیری ساختاری ایجاد شده یا حذف می شوند.

فصل ۲

سیستم‌های مشارکتی

همان‌طور که در فصل قبل اشاره شد، ترکیب مشارکتی^۱ به صورت تعیین پارامترهای مشخص یک سیستم فازی (FS) توسط یک شبکه عصبی (NN) و برعکس قابل تعریف است که هر دوی آنها به صورت مستقل از یکدیگر کار کرده و پشتیبانی لازم را فراهم می‌آورد. در اواخر دهه ۱۹۸۰، هر دو فناوری عصبی و فازی به خوبی پابرجا شده بود. در آن زمان، با هدف ارتقای عملکرد کلی سیستم و بهبود زمان طراحی آن در کالاهای مصرفی، پژوهش‌هایی به منظور ترکیب شبکه عصبی و سیستم فازی آغاز شد. از دهه ۱۹۹۰ محصولات زیادی در بازار وجود داشته است که از هر دو شبکه عصبی و سیستم فازی با انواع گوناگونی از ترکیب‌های مشارکتی استفاده کرده‌اند. با روش‌های مختلفی می‌توان این دو سیستم را ترکیب کرد، اما ترکیب‌های زیر از دهه ۱۹۹۰ در کالاهای مصرفی متعددی کاربردی شده و استفاده گسترده‌ای داشته است (تاکاگی^۲، ۱۹۹۲، ۱۹۹۵، ۱۹۹۷):

- شبکه عصبی و سیستم فازی به عنوان ابزار توسعه
- شبکه عصبی و سیستم فازی به عنوان سازوکار اصلاحی
- شبکه عصبی و سیستم فازی با ترکیب آبخاری

1. Cooperative

2. Takagi, H.

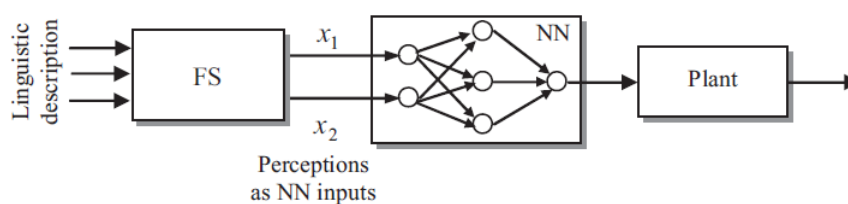
این سه دسته، با دو رویکرد قابل تشریح است: سیستم‌های عصبی فازی مشارکتی^۱ و سیستم‌های فازی عصبی مشارکتی^۲. در سیستم‌های عصبی فازی، سیستم فازی اطلاعات پیش‌نیاز را برای شبکه عصبی به‌منظور کنترل (حل) یک واحد یا مجموعه کاری (مسئله) فراهم می‌آورد. در سیستم‌های فازی عصبی، شبکه عصبی اطلاعات پیش‌نیاز را برای یک سیستم فازی فراهم می‌آورد و یا پارامترها، توابع عضویت و قواعد سیستم فازی را یاد می‌گیرد. شاید ترکیب همزمان دو رویکرد فوق جذاب به نظر برسد، اما از لحاظ محاسباتی ممکن است مزیت یا رویکردی با دوام نباشد.

۱-۲ سیستم‌های عصبی فازی مشارکتی

در این نوع سیستم‌های مشارکتی، عبارات و توصیف‌های زبانی توسط یک سیستم فازی به تعبیرهای مناسب ترجمه شده و یا پارامترهای سیستم تخمین زده می‌شود. این کار به‌منظور ایجاد داده‌های ورودی قابل استفاده برای شبکه عصبی انجام می‌گیرد.

۱-۱-۲ ترکیب آبخاری

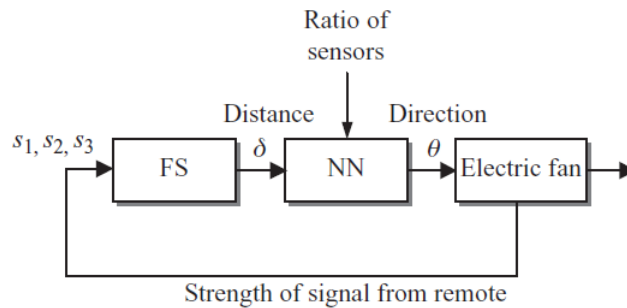
این ترکیب عصبی فازی توسط Sanyo، Hitachi و Toshiba در محصولات مختلف به‌کارگرفته شده است. تصویری کلی از ترکیب آبخاری در شکل ۱-۲ به‌نمایش درآمده است.



شکل ۱-۲: ترکیب آبخاری سیستم عصبی فازی

شکل ۲-۲ نیز یک ترکیب آبخاری را از سیستم فازی و شبکه عصبی نشان می‌دهد که در آن،

1. Cooperative FS-NN systems
2. Cooperative NN-FS systems



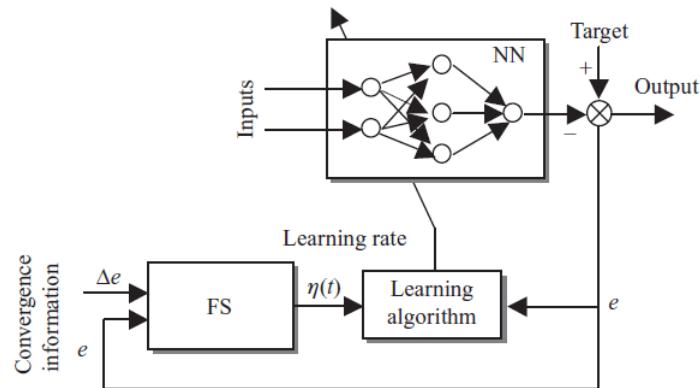
شکل ۲-۲: ترکیب آبخاری: همکاری سیستم فازی با شبکه عصبی

سیستم فازی پارامترهای مورد نیاز شبکه عصبی را برآورد می‌کند. شبکه عصبی، خروجی کنترل برای واحد مورد بررسی را با استفاده از پارامترهای تأمین شده توسط سیستم فازی و اطلاعات حس‌گر ارائه می‌کند. به‌عنوان یک مثال کاربردی می‌توان به نوع خاصی از پنکه‌های برقی که توسط Sanyo ساخته شد، اشاره کرد. این پنکه با استفاده از سه حس‌گر مادون قرمز s_1, s_2, s_3 ، مکان ریموت کنترل خود را تشخیص داده و جهت پنکه را به سمت مکان کاربر تغییر می‌دهد. فاصله بین ریموت کنترل تا پنکه توسط یک سیستم فازی برآورد می‌شود. با فرض اینکه کاربر در نزدیکی ریموت کنترل قرار دارد، شبکه عصبی با استفاده از فاصله برآوردشده و نسبت اطلاعات حس‌گرها، زاویه بین پنکه و دورفرمان را محاسبه می‌کند. تخمین مورد نظر به صورت دو مرحله‌ای انجام می‌شود زیرا مقادیر این سه حس‌گر مادون قرمز با تغییر یافتن فاصله از دورفرمان تغییر می‌کند، حتی اگر زاویه یکسان باقی بماند.

۲-۱-۲ نوع ابزار توسعه

در خلال آموزش شبکه‌های عصبی که از روش پس‌انتشار^۱ یا سایر روش‌ها استفاده می‌کنند، معمولاً نرخ یادگیری ثابت است. سرعت همگرایی چنین الگوریتم‌های آموزشی را می‌توان از طریق تنظیم نرخ یادگیری افزایش داد. در همین راستا، با استفاده از یک سیستم فازی می‌توان به وسیله اطلاعات خطا، نرخ یادگیری را به صورت پویا تعدیل کرد. برخی پژوهشگران قواعدی ابتکاری را به منظور تنظیم پارامترهای یادگیری

1. Backpropagation



شکل ۲-۳: سیستم فازی برای کنترل نرخ یادگیری در یک شبکه عصبی

ارائه کرده‌اند (جیکوبز^۱، ۱۹۸۸؛ هیکن^۲، ۱۹۹۹). آنها پیشنهاد کرده‌اند که هر پارامتر قابل تنظیم تابع هزینه در شبکه باید پارامتر نرخ یادگیری خود را از هر تکرار به تکرار دیگر متفاوت گرداند. زمانی که مشتق تابع هزینه نسبت به وزن، طی تکرارهای متوالی دارای علامت یکسان باشد، پارامتر نرخ یادگیری باید افزایش یابد و زمانی که تغییر علامت وجود داشته باشد، باید کاهش داده شود. هالگاموگی^۳ و همکارانش یک رویکرد آموزشی سریع را ارائه داده‌اند که در آن، نرخ یادگیری طی یک الگوریتم آموزش شبکه عصبی با استفاده از سیستم فازی کنترل می‌شود.

ترکیب ساده‌ای از ابزار توسعه بین سیستم فازی و شبکه عصبی در شکل ۲-۳ نمایش داده شده است. در این ترکیب، با بهره‌گیری از اطلاعات همگرایی برپایه خطا و تغییرات آن یا سایر اطلاعات مشتق، پارامترهای یادگیری مناسبی برای شتاب‌دهی به آموزش شبکه عصبی فراهم می‌شوند. در ادامه، خلاصه‌ای از روش کنترل فازی برای بهبود یادگیری در شبکه‌های پس‌انتشار خطا براساس نگارش لین^۴ و لی^۵ (۱۹۹۶) ارائه می‌شود.

1. Jacobs, R. A.
2. Haykin, S.
3. Halgamuge, S. K.
4. Lin C. T.
5. Lee C. G.

کنترل فازی شبکه‌های پس‌انتشار

قاعده عمومی برای به‌روزرسانی وزن در الگوریتم پس‌انتشار به صورت زیر است:

$$\Delta \mathbf{w}(t+1) = -\alpha \nabla E(\mathbf{w}(t)) + \eta \Delta \mathbf{w}(t) \quad (1-2)$$

E نشان‌گر تابع هزینه است و α و η نیز به ترتیب نرخ یادگیری و بهره تکانه (مومنتوم)^۱ را نشان می‌دهند. هدف این است که یک کنترل‌کننده فازی طراحی شود تا به منظور همگرایی سریع‌تر بتواند پارامترهای یادگیری را بر اساس شکل رویه (سطح) خطا^۲ به صورت خودکار تنظیم کند. به عنوان نمونه، چند قاعده اگر-آنگاه (IF-THEN) فازی برای چنین کنترل‌کننده فازی در ادامه مورد اشاره واقع شده است. در این قاعده‌ها، تغییر در خطا (CE) تخمینی از گرادیان است و تغییر در CE (CEE) نیز نشان‌گر اطلاعات گرادیان مرتبه دوم است که با شتاب همگرایی ارتباط دارد:

- اگر CE کوچک باشد و طی چند تکرار متوالی تغییری در علامت ایجاد نشود، آنگاه مقدار پارامتر یادگیری افزایش می‌یابد.
- اگر طی چند تکرار متوالی تغییری در CE ایجاد شود، آنگاه مقدار پارامتر یادگیری - بدون توجه به مقدار CEE - کاهش می‌یابد.
- اگر CE خیلی کوچک باشد و CE نیز خیلی کوچک باشد و طی چند تکرار متوالی تغییری در علامت ایجاد نشود، آنگاه مقدار پارامترهای یادگیری و تکانه افزایش می‌یابد.

به منظور کمی‌سازی تغییرات علامت در قاعده‌های ابتکاری فوق، پارامتر تغییر علامت به صورت زیر

تعریف می‌شود:

$$SC(t) = 1 - \left\| \frac{1}{\eta} [\text{sgn}(CE(t-1)) + \text{sgn}(E(t))] \right\| \quad (2-2)$$

sgn تابع علامت را نشان می‌دهد و ضرب $\frac{1}{\eta}$ نیز تضمین می‌کند که SC مقدار ۰ (بدون تغییر علامت) یا مقدار ۱ (یک تغییر علامت) را نتیجه دهد. به این ترتیب جمع تجمعی SC، که با CSC نشان داده

1. Momentum

2. Error surface

می‌شود، می‌تواند سابقه تغییر علامت را منعکس کند. به عبارت دیگر:

$$CSC(t) = SC(t) + SC(t - 1) + SC(t - 2) + \dots \quad (3-2)$$

هرچه CSC بزرگتر باشد، تغییرات خطا به طور مکرر رخ می‌دهد. اگر تغییرات خطا را طی ۵ تکرار بررسی کنیم، خواهیم داشت:

$$CSC(t) = \sum_{m=t-4}^t SC(m) \quad (4-2)$$

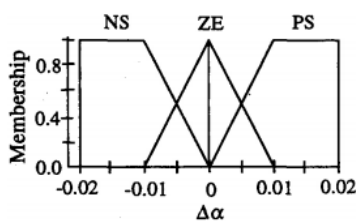
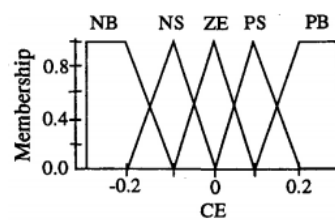
مجموعه کامل قواعد فازی برای الگوریتم پس‌انتشار در جداول تصمیم برای کنترل فازی الگوریتم پس‌انتشار نمایش داده شده است. جدول ۱-۲ قواعد کنترل فازی برای پارامتر نرخ یادگیری است که مقدار متغیر فازی $\Delta\alpha$ را برای مقادیر مفروض CE و CCE با در نظر گرفتن $CSC(t) \leq 2$ نشان می‌دهد. جدول ۲-۲ نیز جدول تصمیم برای کنترل فازی پارامتر تکانه (η) را نمایش می‌دهد. در این جدول متغیر فازی $\Delta\eta$ را برای مقادیر مفروض CE و CCE نشان می‌دهد که — نشان‌گر عدم اصلاح یا تغییر است. حداکثر مقداری که η می‌تواند بگیرد برابر با ۱ قرار داده شده است. در هر دو جدول مذکور، NB، NS، ZE، PS و PB: به ترتیب مقادیر منفی بزرگ، منفی کوچک، صفر، مثبت کوچک و مثبت بزرگ را نشان می‌دهند.

جدول ۱-۲: جدول تصمیم برای کنترل فازی نرخ یادگیری

PB	PS	ZE	NS	NB	CE	CCE
NS	NS	NS	NS	NS		NB
NS	ZE	PS	ZE	NS		NS
ZE	PS	PS	PS	ZE		ZE
NS	ZE	PS	ZE	NS		PS
NS	NS	NS	NS	NS		PB

جدول ۲-۲: جدول تصمیم برای کنترل فازی پارامتر تکانه

PB	PS	ZE	NS	NB	CE	CCE
—	—	—	-۰/۰۱	-۰/۰۱	—	NB
—	—	—	—	-۰/۰۱	—	NS
—	۰/۰۱	۰/۰۱	۰/۰۱	—	—	ZE
-۰/۰۱	—	—	—	—	—	PS
-۰/۰۱	-۰/۰۱	—	—	—	—	PB

(ب) توابع عضویت برای $\Delta\alpha$ 

(آ) توابع عضویت برای CE

شکل ۲-۴: توابع عضویت تعریف شده برای کنترل فازی الگوریتم پس‌انتشار

مقادیر فازی مفروض برای CE و CCE در شکل ۲-۴ تعریف شده است. به‌طور مشابه برای CCE نیز توابع عضویت مشابه با CE مورد استفاده قرار می‌گیرد. براساس جدول مثلاً می‌توان بیان کرد که اگر CE منفی کوچک و CCE صفر باشد، آنگاه $\Delta\alpha$ مثبت کوچک است. نتایج شبیه‌سازی نشان می‌دهد که پس‌انتشار مبتنی بر کنترل فازی به‌طور قابل‌توجه از پس‌انتشار مرسوم هم‌گرایی سریع‌تری دارد.

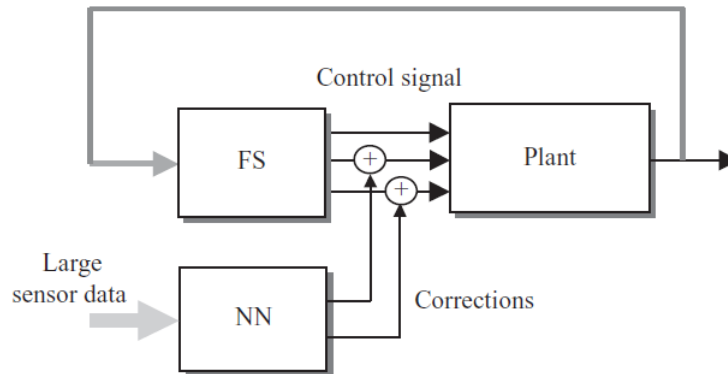
۲-۲ سیستم‌های فازی عصبی مشارکتی

در ترکیب فازی عصبی، شبکه عصبی به‌عنوان فناوری پشتیبان تلقی می‌شود که پارامترهای مختلف سیستم فازی نظیر توابع عضویت، پایگاه قواعد و عوامل مقیاس‌گذاری و وزندهی قواعد حاصل از داده‌های در دسترس تجربی یا حس‌گر را تعیین کرده یا برآورد می‌کند. این موضوع اهمیت دارد که برای استخراج پارامترهای مطلوب و یادگیری شبکه عصبی، از کفایت داده‌های موجود اطمینان حاصل کنیم. در خلال عملکرد سیستم فازی، فرآیند تعیین پارامترها می‌تواند آفلاین یا آنلاین باشد. در کنار این مشارکت، شبکه عصبی را می‌توان به‌عنوان یک پیش‌پردازنده یا پس‌پردازنده برای سیستم فازی در نظر گرفت که در آن، سیستم فازی ساختار ثابت و از پیش تعیین‌شده‌ای دارد. نقش شبکه عصبی در اینجا، بهبود عملکرد سیستم ترکیب‌شده است. علاوه بر این، موضوعات مهم دیگری نیز باید در نظر گرفته شود؛ از جمله اینکه این مشارکت بر مبنای یک سیستم فازی موجود شکل گیرد یا سیستم فازی باید به‌صورت کامل طراحی شود؟ طی دو دهه اخیر، روش‌ها، الگوریتم‌های یادگیری و رویکردهای ابتکاری مختلفی در ادبیات مرتبط با این موضوع گزارش شده است (تاکاگی^۱ و هایاشی^۲، ۱۹۹۱؛ یاگر^۳، ۱۹۹۴؛ بی^۴ و همکاران، ۱۹۹۴؛ تاکاگی، ۱۹۹۵). در ادامه، برخی از روش‌های سودمند و مهم که از شبکه عصبی استفاده می‌کنند، مورد بررسی قرار می‌گیرند.

۱-۲-۲ شبکه عصبی به‌عنوان سازوکار اصلاحی

به‌منظور تأمین تقاضای افزایش‌یافته کاربران برای کنترل هموار لوازم خانگی و حساسیت و دقت آنها، نیاز به داده‌های حس‌گر بیشتری است. افزایش تعداد ورودی‌ها موجب پیچیدگی طراحی کنترل فازی می‌شود و همچنین به زمان محاسبات نیاز دارد. به‌منظور کاهش زمان پردازش کنترل‌کننده فازی، ورودی‌های کمتری توسط سیستم فازی استفاده شده و تعداد ورودی‌های حس‌گر بیشتری به شبکه عصبی اختصاص می‌یابد. این کار، زمان پردازش قابل ملاحظه‌ای را برای سیستم فازی ذخیره می‌کند. با استفاده از شبکه عصبی،

1. Takagi, H.
2. Hayashi, I
3. Yager, R. R.
4. Yea, B.



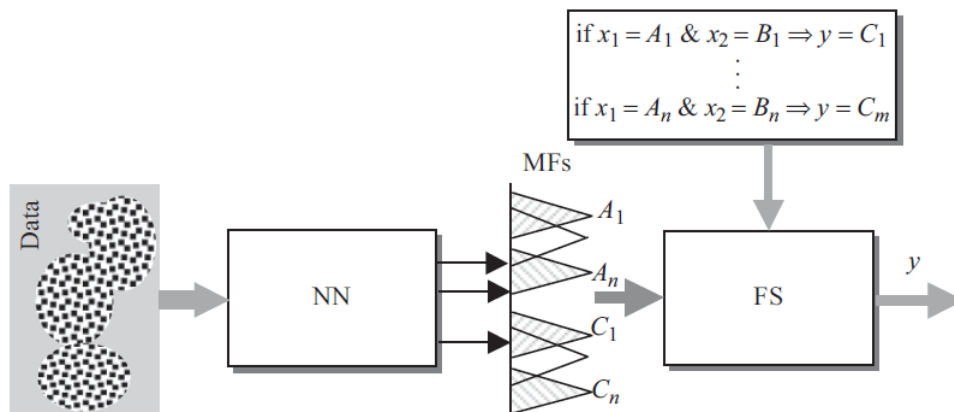
شکل ۲-۵: سازوکار اصلاح خروجی

داده‌های حس‌گر بیشتری به کار رفته و اصلاحات لازم در سیستم فازی ایجاد می‌شود. شکل ۲-۵ ترکیب فازی عصبی را نشان می‌دهد که در آن، خروجی‌های سیستم فازی به وسیله شبکه عصبی اصلاح شده است تا دقت افزایش یابد. این نوع ترکیب، توسط شرکت‌های متعددی در کالاهای مصرفی به کار گرفته شده است؛ نظیر ماشین لباس‌شویی و خشک‌کن (توسط هیتاچی) یا اجاق ماکروویو (توسط سانیو).

۲-۲-۲ شبکه عصبی در تعیین توابع عضویت (نوع ابزار توسعه)

یکی از ویژگی‌های سیستم‌های فازی این است که بخش موتور استنتاج را از حالت فازی^۱ عبارات زبانی مورد استفاده جدا می‌کند. موتور استنتاج در بردارنده قواعد «اگر ... آنگاه» است و حالت فازی به وسیله عبارات زبانی که با توابع عضویت مناسب بیان می‌شوند، ارائه می‌گردد. بنابراین بهره‌گیری از دانشی که با استفاده از عبارات کیفی (یا زبانی) در قالب قواعد بیان می‌شود، ساده است. با این حال، ساخت قواعد فازی (معنای عبارات) وظیفه‌ای دشوار برای طراحان تلقی می‌شود. عملکرد ضعیف یک سیستم فازی عمدتاً به دلیل تعریف نامناسب توابع عضویت است. اگرچه روش سعی و خطا در این زمینه، یک رویکرد پذیرفته‌شده و رایج محسوب می‌شود اما بسیار زمان‌بر است. بنابراین مسئله ساخت توابع عضویت به یک موضوع مرکزی در طراحی سیستم فازی تبدیل شده که تعدادی رویکرد ذهنی، آماری و عصبی برای آن

1. Fuzziness



شکل ۲-۶: یادگیری پارامترهای توابع عضویت از داده‌های در دسترس به کمک شبکه عصبی

پیشنهاد شده است. اگر داده‌های تجربی برای سیستم در دسترس باشد، می‌توان به منظور استخراج مقادیر پارامتر توابع عضویت از رویکرد خوشه‌بندی با شبکه عصبی استفاده کرد. به طور کلی، یک شبکه عصبی در سیستم مشارکتی فازی عصبی تعداد قواعد را با استفاده از خوشه‌بندی به منظور طراحی سیستم فازی تعیین می‌کند. با استفاده از این داده‌های خوشه‌بندی شده، یک شبکه عصبی در تابع عضویت چند بُعدی و غیرخطی به تصمیم‌سازی می‌پردازد که در نتیجه به عنوان تولیدکننده تابع عضویت مورد استفاده قرار می‌گیرد. توابع عضویت یک بُعدی می‌توانند براساس پارامترهایی مثل مراکز خوشه و فاصله از خوشه‌های داده‌های چند بُعدی بنا شوند. یکی از آورده‌های سودمند این رویکرد، وارد کردن شبکه‌های عصبی به فرآیند طراحی سیستم‌های فازی است. علاوه بر این توابع عضویت را می‌توان به صورت کامل و به طور یکجا طراحی کرد به جای اینکه به طور جداگانه در راستای هر محور ورودی انجام گردد. در این ترکیب مشارکتی، پارامترهای توابع عضویت توسط یک شبکه عصبی فراهم می‌شود. چنین ترکیب مشارکتی در شکل ۲-۶ نمایش داده شده است. در همین راستا، عادل و هانگ روشی را برای ایجاد توابع عضویت پیشنهاد کردند که در ادامه تشریح می‌شود.

الگوریتم عادل‌ی و هانگ

عادل‌ی^۱ و هانگ^۲ (۱۹۹۵) الگوریتمی را به منظور تعیین توابع عضویت ارائه دادند که در آن از یک طبقه‌بندی با تغییر وزن و توپولوژی در یک شبکه عصبی دو لایه (شامل لایه ورودی) استفاده شده است. تعداد گره‌های ورودی با تعداد الگوها (M) در هر نمونه آموزشی برابر است. نمونه‌های آموزشی به صورت X_1, X_2, \dots, X_N فرض می‌شوند و تعداد گره‌های خروجی با تعداد خوشه‌ها یکسان و برابر با N است. این الگوریتم به الگوریتم عادل‌ی-هانگ (AHA)^۳ معروف است.

الگوریتم با M ورودی و ۱ خروجی شروع می‌شود. این شبکه به طور اختصار شبکه عصبی $(M, 1)$ یا $NN(M, 1)$ نامیده می‌شود. تعداد نهایی گره‌های خروجی هنوز تعیین نشده است. اولین نمونه آموزشی به اولین خوشه اختصاص می‌یابد. سپس، برای دومین نمونه:

- اگر در اولین خوشه طبقه‌بندی شود، گره خروجی که اولین خوشه را ارائه می‌کند فعال می‌گردد.
- اگر در قالب یک خوشه جدید طبقه‌بندی شود، یک گره خروجی اضافی به شبکه افزوده می‌شود.

این فرآیند تا جایی که همه نمونه‌های آموزشی طبقه‌بندی شوند، ادامه می‌یابد. در ادامه به طور خلاصه به جزئیات مربوط به AHA اشاره می‌شود.

بردار میانگین نمونه‌های آموزشی به صورت زیر تعریف می‌شود.

$$\bar{X}_N = \frac{1}{N} \sum_{i=1}^N x_i \quad (5-2)$$

که برای نمونه آموزشی $N + 1$ به صورت زیر رفتار می‌شود.

$$\bar{X}_{N+1} = \frac{N}{N+1} \bar{X}_N + \frac{1}{N+1} X_{N+1} \quad (6-2)$$

1. Adeli, H.

2. Hung, S. L.

3. Adeli-Hung Algorithm

درجه تفاوت^۱ با استفاده از فاصله اقلیدسی در قالب یک تابع تعریف می‌شود:

$$\text{diff}(X, C_i) = \sqrt{\sum_{j=1}^M (x_j - c_{ij})^2} \quad (7-2)$$

این تابع تفاوت میان یک نمونه آموزشی (X) و یک خوشه (C) (میانگین الگوهای نمونه‌های خوشه) را در یک شبکه $NN(M, 1)$ (یک شبکه عصبی با M گره ورودی و P گره خروجی) نشان می‌دهد.

خوشه‌ای که کمترین درجه تفاوت را داشته باشد، فعال می‌شود:

$$C_{\text{active}} = \left\{ C \mid \min [\text{diff}(X, C_i)], i = 1, 2, \dots, p \right\} \quad (8-2)$$

هنگامی که خوشه‌ها به طور کامل از یکدیگر مجزا شوند، یک ماتریس باینری (Z) به منظور ثبت خوشه هر نمونه مورد استفاده قرار می‌گیرد:

$$\begin{cases} z_{ij} = 1 & \text{اگر نمونه } i \text{ متعلق به خوشه } j \text{ باشد} \\ z_{ij} = 0 & \text{اگر نمونه } i \text{ متعلق به خوشه } j \text{ نباشد} \end{cases} \quad (9-2)$$

اگر خوشه‌ها تا حدی دارای هم‌پوشانی باشند، مرزهای خوشه‌ها فازی می‌شود. با این حال، یک ماتریس باینری همچنان مورد استفاده قرار می‌گیرد. درجه عضویت هر نمونه در هر خوشه براساس فاصله بین آن نمونه و نمونه اولیه^۲ آن خوشه تعیین می‌گردد. نمونه اولیه برای هر خوشه توسط میانگین همه نمونه‌های آن خوشه تعریف می‌شود. با فرض اینکه n_p تعداد نمونه‌های خوشه p است، بردار الگوی متعلق به نمونه i در این خوشه به صورت $X_i^p = (x_{i1}^p, x_{i2}^p, \dots, x_{iM}^p)$ نمایش داده می‌شود. بنابراین، نمونه اولیه این خوشه نیز به صورت زیر محاسبه می‌شود

$$C_p = \left(c_{p1}, c_{p2}, \dots, c_{pM} \right) = \frac{1}{n_p} \sum_{i=1}^{n_p} X_i^p \quad (10-2)$$

1. Degree of difference

2. Prototype

$$c_{pi} = \frac{1}{n_p} \sum_{j=1}^{n_p} x_{ij}^p, \quad j = 1, 2, \dots, M \quad (11-2)$$

درجه عضویت هر نمونه در خوشه p مبتنی بر میزان شباهت بین X_i^p و C_p است. این میزان شباهت با یک نرم موزون از تابع شباهت تعریف می‌شود:

$$D^w(X_i^p, C_p) = \left\| w_p(X_i^p, C_p) \right\|^w \quad (12-2)$$

پارامترهای وزن w_p و w در آن به صورت دلخواه براساس شرایط مورد نظر انتخاب می‌شوند. با استفاده از رابطه $\left[D^w(X_i^p, C_p) \right]$ می‌توان یک تابع عضویت فازی را برای نمونه نام متعلق به p امین خوشه تعریف نمود. با فرض تابع عضویت مثلثی، به صورت زیر تعریف می‌شود:

$$\mu_p(X_i^p) = \begin{cases} 0 & D^w(X_i^p, C_p) > \sigma \\ 1 - \frac{D^w(X_i^p, C_p)}{\sigma} & D^w(X_i^p, C_p) \leq \sigma \end{cases} \quad (13-2)$$

در این رابطه، σ ضریب تقاطع یا حد آستانه را برای توابع عضویت دارای هم‌پوشانی نشان می‌دهد.

موردکاوی

مدیریت پورتفولیو به دنبال ایجاد و کنترل مجموعه‌ای از دارایی‌ها است که به طور مؤثر برای دستیابی به اهداف بازده و ریسک ترکیب می‌شود. در همین راستا، سیستم‌های معاملاتی به طور گسترده در تصمیم‌گیری‌های سرمایه‌گذاری استفاده می‌شوند و بسیاری از معامله‌گران خروجی سیستم‌های معاملاتی را با دیدگاه‌هایشان تنظیم و اصلاح می‌کنند تا به کنترل سبد دارایی‌های خود بپردازند. بنابراین، قابل درک بودن این سیستم‌ها اهمیت داشته و یک مسئله اساسی در بهبود تصمیم‌گیری فعالان بازار سرمایه است. به همین منظور، موسوی و همکاران (۲۰۱۵) طی

مطالعه‌ای به توسعه یک سیستم مبتنی بر قوانین فازی پرداختند که مستقیماً تصمیمات معاملات پرتفوی را از سری‌های زمانی قیمت دارایی‌ها و شاخص‌های فنی استنتاج می‌کند.

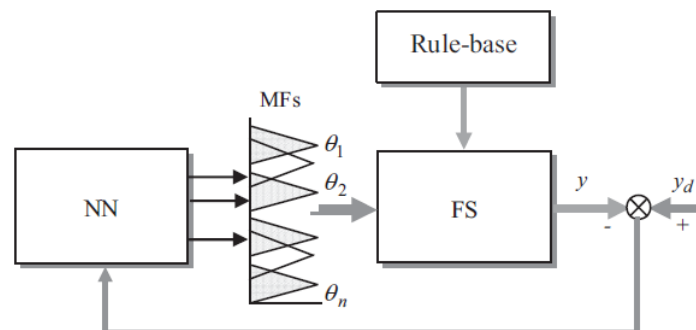
در این پژوهش، ضمن استفاده از الگوریتم عادل‌ی-هانگ، از یک روش مبتنی بر برنامه‌ریزی ژنتیک برای یادگیری یک سیستم تفسیرپذیر و دقیق تاکاگی سوگنو کانگ (TSK) فازی قاعده‌بنیان استفاده شده است. روش پیشنهادی از یک مدل TSK با ساختار جدید بهره می‌گیرد تا بتواند یک سیستم دقیق و قابل تفسیر را برای معاملات پویای پرتفوی ارائه دهد. مدل موردنظر شامل چهار مرحله برای توسعه سیستم فازی TSK است. در مرحله اول، سری زمانی قیمت سهام با استفاده از تبدیل موجک تغییر می‌یابد. اطلاعات غیرضروری و نویزهای داده‌ها در این مرحله حذف می‌شود. در مرحله دوم، مؤثرترین شاخص‌های تکنیکی با استفاده از تحیل رگرسیون گام‌به‌گام انتخاب می‌شود. در مرحله سوم، توابع عضویت مثلثی با استفاده از الگوریتم اصلاح‌شده عادل‌ی-هانگ به شاخص‌های منتخب اختصاص می‌یابد. در نهایت و در گام آخر، سیستم‌های معاملات پرتفوی TSK با استفاده از الگوریتم ژنتیک multitree آموزش می‌بینند. در گام سوم مدل پیشنهادی این پژوهش، الگوریتم عادل‌ی-هانگ با توسعه‌ای همراه شده است. در مراحل AHA به هر نمونه آموزشی یک درجه عضویت اختصاص داده می‌شود؛ این در حالی است که در این پژوهش نیاز به تخصیص یک عدد مشخص از مجموعه‌های فازی به متغیر ورودی وجود دارد. بنابراین، AHA در این مرحله تغییر داده شده و متغیرهای ورودی ابتدا توسط تغییر وزن و توپولوژی شبکه عصبی خوشه‌بندی می‌شوند. گام‌های روش معمولی AHA انجام می‌شود تا جایی که اگر تعداد مطلوب برای خوشه‌ها (برچسب‌های زبانی) حاصل شد، الگوریتم متوقف شده و اگر نه، به گام تعیین مرکز خوشه بازمی‌گردد. به این ترتیب، هر متغیر ورودی به تعدادی از مجموعه‌های فازی تقسیم می‌شود و یک برچسب زبانی به هر مجموعه فازی اختصاص می‌یابد تا برای متخصصان و کاربران معنادار بوده و به راحتی قابل فهم باشد. علاوه بر این، این مجموعه‌های فازی متقارن و مثلثی هستند و مجموعه‌های فازی متعلق به برچسب‌های مجاور در

برخی زوایا هم‌پوشانی دارد جایی که نقاط رأس آنها متقاطع نیستند. این موضوع باعث شفافیت هرچه بیشتر برای این سیستم پیشنهادی می‌شود. موردکاوای این پژوهش در معاملات سبد سهام در بورس اوراق بهادار تهران و همچنین در مبادلات بورس تورنتو و فرانکفورت اجرا شده است.

Mousavi, S., Esfahanipour, A. & Fazel Zarandi, M. H. (2015). MGP-INTACTSKY: Multitree Genetic Programming-based learning of INTERpretable and ACCurate TSK sYSTEMS for dynamic portfolio trading. *Applied Soft Computing*, 34, 449-462.

اگرچه یک سیستم فازی را می‌توان با استفاده از دانش خبره ایجاد کرده و توابع عضویت را با روش کیفی تعریف نمود، اما تعریف دقیق پارامترهای توابع عضویت حاصل از خوشه‌بندی داده‌ها همیشه شدنی نیست زیرا ممکن است که توزیع داده‌ها همه فضای ورودی را نمایندگی نکند. به‌عنوان اولین تلاش، یک شبکه عصبی به‌منظور کسب دانش از مجموعه داده‌های تجربی به‌کار گرفته می‌شود و سپس یک تابع چندبُعدی به توابع یک‌بُعدی تجزیه می‌گردد. خطای بین سیستم فازی طراحی شده و داده‌های واقعی به پارامترهای توابع عضویت یک بُعدی بستگی دارد. این توابع عضویت به طریقه‌ای شبیه به یادگیری پس‌انتشار تنظیم می‌شوند تا خطا کمینه گردد. در مدل شکل ۲-۷ از یک شبکه عصبی برای بهینه‌سازی پارامترهای سیستم فازی (یعنی پارامترهای توابع عضویت) به وسیله حداقل‌سازی خطا بین مشخصات و خروجی سیستم فازی استفاده شده است.

این نوع ترکیب به‌طور گسترده در بسیاری از کاربردها و محصولات مصرفی نظیر ماشین لباس‌شویی، جاروبرقی، پلوپز، ماشین ظرف‌شویی و دستگاه فتوکپی که توسط شرکت‌های ژاپنی توسعه یافته‌اند، مورد استفاده قرار گرفته است.



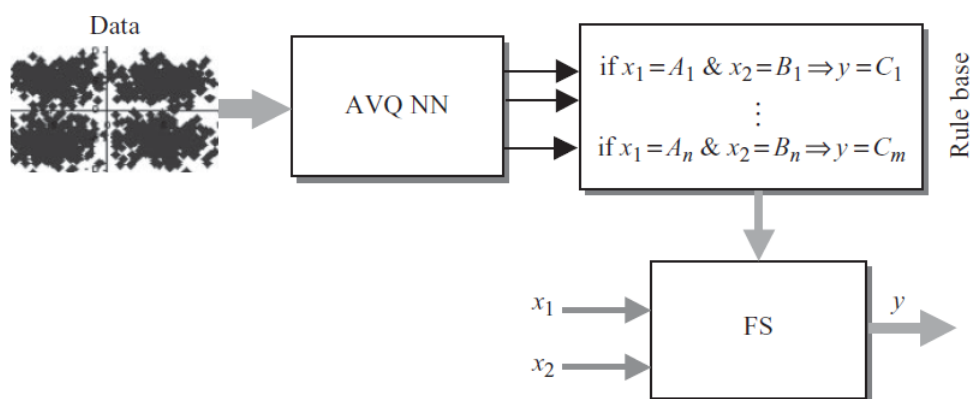
شکل ۲-۷: ترکیب فازی عصبی از نوع ابزار توسعه

۳-۲-۲ شبکه عصبی برای یادگیری قواعد فازی

شبکه عصبی با استفاده از داده‌های آموزشی، قواعد فازی را تعیین می‌کند. یک رویکرد خوشه‌بندی معمولاً از پیاده‌سازی نگاشت‌های خودسازمان‌ده (SOM)^۱ بهره می‌گیرد. SOM به صورت آفلاین آموزش داده می‌شود و سپس در سیستم فازی به کار می‌رود. توابع عضویت و سیستم فازی از پیش تعیین می‌شوند. این سیستم فازی مشارکتی در شکل ۲-۸ نمایش داده شده است. پدريچ^۲ و کارد^۳ به منظور استخراج قواعد فازی از داده‌ها، روش SOM را مورد استفاده قرار دادند (۱۹۹۲). بهره‌گیری از حافظه انجمنی فازی (FAM)^۴ که توسط کاسکو^۵ ارائه شده است (۱۹۹۲)، روش دیگری برای ساخت پایگاه قواعد فازی است که قواعد فازی در آن به صورت پیوندی بین شرط (مقدم) و نتیجه‌ها (تالی) تفسیر می‌شود. اگر مجموعه‌های فازی به صورت نقاطی در مکعب واحد دیده شوند و قواعد نیز هم‌پیوند (انجمن) باشند، بهره‌گیری از حافظه انجمنی عصبی^۶ به منظور نمایندگی قواعد فازی ممکن خواهد بود. حافظه انجمنی عصبی به حافظه انجمنی دو سو به (BAM)^۷ نیز مشهور است؛ زیرا ایجاد ماتریس اتصال^۸

1. Self-Organizing [feature] Maps
2. Pedrycz, W.
3. Card, H. C.
4. Fuzzy Associative Memory
5. Kosko, B.
6. Neural Associative Memory
7. Bidirectional Associative Memory
8. Connection matrix

آن با قاعده یادگیری هب^۱ همخوانی دارد. گونه‌ای از چندی‌سازی بردار انطباقی (AVQ)^۲ به منظور یادگیری FAM از داده‌های موجود توسط کاسکو پیشنهاد شده است. AVQ با عنوان چندی‌سازی بردار یادگیری (LVQ)^۳ نیز شناخته می‌شود. یادگیری AVQ یا LVQ شبیه به SOM است و با استفاده از یادگیری رقابتی^۴ تحقق می‌یابد.



شکل ۲-۸: تعیین قواعد برای سیستم فازی توسط شبکه عصبی

اگر یک سیستم فازی از چندین متغیر ورودی و چندین متغیر خروجی استفاده کند، فرموله کردن قواعد فازی «اگر ... آنگاه» برای خبره دشوار خواهد بود. در چنین حالتی، مطلوب است که قواعد با استفاده از داده‌های در دسترس ناشی از سیستم‌های فیزیکی نیازمند مدل‌سازی استخراج شود. تاکاگی و هایاشی یک روش تعمیم‌یافته با کمک شبکه‌های عصبی مصنوعی را برای سیستم فازی از نوع سوگونوارائه کرده‌اند که در ادامه تشریح می‌شود.

1. Hebbian learning rule
2. Adaptive Vector Quantization
3. Learning Vector Quantization
4. Competitive learning

الگوریتم تاکاگی و هایاشی: استدلال فازی شبکه عصبی محور

ایده اصلی استفاده از شبکه عصبی برای تحقق و تعمیم مدل استنتاج فازی تاکاگی-سوگنو-کانگ (مدل TSK)، پیاده‌سازی توابع عضویت در مقدم‌ها و همچنین تابع استنتاج در تالی‌ها به وسیله شبکه‌های عصبی مناسب است. چنین رویکردی می‌تواند دو مسئله اصلی را در استدلال فازی حل کند: فقدان یک روش معین و صریح برای تعیین توابع عضویت و همچنین نبود یک تابع یادگیری برای تنظیم خودکار قواعد استنتاج. این شبکه‌های عصبی در بخش مقدم می‌توانند توابع عضویت مناسب را یاد بگیرند و نیز در بخش تالی می‌توانند اقدام مناسب یک قاعده را بیاموزند.

استدلال فازی شبکه عصبی محور (NDF)^۱ طرحی برای تعمیم مدل TSK با استفاده از شبکه‌های عصبی است که توسط تاکاگی^۲ و هایاشی^۳ (۱۹۹۱) معرفی شده است. الگوریتم این مدل، مجموعه‌های فازی بخش مقدم را به وسیله شبکه پس‌انتشار می‌سازد. تابع استنتاج تالی متناظر نیز با استفاده از یک شبکه پس‌انتشار ساخته می‌شود.

روش تاکاگی-هایاشی به‌طور کلی شامل سه مرحله زیر است:

۱. افزایش فضای ورودی به تعداد قوانین: این کار از طریق خوشه‌بندی داده‌های موجود انجام می‌شود.
۲. تعیین توابع عضویت با کمک شناسایی مقادیر شرط (مقدم) قواعد مفروض: این کار با به‌کارگیری یک شبکه عصبی در استنتاج تابع عضویت هر قاعده انجام می‌شود.
۳. تعیین مقدار نتیجه (تالی) با کمک شناسایی تابع نتیجه: در روش تاکاگی-هایاشی تابع نتیجه متعلق به هر قاعده با یک شبکه عصبی جایگزین می‌شود که این شبکه عصبی به کمک یادگیری بانظارت آموزش می‌بیند. بنابراین قاعده موردنظر به صورت زیر خواهد بود.

اگر $\mathbf{x} = (x_1, x_2, \dots, x_m)$ برابر A_s باشد، آنگاه

$$y_s = NN_s(x_1, x_2, \dots, x_n) \text{ است.}$$

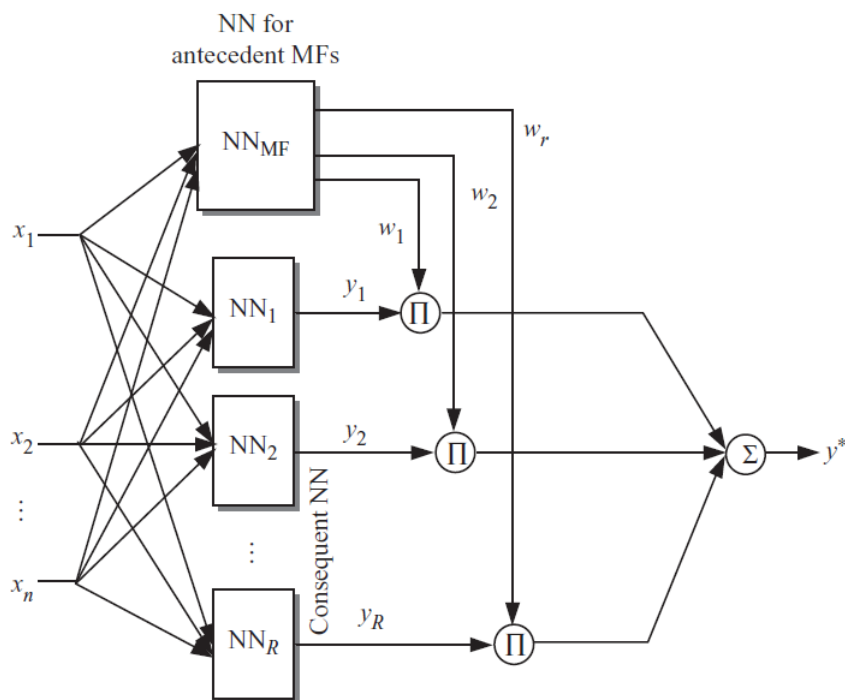
در عبارت فوق، $x = (x_1, x_2, \dots, x_n)$ بردار ورودی را نشان می‌دهد و A_s نمایان‌گر توابع عضویت

1. Neural-network-driven fuzzy reasoning

2. Takagi, H.

3. Hayashi, I.

مقدم است. $NN_s(x_1, x_2, \dots, x_n)$ نیز شبکه عصبی تالی متعلق به قاعده s ام را مشخص می‌کند که تولیدکننده خروجی است. $s = 1, 2, \dots, r$ نیز قاعده استنتاج موردنظر را مشخص می‌کند. یک تصویر کلی از نمودار بلوکی روش تاکاگی-هایاشی در شکل ۲-۹ به نمایش درآمده است. شبکه عصبی NN_{mem} توابع عضویت را برای مقدم قاعده ایجاد کرده و شبکه‌های عصبی NN_1 و \dots و NN_r مقادیر تابع تالی را تولید می‌کنند. خروجی نهایی y^* مجموع خروجی‌های موزون y_s است.



شکل ۲-۹: یک نمودار بلوکی از روش تاکاگی-هایاشی

طراحی فرآیند سیستم (NDF) در گام‌های زیر خلاصه می‌شود (تاکاگی^۱ و هایاشی^۲، ۱۹۹۱؛ لین^۳

ولی^۴، ۱۹۹۶):

1. Takagi, H.
2. Hayashi, I.
3. Lin, C. T.
4. Lee, C. G.

گام ۱: انتخاب متغیرهای ورودی/خروجی و داده‌های آموزشی. متغیر خروجی با y_i و متغیرهای ورودی با x_j ($j = 1, 2, \dots, k$) تعریف می‌شوند. در این گام، متغیرهای x_j ($j = 1, 2, \dots, m; m \leq k$) متناظر با مقادیر مشاهده‌شده y_i ($i = 1, 2, \dots, n$) به وسیله شبکه‌های پسانتشار انتخاب می‌شود. این کار با کمک روش حذف پس‌رو با استفاده از تابع هزینه مجموع مربعات خطا (SSE) انجام می‌شود. متغیرهای ورودی که دارای نویز هستند، حذف می‌شوند و فقط متغیرهای ورودی که همبستگی معناداری با مقادیر خروجی مشاهده‌شده دارند، انتخاب می‌شوند. \hat{y}_i آیین مجموعه از مقادیر ورودی متناظر با مقدار خروجی y_i به صورت $\mathbf{x}_i = (x_1, \dots, x_j, \dots, x_m)^T$ فرض می‌شود که $i = 1, 2, \dots, n$ است.

گام ۲: تعیین داده‌های آموزشی و بازرسی. داده‌های ورودی/خروجی (\mathbf{x}_i, y_i) به داده‌های آموزشی (TRD)^۱ و داده‌های ارزیابی (CHD)^۲ تقسیم می‌شوند که مجموع تعداد این دو دسته برابر با کل داده‌های ورودی/خروجی است: $n_t + n_c = n$.

گام ۳: خوشه‌بندی داده‌های آموزشی. با استفاده از یک روش خوشه‌بندی باید TRD را به r دسته R^s تقسیم کرد که $s = 1, 2, \dots, r$ است. برای TRD با (\mathbf{x}_i^s, y_i^s) مشخص می‌شود که $i = 1, 2, \dots, n_s$ و n_s نمایانگر تعداد TRD برای R^s است. تقسیم فضای m بعدی به r ، به معنای این است که تعداد قواعد استنتاج برابر با r در نظر گرفته شده است.

گام ۴: آموزش شبکه عصبی NN_{mem} متناظر با بخش‌های مقدم از قواعد استنتاج فازی. در این مرحله، یک شبکه عصبی عضویت (NN_{mem}) بین هر بردار ورودی و بردار اختصاص‌یافته به خوشه متناظرش آموزش می‌بیند که فرآیند آن در شکل ۲-۱۰ تصویر شده است. برای هر بردار ورودی $x_i \in R^s$ از TRD یک بردار $w_i = (w_i^1, w_i^2, \dots, w_i^r)^T$ تعریف می‌شود. اگر \mathbf{x}_i مقادیر لایه ورودی باشد، w_i^s

1. Training data

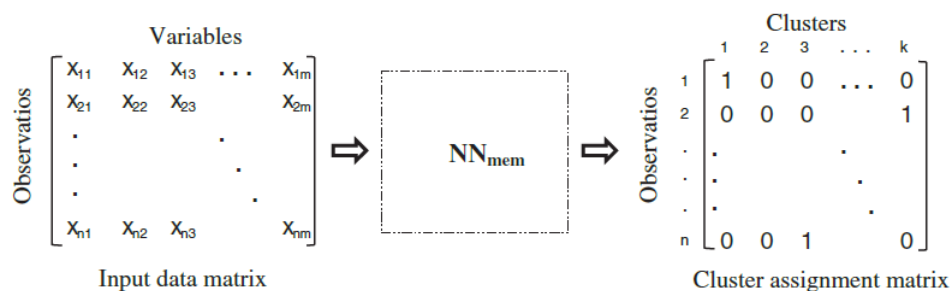
2. Checking data

به‌عنوان داده‌های نظارت‌شده برای لایه خروجی تعیین می‌شوند:

$$w_i^s = \begin{cases} 1 & \mathbf{x}_i \in R^s \\ 0 & \mathbf{x}_i \notin R^s \end{cases} \quad i = 1, 2, \dots, n_i, \quad s = 1, \dots, r \quad (14-2)$$

سپس یک شبکه عصبی با m گره ورودی و r گره خروجی براساس زوج‌های (\mathbf{x}_i, w_i) آموزش داده می‌شود. بنابراین شبکه عصبی قادر خواهد بود که درجه نسبت (w_i^s) هر داده ورودی (\mathbf{x}_i) به R^s را استنتاج کند. به این ترتیب تابع عضویت بخش مقدم در قالب مقدار استنتاج‌شده w_i^s - که خروجی شبکه عصبی NN_{mem} است - تعریف می‌شود. یعنی:

$$\mu_{A_s}(x_i) = \hat{w}_i^s \quad i = 1, 2, \dots, n, \quad s = 1, \dots, r \quad (15-2)$$



شکل ۲-۱۰: آموزش شبکه عصبی عضویت

گام ۵: آموزش شبکه عصبی NN_s متناظر با بخش تالی قاعده استنتاج فازی s^m . مقادیر ورودی TRD $(x_{i1}^s, x_{i2}^s, \dots, x_{im}^s)$ و مقدار خروجی آن (y_i^s) به ورودی و خروجی NN_s تخصیص داده می‌شود. این شبکه عصبی بخش تالی در R^s است. با آموزش NN_s می‌توان مقدار کنترلی را استنتاج نمود. مقادیر ورودی CHD $(x_{i1}, x_{i2}, \dots, x_{im})$ در این شبکه عصبی جایگزین می‌شوند تا بتوان مجموع مربعات خطا را به دست آورد:

$$E_m^s = \sum_{i=1}^{n_c} \left\{ y_i - u_s(\mathbf{x}_i) \cdot \mu_{A_s}(\mathbf{x}_i) \right\}^2 \quad (16-2)$$

که مقدار تخمین زده شده $u_s(\mathbf{x}_i)$ به عنوان خروجی NN_s حاصل می‌شود. همچنین می‌توان خطا را با وزن‌ها تعریف کرد:

$$E_m^s = \sum_{i=1}^{n_c} \mu_{A_s}(\mathbf{x}_i) \left\{ y_i - u_s(\mathbf{x}_i) \cdot \mu_{A_s}(\mathbf{x}_i) \right\}^2 \quad (17-2)$$

گام ۶: ساده‌سازی بخش تالی با استفاده از یک روش حذف پس‌رو. از میان m متغیر ورودی یک شبکه عصبی که مقادیر کنترلی بخش‌های تالی را برای هر قاعده استنتاج می‌کند، یکی از متغیرهای ورودی (x^p) به‌طور دلخواه حذف می‌شود. سپس شبکه عصبی با استفاده از TRD مجدداً مطابق گام ۵ آموزش می‌بیند. رابطه زیر مربعات خطای (E_{m-1}^{sp}) مقدار کنترلی قاعده s ام را در شرایط حذف x_p به دست می‌دهد. مقدار E_{m-1}^{sp} را می‌توان با به‌کارگیری CHD برآورد کرد:

$$E_{m-1}^{sp} = \sum_{i=1}^{n_c} \left\{ y_i - u_s(\mathbf{x}_i) \cdot \mu_{A_s}(\mathbf{x}_i) \right\}^2 \quad p = 1, 2, \dots, m \quad (18-2)$$

که $\hat{\mathbf{x}}_i = (x_{i,1}, \dots, x_{i,p-1}, x_{i,p+1}, \dots, x_{i,m})^T$ است. در مقایسه تساوی‌های ۱۶-۲ و ۱۸-۲ اگر رابطه

$$E_m^s > E_{m-1}^{sp} \quad (19-2)$$

برقرار باشد، آنگاه می‌توان معناداری متغیرهای ورودی حذف شده (x_p) را اندک در نظر گرفت و x_p را حذف کرد.

این کار به‌طور مشابه برای $m - 1$ متغیر ورودی باقی‌مانده نیز انجام می‌گردد. گام‌های ۵ و ۶ به‌طور مداوم تکرار می‌شوند تا زمانی که دیگر رابطه ۱۹-۲ برای هیچ متغیر ورودی باقی‌مانده‌ای برقرار نگردد. مدلی که کمترین مقدار E^s را حاصل کند، بهترین شبکه عصبی است.

به این ترتیب گام‌های ۱ تا ۶ بخش‌های مقدم و بخش‌های تالی را برای هر قاعده استنتاج تعیین می‌کند. لذا فرآیند شناخت برای مدل فازی کامل می‌شود.

گام ۷: تصمیم بر خروجی نهایی. با استفاده از رابطه زیر می‌توان مقدار کنترلی نهایی (y_i^*) را نتیجه گرفت.

$$y_i^* = \frac{\sum_{s=1}^r \mu_{A_s}(\mathbf{x}_i) \cdot \overline{u_s(\mathbf{x}_i)}}{\sum_{s=1}^r \mu_{A_s}(\mathbf{x}_i)} \quad i = 1, 2, \dots, n \quad (2-20)$$

$\overline{u_s(\mathbf{x}_i)}$ مقدار استنتاج‌شده‌ای است که از بهترین شبکه عصبی که با جایگزین کردن CHD در گام قبل به دست آمد، حاصل می‌شود.

روند فوق در مثال زیر روشن می‌شود. در اینجا منظور از عبارت $[u_1 \times u_2 \times \dots \times u_l]$ اندازه

شبکه پس‌انتشار است که تعداد نرون‌ها در هر یک از لایه‌ها توسط u_i نشان داده می‌شود.

مثال ۲-۱. در این مثال، روش استدلال فازی شبکه عصبی محور بر روی داده‌های حاصل از $y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2$ و نویز تصادفی x_4 آموزش داده می‌شود.

گام ۱ و ۲: مجموعه‌ای از ۴۰ الگوی ورودی/خروجی $(x_1, x_2, x_3, x_4; y)$ را که به صورت تصادفی با در نظر گرفتن $x_1 = 1, 5$, $x_2 = 1, 3, 5$, $x_3 = 1, 2, 3, 4, 5$ و $x_4 = 1, 5$ ایجاد شده‌اند را در نظر می‌گیریم. این الگوها به دو بخش تقسیم می‌شود: ۲۰ الگو برای داده‌های آموزشی و ۲۰ الگو برای داده‌های ارزیابی. بنابراین در اینجا، $n_t = n_c = 20$ است. جدول ۲-۳ این ۴۰ داده را نمایش می‌دهد که الگوهای ۱ تا ۲۰ داده‌های آموزشی و الگوهای ۲۱ تا ۴۰ داده‌های ارزیابی هستند. نتیجه آموزش پس از طی ۱۵۰۰۰ تکرار با یک شبکه $[1 \times 3 \times 3 \times 4]$ که تمام متغیرها را در بردارد و یک شبکه $[1 \times 3 \times 3 \times 3]$ که برای انتخاب متغیرهای ورودی به کار می‌رود، در جدول ۲-۴ ارائه شده است. هر دو مجموعه داده‌های آموزشی و ارزیابی برای این مدل‌های یادگیری مورد استفاده قرار گرفته‌اند. عملکرد برآوردشده برای مدلی که در آن x_4 حذف شده است، نسبتاً شبیه به مدلی است که از تمام متغیرها استفاده می‌کند؛ از این رو متغیر ورودی x_4 اندک و قابل چشم‌پوشی است و می‌توان از آن صرف‌نظر کرد.

گام ۳: با استفاده از یک روش خوشه‌بندی متداول، TRD به دو بخش افزایش می‌شود. این داده‌های

آموزشی بخش‌بندی شده در جدول ۲-۵ به نمایش درآمده است.

جدول ۲-۳: داده‌های ورودی و خروجی مثال

داده‌های ارزیابی (CHD)						داده‌های آموزشی (TRD)					
y	x_4	x_3	x_2	x_1		y	x_4	x_3	x_2	x_1	
۹/۵۴	۱	۵	۱	۱	۲۱	۱۱/۱۱	۱	۱	۳	۱	۱
۶/۰۴	۱	۴	۳	۱	۲۲	۶/۵۲	۱	۲	۵	۱	۲
۵/۷۲	۵	۳	۵	۱	۲۳	۱۰/۱۹	۵	۳	۱	۱	۳
۱۱/۲۵	۵	۲	۱	۱	۲۴	۶/۰۴	۵	۴	۳	۱	۴
۱۱/۱۱	۱	۱	۳	۱	۲۵	۵/۲۴	۱	۵	۵	۱	۵
۱۴/۳۶	۱	۲	۵	۵	۲۶	۱۹/۰۲	۱	۴	۱	۵	۶
۱۹/۶۱	۵	۳	۱	۵	۲۷	۱۴/۱۵	۵	۳	۳	۵	۷
۱۳/۶۵	۵	۴	۳	۵	۲۸	۱۴/۳۶	۵	۲	۵	۵	۸
۱۲/۴۳	۱	۵	۵	۵	۲۹	۲۷/۴۲	۱	۱	۱	۵	۹
۱۹/۰۲	۱	۴	۱	۵	۳۰	۱۵/۳۹	۱	۲	۳	۵	۱۰
۶/۳۸	۵	۳	۳	۱	۳۱	۵/۷۲	۵	۳	۵	۱	۱۱
۶/۵۲	۵	۲	۵	۱	۳۲	۹/۷۷	۵	۴	۱	۱	۱۲
۱۶/۰۰	۱	۱	۱	۱	۳۳	۵/۸۷	۱	۵	۳	۱	۱۳
۷/۲۲	۱	۲	۳	۱	۳۴	۵/۴۱	۱	۴	۵	۱	۱۴
۵/۷۲	۵	۳	۵	۱	۳۵	۱۰/۱۹	۵	۳	۱	۱	۱۵
۱۹/۰۲	۵	۴	۱	۵	۳۶	۱۵/۳۹	۵	۲	۳	۵	۱۶
۱۳/۳۹	۱	۵	۳	۵	۳۷	۱۹/۶۸	۱	۱	۵	۵	۱۷
۱۲/۶۸	۱	۴	۵	۵	۳۸	۲۱/۰۶	۱	۲	۱	۵	۱۸
۱۹/۶۱	۵	۳	۱	۵	۳۹	۱۴/۱۵	۵	۳	۳	۵	۱۹
۱۵/۳۹	۵	۲	۳	۵	۴۰	۱۲/۶۸	۵	۴	۵	۵	۲۰

جدول ۲-۴: نتایج حذف پس‌رو با استفاده از یک شبکه عصبی

مجموع مربعات خطا

۰/۰۰۰۰۷	زمانی که از همه متغیرها استفاده می‌شود
۰/۳۹۳۶	زمانی که x_1 حذف می‌شود
۰/۱۴۸۲	زمانی که x_2 حذف می‌شود
۰/۰۸۷۲	زمانی که x_3 حذف می‌شود
۰/۰۰۰۱۹	زمانی که x_4 حذف می‌شود

جدول ۲-۵: افزاز قواعد بر اساس داده‌های آموزشی

شماره داده آموزشی	قاعده کنترلی
۱, ۲, ۳, ۴, ۵, ۱۱, ۱۲, ۱۳, ۱۴, ۱۵	R^1
۶, ۷, ۸, ۹, ۱۰, ۱۶, ۱۷, ۱۸, ۱۹, ۲۰	R^2

گام ۴: یک شبکه عصبی $[2 \times 3 \times 3 \times 3]$ طی ۵۰۰۰ بار آموزش اجرا می‌شود تا بتواند $w_i^s \in \{0, 1\}$ (یعنی درجه نسبت داده آموزشی x_i به A_s) را با مقدار $\hat{w}_i^s \in [0, 1]$ استنتاج کند. به وسیله این آموزش، عدد فازی A_s در بخش مقدم حاصل می‌شود. جدول ۲-۶ توابع مقادیر عضویت این عدد فازی را برای قواعد کنترلی R^s نشان می‌دهد.

گام ۵: از فرمول استنتاج برای تعیین مقدار کنترلی در بخش تالی قواعد کنترلی مختلف استفاده می‌شود. جدول ۲-۷ خطاهای خروجی (E_p^s) که پس از ۲۰۰۰۰ تکرار آموزش شبکه $[3 \times 8 \times 8 \times 1]$ حاصل شده است را نشان می‌دهد.

گام ۶: زمانی که متغیرهای ورودی به طور دلخواه از بخش مقدم حذف می‌شوند، مجموع مربعات خطا (E_p^{sp}) با قاعده کنترلی R^s به دست می‌آید. این مجموع در جدول ۲-۸ نشان داده شده است که از قواعد کنترلی R^1 و R^2 پس از یادگیری شبکه $[2 \times 8 \times 8 \times 1]$ طی ۱۰۰۰۰ تا ۲۰۰۰۰ تکرار حاصل شده است.

جدول ۲-۶: مقادیر عضویت برای قاعده R^s

مقدار عضویت		داده‌های آموزشی		
قاعده ۱	قاعده ۲	x_3	x_2	x_1
۰٫۹۹۷۰	۰٫۰۰۳۱	۱	۳	۱
۰٫۹۹۷۲	۰٫۰۰۲۸	۲	۵	۲
۰٫۹۹۷۲	۰٫۰۰۲۸	۳	۱	۳
۰٫۹۹۷۳	۰٫۰۰۲۷	۴	۳	۴
۰٫۹۹۷۴	۰٫۰۰۲۶	۵	۵	۵
۰٫۰۰۲۸	۰٫۹۹۷۱	۴	۱	۵
۰٫۰۰۲۸	۰٫۹۹۷۲	۳	۳	۵
۰٫۰۰۲۷	۰٫۹۹۷۲	۲	۵	۵
۰٫۰۰۲۷	۰٫۹۹۷۳	۱	۱	۵
۰٫۰۰۲۷	۰٫۹۹۷۳	۲	۳	۵
۰٫۹۹۷۳	۰٫۰۰۲۸	۳	۵	۱
۰٫۹۹۷۳	۰٫۰۰۲۷	۴	۱	۱
۰٫۹۹۷۴	۰٫۰۰۲۶	۵	۳	۱
۰٫۹۹۷۳	۰٫۰۰۲۷	۴	۵	۱
۰٫۹۹۷۲	۰٫۰۰۲۸	۳	۱	۱
۰٫۹۹۷۳	۰٫۰۰۲۷	۲	۳	۵
۰٫۹۹۷۳	۰٫۰۰۲۷	۱	۵	۵
۰٫۹۹۷۳	۰٫۰۰۲۷	۲	۱	۵
۰٫۹۹۷۲	۰٫۰۰۲۸	۳	۳	۵
۰٫۹۹۷۱	۰٫۰۰۲۹	۴	۵	۵

جدول ۷-۲: خطاهای خروجی

قاعده کنترلی ۱: E_p^1 ۲۷/۸۶

قاعده کنترلی ۲: E_p^2 ۱/۹۳

جدول ۸-۲: خطاهای خروجی پس از حذف متغیرها

قاعده ۲	قاعده ۱	
$E_p^2 = ۱/۹۳$	$E_p^1 = ۲۷/۸۶$	از جدول ۷-۲
$E_p^{21} = ۰/۹۳$	$E_p^{11} = ۴۲/۸۴$	زمانی که x_1 حذف می‌شود
$E_p^{22} = ۱۱۹/۶۱$	$E_p^{12} = ۷۴/۷۱$	زمانی که x_2 حذف می‌شود
$E_p^{23} = ۷۳/۲۸$	$E_p^{13} = ۵۵/۲۷$	زمانی که x_3 حذف می‌شود

با مقایسه گام ۵ و ۶ برای هر قاعده کنترلی، وضعیت‌های زیر قابل مشاهده است:

$$E_p^{1p} > E_p^1 (= ۲۷/۸۶)$$

و

$$E_p^{21} (= ۰/۹۳) < E_p^2 (= ۱/۹۳)$$

بنابراین شبکه عصبی گام ۵ به‌عنوان مدل بخش‌های تالی قاعده کنترلی ۱ طراحی می‌شود. این محاسبات برای قاعده ۲ ادامه می‌یابد و محاسبات تکراری در گام ۲ به پایان می‌رسد. لذا شبکه عصبی حاصل - که ورودی‌های (x_2, x_3) را شامل می‌شود - به‌عنوان مدل پایانی طراحی می‌شود. بنابراین مدل فازی به‌دست آمده به‌صورت زیر بیان می‌شود.

$$R^1: \text{اگر } \mathbf{x} = (x_1, x_2, x_3) \text{ برابر } A_1 \text{ باشد، آنگاه } y_1 = NN_1(x_1, x_2, x_3)$$

است.

$$R^2: \text{اگر } \mathbf{x} = (x_1, x_2, x_3) \text{ برابر } A_2 \text{ باشد، آنگاه } y_2 = NN_2(x_2, x_3) \text{ است.}$$

جدول ۲-۹: خروجی استدلال فازی شبکه عصبی محور آموزش دیده

داده‌های ارزیابی				داده‌های آموزشی					
W_i^2	W_i^1	y^*	y	W_i^2	W_i^1	y^*	y		
۰/۰۰۰۲۷	۰/۹۹۷۴	۸/۸۸	۹/۵۵	۲۱	۰/۰۰۰۳۱	۰/۹۹۹۷	۱۱/۱۴	۱۱/۱۱	۱
۰/۰۰۰۲۷	۰/۹۹۷۳	۶/۱۴	۶/۰۴	۲۲	۰/۰۰۰۲۸	۰/۹۹۷۲	۶/۵۳	۶/۵۲	۲
۰/۰۰۰۲۸	۰/۹۹۷۳	۵/۷۱	۵/۷۲	۲۳	۰/۰۰۰۲۸	۰/۹۹۷۲	۱۰/۲۱	۱۰/۱۹	۳
۰/۰۰۰۰۳	۰/۹۹۷۱	۱۰/۵۵	۱۱/۲۵	۲۴	۰/۰۰۰۲۷	۰/۹۹۷۳	۶/۱۴	۶/۰۴	۴
۰/۰۰۰۳۱	۰/۹۹۹۷	۱۱/۱۴	۱۱/۱۱	۲۵	۰/۰۰۰۲۶	۰/۹۹۷۴	۵/۳۷	۵/۲۴	۵
۰/۹۹۷۲	۰/۰۰۰۲۷	۱۴/۳۳	۱۴/۳۶	۲۶	۰/۹۹۷۱	۰/۰۰۰۲۸	۱۹/۰۰	۱۹/۰۲	۶
۰/۹۹۷۲	۰/۰۰۰۲۸	۱۹/۰۶	۱۹/۶۱	۲۷	۰/۹۹۷۲	۰/۰۰۰۲۸	۱۴/۱۳	۱۴/۱۵	۷
۰/۹۹۷۱	۰/۰۰۰۲۹	۱۳/۹۲	۱۳/۶۵	۲۸	۰/۹۹۷۲	۰/۰۰۰۲۷	۱۴/۳۳	۱۴/۳۶	۸
۰/۹۹۶۹	۰/۰۰۰۰۳	۱۲/۲۹	۱۲/۴۳	۲۹	۰/۹۹۷۳	۰/۰۰۰۲۷	۲۷/۳۷	۲۷/۴۲	۹
۰/۹۹۷۱	۰/۰۰۰۲۸	۱۹/۰۰	۱۹/۰۲	۳۰	۰/۹۹۷۳	۰/۰۰۰۲۷	۱۵/۳۸	۱۵/۳۹	۱۰
۰/۰۰۰۲۸	۰/۹۹۷۲	۷/۱۸	۶/۳۸	۳۱	۰/۰۰۰۲۸	۰/۹۹۷۳	۵/۷۱	۵/۷۲	۱۱
۰/۰۰۰۲۸	۰/۹۹۷۲	۶/۵۳	۶/۵۲	۳۲	۰/۰۰۰۲۷	۰/۹۹۷۳	۹/۷۹	۹/۷۷	۱۲
۰/۰۰۰۳۲	۰/۹۹۶۹	۱۱/۲۴	۱۶/۰۰	۳۳	۰/۰۰۰۲۶	۰/۹۹۷۴	۵/۷۵	۵/۸۷	۱۳
۰/۰۰۰۲۹	۰/۹۹۷۱	۹/۰۲	۷/۲۲	۳۴	۰/۰۰۰۲۷	۰/۹۹۷۳	۵/۴۵	۵/۴۱	۱۴
۰/۰۰۰۲۸	۰/۹۹۷۳	۵/۷۱	۵/۷۲	۳۵	۰/۰۰۰۲۸	۰/۹۹۷۲	۱۰/۲۱	۱۰/۱۹	۱۵
۰/۹۹۷۱	۰/۰۰۰۲۸	۱۹/۰۰	۱۹/۰۲	۳۶	۰/۹۹۷۳	۰/۰۰۰۲۷	۱۵/۳۸	۱۵/۳۹	۱۶
۰/۹۹۹۷	۰/۰۰۰۰۳	۱۳/۸۹	۱۳/۳۹	۳۷	۰/۹۹۷۳	۰/۰۰۰۲۷	۱۹/۶۵	۱۹/۶۸	۱۷
۰/۹۹۷۱	۰/۰۰۰۲۹	۱۲/۶۷	۱۲/۶۸	۳۸	۰/۹۹۷۳	۰/۰۰۰۲۷	۲۱/۰۵	۲۱/۰۶	۱۸
۰/۹۹۷۲	۰/۰۰۰۲۸	۱۹/۰۶	۱۹/۶۱	۳۹	۰/۹۹۷۲	۰/۰۰۰۲۸	۱۴/۱۳	۱۴/۱۵	۱۹
۰/۹۹۷۳	۰/۰۰۰۲۷	۱۵/۳۸	۱۵/۳۹	۴۰	۰/۹۹۷۱	۰/۰۰۰۲۹	۱۲/۶۷	۱۲/۶۸	۲۰

جدول ۲-۹ مقادیر y^* حاصل از رابطه ۲-۲۰ را نشان می‌دهد.

موردکاوی

میراحدی و زاید (۲۰۱۶) ضمن توسعه بخش خوشه‌بندی روش استدلال فازی شبکه عصبی محور (NNDFR)^T، این مدل را در مطالعه‌ای پیرامون تخمین و برآورد بهره‌وری ساخت به‌کار بستند و توسعه دادند. توسعه مذکور مبتنی بر یافتن پارامترها (فازی‌کننده و تعداد خوشه‌ها) است. به‌همین منظور از الگوریتم ژنتیک برای دستیابی به بالاترین سطح تنظیم استفاده شده است.^۳

برآورد بهره‌وری عملیات ساخت - به‌عنوان یک عنصر برنامه‌ریزی و کنترل پروژه - به هدف قابل‌توجه در مدل‌های پیش‌بینی تبدیل شده است. در این پژوهش، یک مدل هوشمند ترکیبی برای افزایش دقت در برآورد بهره‌وری در عملیات ساخت پیشنهاد شده است. فرآیند بتن‌ریزی به‌عنوان یک فرآیند ساخت در نظر گرفته شده و عوامل وضعیت جوی (دما، رطوبت، سرعت باد و بارش)، نیروی انسانی (اندازه جمعیت و درصد کارگران) و پروژه (نوع کار، ارتفاع کف و روش کار) به‌عنوان متغیرهای منتخب آن برشمرده شده‌اند. سپس مدل موردنظر براساس متغیرهای موردنظر و میزان بهره‌وری (بازدهی) روزانه پیاده‌سازی شده است. دستاوردهای پژوهشی این کار تحقیقاتی می‌تواند در مرتبط کردن محیط‌زیست و مدیریت موفقیت‌آمیز آن مؤثر باشد.

نوآوری این پژوهش، ایجاد یک نسخه اصلاح‌شده از ساختار NNDFR است که قابلیت تنظیم بهتر و ارتقای بیشتری پیدا کرده است. این توسعه براساس ایجاد رابطه‌ای است که به مدل اجازه می‌دهد تا با هر دو نوع متغیر ورودی - یعنی قطعی و فازی - سروکار داشته باشد. ساختار NNDFR ابتدا با جانشین کردن FCM برای الگوریتم K-means اصلاح می‌شود که موجب تنظیم حالت فازی توابع عضویت می‌گردد. سپس پارامترهای مدل توسط یک الگوریتم ژنتیک بهینه می‌گردد. در نهایت، روشی موسوم به رویکرد ترکیبی برای هم‌زمانی پایگاه‌های داده مدل به‌کار گرفته می‌شود که هر دو نوع اعداد فازی و مقادیر قطعی را در نظر می‌گیرد. در اینجا مؤلفه‌های دما و رطوبت فازی است و سایر مؤلفه‌ها نظیر سرعت باد، روش کار و ... مقادیر عددی قطعی دارند.

در نهایت، خروجی مدل پیشنهادی با خروجی NNDFR معمولی، شبکه عصبی مصنوعی (ANN) و مدل سیستم استنتاج عصبی فازی انطباقی (ANFIS) در قالب مقدار خطای MSE مقایسه شده است که نسبت به آنها به ترتیب ۶۹٪، ۸۳٪ و ۷۲٪ بهبود داشته است.

Mirahadi, F. & Zayed, T. (2016) Simulation-based construction productivity forecast using Neural-Network-Driven Fuzzy Reasoning. *Automation in Construction*, 65, 102-115.

آ. در این پژوهش به جای NDF از NNDFR برای نام اختصاری روش استفاده شده است.

رویکرد نوآورانه فازی‌سازی در روش NDF به‌طور خودکار وابستگی متقابل متغیرهای ورودی را با استفاده از تابع عضویت سطح بالا^۱ در نظر می‌گیرد و ظرفیتی را ایجاد می‌کند که در فرآیند فازی‌سازی مرسوم وجود ندارد. این روش، یک تابع عضویت سطح بالای شبکه عصبی محور می‌سازد که با ترکیب تمام توابع عضویت دو بُعدی از متغیرهای مختلف، یک تابع عضویت چند بُعدی را ایجاد می‌کند. توابع عضویتی که با متغیرهای وابسته سروکار دارند باید با یک محور که مقدار عضویت همه متغیرها را نمایش می‌دهد، با سطوح بالا به شکل منحنی درآید. به این ترتیب، تغییرات در یک متغیر می‌تواند مقادیر عضویت دیگر را دست‌خوش تغییر کند. ساخت چنین توابع عضویتی فقط از طریق به‌کارگیری الگوریتم‌های خوشه‌بندی فازی ممکن است. با این وجود، محدودیت‌هایی نیز در روش NDF وجود دارد. دقت عملکرد این روش به تعداد خوشه‌ها - و تبعاً به تعداد شبکه‌های عصبی تالی - حساس است. با این حال، هیچ رویکردی برای بهینه‌سازی تعداد خوشه‌ها اتخاذ نشده است. از سوی دیگر، NN_{mem} که خروجی الگوریتم خوشه‌بندی سخت است، یک حالت فازی را ایجاد می‌کند که نمی‌تواند کنترل یا تنظیم شود زیرا بخش نظارت‌شده از فرآیند یادگیری به‌قدر کافی انعطاف‌پذیر نیست و تنها مقدار ۰ یا ۱ را می‌پذیرد. در نتیجه، توابع عضویت و حالت فازی هر نقطه قابل کنترل نیست. به عبارت دیگر، این سیستم همه نقاط را در یک خوشه یکسان در نظر گرفته و فاصله از مرکزوار آنها را در این تصمیم‌گیری لحاظ نمی‌کند.

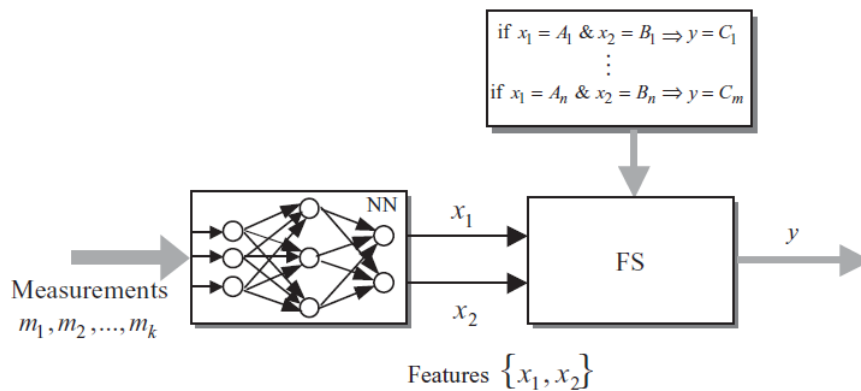
1. Hyper-surface membership function

2. Centroid

اطلاعات بیشتر پیرامون مدل NDF را می‌توان در پژوهش‌های تاکاگی^۱ و هایاشی^۲ (۱۹۹۱) و تسوکالاس^۳ و یوریگ^۴ (۱۹۹۷) مشاهده کرد.

۴-۲-۲ شبکه عصبی برای انتخاب ویژگی

در بسیاری از سیستم‌های دنیای واقعی، داده‌هایی که از سنسورها و اندازه‌گیری‌ها به دست می‌آیند، دارای نویز (نوفه) بوده و اطلاعات بی‌فایده و زائدی دارند. این موضوع باعث می‌شود که سیستم فازی نتواند از آنها به طور مؤثری استفاده کند. این مقادیر اندازه‌گیری شده به یک شبکه عصبی خورنده می‌شود تا اطلاعات مفیدی استخراج شده و مقادیر عددی در قالب ورودی‌های سیستم فازی فراهم شوند. شکل ۱۱-۲ یک سیستم فازی اشتراکی را نشان می‌دهد که در آن، فضای اندازه‌های $\{m_1, m_2, \dots, m_k\}$ به فضای ویژگی $\{x_1, x_2\}$ تصویر شده است (در قالب ورودی‌ها به سیستم فازی).

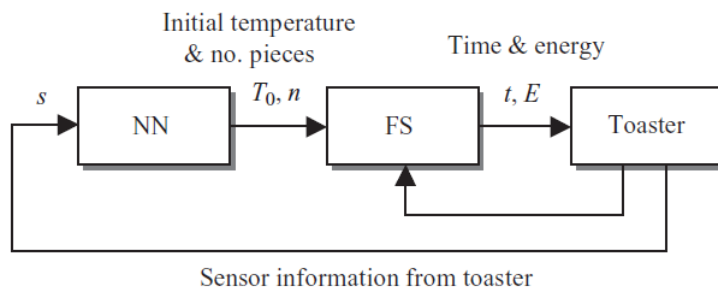


شکل ۱۱-۲: استخراج ویژگی‌ها در قالب ورودی یک سیستم فازی

1. Takagi, H.
2. Hayashi, I.
3. Tsoukalas L. H.
4. Uhrig, R. E.

۵-۲-۲ ترکیب آبشاری

می‌توان پیکربندی مشابهی با آنچه در شکل ۱۱-۲ آمده است را تصور نمود که شبکه عصبی در آن پیش‌برآورد^۱ پارامترهای ورودی سیستم فازی را انجام داده و پارامترهای سیستم توسط سیستم فازی کنترل می‌شود. چنین ترکیبی توسط توشیبا در دسته‌ای از برشته‌کن^۲ها به‌کارگرفته شده که شکل ۱۲-۲ نمایشی از آن است. شبکه عصبی با استفاده از اطلاعات حس‌گر، دمای اولیه و تعداد قطعات نان را برآورد می‌کند. سیستم فازی با استفاده از این اطلاعات و سایر ورودی‌های حس‌گر، زمان و انرژی بهینه مورد نیاز برای فرآیند برشته‌کردن را تعیین می‌کند.

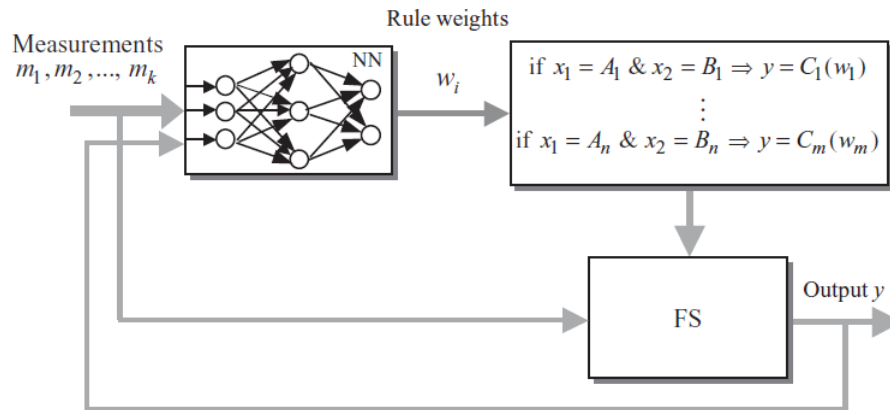


شکل ۱۲-۲: ترکیب آبشاری - همکاری شبکه عصبی با سیستم فازی

۶-۲-۲ شبکه عصبی برای تعیین پارامتر

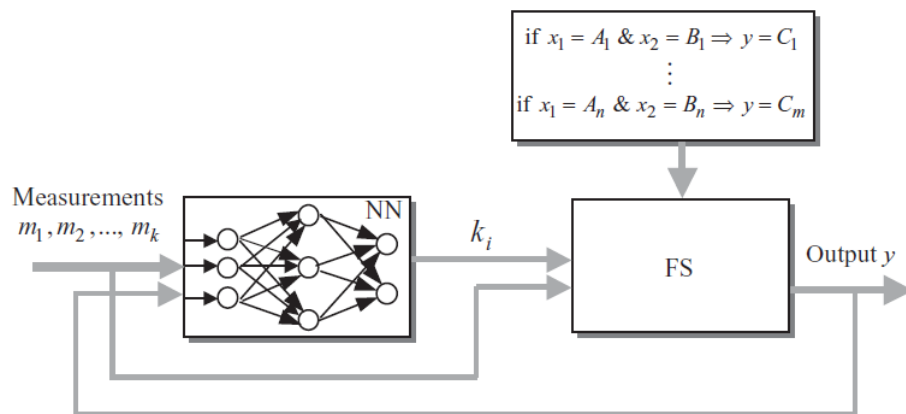
یک شبکه عصبی، پارامترها (پارامترهای مقیاس‌گذاری) را برای اصلاح توابع عضویت به‌صورت آنلاین (یعنی در خلال استفاده از سیستم فازی) تعیین می‌کند و همچنین می‌تواند وزن‌های قواعد را به‌صورت آنلاین یا آفلاین یاد بگیرد. شکل ۱۳-۲ این ترکیب مشارکتی شبکه عصبی و سیستم فازی را نشان می‌دهد که در آن، وزن‌های قواعد سیستم فازی (یعنی $\{w_1, w_2, \dots, w_m\}$) توسط شبکه عصبی تعیین می‌شوند. با استفاده از تابع خطای ناشی از اختلاف بین خروجی مطلوب و خروجی y حاصل از سیستم فازی، می‌توان شبکه عصبی را به‌صورت آفلاین آموزش داد.

1. Preestimation
2. Toaster



شکل ۲-۱۳: یادگیری مشارکتی وزن‌های قواعد فازی از یک شبکه عصبی

گاهی لازم است که پس از طراحی یک سیستم فازی، توابع عضویت را اصلاح و تعدیل نمود. این اصلاح را می‌توان با بهره‌گیری از توزیع داده‌ها مانند شکل ۲-۶ انجام داد. اصلاح توابع عضویت از طریق تنظیم یا یادگیری پارامترهای مقیاس‌گذاری شده سیستم فازی نیز ممکن است. در بسیاری از موارد، تنظیم عوامل مقیاس‌گذاری یا تعدیل و اصلاح کردن توابع عضویت می‌تواند به نتیجه مشابهی منجر شود. اصلاح توابع عضویت به چندین پارامتر نیاز دارد و لذا تنظیم عوامل مقیاس‌گذاری نسبت به اصلاح پارامترهای تابع عضویت کار بسیار ساده‌تری است. در شکل ۲-۱۴ ترکیب مشارکتی فازی عصبی به‌نمایش درآمده است که در آن، شبکه عصبی عوامل مقیاس‌گذاری را برای سیستم فازی تعیین می‌کند. با استفاده از تابع خطای ناشی از اختلاف بین خروجی مطلوب و خروجی y حاصل از سیستم فازی، می‌توان شبکه عصبی را به‌صورت آفلاین آموزش داد.



شکل ۲-۱۴: یادگیری پارامترهای مقیاس‌گذاری از یک شبکه عصبی

فصل ۳

سیستم‌های همزمان

سیستم‌های فازی و شبکه‌های عصبی می‌توانند بدون همکاری دو طرفه، به صورت موازی بر روی یک واحد کار کنند. این شیوه همکاری به مدل همزمان^۱ مشهور است. در این مدل، شبکه عصبی به‌طور پیوسته و مداوم به سیستم فازی کمک می‌کند (یا برعکس) تا پارامترهای مورد نیاز سیستم را تعیین کند. این کار به‌ویژه در شرایطی انجام می‌شود که پارامترهای مورد نیاز را نتوان به‌صورت مستقیم سنجید. این شیوه ترکیب، سیستم فازی یا عصبی را بهینه نمی‌کند بلکه موجب بهبود عملکرد سیستم کلی می‌شود. یادگیری که فقط در شبکه عصبی و سیستم فازی صورت می‌گیرد در خلال این مرحله بدون تغییر باقی می‌ماند.

به‌عنوان نمونه، دستگاه‌های ژاپنی تهویه هوا از یک سیستم فازی برای جلوگیری از یخ‌زدن کمپرسور در زمستان استفاده می‌کنند و از یک شبکه عصبی برای تخمین پارامترهای شاخص آسایش حرارتی - که با میانگین آرای پیش‌گویانه (PMV)^۲ شناخته می‌شود - بهره می‌گیرند. PMV را می‌توان تابعی از دمای اتاق، میانگین دمای تابشی، سرعت نسبی هوا، رطوبت نسبی، مقاومت گرمایی ناشی از پوشش افراد و میزان سوخت و ساز تعریف کرد. برخی از پارامترهای PMV مثل مقاومت گرمایی پوشش یا میزان سوخت و ساز را نمی‌توان با استفاده از حس‌گرها اندازه‌گیری کرد.

با استفاده از یک شبکه عصبی، می‌توان شاخص PMV را به کمک مجموعه‌ای از مقادیر اندازه‌گیری

1. Concurrent

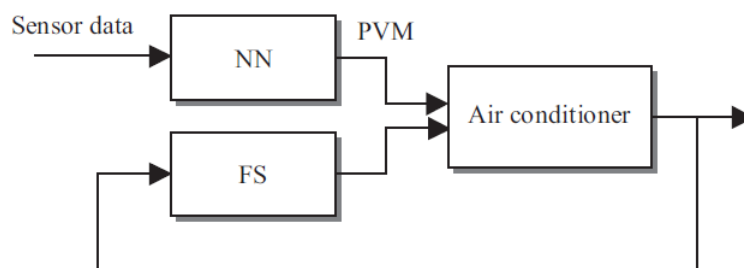
2. Predictive Mean Vote

شده نظیر دمای اتاق، اختلاف زمان دمای اتاق، دمای هوای خارجی، جریان هوا، دمای تنظیم‌شده و جهت جریان هوا برآورد کرد. داده‌های حس‌گر و PMV به‌عنوان ورودی‌ها و خروجی‌های شبکه عصبی استفاده می‌شوند که در قالب ترکیب‌های شش‌گانه زیر قابل تعریف‌اند:

$$\text{PMV} = \{\text{دمای اتاق، میانگین دمای تابشی، سرعت نسبی هوا، رطوبت نسبی، مقاومت گرمایی ناشی از پوشش افراد، میزان سوخت و ساز}\}$$

$$\text{داده‌های حس‌گر} = \{\text{دمای اتاق، دمای هوای خارجی، جریان هوا، دمای تنظیم‌شده، جهت جریان هوا}\}$$

شبکه عصبی یک نگاهت غیرخطی را از مقادیر اندازه‌گیری شده حس‌گر به شاخص PMV ایجاد می‌کند. روشی که توسط شرکت Matsushita Electric در دستگاه‌های تهویه‌کننده هوا به کار گرفته شده است، در شکل ۱-۳ به نمایش درآمده است.



شکل ۱-۳: فعالیت همزمان سیستم فازی و شبکه عصبی بر روی یک واحد

در برخی از منابع با نگاهی سخت‌گیرانه بیان می‌شود که سیستم‌های همزمان را نمی‌توان در زمره سیستم‌های فازی عصبی محسوب کرد زیرا شبکه عصبی همراه با سیستم فازی کار می‌کند. به عبارت دیگر، ورودی‌هایی که داخل سیستم فازی می‌شوند، پیش‌پردازش شده و سپس شبکه عصبی خروجی‌های این سیستم همزمان را پردازش می‌کند و یا بالعکس. در سیستم‌های فازی عصبی همزمان، نتایج به‌طور کامل قابل تفسیر نیستند و این موضوع به‌عنوان یک نقطه ضعف برای چنین سیستم‌هایی تلقی می‌شود.

موردکاوی

گونی و ساریکایا (۲۰۰۸) روشی را بر اساس سیستم فازی عصبی همزمان به منظور محاسبه همزمان فرکانس‌های تشدید حاصل از آنتن‌های میکرواستریپ مستطیلی، دایره‌ای و مثلثی ارائه کرده‌اند. این سیستم پیشنهادی شامل یک شبکه عصبی مصنوعی (ANN) و یک سیستم استنتاج فازی عصبی انطباقی (ANFIS) می‌شود که در آن، شبکه عصبی به طور پیوسته به سیستم فازی کمک می‌کند تا فرکانس تشدید محاسبه گردد و به طور برعکس، سیستم فازی نیز می‌تواند به شبکه عصبی برای تحقق همین منظور کمک کند. از این رو، در این پژوهش دو مدل سیستم عصبی فازی همزمان مورد بررسی قرار گرفته است. در مدل اول، فرکانس‌های تشدید ابتدا توسط ANN محاسبه شده و سپس اشتباهات و خطاهای محاسبات ANN توسط ANFIS اصلاح می‌گردد. در مدل دوم، برعکس مدل اول رفتار می‌شود؛ یعنی در ابتدا فرکانس‌های تشدید توسط ANFIS به دست آمده و سپس خطاهای محاسبات ANFIS با کمک ANN اصلاح می‌شود. جزئیات مربوط به ساختار ANN و ANFIS برای هر یک از مدل‌های یک و دو در متن مقاله این پژوهش درج شده است.

در نهایت، نشان داده است که نتایج حاصل از سیستم فازی عصبی همزمان برای فرکانس تشدید از آنتن‌های مذکور با نتایج تجربی موجود در ادبیات این موضوع در سطح بسیار خوبی دارای تطابق است.

Guney, K. & Sarikaya, N. (2008) Concurrent neuro-fuzzy systems for resonant frequency computation of rectangular, circular, and triangular microstrip antennas. Progress In Electromagnetics Research (PIER), 84, 253–277.

فصل ۴

سیستم‌های یکپارچه فازی عصبی

زمان‌برترین بخش در هر سیستم فازی، استنتاج با استفاده از پایگاه قواعد و فازی‌زدایی با کمک روش‌های مختلفی مثل مرکز ثقل است. ایده رویکرد هیبریدی بر تفسیر یک سیستم فازی در قالب یک شبکه عصبی مبتنی است. سیستم‌های فازی عصبی هیبریدی^۱ با عناوین سیستم‌های فازی عصبی یکپارچه^۲ یا سیستم‌های فازی عصبی آمیخته^۳ نیز شناخته می‌شود.

استراتژی که در این نوع سیستم‌ها بر پایه یک سیستم فازی عصبی اتخاذ می‌شود، دو محور دارد. یکم، جایگزینی پایگاه قواعد با شبکه‌های عصبی است که ساده‌سازی فرآیند استنتاج را به دنبال خواهد داشت و دوم، پیدا کردن پارامترهای سیستم فازی به وسیله روش‌های یادگیری حاصل از شبکه‌های عصبی است. یک راه معمول برای پیاده‌سازی یک الگوریتم یادگیری در سیستم‌های فازی، به‌کارگیری آن در ساختار خاصی شبیه به شبکه است تا بتوان یک الگوریتم یادگیری نظیر پس‌انتشار را برای آموزش سیستم استفاده کرد.

در اولین گونه از سیستم‌های فازی عصبی، سه نوع شبکه عصبی فازی وجود دارد که در آنها فقط جایگزین کردن پایگاه قواعد انجام می‌پذیرد و توابع عضویت ورودی و خروجی به همان صورت نگه داشته

-
1. Hybrid Neuro-Fuzzy Systems
 2. Integrated Neuro-Fuzzy Systems
 3. Fused Neuro-Fuzzy Systems

می‌شوند. به منظور حداقل کردن زمان فرآیند، روش‌های پیش‌فازی‌زدایی^۱ مختلفی به کار گرفته می‌شود. هیچ الگوریتم یادگیری در تنظیم توابع عضویت یا پارامترهای سیستم به کار نمی‌رود. در تنظیم و اصلاح پارامترهای این سه نوع سیستم از روش‌های ابتکاری یا سعی و خطا استفاده می‌شود. این دسته از سیستم‌های فازی عصبی در بخش نخست این فصل با عنوان شبکه‌های عصبی فازی با سیستم استنتاج فازی مورد بررسی قرار می‌گیرد.

در دومین گونه از سیستم فازی عصبی، الگوریتم‌های یادگیری نظیر پس‌انتشار یا آموزش ترکیبی به منظور تنظیم یا اصلاح پارامترهای سیستم به کار می‌رود. انواع مختلفی از این گونه سیستم‌های فازی عصبی در ادبیات این موضوع گزارش شده است که طی بخش‌های مختلف دیگر این فصل به آنها پرداخته می‌شود.

۱-۴ شبکه‌های عصبی فازی با سیستم استنتاج فازی

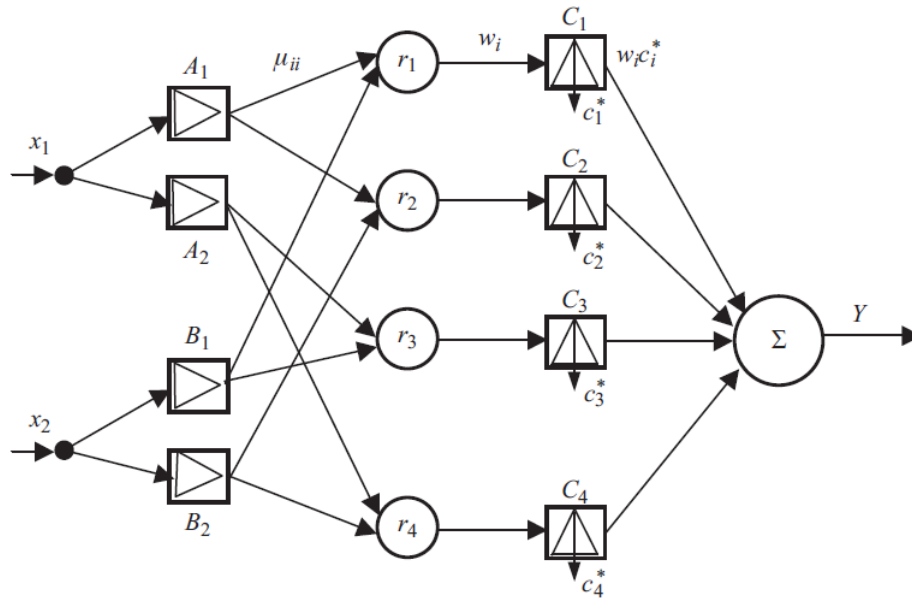
۱-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع مددانی

شبکه عصبی فازی که در این بخش مورد بحث قرار می‌گیرد، یک سیستم فازی از نوع مددانی است که پایگاه قواعد آن با یک شبکه عصبی جایگزین می‌شود. سیستم استنتاج فازی مددانی یک سیستم استنتاج بسیار متداول مبتنی بر منطق فازی است که در آن بخش مقدم قواعد و بخش تالی قواعد فازی هستند. ادامه، برای سادگی، یک سیستم ساده با دو ورودی و یک خروجی در شکل ۱-۴ نمایش داده شده است. این شبکه عصبی فازی شامل پنج لایه است که در ادامه تشریح می‌شود.

لایه ۱: گره‌های این لایه مشخص‌کننده توابع عضویت فازی $\{A_j, B_j\}$ هستند؛ x_1 و x_2 نیز دو ورودی را نشان می‌دهند. این گره‌ها، درجه عضویت ورودی‌ها را با عمل فازی‌سازی محاسبه می‌کنند:

$$\mu_{A_j}(x_1), \mu_{B_j}(x_2) \quad j = 1, 2 \quad (1-4)$$

در اینجا برای هر ورودی دو تابع عضویت وجود دارد.



شکل ۴-۱: سیستم عصبی فازی یکپارچه از نوع ممدانی

لایه ۲: هر گره در این لایه، یک قاعده از سیستم فازی را بیان می‌کند که در اینجا چهار قاعده با برجسب‌های r_1 ، r_2 ، r_3 و r_4 وجود دارند. هر گره، شدت برانگیختگی^۱ یک قاعده را براساس تعریف زیر تعیین می‌کند.

$$w_i = \Gamma\{\mu_{A_j}(x_1), \mu_{B_j}(x_2)\} \quad i = 1, 2, 3, 4; j = 1, 2 \quad (2-4)$$

تابع $\Gamma(\cdot)$ عمل استنتاج را با استفاده از قاعده ضرب^۲ یا قاعده \min تعیین می‌کند. به‌عنوان مثال، شدت برانگیختگی در رابطه ۴-۳ با استفاده از قاعده ضرب تعریف شده است.

$$w_i = \mu_{A_j}(x_1) * \mu_{B_j}(x_2) \quad i = 1, 2, 3, 4; j = 1, 2 \quad (3-4)$$

با نرمال کردن وزن‌ها می‌توان شدت برانگیختگی نسبی را برای هر یک از قواعد مجزا

1. Firing strength
2. Product rule

به دست آورد. نرمال کردن براساس رابطه ۴-۴ انجام می‌شود. اگرچه، وزن‌های نرمال شده در اینجا (ساختار مذکور) مورد استفاده قرار نگرفته‌اند.

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^4 w_i} \quad i = 1, 2, 3, 4 \quad (4-4)$$

لایه ۳: هر گره در این لایه، نماینده تابع عضویت فازی C_k برای خروجی است. توابع عضویت خروجی بر اساس رابطه زیر، پیش‌فازی‌زدایی می‌شوند.

$$c_i^* = \Psi(C_k) \quad k = 1, 2, 3, 4 \quad (5-4)$$

$\Psi(\cdot)$ تابع فازی‌زدایی است که بر روی توابع عضویت C_k اعمال می‌شود. مقادیر غیرفازی‌شده توابع عضویت تالی برای هر قاعده با c_i^* نشان داده شده است. انواع دیگری از روش‌های فازی‌زدایی را نیز می‌توان در اینجا به کار گرفت.

لایه ۴: تنها گره این لایه، خروجی نهایی را با استفاده از انبوه‌سازی همه مقادیر قواعد فعال به صورت زیر تولید می‌کند.

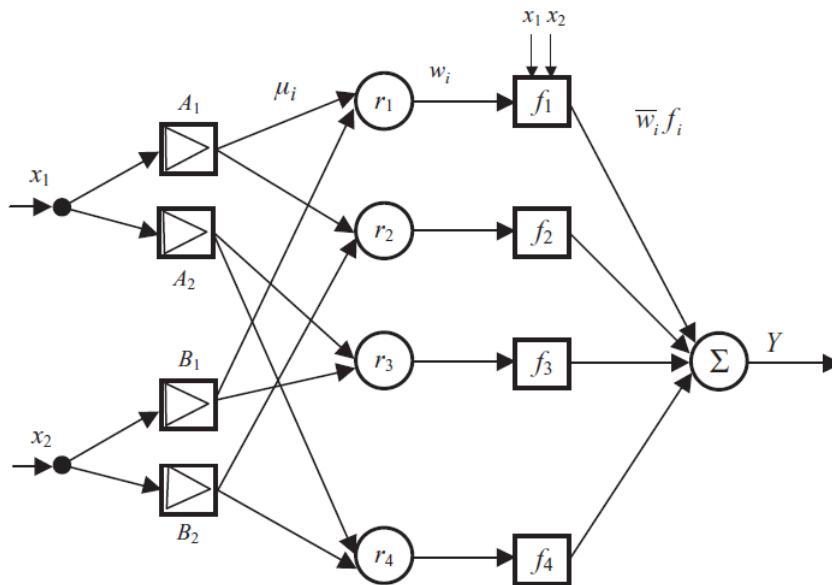
$$Y = \sum_i w_i \cdot c_i^* \quad i = 1, 2, 3, 4 \quad (6-4)$$

همان‌طور که در ابتدا اشاره شد، پایگاه قواعد در یک سیستم عصبی فازی ممدانی با یک ساختار شبیه به شبکه عصبی جایگزین می‌شود تا سازوکار استنتاج کردن با سهولت همراه گردد. بنابراین می‌توان لایه‌ای را که فرآیند نرمال‌سازی شدت برانگیختگی را انجام می‌دهد، بدون افت قابل توجهی در عملکرد سیستم حذف نمود.

۴-۱-۲ شبکه‌های مبتنی بر استنتاج فازی از نوع تاکاگی-سوگنو

شبکه عصبی فازی مورد بررسی در این بخش، یک سیستم از نوع سوگنو - یا همان سیستم فازی تاکاگی-سوگنو-کانگ - است. چنین سیستم‌های عصبی فازی اساساً سیستم‌های فازی از نوع سوگنو هستند که در آنها یک شبکه عصبی جایگزین پایگاه قواعد شده و به جای توابع عضویت فازی در تشریح توابع عضویت خروجی از توابع خطی استفاده می‌شود. برای سهولت در بیان موضوع، یک سیستم ساده با

دو ورودی و یک خروجی در شکل ۴-۲ به نمایش درآمده است. شبکه عصبی فازی نمایش داده شده در این شکل، شامل چهار لایه است که در ادامه تشریح می‌شوند.



شکل ۴-۲: سیستم عصبی فازی یکپارچه از نوع سوگنو

لایه ۱: هر گره در این لایه، گره‌ای با تابع عضویت فازی است و x_1 و x_2 نیز دو ورودی هستند. درجه عضویت ورودی‌ها توسط گره‌ها محاسبه می‌شود:

$$\mu_{A_j}(x_1), \mu_{B_j}(x_2) \quad j = 1, 2 \quad (7-4)$$

لایه ۲: گره‌های این لایه، گره‌های ثابتی برای بیان قواعد r_1, r_2, r_3, r_4 هستند. هر گره، شدت برانگیختگی یک قاعده را براساس تعریف زیر تعیین می‌کند.

$$w_i = \Gamma\{\mu_{A_j}(x_1), \mu_{B_j}(x_2)\} \quad i = 1, 2, 3, 4; j = 1, 2 \quad (8-4)$$

تابع $\Gamma(\cdot)$ عمل استنتاج را با استفاده از قاعده ضرب یا \min انجام می‌دهد. به‌عنوان مثال، $\Gamma(\cdot)$ با قاعده \min به صورت $\min\{\mu_{A_j}(x_1), \mu_{B_j}(x_2)\}$ درمی‌آید. با نرمال

کردن وزن‌ها می‌توان شدت برانگیختگی نسبی را برای قواعد مجزا به دست آورد. نرمال کردن را می‌توان براساس رابطه ۴-۴ انجام داد.

لایه ۳: هر گره در این لایه، گره خروجی است که یک تابع خطی را به فرم زیر نشان می‌دهد.

$$f_i = a_i x_1 + b_i x_2 + c_i \quad i = 1, 2, 3, 4 \quad (9-4)$$

در رابطه فوق، a_i ، b_i و c_i پارامترهای بخش تالی هر قاعده هستند. هر گره، مقدار موزون بخش تالی هر قاعده را به کمک رابطه زیر محاسبه می‌کند. اگرچه، این وزن‌های نرمال شده در ساختار شکل ۴-۲ استفاده نشده‌اند.

$$w_i \cdot f_i = w_i (a_i x_1 + b_i x_2 + c_i) \quad i = 1, 2, 3, 4 \quad (10-4)$$

پارامترهای a_i ، b_i و c_i را باید توسط یک روش ابتکاری یا سعی و خطا تخمین زد. اگر $a_i = 0$ و $b_i = 0$ در نظر گرفته شوند، در نتیجه $f_i = c_i$ یک مقدار ثابت خواهد شد. برخی از پژوهشگران، این حالت را سیستم تاکاگی-سوگنوی مرتبه صفر^۱ نامیده‌اند. c_i را می‌توان به طور دلخواه یا به وسیله سعی و خطا انتخاب کرد. اگر c_i با استفاده از مقدار پیش فرض فازی‌زدایی شده از توابع عضویت خروجی سیستم ممدانی (در شکل ۴-۱) مانند c_i^* انتخاب شود، آنگاه سیستم تاکاگی-سوگنو با سیستم ممدانی هم‌ارز خواهد شد.

لایه ۴: تنها گره این لایه، وظیفه تولید خروجی نهایی را با استفاده از انبوه‌سازی همه مقادیر قواعد فعال برعهده دارد:

$$Y = \sum_i w_i \cdot f_i \quad i = 1, 2, 3, 4 \quad (11-4)$$

اگر سیستم تاکاگی-سوگنو از مرتبه صفر در نظر گرفته شود، خروجی مطابق رابطه زیر تعریف می‌شود.

$$Y = \sum_i w_i \cdot c_i \quad i = 1, 2, 3, 4 \quad (12-4)$$

1. Zero-order

اگر c_i مانند c_i^* انتخاب شود، خروجی به صورت زیر تعریف خواهد شد.

$$Y = \sum_i w_i \cdot c_i^* \quad i = 1, 2, 3, 4 \quad (13-4)$$

بدین ترتیب، یک سیستم عصبی فازی ایجاد شده که در عمل با مدل فازی تاکاگی-سوگنو هم‌ارز است. برای یک سیستم استنتاج مددانی با ترکیب بیشینه (max) / کمینه (min) می‌توان یک سیستم عصبی فازی متناظر را ایجاد کرد، چنانچه تخمین‌های گسسته برای جایگزینی انتگرال‌ها در یک نظام فازی‌زدایی مرکزوار^۱ (یا انواع دیگر) مورد استفاده واقع شود.

۳-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع تسوکاموتو

سیستم‌های عصبی فازی از نوع تسوکاموتو اساساً سیستم‌هایی فازی از نوع تسوکاموتو هستند که پایگاه قواعد آنها با یک شبکه عصبی پیش‌خور جایگزین شده و توابع عضویت خروجی به وسیله توابع عضویت یکنوا^۲ توصیف می‌شوند. یک سیستم عصبی فازی تسوکاموتو با دو ورودی و یک خروجی در شکل ۳-۴ به نمایش درآمده که لایه‌های آن در ادامه تشریح می‌شوند.

لایه ۱: هر گره در این لایه، گره‌ای با تابع عضویت فازی است و x_1 و x_2 نیز دو ورودی هستند. درجه عضویت ورودی‌ها توسط گره‌ها محاسبه می‌شود:

$$\mu_{A_j}(x_1), \mu_{B_j}(x_2) \quad j = 1, 2 \quad (14-4)$$

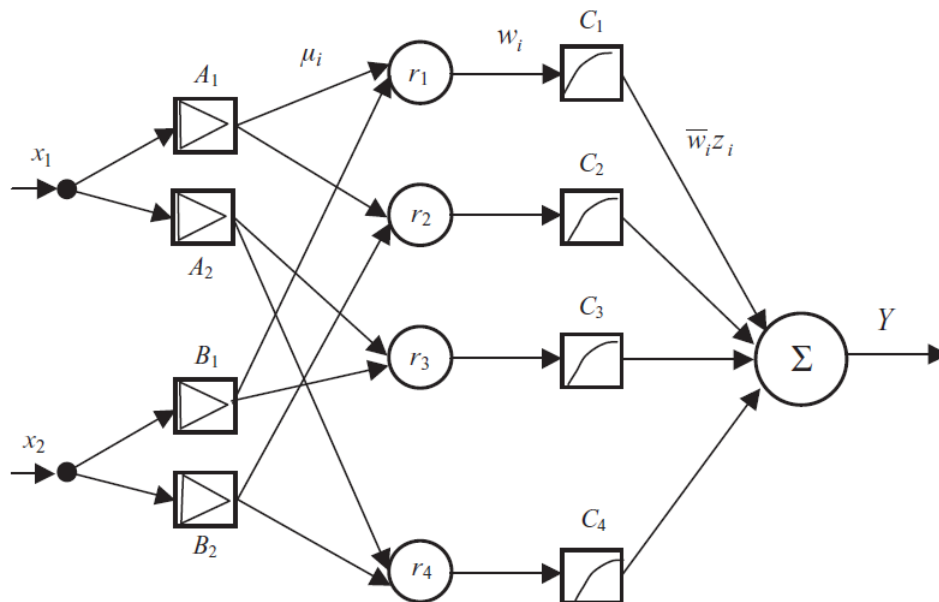
لایه ۲: گره‌های این لایه، گره‌های ثابتی هستند که قواعد r_1, r_2, r_3 و r_4 را مشخص می‌کنند. هر گره، شدت برانگیختگی یک قاعده را براساس تعریف زیر تعیین می‌کند.

$$w_i = \Gamma\{\mu_{A_j}(x_1), \mu_{B_j}(x_2)\} \quad i = 1, 2, 3, 4; j = 1, 2 \quad (15-4)$$

تابع $\Gamma(\cdot)$ عمل استنتاج را با استفاده از قاعده ضرب یا قاعده min تعیین می‌کند. به‌عنوان مثال، اگر $\Gamma(\cdot)$ قاعده ضرب را تعریف کند، به صورت $\{\mu_{A_j}(x_1) * \mu_{B_j}(x_2)\}$ درمی‌آید.

1. Centroid

2. Monotonic MFs



شکل ۴-۳: سیستم عصبی فازی یکپارچه از نوع تسوکاموتو

لایه ۳: هر گره در این لایه، یک تابع یکنوا C_k را برای توابع عضویت خروجی ارائه می‌کند. عمل فازی‌زدایی توابع عضویت خروجی به شکل زیر تبیین می‌شود.

$$z_i = \Psi(C_k) \quad i = 1, 2, 3, 4 \quad (16-4)$$

$\Psi(\cdot)$ یک تابع فازی‌زدایی است که برای توابع عضویت C_k به کار گرفته می‌شود؛ یعنی z_i نشان‌دهنده مقادیر غیرفازی‌شده توابع عضویت مقدم برای هر قاعده است.

لایه ۴: تنها گره این لایه، وظیفه تولید خروجی را با استفاده از انبوه‌سازی همه مقادیر قواعد فعال دارد:

$$Y = \sum_i w_i \cdot z_i \quad i = 1, 2, 3, 4 \quad (17-4)$$

سیستم عصبی فازی که در این بخش آشکار شد، در عمل با یک مدل فازی از نوع تسوکاموتو هم‌ارز

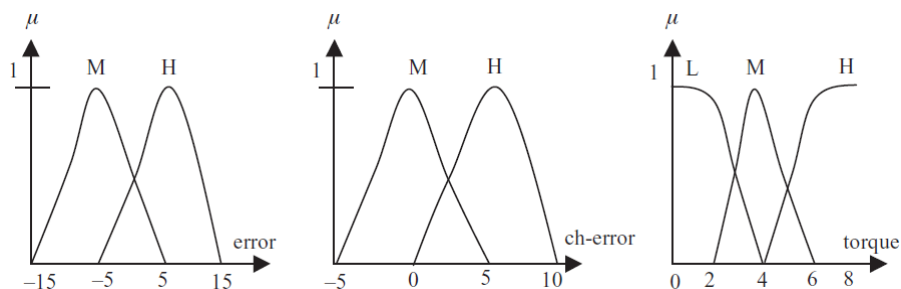
است. باید توجه نمود که این سه مدل (ممدانی، تاکاگی-سوگنو و تسوکاموتو) از هیچ الگوریتم یادگیری

استفاده نمی‌کنند بلکه در ورودی و توابع عضویت خروجی و سایر پارامترها بر روش‌های ابتکاری یا سعی و خطا متکی هستند.

با ترکیب بیشینه (max) / کمینه (min) در یک سیستم استنتاج فازی ممدانی، می‌توان یک سیستم متناظر را ایجاد کرد، چنانچه تخمین‌های گسسته به منظور جایگزینی انتگرال‌ها در یک نظام فازی‌زدایی مرکزوار مورد استفاده واقع شود.

مثال ۴-۱. ساخت یک سیستم عصبی فازی تاکاگی سوگنوی مرتبه صفر با استفاده از تعریف یک سیستم فازی ممدانی

یک سیستم فازی ممدانی را با دو ورودی خطا و تغییر خطا و یک گشتاور خروجی فرض می‌کنیم. برای هر ورودی، دو تابع عضویت و برای خروجی، سه تابع عضویت مطابق شکل ۴-۴ تعریف شده است. پایگاه قواعد نیز در جدول ۴-۱ ارائه شده است.

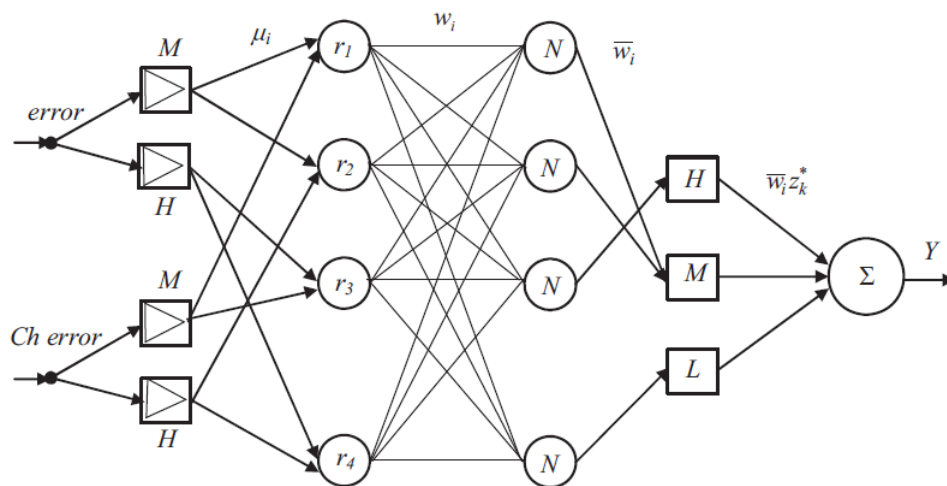


شکل ۴-۴: توابع عضویت، خطا، تغییر خطا و گشتاور

تغییر خطا		
H	M	خطا
$r_2: M$	$r_1: M$	M
$r_4: L$	$r_3: H$	H

جدول ۴-۱: پایگاه قواعد سیستم فازی ممدانی

این سیستم فازی با دو ورودی و یک خروجی در شکل ۴-۵ به نمایش درآمده است. توضیحات مربوط به لایه‌ها در ادامه می‌آید.



شکل ۴-۵: تبدیل سیستم فازی مددانی به سیستم عصبی فازی

لایه ۱: هر گره i در این لایه، گره‌ای ثابت با تابع عضویت مثلثی^۱ است؛ خطا و تغییر خطا نیز با x_1 و x_2 مشخص می‌شود. این گره‌ها، درجه عضویت ورودی‌ها را با فازی‌سازی محاسبه می‌کنند:

$$\mu_{A_j}(x_1), \mu_{B_j}(x_2) \quad (18-4)$$

که $j = 1, 2$ است.

لایه ۲: هر گره در این لایه، گره ثابتی است که چهار قاعده r_1, r_2, r_3 و r_4 را مشخص می‌سازد. هر گره، شدت برانگیختگی یک قاعده را تعیین کرده که در آن، $\Gamma(\cdot)$ عمل ضرب یا کمینه‌سازی را بر روی $\{\mu_{A_j}(x_1) * \mu_{B_j}(x_2)\}$ تعریف می‌کند:

$$w_i = \Gamma[\mu_{A_j}(x_1), \mu_{B_j}(x_2)] \quad i = 1, 2, 3, 4; j = 1, 2 \quad (19-4)$$

1. Triangular membership function

لایه ۳: هر گره لایه، گره‌های ثابتی با برجسب N هستند. شدت برانگیختگی نرمال شده به صورت زیر توسط هر گره محاسبه می‌شود.

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^4 w_i} \quad i = 1, 2, 3, 4 \quad (20-4)$$

لایه ۴: هر گره در این لایه، شامل یک مقدار ثابت پیش‌فازی‌زدایی شده از توابع عضویت نتیجه است که به صورت زیر تعریف می‌گردد.

$$z_k = \Psi(MF_k) \quad k = 1, 2, 3, 4 \quad (21-4)$$

که $\Psi(MF_k)$ یک عمل فازی‌زدایی مورد نظر بر روی توابع عضویت است و z_k مقادیر غیرفازی شده توابع عضویت را در بخش نتیجه سیستم فازی ممدانی نشان می‌دهد. هرگره، مقدار موزون بخش نتیجه را برای هر قاعده به طریقه زیر محاسبه می‌کند:

$$\bar{w}_i \cdot z_k \quad (22-4)$$

لایه ۵: گره یکتای این لایه، با انبوه‌سازی همه مقادیر قواعد فعال به تولید خروجی می‌پردازد:

$$Y = \sum \bar{w}_i \cdot z_k \quad (23-4)$$

۲-۴ سیستم فازی مبتنی بر شبکه عصبی (شبکه Pi-Sigma)

سیستم‌های عصبی فازی که در بخش‌های ۴-۱-۱ تا ۴-۱-۳ تشریح شدند، معمولاً پایگاه قواعد را با یک شبکه عصبی جایگزین کرده و یک قاعده ضرب یا کمینه‌سازی را برای استنتاج به کار می‌برند. به نظر می‌رسد که انعطاف‌پذیری سیستم فازی تاکاگی-سوگنوا از سیستم فازی ممدانی بیشتر است. با این حال، همچنان دو نقطه ضعف وجود دارد. اولاً اینکه شناسایی سیستم فازی کاری کم‌اهمیت و جزئی

نیست، به همین دلیل طی پیاده‌سازی در سیستم‌های بی‌درنگ^۱ دشواری‌هایی ایجاد می‌شود^۲. از سوی دیگر، توابع عضویت فقط به توابع خطی قطعه‌ای^۳ محدود نمی‌شود، اما بخش تالی به صورت خطی فرض می‌گردد. این مشکل برطرف نمی‌شود تا زمانی که شبکه‌های عصبی به منظور افزودن توانایی یادگیری مناسب و ظرفیت نگاشت غیرخطی با سیستم‌های فازی ترکیب شود. جین^۴ و همکاران (۱۹۹۵) یک سیستم فازی عصبی هیبریدی را پیشنهاد کردند که در یک قسمت از آن، شدت برانگیختگی قواعد توسط بخش مقدم سیستم فازی تاکاگی-سوگنو محاسبه شده و در قسمت دیگر، خروجی بخش تالی به کمک یک شبکه عصبی پای-سیگما (PSNN)^۵ تخمین زده می‌شود. شبکه‌های عصبی پای-سیگما، شبکه‌های عصبی سطح بالا هستند که از ضرب مؤلفه‌های ورودی جمع استفاده می‌کند. به همین دلیل، چنین شبکه‌هایی سازوکار یادگیری سریعی دارند و از افزایش احتمالی وزن و ورودی‌ها جلوگیری می‌کنند. PSNN خروجی را در قالب ضرب واحدهای مجموع‌یاب^۶ به لایه خروجی انتقال می‌دهد و از واحدهای وزن کمتری برای پردازش گره‌ها استفاده می‌کند. این کار موجب می‌شود که کارایی و دقت بیشتری نسبت به سایر شبکه‌های عصبی ایجاد گردد. در خلال یادگیری، وزن‌های بین لایه ورودی و لایه پنهان قابل تغییر هستند درحالی‌که یک وزن بدون تغییر از لایه پنهان به لایه خروجی قرار داده شده است تا پیچیدگی شبکه کاهش یافته و پیاده‌سازی مدل ساده‌تر شود. در ساختار این سیستم فازی مبتنی بر شبکه عصبی^۷، یک نرون فازی به کار می‌رود که چند عمل اصلی فازی (مثل عمل‌های min و max) را انجام می‌دهد. این ساختار در شکل ۴-۶ به نمایش درآمده است. ساختار شبکه عصبی پای-سیگمای فازی گسترشی از مدل فازی تاکاگی-سوگنو (قسمت پایین شکل) است که در آن، یک مدل شبکه عصبی لایه‌ای (بخش بالای شکل) خروجی‌های تالی را برآورد می‌کند. این دو مدل با کمک مجموعه‌ای از گره‌های

1. Real-time

۲. در سیستم‌های بی‌درنگ، زمان عامل کلیدی است. برآورد زمان در این سیستم‌ها، باید به‌طور دقیق انجام گردد و کارها دقیقاً در فرجه زمانی موردنظر انجام شوند؛ در غیر این صورت سیستم دچار اختلال خواهد شد. سیستم‌هایی که آزمایش‌های علمی را کنترل می‌کنند، سیستم‌های تصویربرداری پزشکی، سیستم‌های کنترل صنعتی، کنترل‌کننده‌های لوازم خانگی و سیستم‌های جنگ‌افزاری، نمونه‌هایی از سیستم‌های بی‌درنگ هستند.

3. Piecewise linear functions

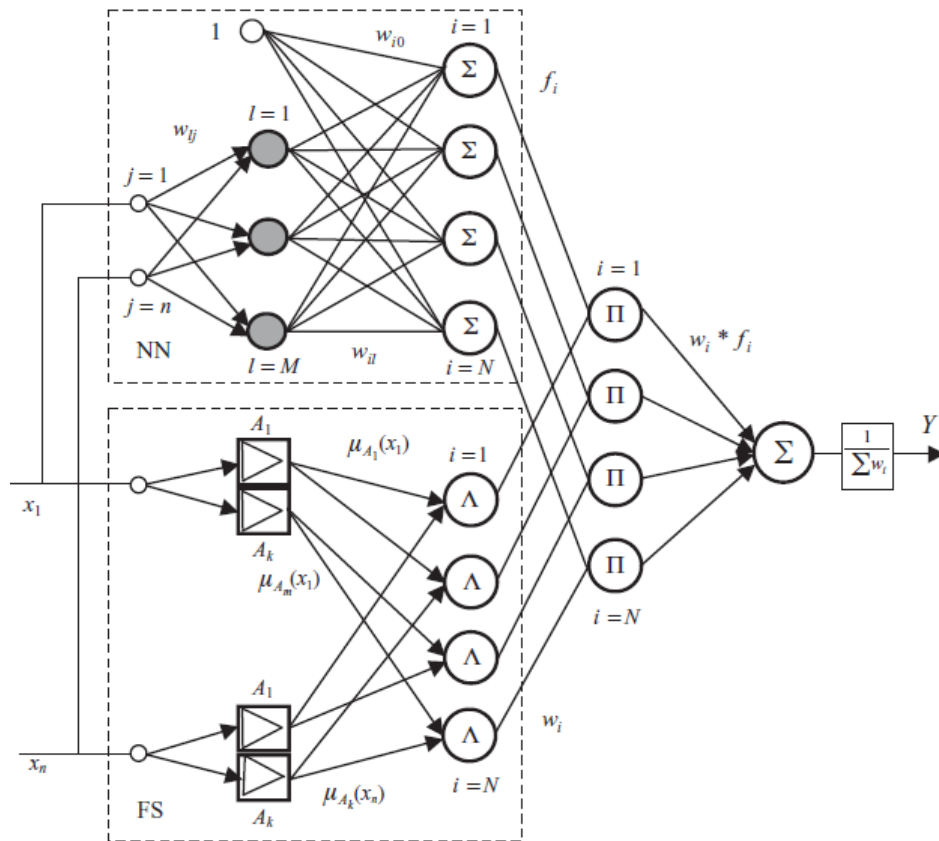
4. Jin, Y.

5. Pi-Sigma Neural Network

6. Summation units

7. Neural Network-Based Fuzzy System

ضرب (گره‌های Π یا π) ترکیب شده‌اند.



شکل ۴-۶: شبکه عصبی فازی یکپارچه پای-سیگما

قسمت سیستم فازی این مدل، ارائه‌کننده بخش مقدم از سیستم فازی تاکاگی-سوگنو است و شدت برانگیختگی قواعد (w_i) را مطابق روند زیر محاسبه می‌کند.

لایه ۱: همه گره‌های این لایه، گره‌هایی با عبارات زبانی فازی (تابع عضویت) $A_k(x_j)$ هستند و $(k = 1, 2, \dots, m)$ هستند و x_j ($j = 1, 2, \dots, n$) نشان‌گر ورودی‌هاست. درجه عضویت ورودی‌ها ($\mu_{A_k}(x_j)$) توسط این گره‌ها محاسبه می‌شود.

لایه ۲: هر گره در این لایه، یک گره قاعده را مشخص کرده و شدت برانگیختگی یک

قاعده در هر گره به شیوه زیر تعیین می‌گردد:

$$w_i = \Gamma(\mu_{A_k}(x_1), \dots, \mu_{A_k}(x_n)) \quad i = 1, 2, \dots, N \quad (24-4)$$

$\Gamma(\cdot)$ یک نرون فازی را مشخص می‌کند که عمل کمینه‌سازی یا ضرب را انجام می‌دهد.

تعداد قوانین سیستم فازی با N نشان داده می‌شود.

قسمت شبکه عصبی در این مدل، بخش تالی از سیستم تاکاگی-سوگنو را ارائه کرده و خروجی (f_i) را بر اساس فرآیند زیر محاسبه می‌کند.

لایه ۱: همه گره‌های این لایه، نرون‌هایی با تابع فعال‌سازی غیرخطی^۱ هستند. خروجی این لایه به وسیله رابطه زیر حاصل می‌شود.

$$o_l = g\left(\sum_{j=1}^n w_{lj}x_j\right) \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, m \quad (25-4)$$

در رابطه فوق، وزن‌های اتصال بین ورودی‌ها و لایه یکم با w_{lj} مشخص شده است. $g(\cdot)$ نیز یک تابع غیرخطی از نوع سیگموئید^۲ است.

لایه ۲: همه گره‌های این لایه، نرون‌های مجموع‌یاب خطی^۳ (sigma-نرون) هستند. خروجی این لایه به صورت زیر محاسبه می‌شود.

$$f_i = w_{i0} + \sum_{l=1}^N w_{il}o_l = w_{i0} + \sum_{l=1}^N \left\{ w_{il} g\left(\sum_{j=1}^n w_{lj}x_j\right) \right\} \quad (26-4)$$

در رابطه فوق، وزن‌های اتصال بین لایه یکم و دوم با w_{il} مشخص شده و w_{i0} نیز نقش بایاس را برای نرون‌های دومین لایه ایفا می‌کند.

لایه ۳: این یک لایه مشترک برای هر دو بخش از سیستم عصبی فازی است. هر گره در این لایه، یک pi-نرون (II) است و حاصل ضرب w_i و f_i (یعنی $w_i \cdot f_i$) را محاسبه می‌کند.

1. Nonlinear activation function

2. Sigmoidal-type nonlinear function

3. Linear summation neuron

لایه ۴: این نیز یک لایه مشترک برای هر دو بخش از سیستم عصبی فازی است و مجموع ضرب‌های انجام‌شده را به وسیله عبارت $\frac{1}{\sum_{i=1}^N w_i}$ به دست می‌آورد تا طبق رابطه زیر به خروجی نهایی سیستم (Y) منجر شود.

$$Y = \frac{\sum_{i=1}^N w_i \cdot f_i}{\sum_{i=1}^N w_i} = \sum_{i=1}^N \bar{w}_i \cdot f_i \quad (27-4)$$

این شبکه عصبی فازی یکپارچه با یک سیستم فازی تاکاگی-سوگنو هم‌ارز است که در آن توابع مقدم خطی به توابع غیرخطی گسترش یافته و پارامترها توسط شبکه عصبی نمایش داده شده در بخش بالای شکل ۴-۶ برآورده می‌شود.

به منظور تنظیم و اصلاح پارامترهای تالی و پارامترهای تابع عضویت، باید الگوریتم پس‌انتشار خطا را در قالب الگوریتم گرادیان که به توابع مشتق‌پذیر نیاز دارد، گسترش داد. بنابراین عامل کمینه‌سازی به تغییر شکل نیاز خواهد داشت. فرض کنید که خروجی مطلوب شبکه پای-سیگما با Y_d نشان داده شود. در این صورت، تابع خطا به شکل زیر تعریف می‌گردد.

$$E = \frac{1}{2} (Y - Y_d)^2 \quad (28-4)$$

بر اساس اصل پس‌انتشار خطا، خطای تعمیم‌یافته برای گره خروجی نهایی (Σ) برابر با $(Y_d - Y) / \sum w_i$ است. از آنجایی که تالی (f_i) و ارزش درستی کلی فرض‌ها (w_i) برای استلزام \dot{w}_i در گره ضرب (II) انجام می‌شود، خطا نمی‌تواند مستقیماً پس‌انتشار یابد. با این حال با در نظر گرفتن w_i به عنوان وزن اتصال گره نتیجه (Σ) و گره خروجی نهایی (Σ) می‌توان گره ضرب را حذف نمود. به این ترتیب خطای تعمیم‌یافته برای \dot{w}_i گره نتیجه (Σ) به طور تخمینی به صورت زیر حاصل می‌شود.

$$\delta_i^1 = (Y_d - Y) w_i / \sum_{i=1}^N w_i \quad (29-4)$$

به طور مشابه، خطای تعمیم‌یافته برای هر گره فازی به صورت زیر به دست می‌آید.

$$\delta_i^2 = (Y_d - Y) f_i / \sum_{i=1}^N w_i \quad (30-4)$$

بنابراین پارامترهای نتیجه براساس روابط زیر اصلاح می‌شوند:

$$\Delta w_i^0 = \eta \delta_i^1 \quad (31-4)$$

$$\Delta w_l^i = \eta x_j \sum_{l=1}^N \left\{ w_{il} g \left(\sum_{j=1}^n w_{lj} x_j \right) \right\} \delta_i^1 \quad (32-4)$$

$$\Delta w_i^l = \eta x_j g \left(\sum_{j=1}^n w_{lj} x_j \right) \delta_i^1 \quad (33-4)$$

در روابط فوق، η یک نرخ یادگیری مثبت است و سایر پارامترها نیز مطابق تعاریف قبلی هستند. تابع عضویتی که در قسمت فازی مورد استفاده قرار گرفت، یک تابع عضویت گاوسی^۱ است که به صورت زیر تعریف می‌شود.

$$\mu_{A_k}(x_j) = \exp \left[- \frac{(x_j - m_k^j)^2}{\sigma_k^j} \right] \quad (34-4)$$

در رابطه فوق، m_k^j و σ_k^j به ترتیب برابر با مرکز و عرض تابع عضویت $A_k(x_j)$ است. پارامترهای عضویت را می‌توان برحسب قواعد زیر تعدیل و اصلاح نمود.

$$\Delta m_k^j = \begin{cases} 2\eta (x_j - m_k^j) w_i \delta_i^1 / \sigma_k^j & \text{اگر } A_k(x_j) \text{ کمینه} \\ 0 & \text{در غیر این صورت} \end{cases} \quad (35-4)$$

$$\Delta \sigma_k^j = \begin{cases} \eta (x_j - m_k^j)^2 w_i \delta_i^1 / \sigma_k^j & \text{اگر } A_k(x_j) \text{ کمینه} \\ 0 & \text{در غیر این صورت} \end{cases} \quad (36-4)$$

η یک نرخ یادگیری مثبت است و سایر پارامترها مطابق تعاریف قبلی هستند. جزئیات مفصلی را درباره عبارات و قواعد مختلف به روزرسانی، می‌توان در پژوهش جین^۲ و جیانگ^۳ (۱۹۹۹) مشاهده نمود.

1. Gaussian MF

2. Jin, Y.

3. Jiang, J.

۳-۴ کنترل هوشمند مبتنی بر استدلال تقریبی

۱-۳-۴ مدل ARIC

کنترل هوشمند مبتنی بر استدلال تقریبی (ARIC)^۱ توسط برنجی^۲ (۱۹۹۲) معرفی شد. این ساختار، یک کنترل‌کننده فازی است که دو شبکه عصبی خاص را شامل می‌شود: شبکه ارزیابی عمل-حالت (AEN)^۳ و شبکه انتخاب عمل (ASN)^۴. AEN ارزیابی ASN را انجام داده و به کنترل‌کننده اصلی اطلاع می‌دهد. ASN نمایش مستقیم کنترل‌کننده فازی است. گره‌های ASN و AEN همراه با ساختار ARIC در شکل ۴-۷ به نمایش درآمده است.

شبکه ارزیابی عمل-حالت (AEN)

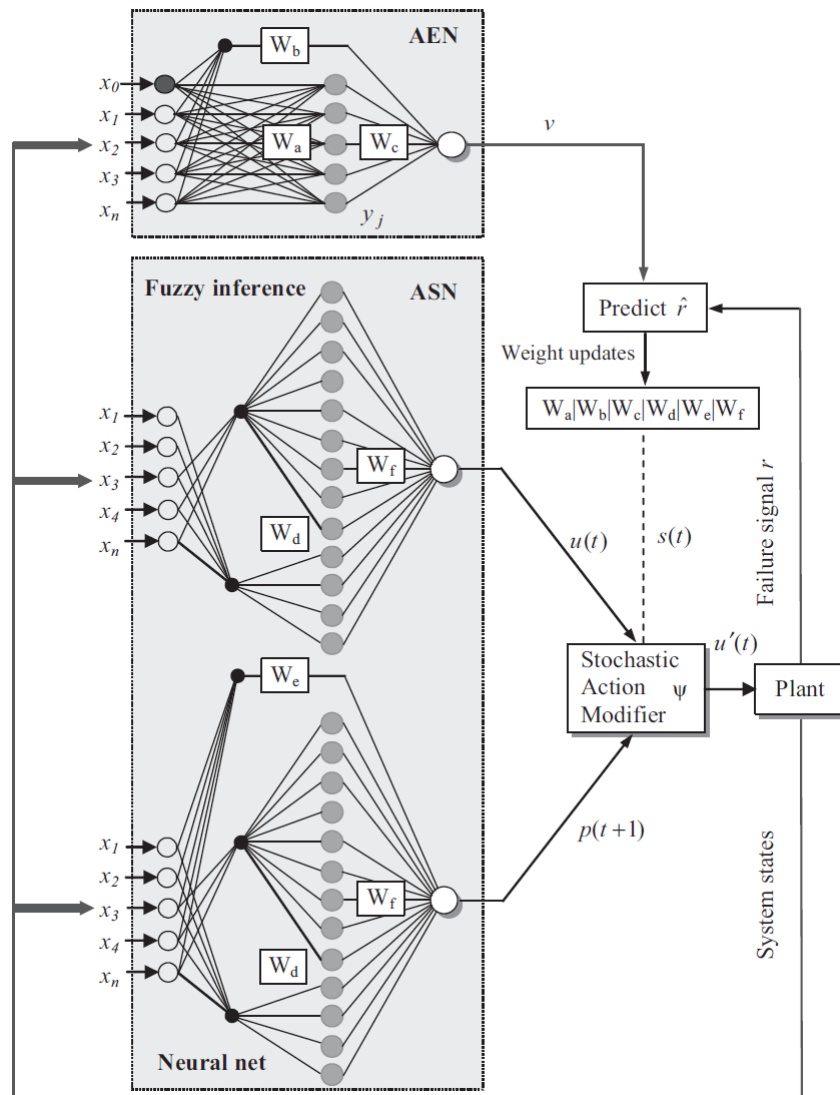
AEN یک شبکه سه لایه است. لایه ورودی شامل n گره ورودی است و اطلاعات را در حالت‌های سیستم دریافت می‌کند که باید متغیرهای وضعیت آن (x_i) و سیگنال شکست (r) کنترل شود. لایه پنهان شامل h_j گره پنهان است. ورودی‌های محیط (x_1, \dots, x_n) و یک بایاس (x_0) به تمامی گره‌های پنهان متصل هستند. وزن‌های اتصال بین ورودی و گره‌های پنهان با W_{aji} $(j = 1, 2, \dots, m$ و $i = 0, 1, 2, \dots, n)$ نشان داده شده است. خروجی‌های گره پنهان (h_j) به صورت زیر تعریف می‌شود.

$$h_j(t+1) = g\left(\sum_{i=0}^n W_{aji}(t)x_i(t+1)\right) \quad (۳۷-۴)$$

در رابطه فوق، $g(\cdot)$ تابع فعال‌سازی سیگموئید است.

لایه خروجی شامل یک گره تنه‌است و ورودی‌ها را از $n+1$ گره لایه ورودی (x_i) و m گره لایه

1. Approximate Reasoning-based Intelligent Control
2. Berenji, H. R.
3. Action-state Evaluation Network
4. Action Selection Network



شکل ۴-۷: ساختار ARIC شامل AEN و ASN

پنهان (h_j) دریافت می‌کند. خروجی v به صورت زیر تعریف می‌شود.

$$v(t+1) = \sum_{i=0}^n Wb_i(t)x_i(t+1) + \sum_{j=1}^m Wc_j(t)h_j(t+1) \quad (38-4)$$

در رابطه بالا، Wb_i وزن‌های اتصال بین لایه ورودی و لایه خروجی بوده و Wc_i نیز وزن‌های بین لایه

پنهان و لایه خروجی است. خروجی v پیش‌بینی تقویت‌ی را نشان می‌دهد. AEN نقش یک عنصر نقدکننده تطبیقی را ایفا کرده و به‌طور مداوم، تقویت‌های آتی را برای وضعیت داده شده پیش‌بینی می‌کند. اقدام معرفی شده توسط ASN در قالب تابعی از سیگنال شکست و تغییر در ارزیابی حالت مبتنی بر حالت سیستم در زمان $t + 1$ توسط AEN ارزیابی می‌شود.

$$\hat{r}(t+1) = \begin{cases} 0 & \text{حالت شروع} \\ r(t+1) - v(t) & \text{حالت شکست} \\ r(t+1) + \alpha v(t+1) - v(t) & \text{حالت‌های دیگر} \end{cases} \quad (4-39)$$

در رابطه فوق، $0 \leq \alpha \leq 1$ نرخ تعدیل است. به‌طور کلی برای آموزش وزن‌های Wb و Wc از الگوریتم AHC متعلق به ساتن^۱ (۱۹۸۴ و ۱۹۸۸) استفاده می‌شود و الگوریتم پس‌انتشار نیز برای آموزش وزن‌های Wa به‌کار می‌رود. وزن‌های Wb و Wc مطابق روندهای زیر به‌روز می‌شوند.

$$Wb_i(t+1) = Wb_i(t) + \beta \hat{r}(t+1)x_i(t) \quad (4-40)$$

$$Wc_j(t+1) = Wc_j(t) + \beta \hat{r}(t+1)h_j(t) \quad (4-41)$$

$\beta > 0$ ثابت یادگیری است و $\hat{r}(t+1)$ تقویت داخلی را در زمان $(t+1)$ نشان می‌دهد. تابع به‌روزرسانی وزن برای Wa ، بریک نسخه اصلاح شده از الگوریتم پس‌انتشار خطا استوار است که در آن، اندازه‌گیری مستقیمی از امکان خطا وجود ندارد. \hat{r} به‌عنوان شاخص خطا به‌کار رفته و وزن‌های Wa برحسب رابطه زیر به‌روز می‌شوند.

$$Wa_{ij}(t+1) = Wa_{ij}(t) + \beta \hat{r}(t+1)h_j(t)[1 - h_j(t)]\text{sgn}[Wc_j(t)]x_i(t) \quad (4-42)$$

یک تغییر مثبت در ارزیابی‌های حالت (یعنی \hat{r} مثبت)، موجب افزایش وزن‌ها خواهد شد و \hat{r} منفی، کاهش وزن‌ها را به‌دنبال خواهد داشت.

1. Sutton, R. S.

شبکه انتخاب عمل (ASN)

ASN دو شبکه را دربردارد. شبکه اول، یک کنترل‌کننده فازی شامل فازی‌ساز، منطق تصمیم‌گیری و پایگاه قواعد و فازی‌زدا است. این کنترل‌کننده فازی، توسط یک شبکه عصبی دو لایه‌ای مدل می‌شود. لایه ورودی، فازی‌سازی متغیرهای ورودی $\{x_1, \dots, x_n\}$ را در برابر برجسب‌هایی که توابع عضویت مشخص کرده‌اند، انجام می‌دهد و درجه توابع عضویت $\{\mu_{i1}(x_i), \dots, \mu_{im}(x_i)\}$ را با فرض $i = 1, 2, \dots, n$ تعیین می‌کند که m تعداد توابع عضویت را برای هر ورودی نشان می‌دهد. لایه پنهان، با قوانین کنترل‌کننده فازی متناظر است و منطق تصمیم‌گیری را شامل می‌شود. در ساختار شکل ۴-۷، مقدم‌ها دیده نمی‌شوند. ورودی‌های نرون، پیش‌شرط‌های یک قاعده و خروجی نرون تالی آن را تشکیل می‌دهند. در اینجا، یک سیستم کنترل چندورودی - تک‌خروجی (MISO)^۱ در نظر گرفته می‌شود. w_j را به‌عنوان درجه‌ای که قاعده j توسط یک متغیر حالت ورودی x_i در X برانگیخته (فعال) می‌شود قرار می‌دهیم:

$$w_j = \min \{Wd_{j1}\mu_{1j}(x_1), \dots, Wd_{ji}\mu_{ij}(x_i), \dots, Wd_{jn}\mu_{nj}(x_n)\} \quad (43-4)$$

که $\mu_{ij}(x_i)$ درجه عضویت ورودی x_i را در یک مجموعه فازی نشان می‌دهد که معرف برجسب استفاده شده در اولین پیش‌شرط قاعده j است. Wd_{ji} وزن ارتباط j امین گره قاعده با i امین گره ورودی است یا به عبارت دیگر اتصال بین نرون i و نرون j (از لایه ورودی به لایه پنهان) را نشان می‌دهد. نتیجه اعمال w_j بر بخش تالی قاعده j با m_j بیان شده به صورت زیر محاسبه می‌شود:

$$m_j = \mu_{C_j}^{-1}(w_j) \quad j = 1, 2, \dots, M \quad (44-4)$$

M تعداد قواعد (یا تعداد گره‌ها در لایه پنهان) را نشان داده و μ_{C_j} تابع عضویت یکنوا را برای بخش تالی قاعده j نمایش می‌دهد. معکوس این تابع عضویت $(\mu_{C_j}^{-1})$ یک عمل فازی‌زدایی مناسب است که در هر قاعده قابل اجراست. لایه خروجی، فرآیند فازی‌زدایی و ترکیب بخش تالی را برای هر قاعده با استفاده از روش مرکز سطح (COA)^۲ انجام می‌دهد. بدین ترتیب می‌توان مقدار اقدام کنترلی $u(t)$ را با

1. Multi-Input Single-Output

2. Centre of Area

در نظر گرفتن توابع عضویت دلخواه مطابق رابطه زیر به دست آورد.

$$u(t) = \frac{\sum_{j=1}^M W f_j \times m_j \times w_j}{\sum_{j=1}^M w_j \times W f_j} \quad (45-4)$$

در این رابطه، $W f_j$ وزن‌های اتصال بین گره‌های لایه پنهان به گره‌های لایه خروجی است. این ساختار اجازه می‌دهد که قواعد موجود در پایگاه دانش کنترلی به سادگی در قالب شبکه انتخاب عمل (ASN) ترجمه شوند. تعدیل وزن‌ها در این شبکه، موجب تنظیم دقیق قواعد کنترلی خواهد شد.

شبکه دوم ASN شبکه عصبی محاسبه‌گر یک مقدار احتمالی p است که نشان‌دهنده معیاری برای اطمینان مربوط به عمل انتخاب شده است. این معیار احتمالی به منظور اصلاح اقدام کنترلی $u(t)$ مورد استفاده قرار می‌گیرد. خروجی گره‌های لایه پنهان برابر است با:

$$z_j(t) = g\left(\sum_{i=1}^n W d_{ji}(t) x_i(t+1)\right) \quad j = 1, 2, \dots, H \quad (46-4)$$

در رابطه فوق، $g(\cdot)$ تابع سیگموئید است و H تعداد گره‌های لایه پنهان است. گره‌های لایه خروجی، ورودی‌ها را از لایه پنهان و لایه ورودی دریافت کرده و سپس احتمال $p(t+1)$ را محاسبه می‌کند:

$$p(t+1) = \sum_{i=1}^n W e_i(t) x_i(t+1) + \sum_{j=1}^H W f_j(t) z_j(t+1) \quad (47-4)$$

$W e_i$ وزن‌های اتصال بین لایه ورودی و لایه خروجی را نشان می‌دهد و $p(t+1)$ نیز به منظور اصلاح و تعدیل اقدام $u(t)$ کنترل‌کننده فازی استفاده می‌شود. تعدیل تصادفی اقدام به صورت زیر قابل انجام است.

$$u'(t) = \Psi[u(t), p(t+1)] \quad (48-4)$$

$[\cdot]$ یک تابع تعدیل تصادفی مبتنی بر احتمال $p(t+1)$ است. در ادامه، می‌توان از یک معیار نهایی برای تعدیل تصادفی عمل استفاده کرده که بر اساس مقایسه $u(t)$ و $u'(t)$ به دست می‌آید:

$$s(t) = k[u(t), u'(t)] \quad (49-4)$$

تابع $k[.]$ باید با توجه به کاربرد موردنظر انتخاب شود. $s(t)$ به منظور به‌روزرسانی وزن‌های ASN به‌کار می‌رود. AHC و الگوریتم پس‌انتشار برای آموزش وزن‌ها مورد استفاده واقع می‌شود. تغییرات وزنی برای We و Wf متناسب با \hat{r} ، s و خروجی متناظر است. وزن‌ها مطابق رابطه‌های زیر به‌روز می‌شوند:

$$We_i(t+1) = We_i(t) + \eta \hat{r}(t+1)s(t)x_i(t) \quad (50-4)$$

$$Wf_j(t+1) = Wf_j(t) + \eta \hat{r}(t+1)s(t)z_j(t) \quad (51-4)$$

به‌روزرسانی وزن Wd مبتنی بر یک نسخه اصلاح‌شده از الگوریتم پس‌انتشار است که در آن از \hat{r} با $s(t)$ به‌عنوان شاخص خطا استفاده می‌شود. با پس‌انتشار این خطا، وزن‌های Wd به این صورت به‌روز می‌شوند:

$$(52-4)$$

$$Wd_{ji}(t+1) = Wd_{ji}(t) + \eta \hat{r}(t+1)z_j(t)[1 - z_j(t)]\text{sgn}[Wf_j(t)]s(t)x_i(t)$$

$\eta > 0$ یک ثابت یادگیری است. تعداد گره‌های لایه پنهان یا لایه خروجی در AEN و ASN یکسان نیست. ساختار ARIC که در بالا تشریح شد، اجازه می‌دهد که قواعد پایگاه دانش کنترلی به ASN ترجمه شوند. همچنین ثابت شده است که این تغییر وزن‌های شبکه منجر به تنظیم دقیق قواعد کنترل خواهد شد.

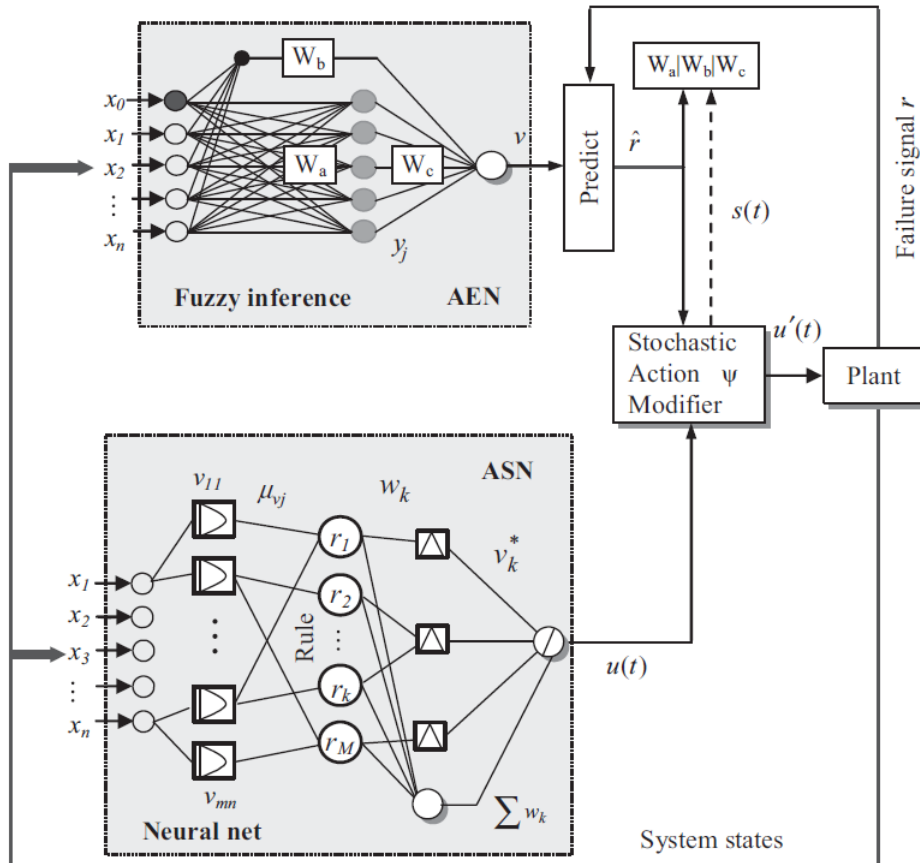
۲-۳-۴ مدل GARIC

کنترل هوشمند مبتنی بر استدلال تقریبی تعمیم‌یافته (GARIC)^۱ گسترشی از ARIC است که توسط برنجی^۲ و کدکار^۳ (۱۹۹۲، ۱۹۹۳) ارائه شد. مشابه ARIC، GARIC نیز شامل یک شبکه ارزیابی (AEN) و یک شبکه عمل (ASN) است. ساختار و الگوریتم یادگیری AEN در GARIC دقیقاً مشابه ARIC است که در بخش ۱-۳-۴ تشریح شد. اما ASN یک شبکه پیش‌خور پنج لایه‌ای اصلاح‌شده است که در آن، تمام وزن‌های اتصال بین لایه‌ها واحد هستند. مدل GARIC پنج لایه‌ای در شکل ۸-۴ به‌نمایش درآمده است. این لایه‌ها در ادامه تشریح می‌شوند.

1. Generalized Approximate Reasoning-based Intelligent Control

2. Berenji, H. R.

3. Khedkar, P.



شکل ۴-۸: ساختار GARIC

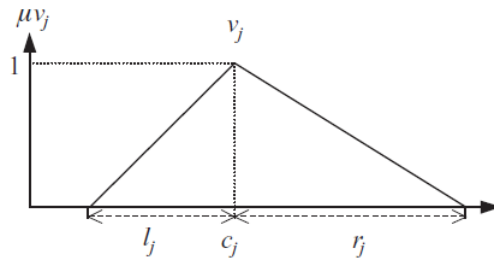
لایه ۱: این لایه، لایه ورودی است که متغیرهای ورودی حقیقی مقدار را دربردارد. در ادامه، ورودی‌ها به لایه دوم منتقل می‌شوند.

لایه ۲: گره‌های این لایه، m تابع عضویت فازی $\{A_{ji}(x_i)\}$ را برای ورودی‌های x_i مشخص می‌کنند. این گره‌ها با کمک عمل فازی‌سازی، محاسبه درجه عضویت ورودی‌ها

را انجام می‌دهند. GARIC از توابع عضویت مثلثی نامتقارن استفاده می‌کند:

$$\mu_{v_j}(x_i) = \begin{cases} 1 - \frac{|x_i - c_j|}{r_j} & x_i \in [c_j, c_j + r_j] \\ 1 - \frac{|x_i - c_j|}{l_j} & x_i \in [c_j - l_j, c_j] \\ 0 & \text{در سایر موارد} \end{cases} \quad (53-4)$$

که c_j ، r_j و l_j به ترتیب مرکز، پهنای راست و پهنای چپ عدد فازی مثلثی را مطابق شکل ۹-۴ نشان می‌دهند. مقادیر عضویت به دومین لایه پنهان انتقال می‌یابد.



شکل ۹-۴: تابع عضویت مثلثی نامتقارن که برای متغیرهای ورودی و خروجی استفاده می‌شود

لایه ۳: هر گره در این لایه، یک قاعده از سیستم فازی را مشخص می‌کند. قواعد با r_1 ، r_2 ، ... و r_M نام‌گذاری شده‌اند که M تعداد قواعد است. هر گره، درجه تحقق (یا شدت برانگیختگی) یک قاعده را تعیین می‌کند. در اینجا از یک عمل حداقل‌کننده دیگر که softmin نامیده می‌شود، استفاده می‌کنیم:

$$\tilde{\min} \{ \mu_1, \mu_2, \dots, \mu_n \} = \frac{\sum_{i=1}^n \mu_i e^{-\alpha \mu_i}}{\sum_{i=1}^n e^{-\alpha \mu_i}} \quad (54-4)$$

در رابطه فوق، $\alpha > 0$ یک پارامتر است که رفتار تابع را معین می‌کند. زمانی که $\alpha = 0$ باشد، $\tilde{\min}$ را برای هر قاعده مطابق رابطه زیر محاسبه می‌کند.

$$w_k = \tilde{\min} \{ \mu_{v_j}^k(x_1), \mu_{v_j}^k(x_2), \dots, \mu_{v_j}^k(x_n) \} \quad (55-4)$$

در این رابطه، $k = 1, 2, \dots, M$ است. شدت برانگیختگی قاعده (w_k) به لایه چهارم و لایه پنجم (خروجی) انتقال داده می‌شود.

لایه ۴: هر گره در این لایه، یک تابع عضویت فازی تالی را ارائه می‌دهد. فرض می‌شود که مجدداً یک تابع عضویت مثلثی نامتقارن به کار می‌رود. از یک تابع عضویت مشابه برای ورودی‌هایی که در شکل ۴-۹ نشان داده شده‌اند، استفاده می‌شود. روند فازی‌زدایی‌ای که در اینجا مورد استفاده است، میانگین حداکثر محلی نام دارد که به صورت زیر تعریف می‌گردد.

$$v_{jk}^* = c_j + \frac{1}{\gamma}(r_j - l_j)(1 - w_k) \quad (56-4)$$

در رابطه بالا، مقدار فازی‌زدایی شده v_{jk}^* تابع عضویت v_{jk} برای k امین قاعده است. مرکز، پهنای راست و پهنای چپ تابع عضویت مثلثی v_j ، به ترتیب با c_j ، r_j و l_j نشان داده می‌شود که $l_j \neq r_j$ و $j = 1, 2, \dots, m$ است؛ m نیز تعداد توابع عضویت در بخش تالی را بیان می‌کند. ویژگی غیرمعمول گره‌ها در لایه چهارم این است که آنها می‌توانند خروجی‌های چندگانه تولید کنند. این ویژگی غیراستاندارد را می‌توان به کمک مقدار یکپارچه یکتا برای گره‌های لایه چهارم (v_j^*) حذف کرد:

$$v_{jk}^* = c_j + \frac{1}{\gamma}(r_j - l_j) \sum_k w_k - \frac{1}{\gamma}(r_j - l_j) \sum_k (w_k)^2 \quad (57-4)$$

اگر انطباق دقیق برای یک مدل شبکه عصبی موردنیاز باشد و مدل GARIC به وسیله مقادیر یکپارچه فوق تحت‌تأثیر قرار نگیرد، مقدار خروجی یکپارچه در رابطه ۴-۵۷ ضروری است.

لایه ۵: تنها گره این لایه خروجی، خروجی نهایی را با کمک انبوه‌سازی همه مقادیر قواعد فعال به صورت زیر حاصل می‌کند.

$$u = \frac{\sum_{k=1}^M w_k \cdot v_{jk}^*}{\sum_{k=1}^M w_k} = \sum_{k=1}^M \bar{w}_k \cdot v_{jk}^* \quad (58-4)$$

همچنین می‌توان از مزیت مقدار یکپارچه یکتا که در لایه چهارم به دست آمد، استفاده نمود:

$$u = \frac{\sum_{k=1}^M v_{jk}^f}{\sum_{k=1}^M w_k} \quad (59-4)$$

بنابراین لایه پنجم یک مقدار پیوسته را تولید می‌کند که عمل انتخاب شده توسط ASN است. اصلاح خروجی شبکه به کمک تعدیل‌گر تصادفی عمل، انجام شده که در آن از پیش‌بینی قبلی $\hat{r}(t-1)$ حاصل از AEN و انتخاب عمل $u(t)$ حاصل از ASN استفاده می‌شود. تعدیل تصادفی عمل از رابطه زیر پیروی می‌کند.

$$u'(t) = \Psi[u(t) \cdot \sigma[\hat{r}(t-1)]] \quad (60-4)$$

$\Psi[\cdot]$ یک تابع تعدیل تصادفی بوده که مبتنی بر یک متغیر تصادفی گاوسی با میانگین $u(t)$ و انحراف معیار $\sigma[\hat{r}(t-1)]$ است. $\sigma[\cdot]$ یک تابع نزولی یکنوای غیرمنفی نظیر $e^{-\hat{r}}$ است. می‌توان یک معیار نهایی برای تعدیل تصادفی عمل به صورت زیر به کار گرفت. در هر گام زمانی از یک اختلال استفاده می‌شود که براساس انحراف نرمال شده از $u(t)$ و $u'(t)$ مطابق رابطه زیر به دست می‌آید.

$$s(t) = \frac{u'(t) - u(t)}{\sigma[\hat{r}(t-1)]} \quad (61-4)$$

این مورد به عنوان یک عامل یادگیری در به روزرسانی‌های وزن در ASN ظاهر شده و کمک می‌کند.

یادگیری وزن‌های Wc و Wb ، Wa مربوط به AEN در GARIC مشابه سازوکار یادگیری در ARIC است. به روزرسانی وزن‌های Wc و Wb ، Wa مربوط به AEN در GARIC به ترتیب براساس روابط ۴-۴۲، ۴-۴۰ و ۴-۴۱ انجام می‌شود. وزن‌های اتصال شبکه عصبی در ASN واحد هستند. این پارامترهای توابع عضویت فازی در لایه دو و چهار ASN متعلق به GARIC است که باید یادگیری آنها

شود. فرض کنید p بردار پارامتری باشد که شامل تمام پارامترهای توابع عضویت (مرکز، پهنای چپ و پهنای راست) مقدم و تالی شود. تابع هدفی که باید نسبت به بردار پارامتری p بیشینه شود را با v نشان می‌دهیم. این کار را می‌توان را با استفاده از یک الگوریتم یادگیری کاهش گرادیان انجام داد:

$$\Delta p = \eta \frac{\partial v}{\partial p} = \eta \frac{\partial v}{\partial u} \frac{\partial u}{\partial p} \quad (۶۲-۴)$$

می‌توان برای جمله اول رابطه فوق از یک تقریب استفاده کرد:

$$\frac{\partial v}{\partial u} \approx \frac{\Delta v}{\Delta u} \approx \frac{v(t) - v(t-1)}{u(t) - u(t-1)} \quad (۶۳-۴)$$

علامت خارج قسمت در رابطه ۶۳-۴ برای به‌روزرسانی پارامترها کافی است. عبارت دوم $\frac{\partial u}{\partial p}$ را می‌توان به‌سادگی با استفاده از رابطه‌های ۵۶-۴ و ۵۸-۴ برای پارامترهای تالی محاسبه کرد:

$$\frac{\partial u}{\partial p_j} = \sum_k \bar{w}_k \frac{\partial v_{jk}^*}{\partial p_j} \quad (۶۴-۴)$$

مشتق v_{jk}^* نسبت به سه پارامتر موجود در $p_j = \{c_j, r_j, l_j\}$ متعلق به تابع عضویت تالی به‌صورت زیر به‌دست می‌آید.

$$\frac{\partial v_{jk}^*}{\partial c_j} = \frac{\partial \left(c_j + \frac{1}{\gamma} (r_j - l_j) (1 - w_k) \right)}{\partial c_j} = 1 \quad (۶۵-۴)$$

$$\frac{\partial v_{jk}^*}{\partial r_j} = \frac{\partial \left(c_j + \frac{1}{\gamma} (r_j - l_j) (1 - w_k) \right)}{\partial r_j} = \frac{1}{\gamma} (1 - w_k) \quad (۶۶-۴)$$

$$\frac{\partial v_{jk}^*}{\partial l_j} = \frac{\partial \left(c_j + \frac{1}{\gamma} (r_j - l_j) (1 - w_k) \right)}{\partial l_j} = -\frac{1}{\gamma} (1 - w_k) \quad (۶۷-۴)$$

بنابراین قاعده به‌روزرسانی برای پارامترهای تالی $p_j = \{c_j, r_j, l_j\}$ مطابق زیر تعریف می‌شود.

$$\Delta p = \eta \operatorname{sgn} \left(\frac{\Delta v}{\Delta u} \right) s(t) \hat{r} \frac{\partial u}{\partial p_j} \quad (۶۸-۴)$$

η نرخ یادگیری است. استفاده از $s(t)$ و \hat{r} به عنوان عوامل یادگیری، یک پاداش دیگر را برای نتایج بزرگ اختلال در یک عمل خوب به دست می‌دهد. در بسیاری از کاربردها، ممکن است فقط به روزرسانی پارامترهای تالی کافی باشد.

به روزرسانی پارامترهای تابع عضویت مقدم را می‌توان به شیوه مشابه انجام داد. این عمل به درجه‌های w_k بستگی دارد که خود آنها به درجه‌های μ_j در لایه دو وابسته هستند. به روزرسانی‌ها را فقط در شرایطی می‌توان حاصل کرد که توابع μ_j (تعریف شده در رابطه ۴-۵۳) نسبت به p_j مشتق‌پذیر باشند. به منظور بیشینه‌سازی u برحسب پارامتر p_j تابع عضویت مقدم، از قاعده یادگیری کاهش گرادیان استفاده می‌شود:

$$\frac{\partial u}{\partial p_j} = \frac{\partial u}{\partial \mu_j} \frac{\partial \mu_j}{\partial p_j} \quad (۶۹-۴)$$

جمله اول $(\frac{\partial u}{\partial \mu_j})$ به صورت زیر محاسبه می‌شود:

$$\frac{\partial u}{\partial \mu_j} = \sum_k \frac{\partial u}{\partial w_k} \frac{\partial w_k}{\partial \mu_j} \quad (۷۰-۴)$$

عبارت‌های $\frac{\partial u}{\partial w_k}$ و $\frac{\partial w_k}{\partial \mu_j}$ به شیوه زیر منتج می‌شوند:

$$\frac{\partial u}{\partial w_k} = \frac{c_j + \frac{1}{\gamma}(r_j - l_j)(1 - 2w_k) - u}{\sum_k w_k} \quad (۷۱-۴)$$

$$\frac{\partial w_k}{\partial \mu_j} = \frac{e^{-\alpha \mu_j(x_i)}(1 + \alpha[w_k - \mu_j(x_i)])}{\sum_i e^{-\alpha \mu_j(x_i)}} \quad (۷۲-۴)$$

فقط هنگامی می‌توان جمله دوم $(\frac{\partial \mu_j}{\partial p_j})$ را محاسبه کرد که μ_j در تمام نقاط مشتق‌پذیر باشد. یک مثال از چنین تابع همواری را در رابطه زیر مشاهده می‌کنید.

$$\mu_j(x_i) = \frac{1}{1 + \left| \frac{x_i - c_j}{s} \right|^b} \quad (۷۳-۴)$$

اگر $x_i < c_j$ باشد، $s = l_j$ است و اگر $x_i \geq c_j$ باشد، $s = r_j$ است. b یک پارامتر است که شکل تابع را کنترل می‌کند. تقریبی برای تابع عضویت مثلثی براساس پژوهش برنجی^۱ و کدکار^۲ (۱۹۹۲) در رابطه ۴-۵۳ ارائه شده است. محاسبه مشتق μ_j نسبت به پارامترهای $\{c_j, r_j, l_j\}$ بسیار ساده است. جزئیات مشتق‌ها برای هریک از پارامترها را می‌توان در پژوهش ناک^۳ و همکاران (۱۹۹۷) یافت. به‌روزرسانی پارامترهای $\{c_j, r_j, l_j\}$ در مقدم، به‌کمک رابطه ۴-۶۸ انجام می‌شود. مشتق‌ها را می‌توان به‌صورت محلی با استفاده از روابط ۴-۶۹ تا ۴-۷۲ محاسبه کرد.

موردکاوی

سیستم‌های پشتیبان سلامت با چالش‌های مهمی مانند کمبود اطلاعات پزشکی، خطاهای قابل پیشگیری، تهدید داده‌ها، تشخیص نادرست و تاخیر مواجه هستند. موتو و همکاران (۲۰۲۰) یک سنسور پوشیدنی یا قابل حمل توسط انسان را پیشنهاد کرده‌اند که به کلان‌داده مبتنی بر اینترنت اشیا (IOT) متصل است و داده‌کاوی در سلامت را دنبال می‌کند.

در این پژوهش، یک شبکه GARIC با قواعد رگرسیونی برای جمع‌آوری اطلاعات مربوط به بیمار از IOT طراحی شده است. مجموعه داده‌های مربوط به سنسورهای IOT به GARIC وارد شده و از GARIC برای استخراج ویژگی از میان شاخص‌های گسترده استفاده می‌شود. به‌منظور وادار کردن مدل به پاسخ‌گویی در برابر تعداد بیشتری از پارامترهای ورودی، از یک شبکه تکثیر برگشتی با وزن‌ها و بایاس‌های بهینه‌سازی شده اولیه استفاده شده است. بنابراین یک سیستم فازی عصبی GARIC توسعه یافته است که طی آن از طراحی GARIC برای بهینه‌سازی وزن‌ها و بایاس‌های اولیه سیستم فازی عصبی استفاده می‌شود. همچنین، برای پیش‌بینی هر بیماری یا اختلال در مرحله اولیه، از کاوش کلان‌داده به کمک روش رگرسیون بهره گرفته شده است. در مرحله بعد از یک شبکه عمیق بولتزمن استفاده می‌شود تا توسط داده‌های پردازش شده در مرحله قبلی آموزش ببیند. سپس از مطالعه هم‌خوانی سراسر ژنوم (GWAS) برای پیش‌بینی

1. Berenji, H. R.

2. Khedkar, P.

3. Nauck, D.

بهره گرفته می‌شود. اگر افراد مبتلا به بیماری‌هایی باشند، از طریق پیامک و ایمیل و ... هشدار دریافت می‌کنند و بعد از آن درمان‌ها و توصیه‌هایی از پزشکان دریافت می‌کنند. نتایج این پژوهش نشان می‌دهد که دقتی نزدیک به ۹۶ درصد در پیش‌بینی توسط روش پیشنهادی به دست آمده است.

Muthu, B., Sivaparthipan, C. B., Manogaran, G., Sundarasekar, R., Kadry, S., Shanthini, A. & Dasel, A. (2020). IOT based wearable sensor for diseases prediction and symptom analysis in healthcare sector. Peer-to-peer networking and applications, 13(6), 2123-2134.

۴-۴ شبکه‌های تابع پایه فازی (FBFN)

شبکه تابع پایه فازی (FBFN)^۱ ساختاری شبیه به شبکه‌های تابع پایه شعاعی (RBF)^۲ دارد که یکی از گونه‌های مشهور شبکه‌های عصبی مصنوعی است. FBFN اولین بار توسط وانگ^۳ و مندل^۴ (۱۹۹۲) با استفاده از نظریه استون-وایرشرس^۵ معرفی شد که نشان می‌دهد ترکیب خطی توابع پایه فازی قادر است که هر تابع پیوسته را تقریب بزند. به‌طور کلی الگوریتم پس‌انتشار در یادگیری توابع پایه فازی نیازمند هزاران تکرار است. بنابراین یک الگوریتم یادگیری کمترین مربعات متعامد (OLS)^۶ برای طراحی سیستم‌های فازی مبتنی بر زوج‌های ورودی/خروجی پیشنهاد می‌شود. این الگوریتم، توابع معنادار پایه فازی را از یک مجموعه اولیه از توابع برمی‌گزیند. سپس، از این‌ها برای ساخت سیستم فازی نهایی استفاده می‌شود. الگوریتم یادگیری OLS، یک روند تک‌گذر^۷ است که در مقایسه با الگوریتم پس‌انتشار سریع‌تر عمل می‌کند.

1. Fuzzy Basis Function Network
2. Radial Basis Function
3. Wang, L. X.
4. Mendel, J. M.
5. Stone-Weierstrass
6. Orthogonal Least Squares
7. One-pass

پیکربندی اصلی FBFN مشابه سیستم‌های فازی متداول است که اجزای اصلی نظیر فازی‌سازی، پایگاه قواعد، موتور استنتاج و فازی‌زدایی را دربردارد. یک FBFN شامل پنج لایه است و از تابع عضویت گاوسی به عنوان ورودی‌ها، از یک سیستم استنتاج فازی با استنتاج ضرب و از تابع عضویت با شیوه فازی‌زدایی مرکزوار در خروجی استفاده می‌کند. یک سیستم چندورودی - تک خروجی (MISO)^۱ در این بخش مفروض است. سیستم n ورودی - تک خروجی را با M قاعده به فرم زیر در نظر بگیرید.

$$(۷۴-۴)$$

R^k : اگر x_1 برابر A_{1j}^k و x_2 برابر A_{2j}^k و ... و x_n برابر A_{nj}^k باشد، آنگاه y برابر B_j^k است.

در اینجا، A_{ij} توابع عضویت را برای ورودی‌ها و B_j توابع عضویت تکی را با $i = 1, 2, \dots, n$ و $j = 1, 2, \dots, m$ نشان می‌دهند. تعداد توابع عضویت برای هر ورودی و خروجی با m مشخص شده و فرض می‌شود هر ورودی و خروجی تعداد توابع عضویت یکسانی دارند. $k = 1, 2, \dots, M$ است و M تعداد قواعد سیستم فازی را نشان می‌دهد. ساختار FBFN با سه ورودی، یک خروجی و چهار قاعده در شکل ۴-۱۰ به نمایش درآمده است.

لایه ۱: این لایه، لایه ورودی است و مسئولیت انتقال مقادیر ورودی را از طریق خود به لایه دوم برعهده دارد.

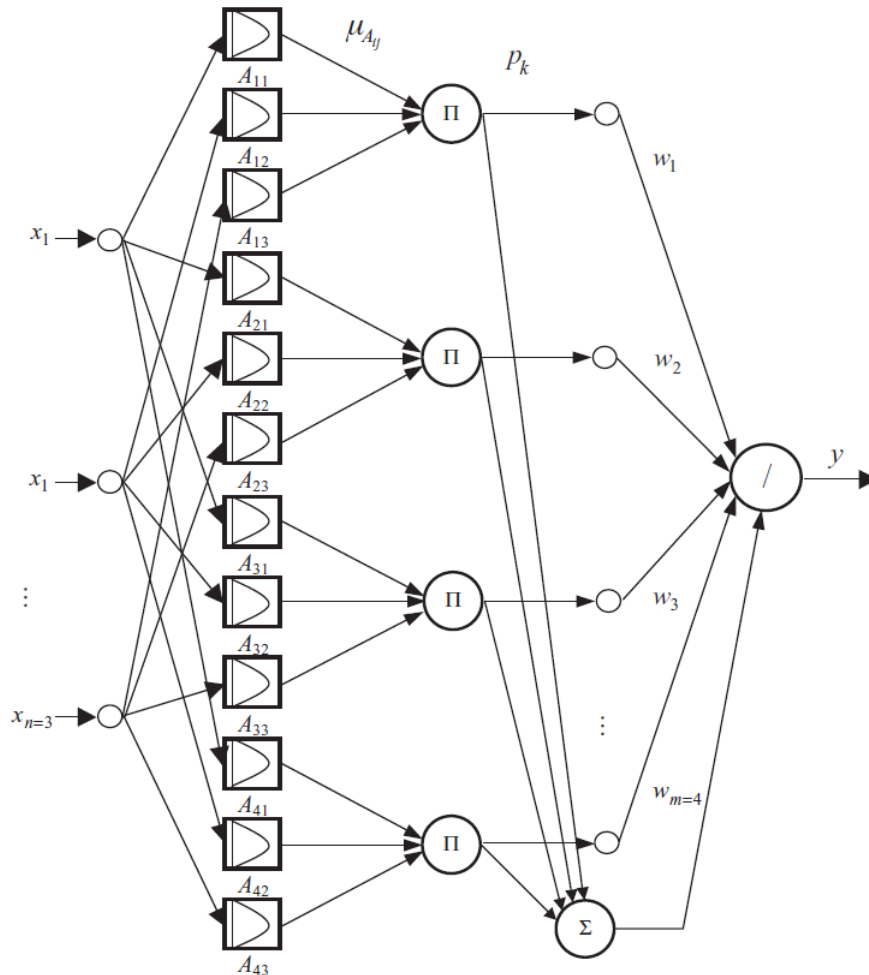
لایه ۲: این لایه، لایه فازی‌سازی است که شامل توابع عضویت گاوسی به فرم زیر می‌شود.

$$\mu_{A_{ji}}(x_i) = \exp \left[-\frac{1}{2} \left(\frac{x_i - m_{ji}}{\sigma_{ji}} \right)^2 \right] \quad (۷۵-۴)$$

m_{ji} و σ_{ji} مرکز و پهنای تابع عضویت گاوسی j ام را برای ورودی x_i نشان می‌دهند. مقدار عضویت $\mu_{A_{ij}}(x_i)$ به لایه قواعد انتقال می‌یابد.

لایه ۳: این لایه، لایه قواعد است و از قاعده ضرب برای استنتاج استفاده می‌کند:

$$p_k(x_i) = \prod_{i=1}^n \mu_{A_{ji}}(x_i) \quad k = 1, 2, \dots, M \quad (۷۶-۴)$$



شکل ۴-۱۰: ساختار FBFN با چهار قاعده

لایه ۴: این لایه، لایه فازی‌زدایی است و با کمک یک فازی‌زدای مرکزی، یک عمل فازی‌زدای مرکزوار را بر تابع عضویت خروجی اعمال می‌کند. وزن‌های اتصال بین لایه چهار و پنج با w_k نشان داده شده‌اند.

لایه ۵: این لایه انبوه‌سازی است که FBF را به صورت زیر تعریف می‌کند.

$$y = f(x) = \frac{\sum_{k=1}^M p_k(x)w_k}{\sum_{k=1}^M p_k(x)} \quad k = 1, 2, \dots, M \quad (77-4)$$

نشان داده می‌شود که این سیستم استنتاج فازی با یک بسط FBF (مثل ترکیب خطی از FBF یا FBFN) هم‌ارز است.

فرض کنید که یک مجموعه داده N ورودی/خروجی به صورت $\{(x_1, d_1), (x_2, d_2), \dots, (x_N, d_N)\}$ برای آموزش موجود باشد. وظیفه آموزش FBFN طراحی $f(x)$ است به طوری که:

$$d(t) = f[x(t)] + e(t) = \sum_{k=1}^M \bar{p}_k(x)w_k + e(t). \quad (78-4)$$

در اینجا، $\bar{p}_k(x) = \frac{p_k(x)}{\sum_{k=1}^M p_k(x)}$ است. رابطه ۷۸-۴ را می‌توان به صورت نمادگذاری ماتریسی نوشت:

$$d = Pw + e \quad (79-4)$$

در رابطه فوق، $d = [d_1, \dots, d_N]$ ، $p = \begin{bmatrix} p_{11} & \dots & p_{1M} \\ \vdots & \ddots & \vdots \\ p_{N1} & \dots & p_{NM} \end{bmatrix}$ و $w = [w_1, \dots, w_N]$

است. هنگامی که P معلوم باشد، FBFN به یک مسأله حداقل مربعات خطی تبدیل می‌شود. بردارهای ستونی P ، بردارهای پاسخ گره‌های FBF هستند. به منظور پیاده‌سازی روند OLS، یادگیری پارامترهای $\lambda = \{(m_1, \sigma_1), \dots, (m_m, \sigma_m)\}$ متعلق به FBF باید بر روی زوج‌های ورودی/خروجی اعمال شود (شین^۱ و سوآ، ۲۰۱۷).

1. Shin, Y. C.

2. Xu, C.

۴-۵ کنترل‌کننده فازی عصبی (NEFCON)

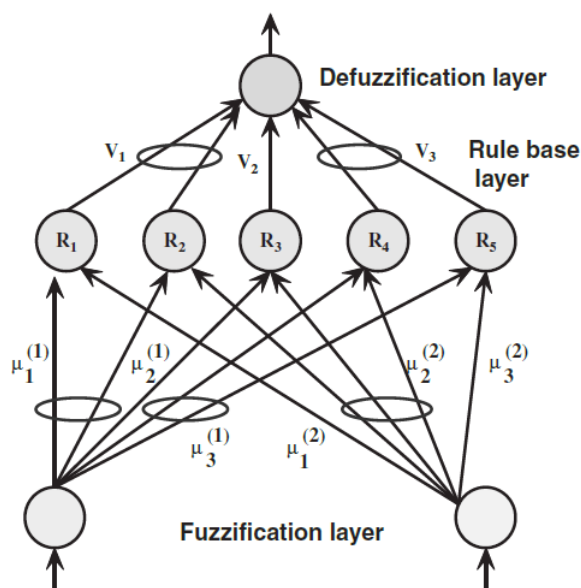
مسئله مهمی که در طراحی یک کنترل‌کننده فازی عصبی با آن مواجه می‌شویم، نیازمندی به مجموعه‌ای از داده‌های ورودی/خروجی برای یادگیری است. یک مسئله کنترل را نمی‌توان با استفاده از یادگیری با نظارت حل کرد زیرا ممکن است مجموعه‌ای از داده‌های ورودی/خروجی در دسترس نباشد. یک رویکرد جایگزین، استفاده از یادگیری تقویتی^۱ است. زمانی از یادگیری تقویتی استفاده می‌شود که سیگنال یاددهنده‌ای وجود نداشته باشد و یادگیری از طریق برهم‌کنش پیوسته با محیط برای کمینه‌سازی یک شاخص عملکرد اسکالر (نرده‌ای) انجام شود (بارتو^۲ و همکاران، ۱۹۸۳). کنترل‌کننده فازی عصبی (NEFCON)^۳ یک سیستم استنتاج فازی از نوع مددانی است که با استفاده از یادگیری تقویتی به کار گرفته می‌شود. این روش را می‌توان به‌عنوان گسترشی از GARIC تلقی کرد که به پایگاه قواعد از پیش تعیین‌شده نیازمند است. ساختار NEFCON با دو ورودی و یک اقدام کنترلی (خروجی) در شکل ۴-۱۱ به نمایش درآمده است. گره‌های R_1 تا R_2 قواعد را ارائه می‌کنند. گره‌های ξ_1 و ξ_2 بیان‌کننده مقادیر ورودی هستند و گره η مقدار خروجی را نمایش می‌دهد. μ_r و V_r نیز مجموعه‌های فازی هستند که مقدم‌ها و تالی‌ها را تشریح می‌کنند. NEFCON دارای ساختاری شبیه به شبکه عصبی با سه لایه است که لایه ورودی، متغیرهای ورودی را ارائه می‌کند، لایه پنهان، بیان‌کننده قواعد فازی است و لایه خروجی، شامل یک گره تنها می‌شود. NEFCON می‌تواند علاوه بر پایگاه قواعد، توابع عضویت را نیز یاد بگیرد. ایده NEFCON بر کاوش و کشف دانش پیشینی نظیر قواعد معلوم و سنج‌های خطا مبتنی است. این ساختار را می‌توان به‌منظور یادگیری یک پایگاه قواعد اولیه - در شرایطی که هیچ دانش پیشینی از سیستم موجود نباشد - به کار برد یا برای بهینه‌سازی یک پایگاه قواعد اولیه که به صورت دستی تعریف شده، مورد استفاده قرار داد. در مقابل شبکه‌های عصبی، اتصال‌ها در NEFCON به جای اعداد حقیقی، توسط مجموعه‌های فازی وزن‌دار می‌شود. قواعدی که دارای مقدم یکسانی هستند از وزن‌های اشتراکی استفاده می‌کنند که در شکل با بیضی‌های کشیده شده دور اتصال‌ها نشان داده شده‌اند. آنها صحت پایگاه قواعد موردنظر را تضمین می‌کند. پایگاه دانش این سیستم فازی به‌طور ضمنی توسط ساختار شبکه عصبی ارائه

1. Reinforcement learning

2. Barto, A. G.

3. Neuro-Fuzzy Controller

می‌گردد. واحدهای ورودی وظیفه فازی‌سازی را بر عهده دارند؛ منطق استنتاج توسط توابع انتشار ارائه می‌شود و واحد خروجی مسئول فازی‌زدایی است.



شکل ۴-۱۱: ساختار NEFCON

فرآیند یادگیری در مدل NEFCON را می‌توان به دو مرحله تقسیم نمود: یادگیری ساختار (یعنی یادگیری قواعد) و یادگیری پارامترها (یعنی یادگیری توابع عضویت). مرحله اول به منظور یادگیری پایگاه قواعد طراحی می‌شود. مرحله دوم بهینه‌سازی قواعد را به وسیله انتقال و اصلاح مجموعه‌های فازی قواعد به انجام می‌رساند. برای آموزش پایگاه قواعد از دو روش استفاده می‌شود. هنگامی که خروجی صحیح نامعلوم باشد از یادگیری افزایشی برای قواعد استفاده شده و قواعد براساس مقادیر تخمینی خروجی ایجاد می‌شوند. با پیش‌رفتن این یادگیری، براساس میزان نیاز قواعد بیشتری افزوده می‌شود. در یادگیری کاهشی، ابتدا به واسطه بخش‌های فازی از متغیرهای فرآیند، قواعد ایجاد می‌شود و قواعد غیرضروری طی یادگیری حذف می‌گردند. یادگیری کاهشی قواعد در مقایسه با رویکرد افزایشی کارایی کمتری دارد. با این حال، می‌توان آن را بدون دشواری در فرآیندهای نامعلوم به کار برد که در آنها نیازی به معلوم بودن یا وجود حدسی برای مقدار بهینه خروجی نیست. به منظور یادگیری یا بهینه‌سازی پایگاه قواعد، هر دو مرحله

از یک خطای فازی (E) استفاده می‌کنند که کیفیت حالت کنونی سیستم را نشان می‌دهد. به‌منظور دستیابی به یک پایگاه قواعد خوب، باید اطمینان ایجاد شود که فضای حالت فرآیند طی فرآیند یادگیری به‌میزان کافی پوشش داده می‌شود. به‌دلیل پیچیدگی محاسبات موردنیاز، یادگیری کاهشی قواعد را تنها زمانی می‌توان به‌کار بست که تعداد کمی متغیر ورودی بدون مجموعه‌های فازی خیلی زیاد وجود داشته باشد. برای سیستم‌های بزرگتر، یادگیری افزایشی قواعد بهینه خواهد بود. هر زمان که دانش قبلی در دسترس باشد می‌توان آن را برای کاهش پیچیدگی یادگیری شرکت داد. توابع عضویت پایگاه قواعد طبق الگوریتم پس‌انتشار خطای فازی (FEBP)^۱ اصلاح می‌شوند. الگوریتم FEBP می‌تواند توابع عضویت را سازگار کند. فقط در شرایطی که پایگاهی از قواعد فازی از قبل وجود داشته باشد، می‌توان این الگوریتم را به‌کار برد. ایده الگوریتم یادگیری یکسان است: افزایش اثر یک قاعده در شرایطی که عملکرد آن در مسیر صحیح قرار دارد (پاداش) و کاهش اثر آن در شرایطی که یک قاعده به‌طور مؤثر برعکس رفتار کند (جزا). در شرایطی که قطعاً هیچ دانشی درباره تابع عضویت اولیه وجود نداشته باشد، باید از یک افراز یکنواخت برای متغیرها استفاده شود. اگر استراتژی کنترل (یعنی قواعد) مشخصی وجود نداشته باشد، کنترل‌کننده فازی عصبی باید قواعد عملیاتش را توسعه دهد. بارتو^۲ و همکاران (۱۹۸۳) نشان داده‌اند که کنترل‌کننده‌های عصبی می‌توانند قواعد را با کمک یادگیری تقویتی یاد بگیرند. جزئیات بیشتر درباره این فرآیند یادگیری را می‌توان در پژوهش ناک^۳ و همکاران (۱۹۹۷) یافت.

دو گونه NEFCON در ادبیات پژوهش‌ها یافت می‌شود: NEFPROX (ناک و کروز^۴، ۱۹۹۹) و NEFCLASS (ناک و کروز، ۱۹۹۵ و ۱۹۹۶). مدل NEFPROX (برای تقریب تابع) شباهت زیادی به NEFCON دارد که می‌تواند بیش از یک خروجی داشته باشد و با استفاده از الگوریتم یادگیری با نظارت آموزش ببیند. الگوریتم یادگیری با نظارت از خطایی استفاده می‌کند که با استفاده از تفاوت بین خروجی مطلوب و خروجی واقعی تعریف می‌شود. در NEFPROX از یک معیار خطای فازی نیز می‌توان استفاده کرد. NEFCLASS (برای موارد طبقه‌بندی) به‌منظور تعیین طبقات برای داده‌های مفروض ایجاد شده است. این مسأله طبقه‌بندی با استفاده از یک تابع نامعلوم بیان می‌شود. پایگاه قواعد مدل

1. Fuzzy Error Backpropagation

2. Barto, A. G.

3. Nauck, D.

4. Kruse, R.

NEFCLASS، این تابع نامعلوم را به وسیله نگاشت الگوی ورودی به یک طبقه خروجی تقریب می‌زند. تشریح این دو گونه از NEFCON در ادامه ارائه می‌گردد.

۴-۵-۱ طبقه‌بندی فازی عصبی (NEFCLASS)

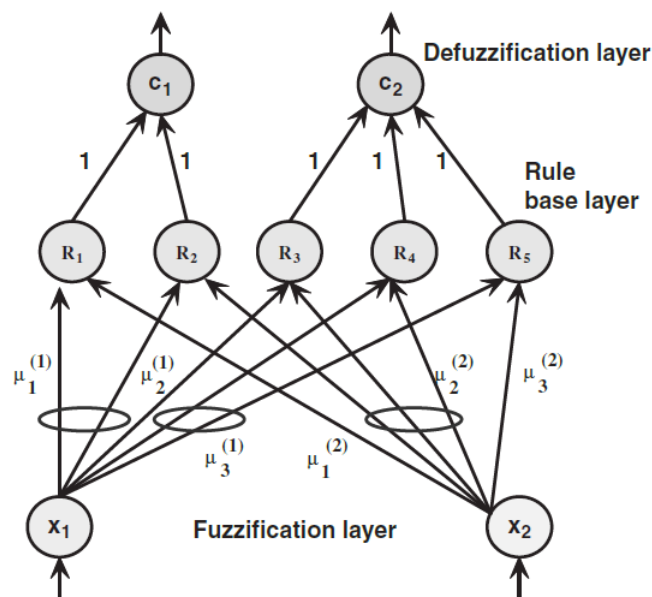
NEFCLASS^۱ به منظور ایجاد قواعد فازی از مجموعه داده‌هایی که بتوان آنها را در طبقات قطعی^۲ مختلف بخش‌بندی کرد، مورد استفاده قرار می‌گیرد. پایگاه قواعد یک سیستم NEFCLASS، تابع نامعلومی (ϕ) را تخمین می‌زند که مسأله طبقه‌بندی را ارائه می‌کند و یک الگوی ورودی (x) را به طبقه خویش (C_i) می‌نگارد.

به دلیل استفاده از رویه‌های پس‌انتشار در NEFCLASS، پایگاه قواعد عملاً تابع ϕ را تخمین نمی‌زند بلکه تخمین تابع ϕ' را فراهم می‌کند. $\phi(x)$ از معادله $\phi(x) = \phi(\phi(x))$ به دست می‌آید که ϕ بازتابی از تفسیر نتیجه طبقه‌بندی حاصل از سیستم NEFCLASS است. یک سیستم NEFCLASS در شکل ۴-۱۲ به نمایش درآمده است که الگوها را به وسیله پنج قاعده زبانی با دو ویژگی به دو طبقه متمایز می‌نگارد. NEFCLASS شباهت زیادی به سیستم NEFCON دارد به استثنای تغییر کوچکی که در الگوریتم یادگیری و تفسیرپذیری قواعد وجود دارد. همانند سیستم NEFCON، در NEFCLASS مقادیر زبانی همسانی برای یک متغیر توسط مجموعه فازی یکسانی ارائه می‌شود. از آنجایی که وظیفه اصلی NEFCLASS طبقه‌بندی است، باید دو قاعده با مقدم‌های یکسان وجود داشته باشند و هر واحد قاعده نیز باید فقط به یک واحد خروجی متصل شود. وزن‌های بین لایه قاعده و لایه خروجی فقط اتصال واحدها را انجام می‌دهند. با استفاده از دانش جزئی درباره الگوها می‌توان یک سیستم NEFCLASS را ایجاد کرد و سپس آن را به وسیله یادگیری اصلاح نمود و یا می‌توان آن را با یادگیری از scratch ساخت. تعریف تعدادی مجموعه فازی ابتدایی که دامنه‌های ویژگی‌های ورودی را بخش‌بندی می‌کند باید توسط کاربر صورت گیرد. وی همچنین باید برای k_i (حداکثر تعداد گره‌های قاعده که می‌تواند در لایه پنهان ایجاد شود) مقداری را مشخص کند. NEFCLASS از توابع عضویت مثلثی استفاده می‌کند. الگوریتم یادگیری این توابع عضویت از یک معیار خطا استفاده کرده که اعلام می‌کند درجه تحقق یک قاعده باید بالاتر یا

1. Neuro-Fuzzy Classification

2. Crisp classes

پایین‌تر باشد. این اطلاعات برای تغییر در مجموعه‌های فازی ورودی به کار گرفته می‌شود. با وجود یک سیستم طبقه‌بندی، علاقمندی زیادی به مقادیر دقیق خروجی وجود نخواهد داشت. علاوه بر این، یک تعبیر «برنده همه چیز را می‌برد» یا «برنده واحد»^۱ برای خروجی در نظر گرفته شده است و بیشترین علاقمندی نسبت به نتایج صحیح طبقه‌بندی خواهد بود. یادگیری افزایشی در NEFCLASS نسبت به یادگیری کاهش‌ی در NEFCON بسیار کم‌هزینه‌تر است. حتی برای مسائلی با ابعاد بالا، پایگاه قواعد پس از حداکثر سه چرخه تکمیل می‌شود. NEFCLASS در مقایسه با شبکه‌های عصبی، از یک استراتژی یادگیری خیلی ساده‌تر بهره می‌گیرد که در آن از چندی‌سازی برداری^۲ در پیدا کردن قواعد (خوشه‌ها) استفاده نشده و همچنین در آن به اطلاعات گزاردیان برای آموزش توابع عضویت نیازی نیست. مزیت‌های دیگری همچون قابلیت تفسیرپذیری، امکان مقداردهی اولیه (با دانش قبلی) و سهولت آن نیز وجود دارد.



شکل ۴-۱۲: ساختار NEFCLASS

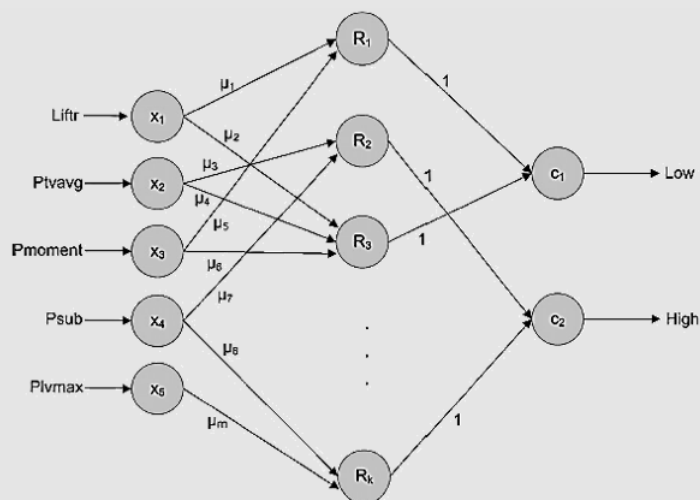
1. Winner takes all.
2. Vector quantization

موردکاوی

با وجود فناوری‌های پیشرفته در صنعت، فرآیندهای دستی و وظایف مرتبط با آن در عرصه‌های مختلف صنعتی و تولیدی همچنان در جریان هستند. یکی از آسیب‌های چنین فرآیندهایی بروز اختلالات اسکلتی-عضلانی مختلف است که هنوز نرخ بالایی در جهان دارد. طبقه‌بندی مشاغل صنعتی براساس اختلالات کمر (Low Back Disorder)، توانایی زیادی در جلوگیری از آسیب‌های احتمالی فراهم خواهد نمود.

آکای، آکایول و کورت (۲۰۰۸) در مطالعه‌ای با استفاده از طبقه‌بندی فازی عصبی، شغل‌های صنعتی را به دو دسته «کم‌خطر» و «پرخطر» تقسیم کرده‌اند. براساس مقایسه انجام شده با پژوهش‌های پیشین، طبقه‌بندی فازی عصبی نتایج بهتری را ایجاد کرده است. علاوه بر آن، به راحتی می‌توان قواعد فازی اگر-آنگاه را از این نتایج حاصل کرد و تحلیل عوامل بالقوه آسیب‌زا به کار برد. با بهره‌گیری از این قواعد، می‌تواند مداخلات ارگونومیکی را با هدف کاهش آسیب‌های کمر طراحی نمود.

در این مطالعه، یک مدل NEFCLASS با استفاده از داده‌های تجربی موجود در پیشینه پژوهش تحت آموزش و آزمون قرار داده شده است. داده‌های مذکور شامل ۲۳۵ پارامتر شغلی جابه‌جایی دستی بار (Manual Material Handling) در ۴۸ صنعت مختلف هستند. پنج عامل ریسک اختلالات کمر (کمر درد) با اختصارهای Liftr، Ptvavg، Pmoment، Psub و Pivmax متغیرهای ورودی مدل را تشکیل می‌دهند. خروجی مدل نیز میزان ریسک اختلالات کمر است که با مقدار ۰ برای سطح «کم‌خطر» و با مقدار ۱ برای سطح «پرخطر» ظاهر می‌شود. از میان ۲۳۵ سطر داده، ۱۴۸ نمونه برای آموزش و ۸۷ نمونه نیز برای آزمون NEFCLASS انتخاب شده است. مدل مذکور در شکل زیر به نمایش درآمده است. لایه پنهان در بردارنده قواعد فازی است که نظیر چنین ساختاری را دارد: «اگر Liftr برابر با μ_1 باشد و Ptvavg برابر با μ_2 باشد و ... Ptvavg برابر با μ_m باشد، آنگاه شغل مورد نظر به طبقه C_i متعلق است.»



به منظور یافتن بهترین طبقه‌بندی برای ریسک اختلالات کمر، الگوریتم NEFCLASS با پارامترهای مختلفی همچون نوع تابع عضویت و تعداد افراز فازی برای هر متغیر ورودی، مورد سعی و خطا قرار گرفت. توابع عضویت دوزنقه‌ای، زنگوله‌ای و مثلثی با ۲، ۳ و ۴ مجموعه برای دست‌یابی به بهترین عملکرد آزمایش شدند. تعداد دوره آموزش برابر با ۱۰۰ و نرخ یادگیری نیز برابر با ۰.۲ فرض شد. به این ترتیب، NEFCLASS با ترکیب‌های مختلفی از این پارامترها اجرا شد. پس از هر اجرا، یک روش هرس کردن (pruning) برای افزایش تفسیرپذیری و عملکرد طبقه‌بندی به کار رفت. در نهایت، پس از اجرای متعدد NEFCLASS با پارامترهای متفاوت، بهترین طرح‌های طبقه‌بندی با بالاترین عملکرد و قواعدی که به سادگی قابل تفسیر هستند، حاصل شدند. طرح برگزیده اول، دقتی معادل ۷۷ درصد در طبقه‌بندی نمونه‌های آزمون داشته است و این دقت برای طرح دوم تقریباً ۷۸ درصد بوده است. طرح اول و دوم به ترتیب دارای ۶ و ۴ قاعده فازی هستند که تفسیرپذیری قابل توجهی را برای تصمیم‌گیری ارائه می‌دهند و می‌توانند اطلاعات ارزشمندی را در تشخیص خطرات اختلالات کمردرد و پیشگیری از آن در صنعت فراهم آورند.

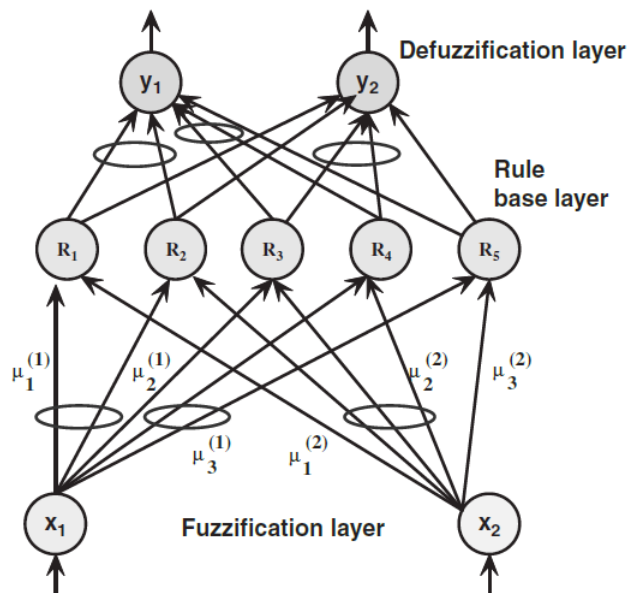
Akay, D., Akcayol, M. A. & Kurt, M. (2008). NEFCLASS based extraction of fuzzy rules and classification of risks of low back disorders. *Expert Systems with Applications*, 35(4), 2107-2112.

۴-۵-۲ تقریب تابع با رویکرد فازی عصبی (NEFPROX)

NEFPROX^۱ براساس یادگیری با نظارت ساده (مسأله یادگیری ثابت) بنا شده و برای تخمین تابع به کار می‌رود. این رویکرد یک نسخه تغییر یافته از NEFCON است که یادگیری تقویتی در آن وجود ندارد. NEFPROX بسیار شبیه به NEFCON و NEFCLASS است با این تفاوت که NEFCON فقط یک گره خروجی دارد و سیستم‌های NEFCLASS نیز در جانب نتیجه‌گیری از توابع عضویت استفاده نمی‌کنند. اگر قواعد مناسب از قبل شناخته شده باشند، می‌توان سیستم NEFPROX را مقداردهی اولیه کرد و اگر نه سیستم قادر است که همه قواعد را به تدریج بیاموزد. ساختار NEFPROX در شکل ۴-۱۳ نشان داده شده است. در حالی که ANFIS قادر است فقط مدل‌های سوگنو با توابع مشتق‌پذیر را به کار گیرد، NEFPROX می‌تواند سیستم فازی متعارف از نوع ممدانی را از داده‌ها بیاموزد. افزون بر این، NEFPROX نسبت به ANFIS در دست‌یابی به نتایج بسیار سریع‌تر عمل می‌کند.

موردکاوی

پیش‌بینی محصولات اسقاطی در بازیافت، به دلیل عدم قطعیت در زمان و مقدار بازگشت قراضه‌ها، مسائل دشواری را برای بنگاه‌های بازیافت و بازسازی ایجاد می‌کند. مارکس-گومز و همکاران (۲۰۰۲) یک روش پیش‌بینی گسترده و توسعه‌یافته را برای پیش‌بینی مقادیر بازگشت محصولات اسقاطی به چرخه بازیافت ارائه کردند. مدل پیشنهادی براساس عوامل مختلف تأثیرگذار و داده‌های چرخه عمر محصول است که در یک مطالعه موردی برای دستگاه‌های فتوکپی به کار گرفته شده است. رویکرد موردنظر، از یک مطالعه شبیه‌سازی، طراحی یک سیستم



شکل ۴-۱۳: ساختار NEFPROX

استنتاج فازی برای پیش‌بینی بازگشت در یک دوره برنامه‌ریزی شده مشخص و یک سیستم فازی عصبی برای پیش‌بینی مقادیر بازگشت برحسب زمان تشکیل شده است.

Marx-Gómez, J., Rautenstrauch, C., Nürnberger, A. & Kruse, R. (2002). Neuro-fuzzy approach to forecast returns of scrapped products to recycling and remanufacturing. Knowledge-Based Systems, 15(1-2), 119-128.

۶-۴ سیستم فازی عصبی انطباقی

شبکه‌های انطباقی، شبکه‌هایی با ساختار منعطف هستند که به وسیله مجموعه‌ای از پارامترهای اصلاح‌پذیر و قابل تعدیل تعریف می‌شود. این شبکه‌ها به دسته‌ای از الگوریتم‌های شبکه اطلاق می‌شود که می‌توانند ساختار، توپولوژی و/یا پارامترهای خود را به صورت پویا در پاسخ به تغییرات داده‌ها و محیط تغییر

دهند. چنین شبکه‌هایی می‌توانند به هم‌افزایی فازی عصبی کمک کنند.

سیستم‌های فازی تحت چارچوب شبکه‌های انطباقی با عنوان ANFIS یا شاخه‌ای از خانواده آن شناخته می‌شوند. محدودیت چنین شبکه انطباقی این است که تابع انتقال در گره‌ها باید قطعه به قطعه مشتق‌پذیر باشد. محدودیت پیکربندی شبکه نیز آن است که باید از نوع پیش‌خور باشد. به‌منظور پیاده‌سازی یک قاعده یادگیری مناسب، تجزیه پارامترهای مقدم و تالی لازم است. علاوه‌براین نشان داده شده است که نظریه استون-وایرشرس چگونه در ANFIS با قواعد ساده‌شده اگر-آنگاه فازی به‌کار گرفته می‌شود و شبکه تابع شعاع‌پایه چگونه با این نوع از ANFIS ساده‌شده ارتباط دارد (جانگ^۱ و سان^۲، ۱۹۹۳). انواع مختلفی از ANFIS در ادبیات ظاهر شده است که برخی از آنها در این فصل تشریح می‌شود.

۴-۶-۱ سیستم استنتاج فازی عصبی انطباقی (ANFIS)

سیستم استنتاج فازی شبکه‌محور انطباقی^۳ یا سیستم استنتاج فازی عصبی انطباقی^۴ (ANFIS) طبقه‌ای از شبکه‌های انطباقی است که از لحاظ عملکرد با یک سیستم استنتاج فازی هم‌ارز است. این سیستم اولین بار توسط جانگ معرفی شد (جانگ، ۱۹۹۳؛ جانگ و سان، ۱۹۹۳؛ جانگ و همکاران، ۱۹۹۷). ساختار ANFIS هر دو مدل فازی سوگنو و تسوکاموتو را بیان می‌کند.

به‌منظور سادگی و سهولت در فهم ساختار ANFIS، در این بخش دو ورودی (x_1 و x_2) و یک خروجی (f) مورد استفاده قرار می‌گیرد. پایگاه قواعدی با چهار قاعده فازی در نظر گرفته شده و سازوکار استنتاج برای یک مدل فازی سوگنوی مرتبه اول و ساختار ANFIS متناظر آن در شکل ۴-۱۴ به‌نمایش

1. Jang, J. S. R.

2. Sun, C. T.

3. Adaptive Network-based Fuzzy Inference System

4. Adaptive Neuro Fuzzy Inference System

درآمده است. پایگاه قواعد به شرح زیر است.

- قاعده ۱: اگر x_1 برابر A_1 و x_2 برابر B_1 باشد، آنگاه $f_1 = p_1x_1 + q_1x_2 + r_1$ است.
 قاعده ۲: اگر x_1 برابر A_1 و x_2 برابر B_2 باشد، آنگاه $f_2 = p_2x_1 + q_2x_2 + r_2$ است.
 قاعده ۳: اگر x_1 برابر A_2 و x_2 برابر B_1 باشد، آنگاه $f_3 = p_3x_1 + q_3x_2 + r_3$ است.
 قاعده ۴: اگر x_1 برابر A_2 و x_2 برابر B_2 باشد، آنگاه $f_4 = p_4x_1 + q_4x_2 + r_4$ است.

در قواعد فوق، A_1 ، A_2 و B_1 ، B_2 مقادیر زبانی x_1 و x_2 را نشان می‌دهند و p ، q و r نیز پارامترهای خروجی خطی در بخش تالی هر قاعده هستند. در ادامه ساختار پنج لایه‌ای ANFIS به تفکیک هر لایه تشریح می‌شود.

لایه ۱: لایه نخست یا لایه ورودی، وظیفه فازی‌سازی مقادیر ورودی را برعهده دارد. هر گره در این لایه، یک گره تطبیق‌پذیر دارای تابع عضویت مشخص است:

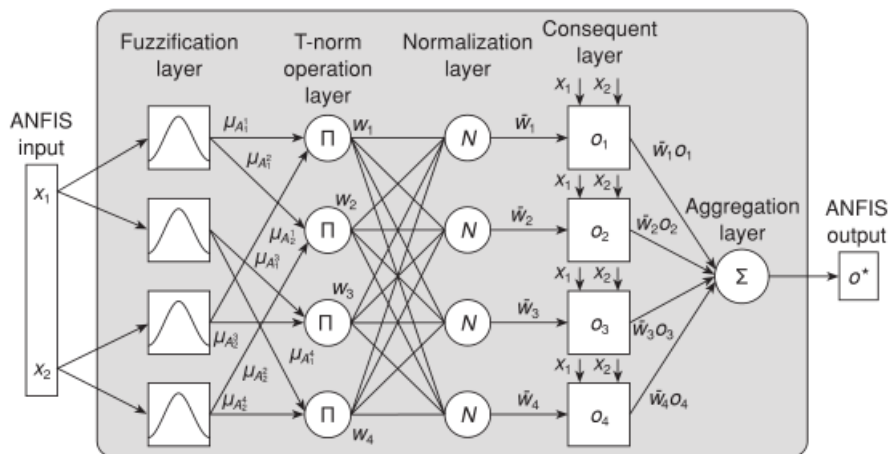
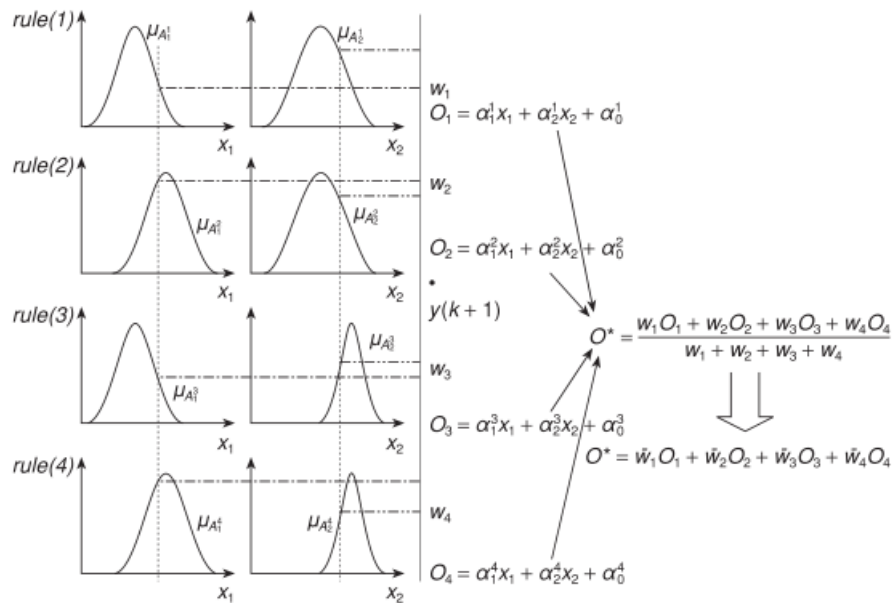
$$O_i^1 = \mu_{A_j}(x_1) \text{ یا } \mu_{B_j}(x_2) \quad (80-4)$$

x_1 و x_2 متغیرهای ورودی هستند و $\mu_{A_j}(x_1)$ و $\mu_{B_j}(x_2)$ نیز درجه تابع عضویت موردنظر را نشان می‌دهند. هر تابع عضویت پارامتری‌شده‌ای را می‌توان برای متغیرهای ورودی به کار برد، اما استفاده از تابع عضویت زنگوله‌ای یا تابع عضویت گاوسی متداول‌تر است:

$$\mu_{A_j}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_j}} \quad (81-4)$$

$$\mu_{A_j}(x) = \exp \left[- \left(\frac{x - c_i}{2a_i} \right)^2 \right] \quad (82-4)$$

$\{a_i, b_i, c_i\}$ مجموعه پارامترهای توابع عضویت ورودی است. پارامترهای این لایه تحت عنوان پارامترهای مقدم شناخته می‌شوند. همان‌طور که در شکل ۴-۱۴ مشخص شده، در اینجا دو تابع عضویت برای هر ورودی به کار گرفته شده است.



شکل ۴-۱۴: ساختار ANFIS و سازوکار استنتاج آن

لایه ۲: هر گره در این لایه، یک گره ثابت (غیر تطبیق‌پذیر) با برجسب Π است که قاعده مناظر را نشان می‌دهد و ضرب سیگنال‌های ورودی را به‌عنوان خروجی به لایه بعد می‌فرستد. به‌عبارت‌دیگر، هر گره شدت برانگیختگی یک قاعده را به‌صورت زیر تعریف

می‌کند.

$$w_i = \mu_{A_j}(x_1) * \mu_{B_j}(x_2) \quad j = 1, 2 \quad (۸۳-۴)$$

به‌طورکلی از هر عملگر t-نرم دیگری که AND فازی را پیاده‌سازی کند می‌توان به‌عنوان تابع در گره‌های این لایه استفاده کرد.

لایه ۳: هر گره در این لایه، یک گره ثابت (غیر تطبیق‌پذیر) است که با N نام‌گذاری شده و شدت برانگیختگی نرمال‌شده قاعده i را مطابق زیر محاسبه می‌کند.

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^4 w_i} \quad i = 1, 2, 3, 4 \quad (۸۴-۴)$$

به‌عبارت‌دیگر، در این لایه نسبت شدت برانگیختگی قاعده i به مجموع شدت برانگیختگی همه قواعد توسط گره i محاسبه می‌شود.

لایه ۴: هر گره در این لایه، یک گره تطبیق‌پذیر مبتنی بر یک تابع خطی است که به‌صورت زیر تعریف می‌شود.

$$f_i = p_i x_1 + q_i x_2 + r_i \quad i = 1, 2, 3, 4 \quad (۸۵-۴)$$

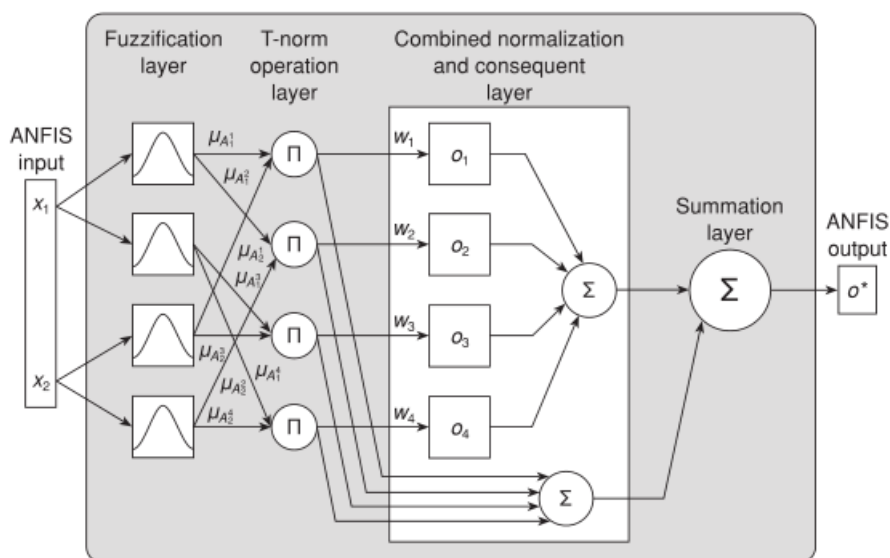
$\{p_i, q_i, r_i\}$ مجموعه پارامترهای این لایه هستند که با عنوان پارامترهای تالی شناخته می‌شود. هر گره مقدار موزون بخش تالی قاعده مربوطه‌اش را محاسبه می‌کند:

$$O_i^4 = \bar{w}_i \cdot f_i = \bar{w}_i \cdot (p_i x_1 + q_i x_2 + r_i) \quad i = 1, 2, 3, 4 \quad (۸۶-۴)$$

لایه ۵: این لایه شامل یک گره ثابت است که خروجی کلی را به‌وسیله انبوه‌سازی همه مقادیر قواعد فعال تولید می‌کند:

$$O^5 = \sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad i = 1, 2, 3, 4 \quad (۸۷-۴)$$

با به‌کارگیری گره‌های انطباقی در لایه ۱ و لایه ۴، یک شبکه انطباقی ایجاد شده است که از لحاظ عملکرد با یک مدل فازی سوگنو هم‌ارز است. این ساختار، منحصر به فرد نیست بلکه می‌توان با ترکیب لایه ۳ و ۴، شبکه مشابهی با چهار لایه ایجاد کرد. به این ترتیب، نرمال‌سازی وزن را می‌توان در لایه چهارم انجام داد. این ساختار در شکل ۴-۱۵ به نمایش درآمده است. در حالت افراطی، حتی می‌توان کل شبکه را به یک گره منفرد تطبیق‌پذیر با مجموعه پارامترهای مشابه تقلیل داد. تا زمانی که هر گره و هر لایه با قابلیت‌های مدولار و معنادار پیاده‌سازی شود، تعیین توابع گره‌ها و پیکره‌بندی شبکه به‌طور دلخواه انجام می‌گردد.



شکل ۴-۱۵: ساختار ANFIS با چهار لایه

با جایگزینی تابع تالی به وسیله یک تابع خطی چندضابطه‌ای پارامتری شده با دو پارامتر، می‌توان یک سیستم استنتاج فازی عصبی از نوع تسوکاموتو را ساخت. در ANFIS از نوع تسوکاموتو خروجی هر قاعده به‌طور توأمان توسط تابع عضویت تالی و شدت برانگیختگی محاسبه می‌شود. برای یک سیستم استنتاج ممدانی با ترکیب \max / \min می‌توان سیستم انطباقی متناظری را ایجاد کرد چنانچه از تقریب‌های گسسته برای جایگزینی انتگرال‌ها در نظام فازی‌زدایی مرکزوار استفاده شود. این ساختار

ANFIS بسیار پیچیده‌تر از هریک از ANFIS‌های سوگنو یا تسوکاموتو است. این پیچیدگی در ساختار یا محاسبات ANFIS ممدانی، لزوماً به قابلیت یادگیری یا قدرت تخمین بهتر منجر نمی‌شود. توابع عضویتی که مورد استفاده قرار می‌گیرند، باید به‌طور منطقی شکل داده شوند تا بتوان به‌منظور انعکاس و نمایش محدودیت‌های لازم، آنها را به‌طور صحیح پارامتری کرد. همچنین این انتظار وجود دارد که توابع عضویت بدون توجه به مقادیر پارامترشان زنگوله‌ای شکل باقی بمانند. سیستم‌هایی که در بخش ۴-۱-۱ تا ۴-۱-۳ تشریح شدند، شباهت زیادی به ANFIS دارند. تنها تفاوت این است که آنها هیچ گره انطباقی ندارند و هیچ سازوکار یادگیری برای تنظیم پارامترهای مقدم و تالی به‌کار گرفته نمی‌شود. اگر توابع عضویت در بخش مقدم ثابت نگه داشته شوند و پارامترهای تالی تطبیق یابند، آنگاه ANFIS به‌عنوان یک شبکه Functional-link (کلاسن^۱ و همکاران، ۱۹۸۸) تلقی شده که نمایش‌های تقویت‌یافته متغیرهای ورودی از طریق توابع عضویت حاصل می‌شود. ساختارهای فازی عصبی انطباقی مشابهی نیز به‌طور مستقل توسط لین^۲ و لی^۳ (۱۹۹۱) و وانگ^۴ و مندل^۵ (۱۹۹۲) پیشنهاد شده است. قابلیت تطبیق در ANFIS، موجب می‌شود که یادگیری و کنترل مسائل به‌طور مستقیم کاربردی شود.

الگوریتم یادگیری ترکیبی

در ساختار ANFIS، لایه اول و لایه چهارم شامل پارامترهایی هستند که می‌توان آنها را در طول زمان اصلاح کرد. لایه اول، یک رابطه غیرخطی از پارامترهای مقدم را دربردارد درحالی که لایه چهارم شامل پارامترهای تالی خطی است. به‌منظور به‌روزرسانی این پارامترها به یک الگوریتم یادگیری نیاز است که بتواند هر دو نوع پارامترها را آموزش دهد و آنها را با محیط‌شان سازگار کند. یک الگوریتم یادگیری ترکیبی^۶ توسط جانگ^۷ (۱۹۹۳) به‌منظور آموزش این پارامترها پیشنهاد شده است. سبب به‌کارگیری این الگوریتم آن است که الگوریتم پس‌انتشاری که برای آموزش پارامترهای موجود در شبکه‌های انطباقی مورد

1. Klassen, M. S.
2. Lin, T. C.
3. Lee, C. S.
4. Wang, L. X.
5. Mendel, J. M.
6. Hybrid learning algorithm
7. Jang, J. S. R.

استفاده قرار می‌گیرد، نرخ همگرایی کُندی دارد و در کمینه محلی به دام می‌افتد. الگوریتم یادگیری ترکیبی دو مرحله دارد که گذر پیش‌رو^۱ و گذر پس‌رو^۲ نامیده می‌شوند. طی گذر پیش‌رو، خروجی‌های گره تا لایه چهار منتشر شده و از یک روش تخمین‌زننده کمترین مربعات بازگشتی (RLSE)^۳ به منظور اصلاح پارامترهای تالی در لایه چهارم استفاده می‌شود. از آنجا که پارامترهای تالی خطی هستند، روش RLSE را می‌توان برای تسریع در نرخ همگرایی در فرآیند یادگیری ترکیبی به کار برد. طی گذر پیش‌رو، پارامترهای مقدم در لایه نخست باید ثابت باشند. پس از حاصل‌شدن پارامترهای تالی، گذر پس‌رو انجام می‌شود. زمانی که گذر پس‌رو اجرا می‌شود، پارامترهای تالی باید ثابت باشند. خطای حاصل از مقایسه بین خروجی ایجادشده با خروجی واقعی رویه عقب تا لایه یکم پس‌انتشار می‌یابد. در همین زمان، پارامترهای مقدم در لایه یکم با استفاده از روش‌های یادگیری کاهش گرادیان یا پس‌انتشار به‌روز می‌شوند. با به‌کارگیری الگوریتم یادگیری ترکیبی که روش‌های RSLE و کاهش گرادیان را ترکیب می‌کند، می‌توان مطمئن بود که نرخ همگرایی سریع‌تر می‌شود زیرا می‌توان فضای جستجوی ابعادی در روش اصلی پس‌انتشار را توسط این ترکیب کاهش داد (نایاک^۴ و همکاران، ۲۰۰۴). هر مرحله از یادگیری ترکیبی، دوره^۵ نامیده می‌شود. جدول ۴-۲ به‌طور خلاصه فرآیند یادگیری ترکیبی در ANFIS را بیان می‌کند.

نوع	گذر پیش‌رو	گذر پس‌رو
پارامتر مقدم	ثابت	کاهش گرادیان
پارامتر تالی	RSLE	ثابت
سیگنال	خروجی گره	نرخ خطا

جدول ۴-۲: فرآیند یادگیری ترکیبی

یادگیری در ANFIS را می‌توان به دو نظام یادگیری تقسیم نمود: یادگیری توابع عضویت مقدم و

1. Forward path
2. Backward path
3. Recursive Least Squares Estimator
4. Nayak, P. C.
5. Epoch

یادگیری پارامترهای تالی. شناسه‌هایی^۱ وجود دارد که توابع عضویت در ANFIS از نوع سوگنو نباید سازوکار یادگیری را به کارگیرند زیرا آنها یک توصیف ذهنی از مسأله ارائه می‌کنند. در صورتی که مجموعه داده‌های ورودی/خروجی موجود کوچک باشد، توابع عضویت باید ثابت نگه داشته شود. یادگیری توابع عضویت در این مورد سودمند نخواهد بود. اگر مجموعه داده‌های ورودی/خروجی بزرگ باشد، تنظیم دقیق توابع عضویت ضروری است به دلیل اینکه توابع عضویت تعیین شده توسط یک خبره ممکن است بهینه نباشد.

یادگیری پس‌انتشار برای پارامترهای مقدم

هسته مرکزی هر قاعده یادگیری در این شبکه انطباقی بر این متمرکز است که چگونه می‌توان به‌طور بازگشتی به یک بردار گرادیان دست یافت به‌گونه‌ای که هر عنصر در قالب مشتق مقدار خطا نسبت به یک پارامتر تعریف شود. این کار با بهره‌گیری از مفهوم قاعده زنجیره‌ای انجام‌پذیر است. این روش با عنوان قانون یادگیری پس‌انتشار^۲ شناخته می‌شود زیرا بردار گرادیان در خلاف جهت جریان خروجی هر گره محاسبه می‌شود. جزئیات این روش در ادامه تشریح شده است.

فرض کنید که یک شبکه انطباقی پیش‌خور^۳ با L لایه وجود داشته باشد که لایه l دارای $N(l)$ گره است که در آن، $l = 0, 1, \dots, L$ بوده و $l = 0$ لایه مربوط به ورودی‌ها را نشان می‌دهد. خروجی و تابع گره i (با در نظر گرفتن $i = 1, \dots, N(l)$) از لایه l را می‌توان به ترتیب با x_i^l و f_i^l نشان داد. از آنجایی که خروجی یک گره به سیگنال‌های ورودی و مجموعه پارامترهای گره بستگی دارد، لذا عبارت عمومی زیر بیان‌گر تابع گره خواهد بود:

$$x_i^l = f_i^l(x_1^{l-1}, \dots, x_{N(l-1)}^{l-1}, \alpha, \beta, \gamma, \dots) \quad (88-4)$$

که α, β, γ و ... پارامترهای مربوط به این گره هستند.

در صورتی که داده‌های آموزشی دارای P عنصر باشند، می‌توان مقدار خطا (شاخص خطا) را برای

1. Argumants
2. Back-propagation learning rule
3. Feedforward

عنصر p ($1 \leq p \leq P$) از داده‌های آموزشی به صورت مجموع مربعات خطا تعریف کرد:

$$E_p = \sum_{k=1}^{N(L)} (d_{k,p} - x_{k,p}^L)^2 \quad (۸۹-۴)$$

$d_{k,p}$ نماینده جزء k از بردار خروجی مطلوب p است و $x_{k,p}^L$ جزء k از بردار خروجی واقعی است که با ارائه بردار ورودی p به شبکه حاصل می‌شود. بدیهی است که هدف در اینجا، حداقل کردن مجموع مقادیر خطای کل است که به صورت $E = \sum_{p=1}^P E_p$ تعریف می‌شود. همچنین باید به یاد داشت که تعریف E_p در رابطه ۴-۸۹ همیشگی نیست و می‌توان برحسب شرایط و کاربردهای موردنظر از تعاریف دیگری نیز استفاده نمود.

به منظور ایجاد یک فرآیند یادگیری که از کاهش گرادیان در این فضای پارامتری استفاده کند، ابتدا باید نرخ خطا $(\frac{\partial E_p}{\partial x})$ را برای داده آموزشی p و برای خروجی محاسبه کرد. نرخ خطا برای گره خروجی i (در لایه L) مستقیماً به صورت زیر محاسبه می‌شود.

$$\epsilon_{i,p}^L = \frac{\partial E_p}{\partial x_{i,p}^L} \quad (۹۰-۴)$$

اگر E_p بر اساس رابطه ۴-۸۹ تعریف شود، رابطه ۴-۹۰ به صورت $\epsilon_{i,p}^L = -2(d_{i,p} - x_{i,p}^L)$ خواهد بود. نرخ خطا برای گره‌های داخلی (غیر خروجی) را می‌توان با استفاده از قاعده زنجیره‌ای به دست آورد. لذا برای گره i از لایه l ، نرخ خطا به فرم زیر در می‌آید.

$$(۹۱-۴)$$

$$\epsilon_{i,p}^l = \frac{\partial E_p}{\partial x_{i,p}^l} = \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial x_{m,p}^{l+1}} \frac{\partial f_{m,p}^{l+1}}{\partial x_{i,p}^l} = \sum_{m=1}^{N(l+1)} \epsilon_{m,p}^{l+1} \frac{\partial f_{m,p}^{l+1}}{\partial x_{i,p}^l} \quad 0 \leq l \leq L-1$$

یعنی نرخ خطای یک گره داخلی در لایه l را می‌توان به صورت یک ترکیب خطی از نرخ خطای گره‌های لایه $l+1$ بیان کرد. بنابراین به ازای هر l و i ($0 \leq l \leq L$ و $l \leq i \leq N(l)$) می‌توان ابتدا مقدار $\epsilon_{i,p}^L = \frac{\partial E_p}{\partial x_{i,p}^L}$ را با استفاده از رابطه ۴-۹۰ یافته و سپس با به کارگیری مکرر رابطه ۴-۹۱ به لایه موردنظر l برسیم. از آنجایی که نرخ‌های خطا به ترتیب از لایه خروجی تا لایه ورودی به دست می‌آیند، این الگوی یادگیری را قاعده یادگیری پس‌انتشار می‌نامند.

بردار گرادیان به صورت مشتق مقدار خطا نسبت به هر پارامتر تعریف می‌شود و لذا باید قاعده زنجیره‌ای را مجدداً برای یافتن بردار گرادیان به کار گرفت. اگر α یک پارامتر از گره i در لایه l باشد،

خواهیم داشت:

$$\frac{\partial E_p}{\partial \alpha} = \frac{\partial E_p}{\partial x_{i,p}^l} \frac{\partial f_{i,p}^l}{\partial \alpha} = \epsilon_{i,p}^l \frac{\partial f_{i,p}^l}{\partial \alpha} \quad (92-4)$$

اگر α را به‌طور اشتراکی بین گره‌های مختلف بپذیریم، آنگاه رابطه ۹۲-۴ به یک صورت عمومی‌تر تغییر می‌کند:

$$\frac{\partial E_p}{\partial \alpha} = \sum_{x^* \in S} \frac{\partial E_p}{\partial x^*} \frac{\partial f^*}{\partial \alpha} \quad (93-4)$$

S مجموعه گره‌هایی است که α را به‌عنوان پارامتر دربردارند و f^* تابع گره برای محاسبه x^* است. بنابراین مشتق مقدار کلی خطا (E) نسبت به α برابر است با:

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad (94-4)$$

به‌این ترتیب و براساس روش تندترین شیب کاهش گرادیان، به‌روزرسانی عمومی α به‌صورت زیر انجام می‌شود.

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \quad (95-4)$$

در رابطه فوق η نرخ یادگیری است که می‌توان براساس تساوی زیر آن را تعریف کرد.

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \quad (96-4)$$

k اندازه گام^۱ است که طول هر انتقال در راستای گرادیان را نشان می‌دهد. به‌منظور سرعت‌بخشیدن به نرخ همگرایی، می‌توان آن را تغییر داد.

یادگیری برای پارامترهای تالی با RSLE

1. Step size

در شرایطی که پارامترهای مقدم در وضعیت پایدار و ثابت هستند، می‌توان خروجی را در قالب یک ترکیب خطی از پارامترهای تالی بیان کرد:

$$Y = \bar{w}_1 y_1 + \bar{w}_2 y_2 + \bar{w}_3 y_3 + \bar{w}_4 y_4 \quad (97-4)$$

$$\begin{aligned} &= \bar{w}_1 (p_1 x_1 + q_1 x_2 + r_1) + \bar{w}_2 (p_2 x_1 + q_2 x_2 + r_2) \\ &\quad + \bar{w}_3 (p_3 x_1 + q_3 x_2 + r_3) + \bar{w}_4 (p_4 x_1 + q_4 x_2 + r_4) \\ &= (\bar{w}_1 x_1) p_1 + (\bar{w}_1 x_2) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x_1) p_2 + (\bar{w}_2 x_2) q_2 + (\bar{w}_2) r_2 \\ &\quad + (\bar{w}_3 x_1) p_3 + (\bar{w}_3 x_2) q_3 + (\bar{w}_3) r_3 + (\bar{w}_4 x_1) p_4 + (\bar{w}_4 x_2) q_4 + (\bar{w}_4) r_4 \end{aligned}$$

زمانی که N داده آموزشی به رابطه فوق داده شود، آنگاه این رابطه به صورت زیر حاصل خواهد شد.

$$(98-4)$$

$$\begin{aligned} &(\bar{w}_1 x_1)_1 p_1 + (\bar{w}_1 x_2)_1 q_1 + (\bar{w}_1)_1 r_1 + (\bar{w}_2 x_1)_1 p_2 + (\bar{w}_2 x_2)_1 q_2 + (\bar{w}_2)_1 r_2 \\ &\quad + (\bar{w}_3 x_1)_1 p_3 + (\bar{w}_3 x_2)_1 q_3 + (\bar{w}_3)_1 r_3 + (\bar{w}_4 x_1)_1 p_4 + (\bar{w}_4 x_2)_1 q_4 + (\bar{w}_4)_1 r_4 = y_1 \\ &\quad \vdots \\ &(\bar{w}_1 x_1)_n p_1 + (\bar{w}_1 x_2)_n q_1 + (\bar{w}_1)_n r_1 + (\bar{w}_2 x_1)_n p_2 + (\bar{w}_2 x_2)_n q_2 + (\bar{w}_2)_n r_2 \\ &\quad + (\bar{w}_3 x_1)_n p_3 + (\bar{w}_3 x_2)_n q_3 + (\bar{w}_3)_n r_3 + (\bar{w}_4 x_1)_n p_4 + (\bar{w}_4 x_2)_n q_4 + (\bar{w}_4)_n r_4 = y_n \end{aligned}$$

برای سادگی، رابطه ۹۸-۴ را می‌توان به فرم ماتریسی زیر بیان کرد.

$$A\theta = B \quad (99-4)$$

در اینجا، θ بردار پارامتری نامعلوم با ابعاد $1 \times M$ است و M به تعداد عناصری اشاره دارد که مجموعه پارامترهای تالی هستند. A ماتریس ضرایب را نشان می‌دهد و ابعاد آن $P \times M$ است. B نیز بردار خروجی با ابعاد $1 \times P$ است. از آنجایی که تعداد زوج‌های آموزشی (P) از تعداد پارامترهای خطی (M) بزرگتر است، جواب یکتایی برای تساوی ۹۹-۴ وجود ندارد. در عوض، یک روش تخمین کمترین مربعات (LSE) به منظور تعیین بردار پارامتری θ که $\|A\theta - B\|^2$ را کمینه می‌کند، مورد استفاده واقع می‌شود:

$$\theta^* = (A^T A)^{-1} A^T B \quad (100-4)$$

استفاده از یک روش LSE بازگشتی، رابطه ۴-۱۰۰ به صورت زیر تبدیل می‌شود.

$$\left. \begin{aligned} \theta_{i+1} &= \theta_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \theta_i) \\ S_{i+1} &= S_i - \frac{S_i + a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, \quad i = 0, 1, \dots, P-1 \end{aligned} \right\} \quad (101-4)$$

a_i^T بردار سطری i از ماتریس A در رابطه ۴-۹۹ است و b_i عنصر i از B را نشان می‌دهد. گاهی S_i را یک ماتریس کوواریانس می‌نامند و به فرم زیر تعریف می‌کنند.

$$S_i = (A^T A)^{-1} \quad (102-4)$$

مقادیر اولیه برای اجرای رابطه ۴-۱۰۱ شامل $\theta_0 = 0$ و $S_0 = \gamma I$ می‌شود که γ یک عدد بزرگ مثبت و I ماتریس همانی (یکه) با ابعاد $M \times M$ است.

رویه یادگیری فوق براساس رویکرد یادگیری دسته‌ای^۱ یا یادگیری آفلاین^۲ بنا شده است. در یادگیری آفلاین، به روزرسانی پارامتر α مبتنی بر رابطه ۴-۹۴ است و فرآیند به روزرسانی فقط هنگامی که همه داده‌های آموزشی ارائه شده باشند (یعنی فقط پس از هر دوره)، انجام می‌شود. از سوی دیگر، اگر بخواهیم پارامترها را بلافاصله پس از ارائه هر زوج ورودی-خروجی به روزرسانی کنیم، فرمول به روزرسانی بر رابطه ۴-۹۳ استوار خواهد بود. این حالت، به یادگیری الگو^۳ یا یادگیری آنلاین^۴ مشهور است.

به منظور تبدیل قاعده یادگیری دسته‌ای به نسخه آنلاین آن، کاهش گرادیان باید براساس E_p صورت بگیرد. در حقیقت، این یک روش جستجوی گرادیان برای کمینه‌سازی E نیست، اما می‌تواند در شرایطی که نرخ یادگیری کوچک باشد، تخمینی را ارائه کند. به منظور لحاظ کردن تفاوت زمانی داده‌های ورودی، نیاز است که اثرات زوج داده‌های قدیمی در قالب ظهور زوج داده‌های جدید از بین برود. یک روش ساده برای این کار، فرموله کردن مقدار مجذور خطا در قالبی موزون است که موجب می‌شود زوج داده‌های اخیر دارای ضرایب وزنی بیشتری شوند. به همین منظور، می‌توان ضریب فراموشی λ را به

1. Batch learning
2. Off-line learning
3. Pattern learning
4. On-line learning

فرمول بازگشتی اصلی اضافه نمود:

$$\left. \begin{aligned} \theta_{i+1} &= \theta_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \theta_i) \\ S_{i+1} &= \frac{1}{\lambda} \left[S_i - \frac{S_i + a_{i+1} a_{i+1}^T S_i}{\lambda + a_{i+1}^T S_i a_{i+1}} \right] \end{aligned} \right\} \quad (103-4)$$

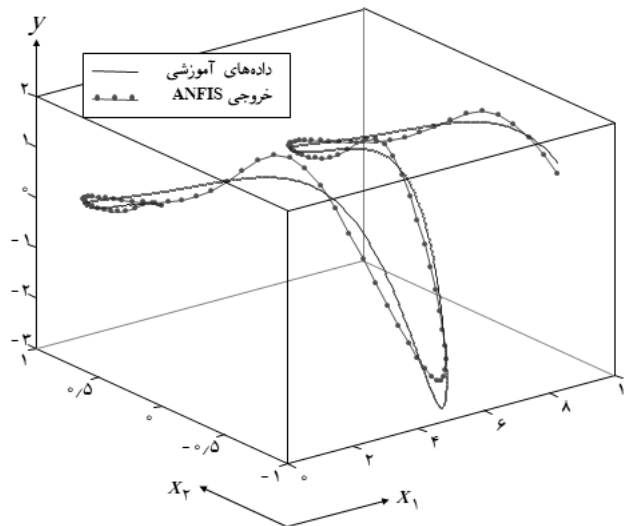
مقدار λ بین ۰/۹ و ۱ است. هرچه مقدار λ کمتر باشد، موجب حذف سریع‌تر اثرات داده‌های قدیمی خواهد شد اما مقدار کوچک λ گاهی موجب ناپایداری عددی می‌شود که باید از آن اجتناب نمود. از آنجایی که در یک قاعده یادگیری ترکیبی پارامترهای مقدم و تالی جدا می‌شوند، با استفاده از روش‌های گرادینان دیگر یا سایر تکنیک‌های بهینه‌سازی امکان تسریع بیشتری در یادگیری وجود دارد.

مثال ۴-۲. در این مثال، تخمین تابع به کمک مدل ANFIS مورد بررسی قرار می‌گیرد که در آن، مسیر یک تابع غیرخطی توسط ANFIS دنبال می‌شود. تعریف این تابع به صورت زیر است:

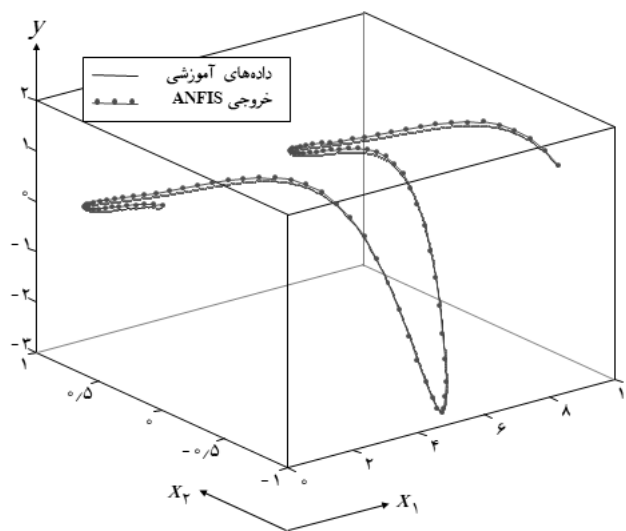
$$y = \frac{\cos(2x_1)}{e^{x_2}}$$

ابتدا یک ساختار مناسب برای ANFIS انتخاب می‌کنیم که باید دو ورودی (x_1, x_2) و یک خروجی (y) داشته باشد. در اینجا، چهار قاعده تعریف می‌شود و داده‌های آموزشی نیز شامل ۱۰۱ نمونه آموزشی است. این مجموعه با یک ماتریس 3×101 به فرمت $[x_1 \ x_2 \ y_d]$ نشان داده می‌شود که x_1 و x_2 بردارهای ورودی و y_d بردار خروجی مطلوب هستند. x_1 از ۰ شروع شده و با گام‌هایی به اندازه ۰/۱ تا عدد ۱۰ افزایش می‌یابد. x_2 با اعمال تابع \sin روی هریک از عناصر بردار x_1 ایجاد می‌شود. در نتیجه، مقدار d_y نیز براساس رابطه فوق حاصل می‌شود.

مدل ANFIS طراحی شده، در دو حالت مجزا مورد ارزیابی قرار می‌گیرد که هریک طی ۱ تکرار و ۱۰۰ تکرار آموزش می‌بینند. فرآیند و خروجی یادگیری برای حالت اول در شکل ۴-۱۶ نمایش داده شده و برای حالت دوم نیز در شکل ۴-۱۷ به تصویر کشیده شده است. در حالت اول از ۲ تابع عضویت زنگوله‌ای برای هریک از ورودی‌ها و از تابع عضویت خطی برای خروجی استفاده می‌شود. در حالت دوم از ۳ تابع عضویت زنگوله‌ای برای هریک از ورودی‌ها و از تابع عضویت خطی برای خروجی استفاده می‌شود.

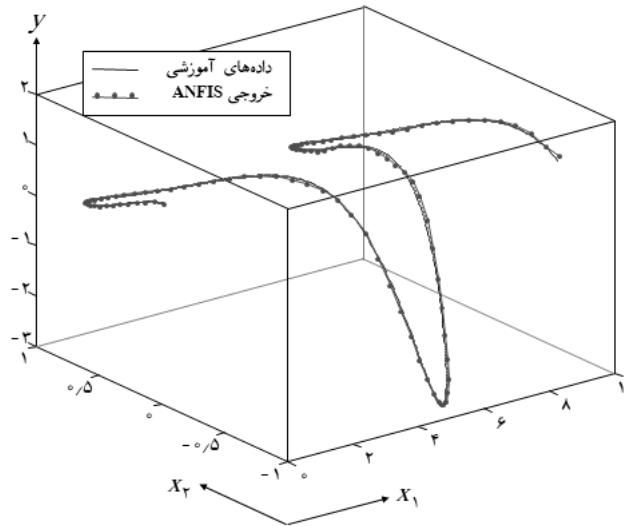


(آ) یادگیری طی ۱ تکرار

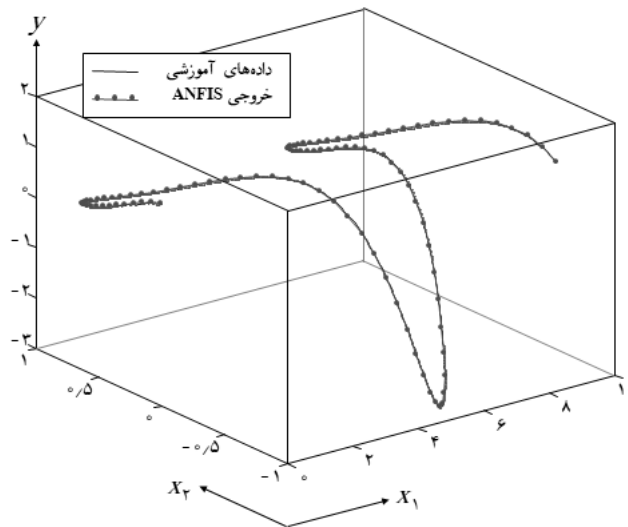


(ب) یادگیری طی ۱۰۰ تکرار

شکل ۴-۱۶: یادگیری در مدل ANFIS با دو تابع عضویت برای هر ورودی



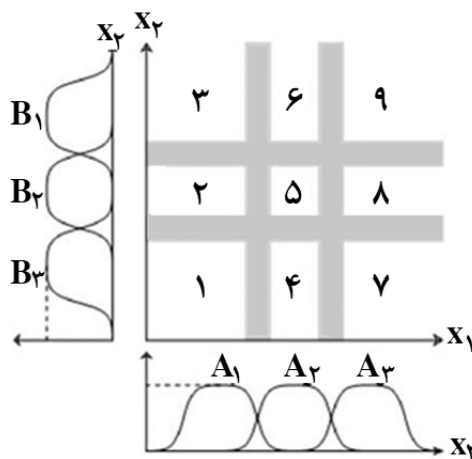
(آ) یادگیری طی ۱ تکرار



(ب) یادگیری طی ۱۰۰ تکرار

شکل ۴-۱۷: یادگیری در مدل ANFIS با سه تابع عضویت برای هر ورودی

نتایج به‌طور آشکار نشان می‌دهد که با افزایش تکرارهای آموزش می‌توانیم بهبود در عملکرد را شاهد باشیم؛ با این حال زمانی که ۳ تابع عضویت به هر متغیر ورودی اختصاص پیدا کند، نتایج بهتری به دست آمده است. در این حالت، از مدل ANFIS با ۹ قاعده استفاده کرده‌ایم. مطابق شکل ۴-۱۸ فضای ورودی دو بُعدی به ۹ ناحیه فازی دارای هم‌پوشانی افزاز شده که هریک توسط یک قاعده اگر-آنگاه فازی تحت تأثیر قرار گرفته‌اند. به عبارت دیگر، بخش مقدم از یک قاعده، یک ناحیه فازی را تعریف می‌کند در حالی که بخش تالی، خروجی را در این ناحیه مشخص می‌کند (شکل ۴-۱۷).



شکل ۴-۱۸: تبدیل فضای ورودی ۲ بُعدی به ۹ ناحیه فازی دارای هم‌پوشانی

موردکاوی

افزایش انتشار گاز کربن دی‌اکسید (CO_2) نقش کلیدی در گرم‌شدن کره زمین دارد که به یکی از چالش‌های اصلی دنیای جدید تبدیل شده و بنابراین، حرکت به سمت کاهش انتشار CO_2 در صنایع مختلف اجتناب‌ناپذیر است. جداسازی CO_2 از گاز طبیعی برای جلوگیری از مشکلات عملیاتی نظیر انجماد قبل از مایع شدن، کاهش کمپرسورهای انتقال گاز طبیعی و کاهش خوردگی در حضور آب ضروری است. برای جذب CO_2 از گاز طبیعی، چندین فناوری

مانند جداسازی غشایی، تقطیر برودتی، جذب سطحی، تشکیل هیدرات و ... وجود دارد. در میان آنها، جذب شیمیایی توسط حلال‌های آمینو اسید رایج‌ترین فناوری در حذف CO₂ است. طراحی فرآیندهای مبتنی بر این حلال‌ها به مدل‌های مناسب برای پیش‌بینی پارامترهای ترمودینامیکی نیاز دارد که در همین رابطه پژوهش‌های متعددی صورت گرفته است.

سروش و همکاران (۲۰۱۹) طی مطالعه‌ای، از ANFIS برای ایجاد یک مدل هوشمند وابسته به دما برای برآورد (پیش‌بینی) ظرفیت بارگذاری CO₂ محلول‌های نمک آمینو اسید اسفاده کرده‌اند. در این مطالعه، هر نمک آمینو اسید در قالب یک ماتریس از ویژگی‌های فیزیکی آن به مدل ارائه شد. متغیرهای ورودی شامل دمای سیستم، غلظت جرمی کل محلول، فشار جزئی تعادلی CO₂، وزن مولکولی آمینو اسید، نقطه ذوب نمک آمینو اسید و وزن مولکولی کاتیون بوده و متغیر خروجی نیز بارگذاری CO₂ بوده است. پس از تعیین ورودی و خروجی‌های مدل، جمعاً ۸۳۶ نمونه روی محلول‌های مختلف آمینو اسید (جاذب) ایجاد شد که ۸۰ درصد از آنها تحت عنوان نمونه‌های آموزشی و ۲۰ درصد تحت عنوان نمونه‌های آزمون تقسیم‌بندی شدند. طی فرآیند آموزش ANFIS، از تقسیم‌بندی شبکه‌ای (grid partitioning) برای ایجاد سیستم استنتاج فازی تاکاگی-سوگنو استفاده شد. نوابغ عضویت هر ورودی به تعداد ۲ عدد و از نوع گاوسی انتخاب شده و تابع عضویت خروجی نیز از نوع خطی تعیین شد.

پس از آموزش شبکه فوق، از R^2 ، MSE و AARD به عنوان شاخص‌های ارزیابی خطا و عملکرد شبکه استفاده شد که به‌طورمثال، شاخص R^2 دقتی بیش از ۹۸ درصد را بر روی نمونه‌های آموزشی و آزمون نشان داده است. شاخص‌های ارزیابی خطا به تفکیک بر روی هر محلول آمینو اسید نیز گزارش شده که در شاخص R^2 دقتی بین ۸۹/۳۴ تا ۹۹/۶۴ درصد در پیش‌بینی را حاصل کرده است. در پایان این پژوهش، میزان اثرگذاری هر متغیر ورودی براساس داده‌های تجربی و نیز داده‌های پیش‌بینی مدل مورد بررسی قرار گرفته است.

Soroush, E., Mesbah, M., Hajilary, N. & Rezakazemi, M. (2019). ANFIS modeling for prediction of CO₂ solubility in potassium and sodium based amino acid Salt solutions. *Journal of Environmental*

Chemical Engineering, 7(1), 102925.

موردکاوی

در یک مقاله پژوهشی که در سال ۲۰۲۱ در Scientific Reports به چاپ رسید، از ANFIS برای تشخیص خودکار عفونت ناشی از کووید ۱۹ (COVID-19) استفاده شده است. در این مقاله، یک فرآیند طبقه‌بندی باینری مبتنی بر GLCM بر روی تصاویر رادیوگرافی قفسه سینه (CXR) انجام شده که در ادامه جزئیاتی از آن ذکر می‌شود.

در این مطالعه، در مرحله پیش‌پردازش تصاویر CXR از ماتریس هم‌رخداد (GLCM) استفاده می‌شود تا ویژگی‌های مورد نیاز از تصاویر استخراج شود. این ماتریس رابطه بین شدت‌های پیکسلی را در فاصله و جهت خاص مشخص می‌سازد. این روش رابطه پیکسلی فضایی را در تصویر در نظر گرفته و اصطلاحاً بافت تصویر را توصیف می‌کند. پیش از اعمال آن در این پژوهش، هر تصویر در مجموعه داده‌ها به خاکستری تبدیل شده و اندازه آن به 512×512 تغییر می‌کند و سپس برای کاهش پیچیدگی محاسباتی نرمال می‌شود. پس از اعمال ماتریس هم‌رخداد، چهار ویژگی همبستگی، کنتراست، انرژی و همگنی استخراج شده که پژوهشگران ویژگی‌های میانگین و واریانس را نیز از هیستوگرام تصویر به آنها اضافه می‌کنند. این شش ویژگی برای هر تصویر محاسبه شده تا داده‌های ورودی به ANFIS آماده طبقه‌بندی شوند. برچسب هر تصویر بودن یا نبودن کووید ۱۹ براساس تشخیص قبلی است.

از آنجا که در این پژوهش، طبقه‌بندی باینری را دنبال شده است، دو طبقه COVID-19 و non COVID-19 در مدل ارائه شده وجود دارد که هریک شامل ۱۰۰۰ تصویر است. بنابراین دیتاست موردنظر مجموعاً ۲۰۰۰ نمونه دارد. این نمونه‌ها در مرحله آموزش و آزمون به ترتیب به دو دسته ۷۰ درصد و ۳۰ درصد تقسیم می‌شوند.

بالترین دقتی که در نتایج این پژوهش گزارش شده است، ۹۸/۶۷ درصد بوده که براساس

نسبت طبقه‌بندی صحیح بر کل طبقه‌بندی‌ها محاسبه شد. از سوی دیگر، مقادیر مربوط به شاخص‌های حساسیت و ویژگی که در ارزیابی مسائل طبقه‌بندی رایج هستند نیز در این مقاله گزارش شده است. بر اساس جدول مقایسه‌ای که در نتایج این پژوهش ذکر شده است، دقت مدل پیشنهادی از بسیاری از پژوهش‌های مشابه بالاتر بوده و حتی در بسیاری از موارد از دقت شبکه‌های پیچیده‌ای همچون شبکه‌های مبتنی بر یادگیری عمیق نیز بهتر عمل کرده است.

Al-Ali, A., Elharrouss, O., Qidwai, U. & Al-Maaddeed, S. (2021). ANFIS-Net for automatic detection of COVID-19. *Scientific Reports*, 11(1), 1-13.

مسئله بیش‌برازش در ANFIS وقتی تعداد پارامترهای ANFIS بیشتر یا مساوی تعداد داده‌های ورودی باشد، همگرایی خیلی خوب در آموزش حاصل می‌شود اما ممکن است قابلیت تعمیم ضعیف باشد و سیستم استنتاج فازی ایجادشده، بر روی داده‌های اعتبارسنجی یا آزمون معتبر نباشد. این مورد با عنوان بیش‌برازش^۱ شناخته می‌شود که باید از آن جلوگیری کرد. همیشه باید تلاش کرد که شبکه ANFIS با کمترین تعداد پارامتر ممکن ایجاد شود تا به حداقل تعداد قواعد اگر-آنگاه فازی بیانجامد. تعداد قواعد اگر-آنگاه فازی به طور مستقیم به تعداد توابع عضویت برای هر متغیر وابسته است. مثلاً اگر ۴ متغیر در فضای ورودی وجود داشته باشد و برای هر یک دو تابع عضویت تعریف شده باشد، تعداد قواعد اگر-آنگاه فازی برابر با $2^4 = 16$ خواهد بود. به همین ترتیب، اگر فقط یک تابع عضویت به متغیر اضافه شود، تعداد قواعد فازی به $2^4 = 16$ افزایش خواهد یافت. بنابراین مسئله بیش‌برازش باید همواره مورد توجه باشد و راهکارهایی نظیر شرط توقف روی داده‌های اعتبارسنجی به کار گرفته شوند.

۲-۶-۴ سیستم استنتاج فازی عصبی هم‌کنش (CANFIS)

یکی از محدودیت‌های ANFIS این است که سیستم استنتاج فازی سوگنو در آن تنها باید یک خروجی داشته باشد. در نتیجه، کاربرد ANFIS فقط به سیستم‌های چندورودی-تک خروجی (MISO)^۲ محدود

1. Overfitting

2. Multiple-Input Single-Output

می‌شود و برای سیستم‌های چندورودی-چندخروجی (MIMO)^۱ با چالش همراه است. برای به‌کارگیری ANFIS چندگانه (MANFIS)^۲ که هر ANFIS یک مجموعه مستقل از قواعد فازی و پارامترهای قابل اصلاح و تعدیل است، تلاش‌هایی انجام شده است. در MANFIS، یافتن همبستگی بین ورودی‌ها و خروجی‌ها دشوار است. همچنان که تعداد خروجی‌ها افزایش می‌یابد، پارامترهای قابل تعدیل نیز شدیداً افزایش پیدا می‌کنند. مسأله این است که چگونه تولید خروجی‌های چندگانه از یک سیستم ANFIS را بیابیم. تعمیمی از ANFIS وجود دارد که CANFIS^۳ نامیده می‌شود و در آن، یک شبکه عصبی و یک سیستم استنتاج فازی نقش فعالی را بازی می‌کنند (میزوتانی^۴ و همکاران، ۱۹۹۴؛ میزوتانی و جانگ^۵، ۱۹۹۵). ساختار CANFIS با دو ورودی و یک خروجی در شکل ۴-۱۹ نمایش داده شده است. CANFIS در اینجا شامل دو قسمت است: یک مدل فازی (قسمت بالای شکل) که وزن‌های نرمال‌شده بخش تالی قواعد را محاسبه می‌کند، و یک مدل شبکه عصبی لایه‌ای (قسمت پایین شکل) که با استفاده از وزن‌های حاصل از سیستم فازی MIMO، خروجی‌های تالی را به دست می‌آورد. مدل سیستم فازی از سه لایه تشکیل شده که در ادامه تشریح می‌شود.

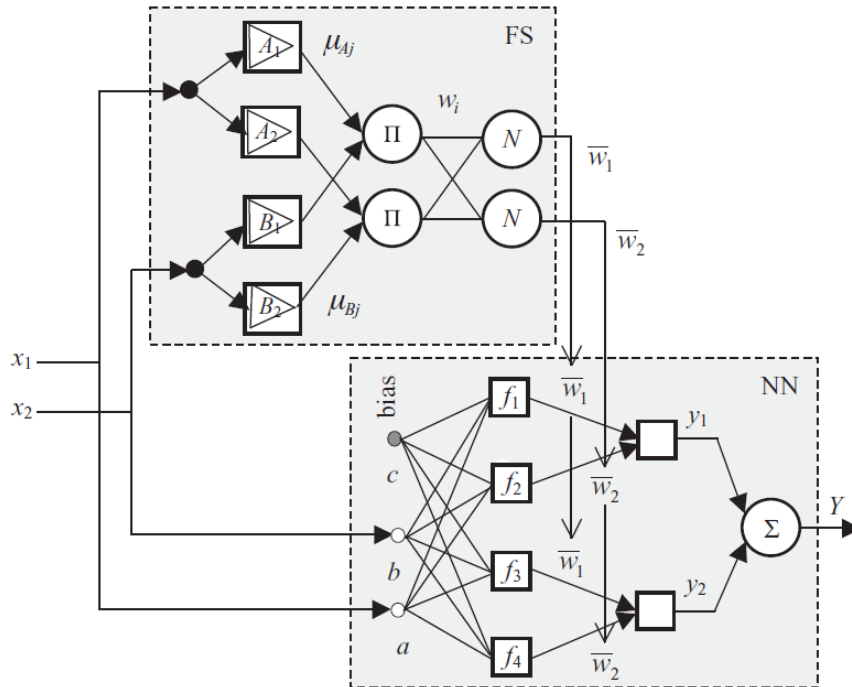
لایه ۱: این لایه، لایه فازی‌سازی است که درجه عضویت $\mu_{A_j}(x_k)$ و $\mu_{B_j}(x_k)$ را برای دو متغیر ورودی $\{x_1, x_2\}$ محاسبه می‌کند.

لایه ۲: لایه دوم شدت برانگیختگی قواعد مجزا را مطابق رابطه زیر تعیین می‌کند.

$$w_i = \Gamma(\mu_{A_j}(x_k), \mu_{B_j}(x_2)) \quad i = 1, 2, \dots, M (= 2 \times 2) \quad (4-104)$$

Γ نرون فازی‌ای را نشان می‌دهد که می‌تواند یک عمل \min یا ضرب را ایفا کند. M تعداد قواعد را در این سیستم فازی نشان می‌دهد.

-
1. Multiple-Input Multiple-Output
 2. Multiple ANFIS
 3. Coactive Neuro-Fuzzy Inference System (CANFIS)
 4. Mizutani, E.
 5. Jang, J. S. R.



شکل ۴-۱۹: ساختار CANFIS

لایه ۳: شدت برانگیختگی (w_i) نرمال شده به صورت زیر حاصل می‌شود.

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^M w_i} \quad (4-105)$$

مدل شبکه عصبی بیان‌کننده بخش تالی قواعد سیستم تاکاگی-سوگنو است و خروجی

لایه ۱: گره‌های این لایه، ورودی‌ها $\{x_1, x_2\}$ را به لایه بعدی انتقال می‌دهند. وزن‌های

اتصال، پارامترهای تالی $\{a_i, b_i, c_i\}$ از سیستم تاکاگی-سوگنو هستند.

لایه ۲: گره‌های این لایه، گره‌های مجموع‌یاب هستند و تابع خطی بخش تالی قواعد را

محاسبه می‌کنند:

$$f_i = a_i x_1 + b_i x_2 + c_i \quad i = 1, 2, 3, 4 \quad (106-4)$$

این گره‌ها بخش‌های تالی تا $f_i = a_1 x_1 + b_1 x_2 + c_1$ تا $f_i = a_4 x_1 + b_4 x_2 + c_4$ را به دست می‌آورند.

لایه ۳: این گره‌ها مطابق مدل شبکه عصبی در شکل ۴-۱۹ با استفاده از توابع تالی (f_i) که به وسیله شدت‌های برانگیختگی نرمال شده قواعد (\bar{w}_i) موزون شده‌اند، خروجی‌ها (y_i) را محاسبه می‌کنند:

$$y_1 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad \text{و} \quad y_2 = \bar{w}_3 f_3 + \bar{w}_4 f_4 \quad (107-4)$$

لایه ۴: خروجی کل مطابق رابطه زیر حاصل می‌شود.

$$Y = y_1 + y_2 \quad (108-4)$$

مدل شبکه عصبی در ساختار CANFIS، یک شبکه MLP پس‌انتشار است که نتایج قواعد را مشخص می‌کند. وزن‌های اتصال، متغیرهای عددی هستند که شدت‌های برانگیختگی را ارائه می‌کنند. تعداد قواعد در CANFIS توسط لایه پنهان بیان می‌شود. وزن‌های اتصال بین لایه پنهان و لایه خروجی با مقادیر عضویت بین لایه تالی و لایه انجمنی فازی متناظر است. مقادیر عضویت که به صورت پویا تغییر می‌یابند، به الگوهای ورودی بستگی دارند. بنابراین CANFIS به طور محلی تنظیم می‌شود. این همان موردی است که قابلیت و توانمندی CANFIS از آنجا سرچشمه می‌گیرد. این امکان وجود دارد که بدون افزایش قواعد یا توابع عضویت، با استفاده از یک تابع تالی غیرخطی نظیر تابع سیگموئید - مانند زیر - عملکرد بهبود یابد.

$$f = \frac{1}{1 + \exp(ax_1 + bx_2 + c)} \quad (109-4)$$

با قراردادن ANFIS در کنار تعداد مورد نیاز خروجی‌ها، می‌توان CANFIS را به خروجی‌های چندگانه گسترش داد. این ترکیب، ANFIS چندگانه (MANFIS) نام دارد. مشکل MANFIS این است که هر

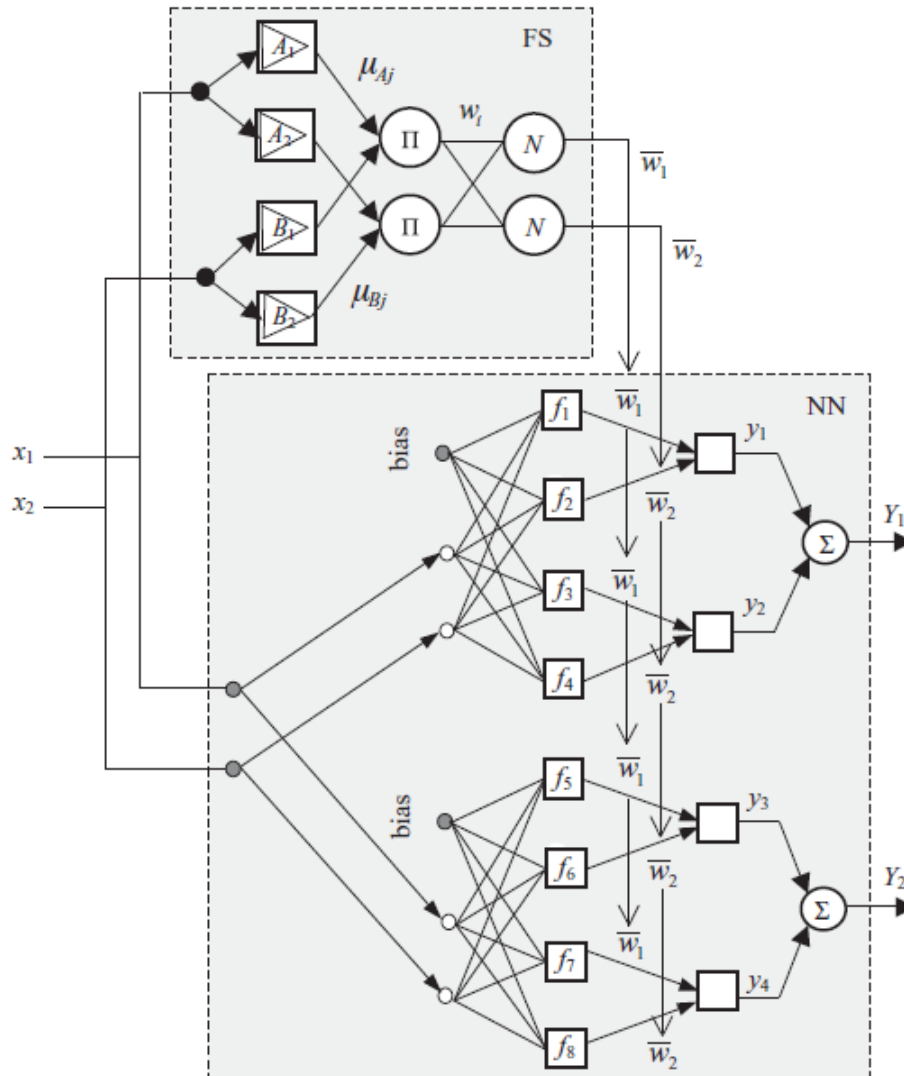
ANFIS دارای قواعد فازی خودش است و هیچ پارامتر قابل تعدیلی توسط ANFIS به اشتراک گذاشته نمی‌شود. با افزایش تعداد خروجی‌ها در MANFIS پارامترهای تعدیل‌پذیر به شدت افزایش می‌یابند. یک روش هوشمندانه برای طراحی خروجی‌های چندگانه، اشتراک‌گذاری پارامترها با استفاده از قواعد فازی مقدم یکسان است. از این رو قواعد فازی با مقادیر عضویت به اشتراک گذاشته شده ساخته می‌شوند تا همبستگی ممکن بین خروجی‌ها را محقق کند. به منظور کسب دقت بیشتر در شبکه عصبی تالی، می‌توان از درهم آمیختن پودمان^۱ های عصبی چندگانه (خبرگان محلی^۲ نیز نامیده می‌شود) برای هر قاعده عصبی استفاده کرد. مزیت‌های پودمان‌های عصبی این است که آنها به کاهش پارامترهای تعدیل‌پذیر کمک می‌کنند. این ساختار با شبکه‌های پودمانی^۳ هم‌ارز است. ساختار یک MIMO CANFIS با دو ورودی، دو خروجی و دو قاعده عصبی در شکل ۴-۲۰ نمایش داده شده است.

۴-۷ سیستم‌های استنتاج فازی عصبی خودسازمان‌ده

۴-۷-۱ شبکه کنترل یادگیری انطباقی فازی (FALCON)

نرخ یادگیری در الگوریتم یادگیری پس‌انتشار محدود است زیرا وزن‌های شبکه براساس حداقل‌سازی تابع خطایی که در قالب یک تابع خروجی مشخص می‌شود، تعیین می‌گردند و این در حالی است که سهم قابل توجهی از زمان صرف بازنمایی درونی شبکه خواهد شد. به منظور کمینه‌سازی این زمان، شبکه کنترل یادگیری انطباقی فازی (FALCON)^۴ توسط لین^۵ و لی^۶ (۱۹۹۱، ۱۹۹۴ و ۱۹۹۶) ارائه شده که مدلی پیوندگرا از یک کنترل‌کننده منطق فازی و سیستم تصمیم‌گیری محسوب می‌شود. گره‌های ورودی و خروجی در این ساختار پیوندگرا، به ترتیب نمایش‌دهنده وضعیت‌های ورودی و سیگنال‌های تصمیم‌کنتری خروجی هستند. در لایه‌های پنهان، گره‌هایی وجود دارند که به عنوان توابع عضویت و قواعد فعالیت می‌کنند. ساختار FALCON از پنج لایه مطابق شکل ۴-۲۱ تشکیل شده است. این لایه‌های مجزا در

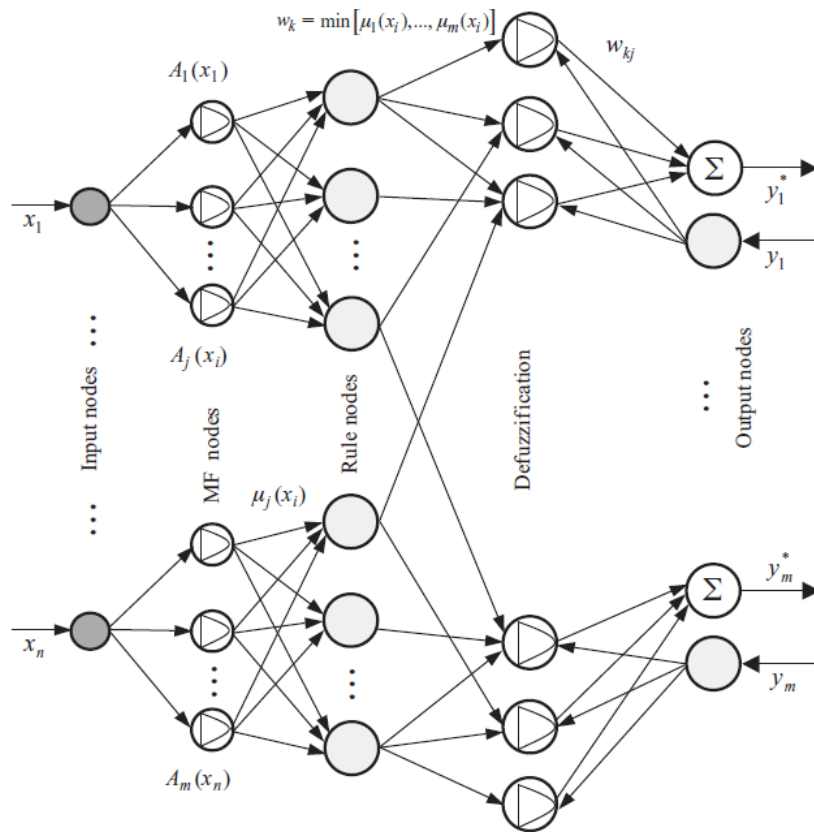
1. Module
2. Local experts
3. Modular networks
4. Fuzzy Adaptive Learning Control Network
5. Lin, C. T.
6. Lee, C. G.



شکل ۴-۲۰: ساختار MIMO CANFIS (MANFIS)

ادامه تشریح شده‌اند.

لایه ۱: گره‌های این لایه، گره‌های ورودی هستند که متغیرهای زبانی را مشخص می‌کنند. مقادیر ورودی، توسط گره‌های این لایه مستقیماً به لایه بعدی فرستاده می‌شود.



شکل ۴-۲۱: ساختار FALCON

لایه ۲: گره‌های این لایه، نقش توابع عضویت را به منظور ارائه عبارات متغیر زبانی مربوطه ایفا می‌کنند. در این لایه می‌توان گره تنهایی داشت که یک تابع عضویت را اجرا کند (مثل تابع مثلثی یا زنگوله‌ای) یا گره‌هایی چندلایه‌ای (یک زیرشبکه) را در نظر گرفت که یک تابع عضویت پیچیده را اجرا کند (مثلاً در یک آشکارساز نشانه صوتی^۱). در این مورد، مجموع تعداد لایه‌ها در FALCON می‌تواند بیشتر از پنج باشد. تابع زنگوله‌ای، تابع عضویتی است که به‌طور معمول در FALCON استفاده می‌شود. یک تابع عضویت زنگوله‌ای را می‌توان با دو پارامتر $A_j(m_{ij}, \sigma_{ij})$ تعریف کرد. مرکز یا

1. Acoustic cue detector

میانگین برای زامین تابع عضویت از زامین ورودی (x_i) با m_{ij} نشان داده شده و پهنا یا واریانس زامین تابع عضویت از زامین ورودی (x_i) با σ_{ij} نمایش داده شده است. در شرایطی که تابع عضویت زنگوله‌ای به کار گرفته شود، وزن‌های اتصال w_{ij} در این لایه را می‌توان به عنوان m_{ij} تلقی کرد. گره‌های این لایه، به طور کامل بین گره‌های زبانی و گره‌های عبارت متناظرشان متصل هستند:

(۱۱۰-۴)

$$\mu_j(x_i) = -\frac{(x_i - m_{ij})^2}{\sigma_{ij}} \quad i = 1, \dots, n; j = 1, \dots, m$$

در رابطه فوق، n تعداد ورودی‌ها و m تعداد توابع عضویت است.

لایه ۳: گره‌های این لایه، گره‌های قواعد هستند. اتصال‌های این لایه به منظور تطابق پیش شرط در قواعد منطق فازی به کار می‌روند. هر گره یک قاعده فازی را ارائه می‌کند که عمل AND فازی - نظیر عمل حداقل سازی - را طی رابطه ۴-۱۱۱ اجرا می‌کند:

$$w_k = \min [\mu_1^k(x_1), \dots, \mu_m^k(x_i)] \quad k = 1, \dots, M \quad (۱۱۱-۴)$$

درجه عضویت برای تابع عضویت j از ورودی i با $\mu_j(x_i)$ مشخص شده و w_k شدت برانگیختگی را برای قاعده k نشان می‌دهد؛ M نیز تعداد قواعد است. پایگاه قواعد توسط این لایه ارائه می‌شود.

لایه ۴: گره‌های این لایه، گره‌های خروجی هستند و به دو شیوه عمل می‌کنند. اولین شیوه، حالت انتقال پایین/بالا است که اتصال‌های این لایه عمل OR فازی را به منظور یکپارچه و ترکیب کردن قواعد فعال انجام می‌دهند. شیوه دوم، حالت انتقال بالا/پایین است که گره‌های لایه چهار و اتصال‌های لایه پنج دقیقاً مشابه لایه دو عمل می‌کند. به منظور ایفای نقش یک تابع عضویت برای متغیرهای زبانی خروجی فقط از یک گره استفاده می‌شود.

لایه ۵: این لایه، لایه خروجی است. برای هر متغیر خروجی، دو نوع گره وجود دارد. گره نوع اول، انتقال بالا/پایین را برای داده‌های آموزشی با هدف خوراندن خروجی‌های

مطلوب y_k به شبکه انجام می‌دهد. گره نوع دوم، انتقال پایین/بالا را برای خروجی‌های سیگنال تصمیم y_k^* پیاده‌سازی می‌کند ($k = 1, \dots, m$). پیکانی که بر روی اتصال قرار دارد، جهت جریان سیگنال نرمال را در شرایطی نشان می‌دهد که شبکه پس از اینکه ساخته شده و آموزش داده شده، در حال استفاده است. این گره‌ها و اتصالات لایه پنجم که به آن‌ها متصل است، به عنوان فازی‌زدا عمل می‌کنند.

بر پایه ساختار FALCON، یک الگوریتم یادگیری دو مرحله‌ای برای تعیین پارامترهای توابع عضویت گره‌ها در لایه‌های ۲ و ۴ توسعه یافته است. همچنین با استفاده از نوع و ماهیت روابط بین لایه‌های ۳ و ۴ به یادگیری قواعد منطق فازی می‌پردازد. این الگوریتم یادگیری هیبریدی به‌طور اختصار FALCON-H نامیده می‌شود و از دو گام مجزا شامل یادگیری بدون نظارت و یادگیری با نظارت (کاهش گرادیان) تشکیل شده که به ایجاد گره‌های قواعد و آموزش توابع عضویت منجر می‌شود. در مرحله نخست، توابع عضویت اولیه تعیین شده و گره‌های قواعد توسط یادگیری خودسازمان‌ده^۱ شکل می‌گیرد. در مرحله دوم، پارامترهای توابع عضویت توسط یادگیری با نظارت اصلاح و تنظیم می‌شوند. به منظور مقداردهی اولیه، باید داده‌های آموزشی و بخش‌بندی فازی مطلوب یا مورد حدس (برای توابع عضویت ورودی/خروجی) از قبل تعیین گردد (مثلاً با نظر خبرگان موضوع). نشان داده شده است که عملکرد الگوریتم یادگیری هیبریدی از عملکرد تنهای الگوریتم با نظارت - مثلاً فقط از الگوریتم پس‌انتشار استفاده کنیم - بهتر است زیرا پیش از یادگیری با نظارت، یک طبقه‌بندی از داده‌های آموزشی به واسطه یک میدان دریافت^۲ دارای هم‌پوشانی ایجاد شده است. برای آشنایی بیشتر با جزئیات FALCON-H به کتاب سیستم‌های فازی عصبی اثر لین^۳ و لی^۴ (۱۹۹۶) مراجعه کنید.

1. Self-organizing learning

2. Receptive field

3. Lin, C. T.

4. Lee, C. G.

موردکاوی

احراز هویت بیومتریک در سال‌های اخیر، به ویژه در سیستم‌هایی با مقیاس بزرگ، بسیار مورد توجه واقع شده است. استفاده از فناوری‌های این حوزه، علاوه بر امنیت بالا، از نظر اجتماعی قابل قبول است و معمولاً با حفظ ویژگی‌های یکسانی در دسترس است. امروزه، به‌کارگیری فناوری‌های بیومتریک به طور گسترده در تلفن‌های همراه هوشمند، گیت‌های فیزیکی و ... مورد استفاده قرار می‌گیرد.

تشخیص چهره، یکی از روش‌های متداول در فرآیند احراز هویت بیومتریک بوده که منجر به پژوهش‌های متعددی در این زمینه شده است. پاروادی و سانکار (۲۰۱۹) در پژوهشی به احراز هویت چندعاملی با استفاده از رویکرد عصبی فازی در تشخیص چهره پرداخته‌اند. آنها از شبکه کنترل یادگیری انطباقی فازی (FALCON) برای طبقه‌بندی استفاده کرده‌اند تا نرخ تشخیص بالایی نسبت به طبقه‌بندی‌کننده‌های دیگر ایجاد شود.

در مطالعه فوق، تصویر چهره هر کاربر در مرحله ثبت نام وی شناسایی و ثبت می‌شود. طی این مرحله، پس از شناسایی چارچوب تصویر، ویژگی‌های موردنظر با استفاده از روش Local Tetra Pattern استخراج می‌شوند. در ادامه، ویژگی‌های استخراج شده به وسیله FALCON در فرآیند آموزش مورد استفاده قرار می‌گیرند. از سوی دیگر، در مرحله احراز هویت نیز تصویر کاربران در قالب نمونه آزمایشی ثبت می‌شود. این تصویر تحت برخی شرایط کنترل‌نشده مانند تغییر در حالت سر، میزان روشنایی یا پوشاندن صورت با عینک، مو و ... گرفته می‌شود بنابراین ممکن است که تصویر کامل و جامعی از صورت وجود نداشته باشد. فرآیند استخراج ویژگی‌ها و نیز آموزش به کمک FALCON در این مرحله نیز مورد تاکید قرار می‌گیرند. سپس، این ویژگی‌های آموزش دیده با مجموعه ویژگی‌های موجود با استفاده از الگوریتم پیشنهادی برای احراز هویت یک شخص مقایسه می‌شوند. مدل پیشنهادی به کمک ارزیابی ۱۱۵۰ تصویر چهره مورد بررسی قرار گرفته که در نهایت دقت ۹۶ درصد برای آن گزارش شده است.

Parvathi, R. & Sankar, M. (2019). An exhaustive multi factor face authentication using neuro-fuzzy approach. *Wireless Personal Communications*, 109(4), 2353-2375.

۲-۷-۴ شبکه استنتاج فازی عصبی خودساخت (SONFIN)

شبکه استنتاج فازی عصبی خودساخت (SONFIN)^۱ یک نمونه تغییر یافته از سیستم فازی تاکاگی-سوگنو است که از سازوکار یادگیری شبکه عصبی استفاده می‌کند. SONFIN با یک پایگاه قواعد خالی شروع به کار می‌کند، یعنی در ابتدا هیچ قاعده‌ای وجود ندارد. این شبکه سپس با بهره‌گیری از سازوکار یادگیری آنلاین به ساخت و تطبیق پایگاه قواعد می‌پردازد. شناسایی و تعیین پارامتر و ساختار به‌طور همزمان به‌منظور شکل‌دهی به یادگیری سریع انجام می‌شود. در تعیین ساختار بخش مقدم، فضای ورودی به شیوه‌ای منعطف بر اساس یک الگوریتم هم‌راستا مبتنی بر خوشه‌بندی افراز می‌شود. به‌منظور تعیین ساختار بخش تالی، ابتدا فقط یک مقدار که به‌وسیله خوشه‌بندی انتخاب شده است، به هر قاعده اختصاص می‌یابد. سپس تعدادی عبارت معنادار دیگر (متغیرهای ورودی) به‌صورت تدریجی در قالب عواید یادگیری به بخش تالی اضافه می‌شود که در قالب شکل‌دهی یک تساوی خطی از متغیرهای ورودی صورت می‌پذیرد. برای تعیین پارامتر، پارامترهای تالی با استفاده از الگوریتم کمترین میانگین مربعات یا الگوریتم بازگشتی کمترین مربعات^۲ به‌طور بهینه تنظیم می‌شود و پارامترهای مقدم (یعنی پارامترهای تابع عضویت) توسط یک الگوریتم پسانتشار تنظیم می‌گردد. به‌منظور تقویت نمایش دانش در SONFIN، می‌توان یک تبدیل خطی از متغیرهای ورودی را برای کاهش بیشتر قواعد در شبکه تلفیق کرد.

ساختار SONFIN از شش لایه تشکیل شده است که یک سیستم فازی TSK به فرم زیر را محقق

می‌کند.

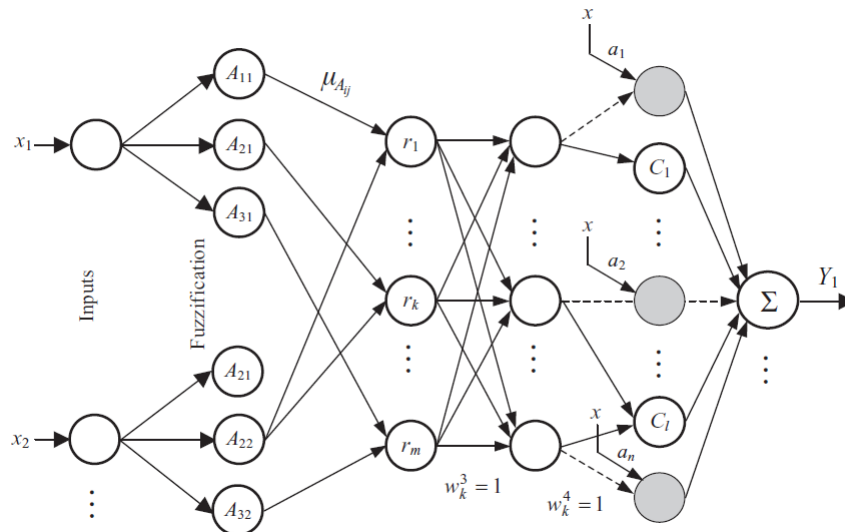
$$\text{If } x_i \text{ is } A_{ij} \text{ then } y_k = m_{ok} + a_{1k}x_1 + \dots + a_{nk}x_n. \quad (112-4)$$

A_{ij} با در نظر گرفتن $i = 1, \dots, n$ و $j = 1, \dots, p$ تعریف می‌شود. n تعداد ورودی‌ها، p تعداد توابع عضویت مقدم و m_{ok} مرکز تابع عضویت متقارن نسبت به y است. a_{ik} نیز پارامترهای تالی را تحت توابع $k = 1, \dots, m$ نشان می‌دهد. یک تفاوت مهم SONFIN با سیستم فازی TSK این است که همه پارامترهای $\{a_{1k}, a_{2k}, \dots, a_{nk}\}$ در تابع خطی خروجی به‌کار گرفته نمی‌شوند. ساختار SONFIN با دو ورودی در شکل ۴-۲۲ به نمایش درآمده است. در این شکل فقط دو ورودی نشان داده شده است اما

1. Self-Constructing Neural Fuzzy Inference Network

2. Recursive Least-Squares Algorithm

می‌توان آن را به n خروجی گسترش داد. این ساختار به صورت لایه‌به‌لایه در ادامه تشریح می‌شود.



شکل ۴-۲۲: ساختار SONFIN

لایه ۱: هر گره در این لایه، متناظر یک متغیر ورودی است. این گره‌ها هیچ فرآیندی را دربر ندارند و فقط مقادیر ورودی را به لایه بعدی منتقل می‌کنند؛ یعنی:

$$x_i = f(x_i) \quad , \quad i = 1, 2, \dots, n \quad (4-113)$$

لایه ۲: هر گره در این لایه، به یک تابع عضویت (برچسب زبانی) مربوط است و ورودی‌ها را فازی می‌کند. توابع عضویت می‌توانند مثلثی، گاوسی، زنگوله‌ای و ... باشند. در SONFIN از یک تابع عضویت گاوسی برای ورودی‌های $\mu_{ji}(x_i)$ استفاده می‌شود:

$$\mu_{ji}(x_i) = f\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji}}\right) \quad (4-114)$$

که m_{ji} و σ_{ji} به ترتیب مرکز و پهنای تابع عضویت گاوسی j ام از ورودی i ام را نشان می‌دهد و $i = 1, \dots, n$ و $j = 1, \dots, p$ هستند. p تعداد توابع عضویت را برای

ورودی x_i مشخص می‌کند. $f(\cdot)$ نشان‌گر تابع فعال‌سازی انتخاب‌شده برای گره‌ها است. نکته مهمی که باید یادآور شویم این است که ضرورتی ندارد توابع عضویت برای هر ورودی در SONFIN یکسان و به یک تعداد باشد. حداکثر تعداد گره‌ها در لایه ۲ برابر با $m = n \times p$ است.

لایه ۳: این لایه، لایه قواعد است که طی آن، بخش مقدم قاعده توسط هر گره ارائه می‌شود:

$$\tau_k = \prod_{k=1}^m \mu_{k=i \times j} \quad (4-115)$$

$k = 1, \dots, m$ است و m تعداد گره‌های حاصل از لایه ۲ را نشان می‌دهد که در تطبیق پیش‌شرط یک قاعده (شدت برانگیختگی τ_k برای قاعده $k^{\text{ام}}$) مشارکت داشته‌اند. در SONFIN از عمل AND استفاده می‌شود. وزن‌های اتصال لایه ۳ برابر واحد است: $w_k^2 = 1$.

لایه ۴: تعداد گره‌های این لایه با تعداد گره‌های لایه ۳ برابر است. در این لایه، شدت برانگیختگی τ_k ($k = 1, \dots, m$) نرمال شده و به صورت زیر محاسبه می‌شود:

$$\bar{\tau}_k = \frac{\tau_k}{\sum_{k=1}^m \tau_k} \quad (4-116)$$

وزن‌های اتصال لایه ۴ نیز برابر واحد است: $w_k^4 = 1$.

لایه ۵: این لایه، لایه نتیجه است که شامل دو نوع گره می‌شود: گره‌های خالی^۱ و سایه‌دار^۲. نوع اول (خالی)، توابع عضویت تالی (یعنی توابع عضویت برای متغیر خروجی) را ارائه می‌کند. میانگین محلی حداکثر فازی‌زدایی در توابع عضویت گاوسی به کارگرفته شده و پهنا در خوشه‌بندی خروجی استفاده می‌شود. $m_{o,k}$ مرکز توابع عضویت گاوسی است. این نوع گره سرانجام مرکز هر تابع عضویت گاوسی ($m_{o,k}$) را در قالب

1. Blank
2. Shaded

خروجی به لایه بعدی تحویل می‌دهد. هر گره خالی خروجی y_k^b را به صورت زیر ارائه می‌کند.

$$y_k^b = m_{\circ k} \quad (117-4)$$

نوع دیگر گره (سایه‌دار)، بخش تالی قواعد را تولید می‌کند. هر گره، یک ورودی را از خروجی‌های لایه ۴ و ورودی دیگر را از خروجی‌های لایه ۱ دریافت می‌کند. هر گره سایه‌دار، خروجی y_k^s را به صورت زیر ارائه می‌کند.

$$y_k^s = m_{\circ k} + (a_{1k}x_1 + \dots + a_{nk}x_n) = m_{\circ k} + \sum_{i=1}^n a_{ik}x_i \quad (118-4)$$

a_{ik} پارامتر متناظر ورودی x_i است. با ترکیب این دو نوع گره در لایه ۵، یک تابع کلی به فرم زیر ایجاد شده که توسط این لایه اجرا می‌شود:

$$y_k = (y_k^s)y_k^b = \left(m_{\circ k} + \sum_{i=1}^n a_{ik}x_i \right) y_k^b \quad (119-4)$$

لایه ۶: هر گره در این لایه با یک متغیر خروجی متناظر است. چنین گره‌ای در این لایه، فازی‌زدایی را به وسیله انبوه‌سازی همه اقدام‌های حاصل از لایه ۵ انجام می‌دهد:

$$Y_1 = \sum_{k=1}^m y_k \quad (120-4)$$

دو نوع یادگیری در SONFIN رخ می‌دهد: یادگیری ساختار و یادگیری پارامتر. یادگیری ساختار، تعیین ساختار هر دو بخش مقدم و تالی قاعده اگر-آنگاه فازی را دربردارد. تعیین ساختار مقدم با بخش‌بندی فضای ورودی متناظر است که می‌تواند در قالب یک مسأله بهینه‌سازی فرموله شود. تعیین ساختار تالی با ساخت یک تابع عضویت جدید برای متغیرهای خروجی و شمول متغیرهای ورودی در تابع خطی بخش تالی متناظر است. یادگیری پارامتر، الگوریتم‌های با نظارت نظیر پس‌انتشار را برای پارامترهای مقدم و الگوریتم‌های LMS و RLS را برای پارامترهای تالی پوشش می‌دهد. جزئیات این فرآیند یادگیری و سایر اجزای ساختار SONFIN در مقاله جوانگ^۱ و لین^۲ (۱۹۹۸) تشریح شده است.

1. Juang, C. F.

2. Lin, C. T.

موردکاوی

میگرن یک بیماری عصبی مزمن است که با سردردهای متوسط تا شدید عودکننده طی یک دوره زمانی ظاهر می‌شود. بروز این بیماری، اغلب با علائمی در مغز انسان و سیستم عصبی خودمختار همراه است. به طور معمول، علائم میگرن را می‌توان به چهار مرحله مختلف طبقه بندی کرد: فاصله بین حمله‌ها (inter-ictal)، پیش از حمله (pre-ictal)، حمله (ictal) و پس از حمله (post-ictal). از آنجایی که بیماران میگرنی در دانستن اینکه چه زمانی دچار حملات میگرنی می‌شوند مشکل دارند. بنابراین تشخیص زودهنگام به یک موضوع مهم تبدیل می‌شود؛ به ویژه برای بیماران میگرنی با فرکانس پایین که کمتر از ۵ بار در ماه حمله‌های میگرنی دارند. تسائو و همکاران (۲۰۱۵) در مطالعه‌ای پیرامون طبقه‌بندی مراحل بیماری میگرن، از SONFIN به‌عنوان ابزار طبقه‌بندی استفاده کرده‌اند. هدف اصلی این پژوهش ایجاد یک سیستم طبقه‌بندی مراحل میگرن بر اساس قدرت EEG در حالت استراحت بیماران میگرنی است.

در این مطالعه، برخی فعالیت‌های EEG میگرنی‌ها در حین بستن چشم‌ها از لوب اکسیپیتال جمع‌آوری شده تا مرحله «پیش از حمله» شناسایی شود. در گام پیش‌پردازش داده‌ها، سیگنال‌های EEG مورد پردازش قرار گرفته و قدرت EEG در پنج باند فرکانسی مختلف حاصل می‌شود. نتایج این پردازش نشان می‌دهد که طیف قدرت مرحله «پیش از حمله» با مراحل دیگر در باندهای دلتا، تتا، آلفا، بتا و گاما متفاوت است. بنابراین می‌توان از این شاخص به‌عنوان نشان‌گرهای زیستی برای شناسایی مرحله «پیش از حمله» در فرآیند طبقه‌بندی استفاده کرد.

در ادامه از SONFIN در طبقه‌بندی مراحل میگرن استفاده شده است. بهره‌گیری از SONFIN در بستر انتخاب ویژگی توسط روش Quadratic Bayes Normal Classifier بهترین ترکیب را حاصل کرده است. براساس نتایج گزارش‌شده، این ترکیب به دقتی در حدود ۶۶ درصد دست یافته که در مقایسه با طبقه‌بندی‌کننده‌های دیگر نظیر KNN یا SVM بالاتر بوده است.

Cao, Z. H., Ko, L. W., Lai, K. L., Huang, S. B., Wang, S. J. & Lin, C. T. (2015, July). Classification of migraine stages based on resting-state EEG power. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-5). IEEE.

۳-۷-۴ سیستم استنتاج فازی عصبی تکاملی پویا (DENFIS)

سیستم استنتاج فازی عصبی تکاملی پویا (DENFIS)^۱ ساختاری شبیه به برخی از سیستم‌های فازی عصبی نظیر ANFIS دارد. نوآوری DENFIS مربوط به استفاده آن از روش خوشه‌بندی تکاملی (ECM)^۲ در ایجاد توابع عضویت فازی است. DENFIS دو مدل یادگیری آنلاین و آفلاین دارد که به ترتیب از ECM و ECM با محدودیت (ECMc) در خوشه‌بندی خود بهره می‌گیرد. ECM در مقاله کاسابوف^۳ و سانگ^۴ (۲۰۰۲) معرفی شده است.

در ادامه، ساختار DENFIS و فرآیند یادگیری آن براساس مقاله اصلی DENFIS مربوطه به کاسابوف و سانگ (۲۰۰۲) و نیز مطالعه کاسابوف (۲۰۱۳) و جیانگ^۵ و همکاران (۲۰۱۹) تشریح می‌شود. مدل DENFIS در هر دو حالت یادگیری آنلاین و آفلاین خود از روش استنتاج تاکاگی-سوگنو استفاده می‌کند. استنتاج در DENFIS بر پایه m قاعده فازی به صورت زیر است:

$$\left\{ \begin{array}{l} \text{If } x_1 \text{ is } R_{11} \ \& \ x_2 \text{ is } R_{12} \ \& \ \dots \ \& \ x_q \text{ is } R_{1q}, \text{ then } y \text{ is } f_1(x_1, x_2, \dots, x_q). \\ \text{If } x_1 \text{ is } R_{21} \ \& \ x_2 \text{ is } R_{22} \ \& \ \dots \ \& \ x_q \text{ is } R_{2q}, \text{ then } y \text{ is } f_2(x_1, x_2, \dots, x_q). \\ \vdots \\ \text{If } x_1 \text{ is } R_{m1} \ \& \ x_2 \text{ is } R_{m2} \ \& \ \dots \ \& \ x_q \text{ is } R_{mq}, \text{ then } y \text{ is } f_m(x_1, x_2, \dots, x_q). \end{array} \right.$$

R_{ij} ($j = 1, 2, \dots, q$ و $i = 1, 2, \dots, m$) بیانگر مجموعه‌های فازی است که توسط توابع عضویت $\mu_{R_{ij}}$ تعریف می‌شوند. x_j ($j = 1, 2, \dots, q$) نیز متغیر ورودی را نشان می‌دهد. به این ترتیب، تعداد $m \times q$ گزاره فازی به صورت $x_j \text{ is } R_{ij}$ داریم که m مقدم را در m قاعده فازی فوق شکل داده‌اند. از سوی دیگر، در بخش تالی هر قاعده فازی، y متغیر تالی را نشان می‌دهد که تابع قطعی f_i ($i = 1, 2, \dots, m$) را به کار می‌گیرد.

در هر دو مدل آنلاین و آفلاین DENFIS، همه توابع عضویت از نوع مثلثی و براساس سه پارامتر a ,

1. Dynamic Evolving Neuro-Fuzzy Inference System
2. Evolving Clustering Method
3. Kasabov, N. K.
4. Song, Q.
5. Jiang, H.

b و c تعریف می‌شوند:

$$\mu(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > c \end{cases} \quad (۱۲۱-۴)$$

که b مرکز خوشه i است و $a = b - d \times D_{thr}$ و $c = b + d \times D_{thr}$ نیز به ترتیب مقادیر سمت چپ و راست تابع عضویت را نشان می‌دهند که در آن $1/2 \leq d \leq 2$ و D_{thr} یک پارامتر خوشه در روش ECM است.

اگر تابع هر بخش تالی یک عدد ثابت باشد (یعنی $f_i(x_1, x_2, \dots, x_q) = C_i$)، این سیستم یک سیستم استنتاج فازی تاکاگی-سوگنو از مرتبه صفر نامیده می‌شود. اگر هر $f_i(x_1, x_2, \dots, x_q)$ یک تابع خطی باشد، آن را استنتاج فازی تاکاگی-سوگنو از مرتبه یک نامیده و اگر توابع غیرخطی باشند، آن را سیستم استنتاج فازی تاکاگی-سوگنو از مرتبه بالا می‌نامند.

برای هر بردار ورودی $x^\circ = [x_1^\circ, x_2^\circ, \dots, x_q^\circ]$ ، نتیجه استنتاج (y° یا همان خروجی سیستم)

به کمک محاسبه میانگین وزنی هر خروجی قاعده به صورت حاصل می‌شود:

$$y^\circ = \frac{\sum_{i=1}^m w_i f_i(x_1^\circ, x_2^\circ, \dots, x_q^\circ)}{\sum_{i=1}^m w_i} \quad (۱۲۲-۴)$$

که $w_i = \prod_{j=1}^q \mu_{R_{ij}}(x_j^\circ)$ است.

یادگیری آنلاین در مدل DENFIS

در مدل آنلاین DENFIS، قواعد فازی تاکاگی-سوگنو از مرتبه اول مورد استفاده قرار می‌گیرد. توابع خطی در بخش تالی به کمک برآوردگر حداقل مربعات (LSE) خطی روی داده‌های آموزشی ایجاد شده و به روز می‌شوند. تعریف هر تابع خطی به صورت زیر است:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q x_q \quad (۱۲۳-۴)$$

برای به دست آوردن این توابع، از یک فرآیند یادگیری روی مجموعه داده‌های تشکیل شده از p جفت داده $([x_{i1}, x_{i2}, \dots, x_{iq}], y_i)$ با $i = 1, 2, \dots, p$ استفاده می‌شود. برآوردگر حداقل مربعات $b = [b_0, b_1, b_2, \dots, b_q]^T$ به وسیله فرمول زیر در قالب ضرایب $\beta = [\beta_0, \beta_1, \beta_2, \dots, \beta_q]^T$ محاسبه می‌شود:

$$b = (A^T A)^{-1} A^T y \quad (124-4)$$

که

$$A = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1q} \\ 1 & x_{21} & x_{22} & \dots & x_{2q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{p1} & x_{p2} & \dots & x_{pq} \end{pmatrix}$$

و $y = [y_1, y_2, \dots, y_p]^T$ است. در اینجا از برآوردگر حداقل مربعات به صورت زیر استفاده می‌شود:

$$b_w = (A^T W A)^{-1} A^T W y \quad (125-4)$$

که

$$W = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & w_p \end{pmatrix}$$

و w_j برابر با فاصله بین نمونه j از مرکز خوشه متناظر آن است ($j = 1, 2, \dots, p$).

روابط ۴-۱۲۴ و ۴-۱۲۵ را می‌توان به صورت زیر بازنویسی کرد:

$$\begin{cases} P = (A^T A)^{-1} \\ b = P A^T y \end{cases} \quad (126-4)$$

$$\begin{cases} P_w = (A^T W A)^{-1} \\ b_w = P_w A^T W y \end{cases} \quad (127-4)$$

اگر سطر k از ماتریس A در رابطه ۴-۱۲۴ به صورت $a_k^T = [1, x_{k1}, x_{k2}, \dots, x_{kq}]$ تعریف شود و عنصر k از y با نماد y_k نمایش داده شود، مقدار b را می‌توان برای $k = n, n+1, \dots, p-1$ به شیوه زیر محاسبه کرد:

$$\begin{cases} b_{k+1} = b_k + P_{k+1} a_{k+1} (y_{k+1} - a_{k+1}^T b_k) \\ P_{k+1} = P_k - \frac{P_k a_{k+1} a_{k+1}^T P_k}{1 + a_{k+1}^T P_k a_{k+1}} \end{cases} \quad (4-128)$$

مقادیر اولیه برای P_n و b_n را می‌توان مستقیماً با استفاده از اولین n جفت داده از مجموعه داده‌های آموزشی توسط رابطه ۴-۱۲۷ حاصل کرد.

رابطه ۴-۱۲۸ فرمول LSE بازگشتی است. در مدل آنلاین DENFIS از یک LSE بازگشتی موزون با ضریب فراموشی به فرم زیر استفاده می‌شود:

$$\begin{cases} b_{k+1} = b_k + w_{k+1} P_{k+1} a_{k+1} (y_{k+1} - a_{k+1}^T b_k) \\ P_{k+1} = \frac{1}{\lambda} \left[P_k - \frac{w_{k+1} P_k a_{k+1} a_{k+1}^T P_k}{\lambda + a_{k+1}^T P_k a_{k+1}} \right] \end{cases} \quad (4-129)$$

که w وزن تعریف شده در رابطه ۴-۱۲۵ است و λ نیز ضریب فراموشی است که معمولاً بین ۰/۸ و ۱ در نظر گرفته می‌شود.

در مدل DENFIS آنلاین، با استفاده از بخش بندی فضای ورودی به کمک ECM و روابط ۴-۱۲۳ و ۴-۱۲۹، قواعد به طور همزمان ایجاد شده و به روز می‌شوند. گام‌های الگوریتم به منظور ایجاد اولین m قاعده فازی برای محاسبه مقادیر اولیه P و b به کار گرفته می‌شوند. در گام‌های این الگوریتم، m ، n و p پارامترهای مدل یادگیری آنلاین DENFIS هستند. مقدار p_i باید بزرگتر از تعداد عناصر ورودی (q) باشد. هنگامی که جفت داده‌های جدید به سیستم وارد می‌شوند، ممکن است قواعد جدیدی ایجاد شده و برخی قواعد فعلی نیز به روز شوند. یک قاعده فازی جدید زمانی ایجاد می‌شود که یک مرکز خوشه جدید توسط ECM یافته شود. بخش مقدم از قاعده فازی جدید با استفاده از رابطه ۴-۱۲۳ و موقعیت مرکز خوشه در قالب یک گره قاعده شکل می‌گیرد. براساس نزدیک‌ترین گره به گره جدید، یکی از قواعد فازی فعلی انتخاب شده و تابع بخش تالی آن به عنوان تابع بخش تالی قاعده فازی جدید در نظر گرفته می‌شود. برای هر جفت داده، چند قاعده فازی فعلی با استفاده از رابطه ۴-۱۲۹ به روز می‌شوند اگر

الگوریتم الگوریتم DENFIS آنلاین

- ۱: اولین جفت داده‌های آموزشی (n_0) از مجموعه داده‌ها دریافت می‌شود.
- ۲: خوشه‌بندی n_0 داده با ECM به منظور دستیابی به m مرکز خوشه انجام می‌شود.
- ۳: نقاط P_i برای هر مرکز خوشه C_i براساس نزدیک‌ترین نقطه در فضای ورودی به مرکز انجام می‌شود ($i = 1, 2, \dots, m$).
- ۴: به منظور دستیافتن به قاعده فازی متناظر با مرکز خوشه، بخش مقدم از هر قاعده به کمک موقعیت مرکز خوشه در رابطه ۴-۱۲۳ به دست می‌آید. مقادیر P و b مربوط به تابع بخش تالی با استفاده از رابطه ۴-۱۲۷ برای جفت داده p_i حاصل می‌شود. فاصله بین p_i و مرکز خوشه به عنوان وزن در رابطه ۴-۱۲۷ در نظر گرفته می‌شود.

گره‌های آنها فاصله‌ای بیشتر از $2 \times D_{thr}$ از نقطه مورد نظر نداشته باشند. فاصله بین این گره‌های قواعد و نقطه مورد نظر به عنوان وزن در رابطه ۴-۱۲۹ لحاظ می‌شوند. علاوه بر این، امکان دارد که یکی از این قواعد از طریق تغییر در مقدم خود به روز شود به طوری که اگر موقعیت گره آن توسط ECM تغییر یابد، یک مقدم جدید به وسیله رابطه ۴-۱۲۳ ایجاد خواهد شد.

یادگیری آفلاین در مدل DENFIS

مدل آنلاین DENFIS که در بخش قبل معرفی شد، می‌تواند به جای مدل آفلاین (آموزش دسته‌ای^۱) نیز به کار گرفته شود اما ممکن است برای مجموعه داده‌های کوچک کارا نباشد. به منظور آموزش دسته‌ای از یادگیری آفلاین استفاده می‌شود که به طور خلاصه دو مدل برای آن توسعه داده شده است: ۱- مدل خطی، ۲- مدل مبتنی بر MLP^۲.

یک موتور استنتاج فازی تاکاگی-سوگنو از مرتبه اول - مشابه مدل آنلاین DENFIS - در مدل ۱ استفاده شده است، و یک موتور استنتاج فازی تاکاگی-سوگنو از مرتبه بالا در مدل ۲ استفاده شده است. مدل دوم، چندین پرسپترون دو لایه (لایه پنهان متشکل از دو یا سه نورون) را به کار می‌گیرد تا به جای

1. Batch training

2. Multi Layer Perceptron

استفاده از تابعی که نوع از پیش‌تعریف‌شده‌ای دارد، تابع f را در بخش تالی هر قاعده فازی محقق کند. جزئیات بیشتر در رابطه با مراحل یادگیری آفلاین در مدل DENFIS را می‌توانید در مقاله کاسابوف^۱ و سانگ^۲ (۲۰۰۲) و نیز کاسابوف (۲۰۱۳) بیابید.

۸-۴ سایر سیستم‌ها

در بخش‌های قبل، سیستم‌های فازی عصبی پراستناد و مشهور مورد بررسی قرار گرفتند. با این حال، مدل‌های دیگری نیز هستند که اگرچه در ادبیات این حوزه به چشم می‌خورند اما کمتر به آنها پرداخته شده است. برخی از این مدل‌ها را در این بخش به صورت خلاصه مورد بررسی قرار می‌دهیم تا تصویر جامع‌تری از انواع ترکیب‌های هیبریدی ممکن برای سیستم‌های فازی و عصبی ارائه شود.

۱-۸-۴ شبکه فازی (FUN)

در بسیاری از موارد طی ایجاد شبکه‌های فازی عصبی، شبکه‌های عصبی به منظور استخراج قواعد به کار گرفته شده و از آنها برای یادگیری قواعد فازی استفاده شده است. با این وجود، شبکه‌های عصبی برای یافتن پارامترهای توابع عضویت نیز مورد استفاده واقع شده است. سولزبرگر^۳ و همکاران (۱۹۹۳) روشی برای ترجمه قواعد فازی و توابع عضویت به یک شبکه را پیشنهاد کردند که FUN^۴ نام دارد. FUN از نرون‌های خاصی استفاده می‌کند که می‌توانند با کمک توابع فعال‌سازی خود، عبارات منطقی را ارزیابی کنند. عملکرد شبکه و کیفیت پایگاه قواعد به وسیله آموزش شبکه عصبی بهبود می‌یابد. این شبکه شامل پنج لایه است که نرون‌های هر لایه دارای توابع فعال‌سازی متفاوتی هستند و مراحل مختلف استنتاج فازی را مشخص می‌کنند. این شبکه شامل یک لایه ورودی، یک لایه خروجی و سه لایه پنهان می‌شود که نمایی از آن در شکل ۴-۲۳ به نمایش درآمده است.

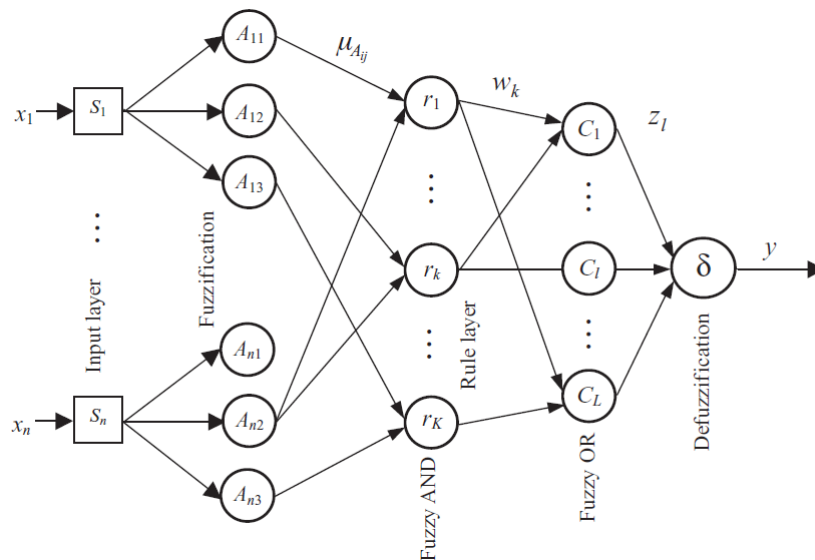
این شبکه با یک پایگاه قواعد فازی و توابع عضویت مربوطه مقداردهی اولیه شده و پس از آن، یک

1. Kasabov, N. K.

2. Song, Q.

3. Sulzberger, S. M.

4. Fuzzy Net



شکل ۴-۲۳: ساختار FUN

روش یادگیری تصادفی را به کار می‌گیرد که به طور تصادفی پارامترهای توابع عضویت و اتصالات درون ساختار شبکه را تغییر می‌دهد. اتصالات بین لایه‌ها، قواعد را ارائه و بیان می‌کنند. به منظور یادگیری قواعد، اتصالات بین لایه فازی‌سازی و لایه قواعد تغییر می‌یابند. الگوریتم یادگیری برای توابع عضویت، ترکیبی از کاهش گرادیان و یک جستجوی تصادفی است. به منظور یادگیری توابع عضویت، داده‌های ورودی به لایه فازی‌سازی (اولین لایه پنهان) و لایه OR فازی (سومین لایه پنهان) تغییر داده می‌شوند. فرآیند یادگیری، با کمک یک تابع هزینه به انجام می‌رسد که پس از تعدیل و اصلاح تصادفی ارزیابی می‌شود. اگر تعدیل و اصلاحات به یک عملکرد بهتر منجر شود، این اصلاحات نگهداری شده و در غیر این صورت به حالت اول بازگردانده می‌شود. این شبکه را می‌توان به وسیله الگوریتم‌های استاندارد آموزش شبکه‌های عصبی نظیر یادگیری تقویتی یا با نظارت، آموزش داد. با به کارگیری FUN در مثال‌های مختلف، نشان داده شده است که توانایی بهینه‌سازی یک پایگاه قواعد و توابع عضویت مفروض در آن وجود دارد.

۲-۸-۴ نرم‌افزار استنتاج فازی با تنظیم‌گری (FINEST)

تنظیم استنتاج فازی، عمدتاً قواعد فازی را بررسی می‌کند مادامی که عوامل دیگر نظیر پارامترهای عملگرهای انبوه‌سازی، توابع استلزام، توابع ترکیب و مسندهای فازی^۱ تحت بررسی باقی بمانند. در همین راستا، ساختار FINEST^۲ توسط تانو^۳ و همکاران (۱۹۹۴، ۱۹۹۶) به منظور ایجاد یک سازوکار تنظیم برای استنتاج فازی و و مسند فازی پیشنهاد شده است. این ساختار، دو نوع فرآیند تنظیم دارد: (۱) تنظیم مسند فازی، (۲) توابع ترکیب و تنظیم یک تابع استلزام. قیاس/استثنایی^۴ تعمیم‌یافته به چهار طریق بهبود می‌یابد: (۱) عملگرهای انبوه‌سازی که هم‌افزایی و ماهیت حذفی دارند؛ (۲) یک تابع استلزام پارامتری‌شده؛ (۳) یک تابع ترکیب که از فازی بودن می‌کاهد؛ (۴) زنجیره‌سازی پس‌سو که مبتنی بر قیاس استثنایی است. الگوریتم پس‌انتشار به منظور تنظیم دقیق پارامترها استفاده شده و چارچوبی برای تنظیم همه پارامترها توسط FINEST ایجاد می‌شود که در گره‌های شبکه ظاهر شده و در صورتی که تابع مشتق نسبت به پارامترها مفروض باشد، بیان‌کننده فرآیند محاسبه داده‌های فازی است.

ساختار عصبی در معماری FINEST چهار لایه دارد که در شکل ۴-۲۴ به نمایش درآمده است. محاسبه خروجی هر لایه به صورت لایه‌به‌لایه و دنباله‌وار تشریح می‌شود.

لایه ۱، خروجی نرون‌ها را با استفاده از یک عمل سوپریمم محاسبه می‌کند. لایه ۲، لایه انبوه‌سازی است و بخش مقدم هر قاعده را حاصل می‌کند. سپس در لایه ۳، استنتاج^۵ مربوط به هر قاعده به کمک پیامد بخش تالی قاعده و تابع استلزام پارامتری‌شده آن انجام می‌شود. در ادامه، ترکیب پیامدهای تمامی قواعد با استفاده از یک تابع ترکیب پارامتری‌شده به دست می‌آید.

ویژگی مهم FINEST، پارامتری‌کردن روند استنتاج است. توابع پارامتری‌شده متعددی وجود دارد که در استنتاج فازی مورد استفاده قرار می‌گیرند. عمل‌های t-norm و t-conorm مختلفی نظیر \max ، \min ،

1. Fuzzy predicates

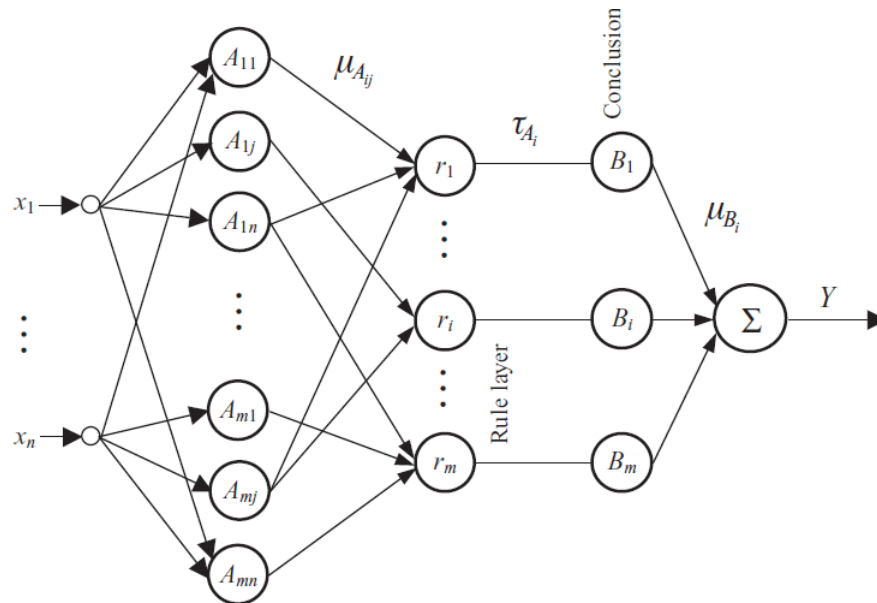
2. Fuzzy Inference Environment Software with Tuning

3. Tano, S. I.

۴. قیاس استثنایی یا Modus ponens، قیاسی است که عین نتیجه یا نقیض آن صراحتاً در مقدم (به‌عنوان جزئی از مقدمات آن) ذکر شده باشد. فرض کنید دو گزاره را با p و q نشان دهیم؛ آنگاه یک صورت از قیاس استثنایی عبارت است از:

$$((p \Rightarrow q) \wedge p) \Rightarrow q$$

5. Deduction



شکل ۴-۲۴: ساختار FINEST چهار لایه‌ای

میانگین و ضرب وجود دارد. یک تابع انبوه‌سازی پارامتری شده جدید (AND) در FINEST تعریف می‌شود که به وسیله افزودن اثر هم‌افزایی به یک t -norm معمولی محقق می‌گردد.

استلزام‌های مختلفی در سیستم‌های فازی استفاده می‌شوند اما انتخاب تابع مناسب در یک کاربرد خاص دشوار است. در FINEST، یک تابع استلزام پارامتری شده تعریف می‌شود که می‌توان آن را به راحتی با تغییر مقادیر پارامترها انتخاب کرد. از سوی دیگر، در اغلب موارد از عملگر \max به عنوان عملگر ترکیب (ایجاد یک مجموعه فازی با نتایج ترکیبی از دو مجموعه فازی حاصل از دو فرآیند استنتاج) استفاده می‌شود که موجب افزایش ثابت در حالت فازی می‌شود. به همین منظور، در FINEST از پارامترهای دیگری همچون تعادل و عوامل همبستگی استفاده می‌شود. توضیحات و جزئیات بیشتر را می‌توانید در پژوهش‌های آرنولد^۱ و تانو^۲ (۱۹۹۴)، اویاما^۳ و همکاران (۱۹۹۴) و تانو و همکاران (۱۹۹۴، ۱۹۹۶) بیابید.

1. Arnould, T.

2. Tano, S. I.

3. Oyama, T.

۹-۴ بحث و بررسی

همان‌طور که واضح است، هر دو مدل مشارکتی و همزمان به دلیل حضور شبکه عصبی (ماهیت جعبه سیاه^۱) به طور کامل قابل تفسیر نیستند. این در حالی است که مدل فازی عصبی هیبریدی (یکپارچه) قابل تفسیر بوده و می‌تواند یادگیری را به طریقه‌ای با نظارت انجام دهد (و یا حتی یادگیری به شیوه تقویتی نظیر NEFCON). فرآیند یادگیری در SONFIN، NEFCON، ANFIS، GARIC، FALCON، NEFCON، FUN و FINEST به انطباق سطح پارامتر در ساختارهای ثابت مربوط است. در مسائل با مقیاس بزرگ، تعیین ساختارهای مقدم-تالی، تعداد قواعد و غیره بسیار پیچیده خواهد بود. کاربر باید جزئیات این ساختار (نوع و کمیت توابع عضویت برای متغیرهای ورودی و خروجی)، نوع عمل‌گرهای فازی و غیره را مشخص کند. FINEST سازوکاری را بر پایه قیاس استثنایی تعمیم‌یافته بهبود یافته برای تنظیم مسندهای فازی^۲ و توابع ترکیب به طور مطلوب و نیز تنظیم یک تابع استلزام فراهم می‌کند. ویژگی مهمی که در EFuNN و dmEFuNN وجود دارد، آموزش تک گذر^۳ (دوره^۴) است که توانمندی بالایی برای یادگیری آنلاین محسوب می‌شود.

زمانی که مسأله شکل می‌گیرد، ساختار و پارامترهای فازی عصبی نیاز به تعاریف دستی بسیار پیچیده‌ای دارند که دشواری‌هایی ایجاد می‌کنند. از این رو، پرسش‌های زیر بدون جواب باقی می‌ماند:

- تعداد و شکل بهینه توابع عضویت برای متغیرهای ورودی/خروجی چگونه تعیین گردد؟
- عمل‌گرهای فازی و ساختار (پایگاه قواعد) بهینه چه خواهد بود؟
- پارامترهای بهینه یادگیری چگونه محاسبه شوند؟
- کدام سیستم استنتاج فازی (نظیر تاکاگی-سوگنو، ممدانی و غیره) برای یک مسأله مفروض بهتر عمل می‌کند؟

1. Black box
 2. Fuzzy predicates
 3. One pass training
 4. Epoch

مقایسه نتایج عملکرد سیستم‌های فازی عصبی در یک سری زمانی

یکی از رویکردهای متداول در مقایسه روش‌های مختلف، بررسی عملکرد آنها در یک یا چند مثال مطرح است؛ همان‌طور که طی بررسی برخی از موردکاوی‌های مندرج در این کتاب به این موضوع اشاره شد. در اینجا به عنوان نمونه، عملکرد چند سیستم فازی عصبی را بر روی یک مثال معروف در پیش‌بینی سری‌های زمانی بررسی می‌کنیم.

نتایج عملکرد برخی از سیستم‌های فازی عصبی در پیش‌بینی سری زمانی آشوبی مکی-گلز^۱ در جدول ۳-۴ به‌نمایش درآمده است. آموزش در اینجا براساس ۵۰۰ مجموعه داده و مدل‌های فازی عصبی مفروض انجام شده که با ۵۰۰ مجموعه دیگر نیز آزمون شده است.

جدول ۳-۴: عملکرد سیستم‌های فازی عصبی در مثال سری زمانی

سیستم	تعداد دوره‌ها	زمان پردازش (s)	RMSE
ANFIS	۷۵	۱۶۵	۰/۰۰۱۷
NEFPROX	۲۱۶	۷۵	۰/۰۳۳۲
EFuNN	۱	۳۰	۰/۰۱۴۰
dmEFuNN	۱	۵۲	۰/۰۰۴۲
SONFIN	۱	—	۰/۰۱۸۰

در میان مدل‌های فازی عصبی مقایسه‌شده، ANFIS کمترین مقدار خطای مجذور میانگین مربعات (RMSE)^۲ را کسب کرده و بیشترین مقدار RMSE نیز متعلق به NEFPROX است. این تفاوت احتمالاً به دلیل استفاده از قواعد تاکاگی-سوگنو در ANFIS در مقایسه با سیستم فازی ممدانی در NEFPROX است. با این حال، NEFPROX برحسب زمان محاسبات، نسبت به ANFIS عملکرد بهتری داشته است. SONFIN، EFuNN و dmEFuNN نیز به دلیل تعداد کم قواعدشان می‌توانند نسبت به ANFIS سریع‌تر

1. Mackey-Glass chaotic time series

2. Root Mean Square Error

عمل کنند. بنابراین، یک مبادله یا بده‌بستان^۱ بین تفسیرپذیری و دقت وجود دارد. سیستم‌های استنتاج از نوع تاکاگی-سوگنو دقیق‌تر هستند اما نیاز به تلاش محاسباتی بیشتری دارند؛ درحالی‌که سیستم‌های استنتاج از نوع ممدانی قابلیت تفسیرپذیری بیشتری داشته و به بار محاسباتی کمتری نیاز دارند اما اغلب دارای مصالحه و کاستی در دقت هستند. جزئیات بیشتر درباره این مثال سری زمانی را می‌توانید در پژوهش ابراهام^۲ و نات^۳ (۲۰۰۰) بیابید. لازم به ذکر است که EFuNN و dmEFuNN در فصل بعد در بخش شبکه‌های عصبی فازی تکاملی معرفی شده‌اند.

از سوی دیگر، ویهاروس^۴ و کیس^۵ (۲۰۱۵) در مطالعه‌ای به مقایسه شش ساختار رایج فازی عصبی براساس مزیت‌ها و ایرادهای آنها پرداختند. جدول ۴-۴ جزئیات این مقایسه را نشان می‌دهد.

بهینه‌سازی سیستم‌های استنتاج فازی با یادگیری شبکه عصبی و محاسبات تکاملی

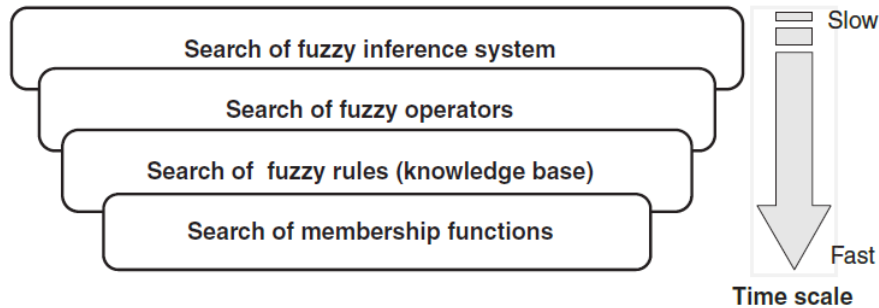
در یک مدل فازی عصبی هیبریدی (یکپارچه)، تضمینی وجود ندارد که الگوریتم یادگیری شبکه عصبی همگرا شود و تنظیم سیستم استنتاج فازی با موفقیت همراه گردد. هوش طبیعی محصولی از تکامل است. بنابراین، می‌توان هوش سطح بالا را نیز با تقلید از تکامل بیولوژیکی شبیه‌سازی کرد. محاسبات تکاملی براساس شبیه‌سازی جمعیتی از افراد، ارزیابی عملکرد آنها و چندین ایجاد تکامل در این جمعیت انجام می‌شود تا زمانی که پاسخ (حل) مورد نیاز حاصل گردد. نقاط ضعف مربوط به شبکه‌های عصبی و سیستم‌های استنتاج فازی مکمل به‌نظر می‌رسند و محاسبات تکاملی می‌تواند به‌منظور بهینه‌سازی این یکپارچه‌سازی مورد استفاده واقع شود تا بهترین هم‌افزایی ممکن برای ساخت یک سیستم منفرد به‌وجود آید. انطباق سیستم‌های استنتاج فازی با استفاده از روش‌های محاسباتی تکاملی به‌طور گسترده‌ای مورد پژوهش واقع شده است. EvONF یک چارچوب انطباقی براساس محاسبات تکاملی و یادگیری عصبی است که در آن توابع عضویت، پایگاه قواعد و عملگرهای فازی برحسب مسأله تطبیق داده می‌شوند. جستجوی تکاملی برای توابع عضویت، پایگاه قواعد، عملگرهای فازی و غیره برحسب محیط مسأله در

1. Tradeoff
2. Abraham, A.
3. Nath, B.
4. Viharos, Zs. J.
5. Kis, K. B.

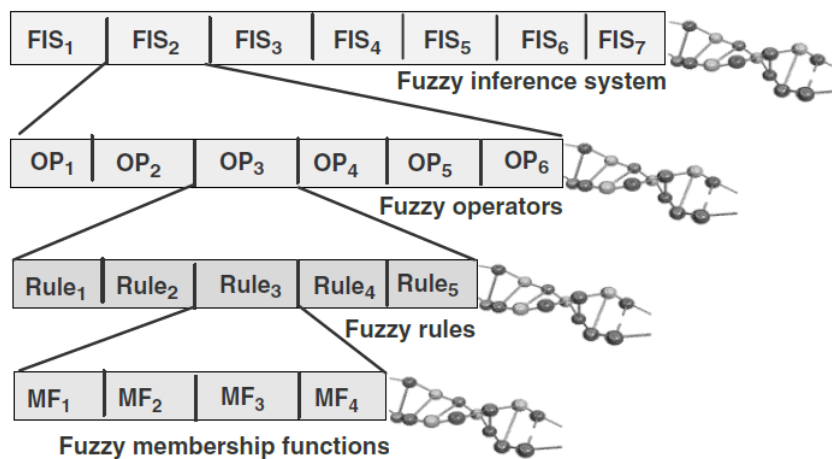
مقیاس‌های زمانی مختلف برای تطبیق سیستم استنتاج فازی پیش می‌رود. توابع عضویت و عملگرهای فازی با استفاده از روش یادگیری عصبی بهتر تنظیم می‌گردد. پارامترهای بهینه در یادگیری عصبی طی فرآیند جستجوی تکاملی مشخص می‌شوند. شکل ۴-۲۵ سازوکار تعاملی عام را برای چارچوب EvoNF با جستجوی تکاملی در سیستم استنتاج فازی (ممدانی، تاکاگی-سوگنو و غیره) نشان می‌دهد که در بالاترین سطح و آهسته‌ترین مقیاس زمانی قرار دارد. در هر جستجوی تکاملی برای عملگرهای فازی (بهترین ترکیب t-نرم، t-کونورم، استراتژی فازی‌زدایی و غیره)، جستجوی پایگاه قواعد فازی در یک مقیاس زمانی سریع‌تر در محیطی که توسط مسأله مشخص می‌شود، پیش می‌رود. جستجوی تکاملی برای توابع عضویت نیز به طریق مشابهی، در یک مقیاس زمانی سریع‌تر (برای هر پایگاه قاعده) در محیطی که توسط مسأله تصمیم‌گیری می‌شود، ادامه می‌یابد. سلسله‌مراتب رویه‌های مختلف انطباقی به دانش قبلی متکی خواهد بود. برای مثال، اگر دانش قبلی بیشتری درباره پایگاه قواعد فازی نسبت به عملگرهای فازی وجود داشته باشد، آنگاه بهتر است که اجرای جستجو برای پایگاه قواعد فازی در سطح بالاتری انجام گیرد. نمایشی از این مسأله (کدگذاری ژنتیک) در شکل ۴-۲۶ آشکار است. برای جزئیات فنی بیشتر مطالعه پژوهش ابراهام^۱ (۲۰۰۵) توصیه می‌شود. به کارگیری یادگیری عصبی و محاسبات تکاملی در سیستم‌های استنتاج فازی با عناوین طراحی تکاملی سیستم‌های فازی عصبی^۲ (EvoNF) یا یادگیری عصبی و تکاملی برای سیستم استنتاج فازی^۳ نیز مطرح شده است.

انطباق خودکار توابع عضویت معمولاً با عنوان خودتنظیمی شناخته می‌شود و کدگذاری پارامترهای توابع عضویت دوزنقه‌ای، مثلثی، لجستیک، تانژانت هایپربولیک، گوسی و ... توسط ژنوم^۴ انجام می‌پذیرد. جستجوی تکاملی برای قواعد فازی را می‌توان با استفاده از سه رویکرد انجام داد. در روش اول (رویکرد میشیگان^۵) به دلیل تضاد نقش‌های رقابت و همکاری در قواعد فازی، پایگاه دانش فازی تطبیق می‌یابد. هر ژنوتایپ^۶ یک قاعده فازی تنها را بیان کرده و همه جمعیت پاسخی را ارائه می‌کند. هرگاه بخش شرط یک قاعده طبقه‌بندی‌کننده با ورودی فعلی منطبق می‌شود، آن قاعده اعمال شده که در

1. Abraham, A.
2. Evolutionary Design of Neuro-Fuzzy Systems
3. Evolutionary and Neural Learning of Fuzzy Inference System
4. Genome
5. Michigan
6. Genotype



شکل ۴-۲۵: اثر متقابل سازوکار جستجوی تکاملی در تطبیق سیستم استنتاج فازی



شکل ۴-۲۶: نمایش کروموزم در سیستم استنتاج فازی انطباقی با استفاده از محاسبات تکاملی و یادگیری عصبی

این صورت اقدام موردنظر به منظور کنترل به فرآیند ارسال می‌گردد. الگوریتم جستجوی کلی، قواعد طبقه‌بندی‌کننده را براساس شدت حاصل شده برای قواعد در خلال کل فرآیند ایجاد می‌کند. این رفتار فازی به وسیله یک رشته (دنباله) فعال‌سازی از قواعد فازی با همکاری دوبه‌دو، ایجاد می‌شود. تمام پایگاه دانش به کمک مشارکت قواعد فازی چندگانه متقابل (رقابتی) ساخته می‌شود.

روش دوم (رویکرد پیتسبورگ^۱) جمعیتی از پایگاه‌های دانش را به جای قواعد فازی مجزا تکامل می‌دهد. عملگرهای ژنتیک به منظور ایجاد ترکیب جدیدی از قواعد و قواعد تازه به کار می‌روند. در برخی از موارد، پایگاه قواعد با طول‌های متفاوت مورد استفاده واقع می‌شوند که برای مواجهه با این تفاوت طول و مکان ژنوم‌های مستقل، از عملگرهای ژنتیک اصلاح‌شده استفاده می‌شود. نقطه ضعف در اینجا، افزایش پیچیدگی در فضای جستجو و بار محاسباتی اضافی به‌ویژه در یادگیری آنلاین است.

روش سوم (رویکرد یادگیری تکرارشونده قواعد^۲) بسیار شبیه به روش اول است که هر کرموزوم نماینده یک قاعده است. اما برخلاف رویکرد میشیگان، فقط بهترین فرد برای شکل‌دهی به بخش پاسخ در نظر گرفته می‌شود و این یعنی مابقی کرموزوم‌های جمعیت کنار گذاشته می‌شوند. فرآیند یادگیری تکاملی، پایگاه قواعد کامل را از طریق یک فرآیند یادگیری تکرارشونده می‌سازد.

برای دسترسی به جزئیات روش‌های فوق، می‌توانید به پژوهش‌های کوردون^۳ و هررا^۴ (۲۰۰۱) و

گونزالس^۵ و هررا (۱۹۹۷) مراجعه کنید.

-
1. Pittsburgh
 2. Iterative rule learning
 3. Cordon, O.
 4. Herrera, F.
 5. Gonzalez, A.

جدول ۴-۴: مقایسه ساختارهای رایج فازی عصبی

ساختار	مزیت‌ها	ایرادها
ANFIS	- از آنجایی که از سازوکار استنتاج تاکاگی - سوگنو - کانگ استفاده می‌کند، مدل خیلی دقیقی است.	- در مدل پیش‌فرض، خروجی چندگانه پشتیبانی نمی‌شود. - فقط امکان آموزش ساختارهای تمام تعریف‌شده وجود دارد. - ایجاد یا کاهش قواعد به صورت پویا وجود ندارد.
FALCON	- یک مرحله یادگیری برای ایجاد ساختار اولیه (تعیین تعداد قواعد و توابع عضویت از طریق آموزش) وجود دارد. - استنتاج از نوع ممدانی موجب می‌شود که تفسیرپذیری بهتری ایجاد شود.	- استنتاج از نوع ممدانی موجب می‌شود که تخمین پارامتر با دقت کمتری انجام شود.
NEFCON	- قادر است که تعداد قواعد را در خلال یادگیری کاهش دهد. - استنتاج از نوع ممدانی موجب می‌شود که تفسیرپذیری بهتری ایجاد شود.	- استنتاج از نوع ممدانی موجب می‌شود که تخمین پارامتر با دقت کمتری انجام شود.
GARIC	- یکی از اولین سیستم‌های فازی عصبی است.	- یک مدل واحد نیست ولی از چند جزء تشکیل شده است.
SONFIN	- هیچ قاعده اولیه‌ای وجود ندارد بلکه قواعد در قالب یادگیری آنلاین از طریق تشخیص همزمان پارامتر و ساختار ایجاد شده و اصلاح می‌شود. - قواعد تولیدشده و توابع عضویت - حتی در مدل‌سازی سیستم‌های پیچیده - تعداد کمی دارند.	- اگرچه شبیه‌ترین ساختار به ANFIS است اما به دقت آن نیست.
dmEFuNN	- طی آموزش آنلاین می‌تواند تعداد قواعد سیستم را افزایش یا کاهش دهد.	- به دقت ANFIS نیست.

فصل ۵

شبکه‌های عصبی مبتنی بر منطق فازی

در این فصل، مدل‌هایی مورد بررسی قرار خواهند گرفت که منطق فازی را در شبکه‌های عصبی ترکیب می‌کنند. سیستم‌هایی که به این شیوه ایجاد می‌شوند را مدل‌های شبکه عصبی مبتنی بر منطق فازی^۱ یا شبکه‌های عصبی فازی^۲ می‌نامند. این سیستم‌ها در دسته سیستم‌های یکپارچه هستند زیرا عملاً یک سیستم واحد تلقی می‌شوند. شبکه‌های عصبی فازی در حقیقت شبکه‌های عصبی فازی‌سازی شده هستند و بنابراین ذاتاً شبکه عصبی محسوب می‌شوند.

شبکه‌های عصبی توانایی و قابلیت‌های خود را در طبقه‌بندی، ذخیره، فراخوانی و یکپارچه‌سازی اطلاعات ثابت کرده‌اند. با ترکیب قوانین فازی در یک شبکه عصبی، انعطاف بیشتری حاصل شده و سیستم استوارتر می‌گردد. به عبارت دیگر امکان تعریف مرزهای سیستم به‌طور مبهم‌تر وجود داشته و سیستم قادر است که پیکره‌بندی خود را به‌منظور بهترین عملکرد تجدید کند. در شبکه‌های عصبی فازی امکان جایگزینی پارامترهای کنترل عددی در گره و وزن‌های اتصال با پارامترهای فازی وجود دارد. شبکه‌های عصبی فازی نسبت به سیستم‌های عصبی دارای قدرت ارائه بیشتر، سرعت آموزش بالاتر و استواری بیشتر هستند. هر بخش از یک شبکه عصبی (نظیر توابع فعال‌سازی، توابع انبوه‌سازی، وزن‌ها، داده‌های ورودی-خروجی و غیره)، هر مدل شبکه عصبی و هر الگوریتم یادگیری عصبی را به‌طور بالقوه می‌توان

1. Fuzzy Logic-Based Neural Network Models

2. Fuzzy Neural Network

فازی نمود. براساس این مفهوم، چند دسته شبکه عصبی فازی شناسایی شده که در این فصل تشریح می‌شوند. این دسته‌ها شامل نورون‌های فازی، مدل‌های عصبی فازی‌سازی‌شده و شبکه‌های عصبی با آموزش فازی هستند.

۱-۵ نورون‌های فازی

نورون‌ها یا گره‌های شبکه‌های عصبی و سیستم‌های عصبی شامل واحدهای پردازنده‌ای هستند که ورودی‌ها و خروجی‌های عددی را پردازش می‌کنند. در دنیای واقعی، موارد متعددی وجود دارد که سنجش‌های عددی در دسترس نیستند یا با نویز مخدوش شده‌اند و یا دچار عدم قطعیت هستند. بنابراین برخی از پژوهشگران توصیه کرده‌اند که این نورون‌ها باید توانایی پردازش چنین موقعیت‌هایی را کسب کنند و از این‌رو تلاش کرده‌اند که آنها را با نورون فازی یکپارچه نمایند. ساختار اصلی نورون فازی مشابه نورون عصبی مصنوعی است به‌استثنای ورودی‌ها، فرآیند پردازش و خروجی‌ها که به‌وسیله منطق فازی تعریف می‌شوند. بنابراین نورون‌های فازی متنوعی ابداع شده و در ادبیات این موضوع یافته می‌شوند.

در این بخش، چند مدل اصلی از نورون‌های فازی معرفی خواهد شد که می‌توان شبکه‌های عصبی فازی را با استفاده از آنها ایجاد کرد. یک نورون فازی^۱ به‌شيوه‌ای کاملاً شبیه به نورون مک‌کلاچ-پیتز^۲ کلاسیک با غیرفازی طراحی می‌شود به‌استثنای آنکه نورون فازی، طبیعت فازی یک نورون را انعکاس می‌دهد و توانایی مواجهه با اطلاعات فازی را دارد. به‌طوراساسی، براساس پژوهش گوپتا^۳ و کیوای^۴ (۱۹۹۱) دو گونه مدل نورون فازی را می‌توان شناسایی نمود. اولی از فازی‌سازی مدل‌های نورون غیرفازی ایجاد می‌شود و دیگری مدلی است که روابط ورودی-خروجی نورون‌ها با استفاده از قواعد اگر-آنگاه فازی تعریف می‌گردد. در این بخش، سه نوع مدل عصبی فازی تشریح می‌شود:

۱. نورون فازی با سیگنال قطعی که برای ارزیابی وزن‌های فازی مورد استفاده قرار می‌گیرد.

۲. نورون فازی با سیگنال‌های فازی که با وزن‌های فازی ترکیب می‌شود.

1. Fuzzy neuron
2. McCulloch-Pitts
3. Gupta, M. M.
4. Qi, J.

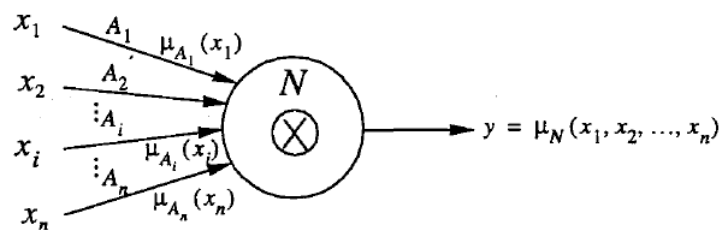
۳. نورون فازی که با معادلات منطقی فازی تعریف می‌شود.

۱-۱-۵ نورون فازی نوع یک

این نوع از نورون‌های فازی در شکل ۱-۵ به نمایش درآمده است. این نورون دارای n ورودی غیرفازی (x_1, x_2, \dots, x_n) است. وزن‌های N مجموعه‌های فازی A_i ($1 \leq i \leq n$) هستند؛ بنابراین عملیات وزن‌دهی با توابع عضویت جایگزین می‌شود. نتایج هر عمل وزن‌دهی، مقدار عضویت متناظر ورودی x_i در مجموعه فازی (وزن) A_i است که با $\mu_{A_i}(x_i)$ نشان داده می‌شود. تمام این مقادیر عضویت با یکدیگر انبوه‌سازی می‌شود تا یک خروجی در بازه $[0, 1]$ حاصل گردد که می‌توان آن را سطح اطمینان تلقی کرد. عمل انبوه‌سازی - که با \otimes نمایش داده می‌شود - می‌تواند از هر عملگر انبوه‌سازی مانند \max , \min یا هر t -نرم یا t -کونورم دیگری استفاده کند. یک بیان ریاضی از چنین نورون فازی N به صورت زیر نوشته می‌شود:

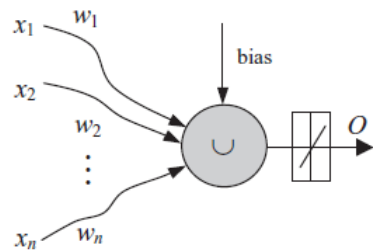
$$\mu_N(x_1, x_2, \dots, x_n) = \mu_{A_1}(x_1) \otimes \mu_{A_2}(x_2) \otimes \dots \otimes \mu_{A_i}(x_i) \otimes \dots \otimes \mu_{A_n}(x_n) \quad (1-5)$$

x_i در رابطه فوق، ورودی (غیرفازی) نام نورون را نشان می‌دهد. μ_{A_i} تابع عضویت i امین وزن (فازی)، μ_N تابع عضویت خروجی نورون و \otimes یک عملگر انبوه‌سازی است.

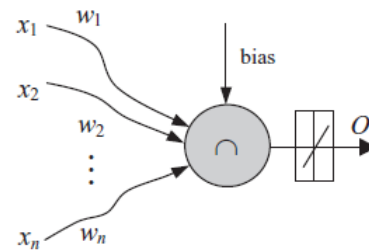


شکل ۱-۵: نورون فازی نوع یک

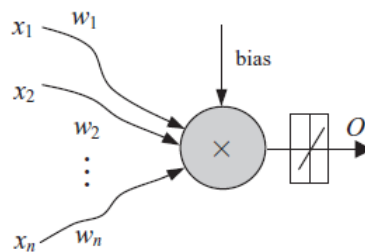
در شکل ۲-۵ تعداد محدودی از نورون‌های نوع یک به نمایش درآمده است و وزن موردنظر در آنها برپایه تابع عضویت است که پیش‌تر ذکر شد. ساختار یک شبکه نورون فازی نیز در شکل ۳-۵ ارائه شده است.



(ب) نورون فازی OR (Max)



(آ) نورون فازی AND (Min)



(ج) نورون فازی ضرب

شکل ۲-۵: چندین نورون فازی مختلف

شکل ۲-۵ (آ) یک نورون min فازی را نشان می‌دهد که عمل انبوه‌سازی را با انتخاب حداقل

ورودی‌های موزون انجام می‌دهد:

$$O = f\left(\bigwedge_{i=1}^n x_i w_i + b\right) \quad (2-5)$$

به‌طور مشابه، شکل ۲-۵ (ب) یک نورون max فازی را نشان می‌دهد که عمل انبوه‌سازی را با انتخاب

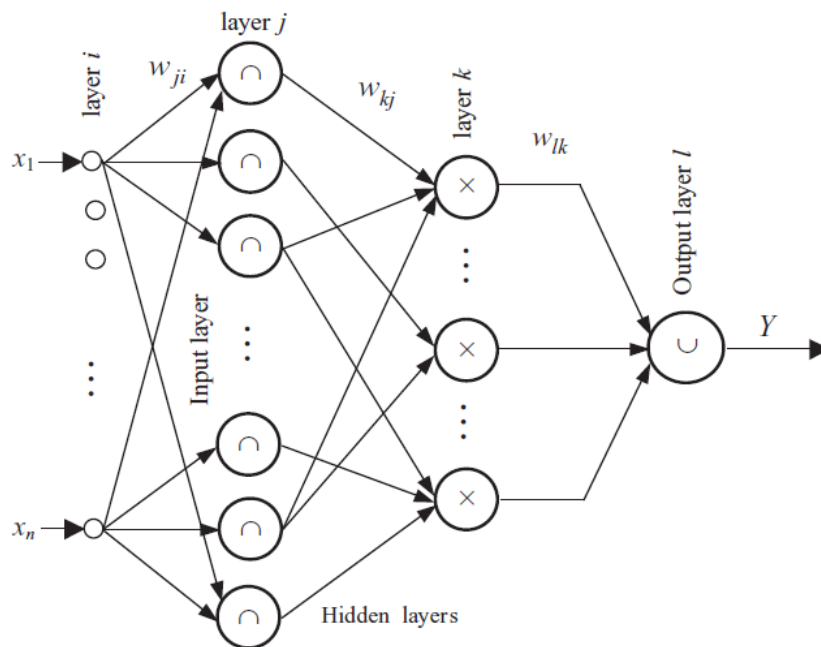
حداکثر ورودی‌های موزون انجام می‌دهد:

$$O = f\left(\bigvee_{i=1}^n x_i w_i + b\right) \quad (3-5)$$

شکل ۲-۵ (ج) نیز یک نورون ضرب فازی را نشان می‌دهد که عمل ضرب همه ورودی‌های موزون را

انجام می‌دهد:

$$O = f\left(\prod_{i=1}^n x_i w_i + b\right) \quad (4-5)$$



شکل ۵-۳: یک شبکه نرون فازی ساده

$f(\cdot)$ تابع فعال‌سازی است. شکل ۵-۳ یک شبکه عصبی فازی چندلایه را نشان می‌دهد. لایه اول، یک لایه ورودی است که ورودی‌ها (x_i) - که توسط w_{ji} موزون شده‌اند - را به لایه دوم انتقال می‌دهد. لایه دوم از نرون‌های AND (\min) تشکیل شده است و بیشینه مقادیر ورودی‌اش را به صورت خروجی بیرون می‌دهد. لایه سوم از نرون‌های ضرب تشکیل شده و خروجی آن، حاصل ضرب ورودی‌هایش است. لایه نهایی، لایه خروجی است که با استفاده از نرون OR (\max) به انبوه‌سازی تمام ورودی‌ها می‌پردازد. آشکار است که الگوریتم‌های آموزش نظیر پس‌انتشار را می‌توان به همراه یک مجموعه از داده‌های ورودی/خروجی برای آموزش به کار برد. تعدادی از کاربردهای شبکه‌های عصبی فازی در ادبیات این

موضوع گزارش شده است (پدريچ^۱ و روشا^۲، ۱۹۹۳؛ تسوکالاس^۳ و یوریگ^۴، ۱۹۹۷؛ ژانگ^۵ و کندل^۶، ۱۹۹۰).

مثالی که در ادامه می‌آید، استفاده از نورون‌های فازی نوع یک را در ساخت یک شبکه عصبی فازی به‌عنوان کنترل‌کننده فازی نشان می‌دهد.

مثال ۵-۱. مثالی از به‌کارگیری نورون‌های فازی نوع یک

یک کنترل‌کننده فازی را فرض کنید که قواعد آن در جدول ۵-۱ و تعاریف اعداد فازی آن در شکل ۵-۴ ارائه شده است. این کنترل‌کننده فازی، خطا (e) و تغییرات خطای (Δe) ورودی‌های منفرد را می‌پذیرد. با فرض مقادیر e و Δe ، ۹ قاعده فازی به‌صورت زیر برآورد می‌شود:

$$\Delta_1 = \min(\mu_{F_1}(e), \mu_{G_1}(\Delta e)),$$

$$\Delta_2 = \min(\mu_{F_2}(e), \mu_{G_2}(\Delta e)),$$

⋮

$$\Delta_9 = \min(\mu_{F_9}(e), \mu_{G_9}(\Delta e)).$$

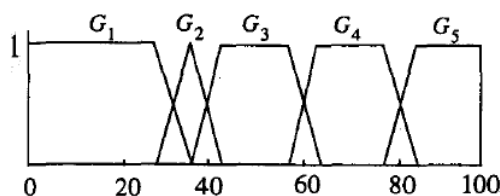
از آنجایی که برخی قواعد اقدام‌های کنترلی مشابه دارند، حداکثرسازی Δ_i متناظر اقدام مشابه A_k به‌صورت $\epsilon_1 = \max(\Delta_1, \Delta_2)$ ، $\epsilon_2 = \max(\Delta_3, \Delta_4)$ ، $\epsilon_3 = \Delta_5$ ، $\epsilon_4 = \max(\Delta_6, \Delta_7)$ و $\epsilon_5 = \max(\Delta_8, \Delta_9)$ انجام می‌شود. بنابراین هر ϵ_k به A_k خویش ($1 \leq k \leq 5$) اختصاص دارد. به‌منظور فازی‌زدایی این نتایج، ابتدا $A = \bigcup(\epsilon_k, A_k)$ محاسبه شده (اجماع در قالب حداکثری انجام می‌شود) و سپس y که با مرکز ثقل A برابر است، یافته می‌شود. y خروجی این کنترل‌کننده است که فازی‌زدایی شده است.

-
1. Pedrycz, W.
 2. Rocha, A. F.
 3. Tsoukalas, L. H.
 4. Uhrig, R. E.
 5. Zhang, Y. Q.
 6. Kandel, A.

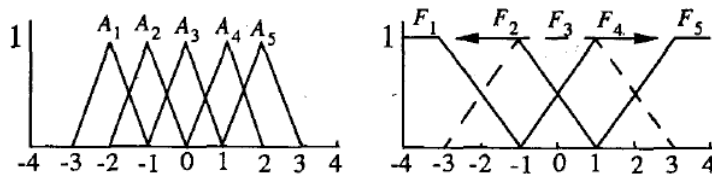
جدول ۵-۱: قواعد به کاررفته در مثال کنترل کننده فازی

تغییر در خطا $\Delta e =$

G_5	G_4	G_3	G_2	G_1	
			$A_1^{(1)}$		F_1
	$A_2^{(2)}$		$A_1^{(2)}$		F_2
$A_4^{(6)}$		$A_3^{(5)}$		$A_2^{(4)}$	F_3 خطا = e
	$A_5^{(8)}$		$A_4^{(7)}$		F_4
	$A_5^{(9)}$				F_5



(ا) اعداد فازی خطا



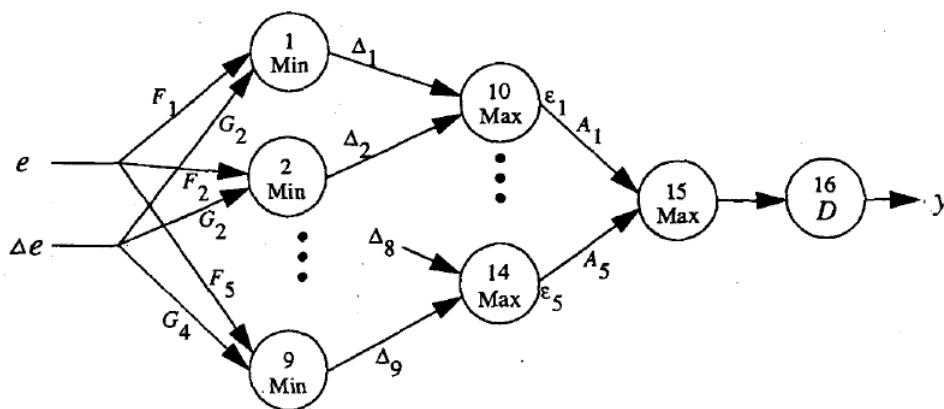
(ج) اعداد فازی خروجی

(ب) اعداد فازی تغییر در خطا

شکل ۵-۴: اعداد فازی استفاده شده در مثال کنترل کننده فازی

این کنترل کننده فازی که در قالب یک شبکه عصبی فازی مدل شده است، در شکل ۵-۵ نمایش داده می شود. در این شبکه، گره های ۱ تا ۹ نشان گر قواعد هستند. گره های ۱۰ تا ۱۴ هیچ وزن ورودی ندارند و گره ۱۶ - بدون هیچ وزنی - نقش فازی زدایی را ایفا می کند. به طور آشکار، گره های ۱ تا ۹ نورون های فازی نوع یک هستند اما گره ۱۵ شباهت زیادی به یک نورون فازی نوع دو دارد که در

بخش بعدی معرفی خواهد شد. دلیل این است که ورودی‌های گره ۱۵، $\epsilon_i A_i$ ($1 \leq i \leq 5$) هستند که مجموعه‌هایی فازی‌اند، درحالی‌که ورودی‌های گره ۱، $\mu_{F_1}(e)$ و $\mu_{G_1}(\Delta e)$ هستند که اسکالر (نرده‌ای) محسوب می‌شوند. انتظار می‌رود که این شبکه عصبی فازی ایجادشده بتواند اعداد فازی مناسب (A_p, G_j, F_i) را بیاموزد در شرایطی که تعدادی داده آموزشی $(e_k, \Delta e_k)$ برای ورودی‌ها و y_k ($k = 1, 2, \dots$) برای خروجی‌های فازی‌زدایی‌شده مطلوب مفروض باشند. به منظور دستیابی به این موضوع، به یک الگوریتم یادگیری نیاز است که شبکه وزن‌های آن را از داده‌های آموزش فراگیرد. جزئیات این موضوع در بخش‌های دیگر این فصل با محوریت آموزش فازی برای شبکه‌های عصبی مصنوعی مورد بررسی قرار می‌گیرد.



شکل ۵-۵: شبکه عصبی فازی به‌عنوان یک کنترل‌کننده فازی

۵-۱-۲ نورون فازی نوع دو

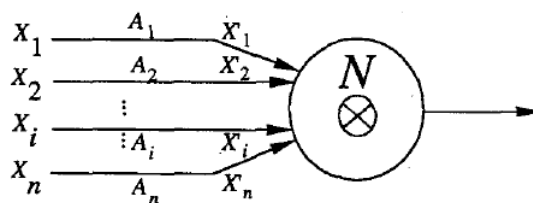
این نوع از نورون‌های فازی در شکل ۵-۶ به‌نمایش درآمده است. این نورون‌ها به نورون‌های فازی نوع یک شبیه هستند با این تفاوت که تمامی ورودی‌ها و خروجی‌ها به‌جای اینکه مقادیر قطعی باشند، از جنس مجموعه‌های فازی هستند. هر ورودی فازی (X_i) دست‌خوش یک عمل وزن‌دهی شده که منجر به یک مجموعه فازی دیگر می‌شود: $X'_i = A_i * X_i$ که $1 \leq i \leq n$ بوده و A_i وزن فازی نام است. همه ورودی‌های اصلاح‌شده به‌منظور تولید یک مجموعه فازی n بعدی (Y) انبوه‌سازی می‌شود. برخلاف

شرایط نورون‌های فازی نوع یک، عمل وزن‌دهی در اینجا یک تابع عضویت نیست بلکه تعدیل‌کننده‌ای برای هر ورودی فازی است. تعریف ریاضی این نورون فازی به صورت زیر انجام می‌شود:

$$X'_i = A_i * X_i, \quad i = 1, 2, \dots, n, \quad (5-5)$$

$$Y = X'_1 \otimes X'_2 \otimes \dots \otimes X'_i \otimes \dots \otimes X'_n,$$

Y مجموعه فازی خروجی مربوط به نورون فازی را بیان می‌کند. ورودی‌های نام قبل و بعد از عمل وزن‌دهی به ترتیب با X_i و X'_i نشان داده شده است. A_i وزن اتصال نام، \otimes عملگر انبوه‌سازی ذکر شده در نورون فازی نوع یک و $*$ عملگر وزن‌دهی است که مانند عمل ضرب دو مجموعه فازی عمل می‌کند. اکنون مثال زیر را در نظر بگیرید که در آن، یک شبکه عصبی فازی شامل تعدادی نورون فازی از نوع دو برای شناسایی سیستم فازی است.



شکل ۵-۶: نورون فازی نوع دو

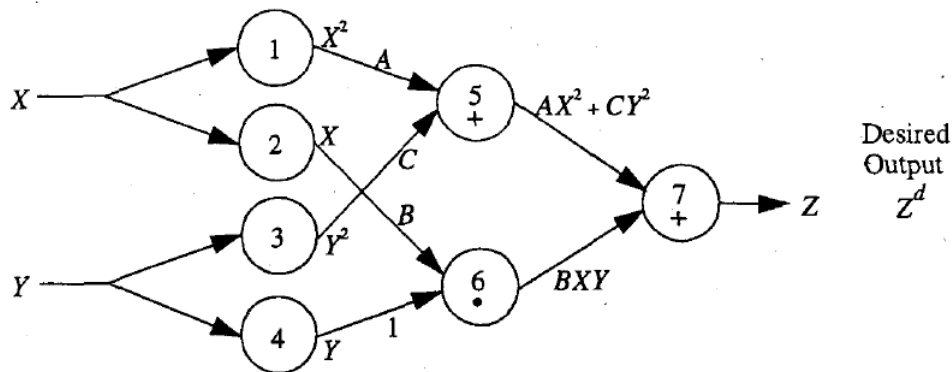
مثال ۵-۲. مثالی از به‌کارگیری نورون‌های فازی نوع دو

یک مسأله شناسایی سیستم^۱ را در نظر بگیرید که می‌خواهیم تابع f را در آن بیابیم. f تابعی است که ورودی‌های فازی X و Y را به خروجی فازی Z تبدیل می‌کند. در اینجا فرض می‌کنیم که یک رابطه کوادراتیک برای اعداد فازی A ، B و C وجود دارد:

$$AX^2 + BXY + CY^2 = Z \quad (6-5)$$

رابطه فوق توسط یک شبکه عصبی فازی (شکل ۵-۷) به‌کارگرفته می‌شود که در آن از نورون‌های فازی نوع دو استفاده شده است. انتظار می‌رود که این شبکه با استفاده از تعدادی داده آموزشی (X_k, Y_k)

1. System identification



شکل ۵-۷: برازش تابع کوادراتیک با شبکه عصبی فازی

برای ورودی‌ها و خروجی متناظر مطلوب Z_k^d ($k = 1, 2, \dots$) بتواند A ، B و C را بیاموزد. چنین فرآیند یادگیری در بخش ۳-۵ تشریح می‌شود.

۳-۱-۵ نورون فازی نوع سه

یک نورون فازی نوع سه با n ورودی فازی و یک خروجی فازی در شکل ۵-۸ نمایش داده شده است. رابطه ورودی-خروجی این نورون فازی توسط یک قاعده اگر-آنگاه بیان می‌شود:

If X_1 AND X_2 AND ... AND X_n , Then Y

که X_1, X_2, \dots, X_n ورودی‌های کنونی و Y خروجی کنونی است. یک نورون فازی نوع سه را می‌توان با استفاده از یک رابطه فازی (R) تعریف کرد. مثلاً:

$$R = X_1 \times X_2 \times \dots \times X_n \times Y \quad (۷-۵)$$

یا در حالت کلی:

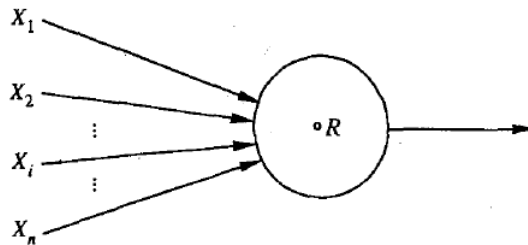
$$R = f(X_1, X_2, \dots, X_n, Y) \quad (۸-۵)$$

که f تابع استلزام را نشان می‌دهد. نورون فازی نوع سه توسط رابطه انتقال فازی (R) تعریف می‌شود. با فرض ورودی‌های کنونی فازی یا غیرفازی x_1, x_2, \dots, x_n ، براساس قاعده ترکیبی استنتاج نورون

فازی خروجی را به صورت زیر حاصل می‌کند:

$$Y'_i = x_1 \circ (x_2 \circ (\dots \circ (x_n \circ R_i) \dots)) \quad (9-5)$$

\circ نماد هر نوع قاعده ترکیبی از عملگر استنتاج مانند t -نرم \max است. باید یادآوری نمود که ورودی‌های این نورون می‌تواند فازی یا غیرفازی باشد؛ مقادیر قطعی، حالت خاصی از مقادیر فازی محسوب می‌شوند.



شکل ۵-۸: نورون فازی نوع سه

به نظر می‌رسد که شبکه عصبی فازی ساخته شده براساس نورون‌های فازی نوع سه برای استخراج قاعده از داده‌های آموزشی در سیستم‌های خبره فازی ایده آل است. مثال زیر این کاربرد را آشکار می‌سازد.

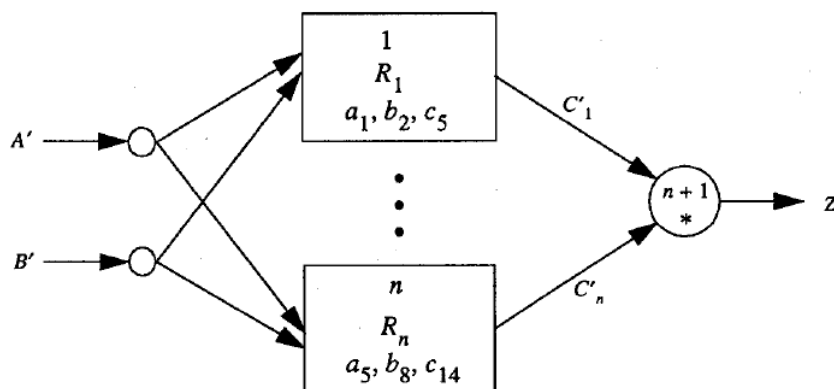
مثال ۵-۳. مثالی از به‌کارگیری نورون‌های فازی نوع سه

مجموعه قواعد زیر مفروض است:

$$c_p \text{ is } z \text{ Then } B_j \text{ is } Y \text{ AND } A_i \text{ is } X \text{ If } :R_r \quad (1 \leq r \leq n)$$

که A_i ، B_j و C_p زیرمجموعه‌های فازی R هستند. برای سادگی، فرض می‌شود که تمام این مجموعه‌های فازی، اعداد فازی مثلثی متقارن با پهنای ثابت هستند و بنابراین توسط مقادیر مرکزی‌شان به طور کامل معین می‌شوند. مقادیر مرکزی به ترتیب برابر با a_i ، b_j و c_p قرار داده می‌شوند.

با فرض داده جدید $X = A'$ و $Y = B'$ ، بعضی روش‌های استدلال تقریبی به‌کار گرفته می‌شود تا این قواعد را ارزیابی کند. ابتدا باید یک عمل‌گر استلزام (M) انتخاب شود تا یک رابطه فازی (R_r) را برای هر قاعده ایجاد نماید. قرار می‌دهیم $R_r(a_i, b_j, c_p) = M(A_i, B_j, C_p)$ که یک رابطه فازی در \mathbb{R}^2 برای قاعده r ($1 \leq r \leq n$) است. با $X = A'$ و $Y = B'$ مقدار $C'_r = (A', B') \circ R_r(a_i, b_j, c_p)$ توسط هر قاعده محاسبه می‌شود ($1 \leq r \leq n$). سپس این نتایج (C'_r) در یک نتیجه نهایی (C') توسط سیستم خبره فازی ترکیب می‌شود. این سیستم خبره فازی را می‌توان به‌وسیله یک شبکه عصبی فازی حاوی نورون‌های فازی نوع سه مدل کرد، همان‌طور که در شکل ۵-۹ به‌نمایش درآمده است. این رابطه فازی (R_r) در r گره قرار دارد به‌طوری که هر گره دارای سه پارامتر a_i ، b_j و c_p است. به‌عنوان مثال، فرض می‌شود که A_1 ، B_2 و C_5 در گره R_1 ظاهر می‌شود و پارامترهای این گره مقادیر مرکزی a_1 ، b_2 و c_5 هستند. $n + 1$ گره ورودی‌ها را با استفاده از عمل * ترکیب می‌کند تا خروجی z حاصل گردد. بنابراین نورون ۱ تا n از جنس نورون‌های فازی نوع سه هستند اما نورون $n + 1$ به یک نورون فازی نوع دو شبیه‌تر است.



شکل ۵-۹: شبکه عصبی فازی برای طراحی یک سیستم خبره فازی

با وجود تعدادی داده آموزشی (A'_k, B'_k) برای ورودی‌ها همراه با خروجی مطلوب Y'_k ($k = 1, 2, \dots$)، انتظار می‌رود که یک الگوریتم یادگیری قواعد صحیح (A_i, B_j) و (C_p) را بیاموزد. طبیعتاً چنین الگوریتم یادگیری به تابع استلزام (M) و قاعده ترکیبی عمل‌گر استنتاج \circ بستگی دارد.

۲-۵ فازی‌سازی مدل‌های شبکه عصبی

در این بخش، چند شبکه عصبی فازی مورد بررسی قرار می‌گیرند که مستقیماً از فازی‌سازی مدل‌های شبکه‌های عصبی غیر فازی حاصل می‌شوند. این نوع از مدل‌های عصبی فازی، جریان اصلی شبکه‌های عصبی فازی را ایجاد می‌کنند.

۱-۲-۵ پرسپترون فازی

پرسپترون^۱، شبکه عصبی با یک نورون مصنوعی است که تحت فرآیند یادگیری بانظارت به مسأله طبقه‌بندی داده‌ها می‌پردازد. این مدل ذاتاً یک طبقه‌بند دودویی^۲ است که ورودی (برداری متشکل از اعداد حقیقی) را به مقدار خروجی (با مقدار باینری) متناظر می‌سازد. این عملیات به کمک نورون عصبی انجام می‌شود که در آن، جمع موزون همه ورودی‌ها با یک مقدار آستانه تطبیق داده شده و سپس خروجی نهایی تولید می‌شود. الگوریتم آموزش پرسپترون یک الگوریتم تکرارشونده است به این صورت که در هر مرحله، الگوریتم با توجه به نقاطی که درست طبقه‌بندی نشده‌اند مقادیر وزن و بایاس را تغییر می‌دهد تا این نقاط به درستی طبقه‌بندی شوند. اگر دو مجموعه از داده‌ها، به‌طور خطی تفکیک‌پذیر باشند، پیدا کردن یک ابرصفحه تفکیک‌کننده طی گام‌هایی محدود توسط الگوریتم پرسپترون تضمین می‌شود؛ اما اگر این دو مجموعه به‌طور خطی تفکیک‌پذیر نباشند، نه تنها الگوریتم پرسپترون قادر نیست که یک ابرصفحه تفکیک‌کننده را بیابد، بلکه روش مشخصی برای خاتمه‌دادن به الگوریتم در حاصل کردن یک مرز تصمیم بهینه یا حتی خوب وجود نخواهد داشت. با توجه به مقادیر الگوهای نمونه، رفتار الگوریتم

1. Perceptron

2. Binary classification

پرسپترون در موارد تفکیک‌ناپذیر خیلی ناسازگار خواهد بود. به‌منظور حل این مشکل روش‌ها و رویکردهای مختلفی در ادبیات یادگیری ماشین یافته می‌شود که در حال توسعه هستند. باین‌حال از منظر شبکه‌های عصبی فازی، کیلر^۱ و هانت^۲ (۱۹۸۵) با افزودن نظریه مجموعه‌های فازی به الگوریتم پرسپترون، پرسپترون فازی^۳ را ارائه کردند.

در الگوریتم قطعی یادگیری پرسپترون، هر الگوی آموزشی (بردار آموزشی) دارای اهمیت یکسانی در تنظیم وزن‌ها است. باین‌حال در حالت تفکیک‌ناپذیر، طبقه‌ها هم‌پوشانی دارند و الگوهایی که سبب این هم‌پوشانی می‌شوند، عامل اصلی رفتار ناسازگار و نامنظم الگوریتم قطعی پرسپترون هستند. در بسیاری از موارد، این الگوهای نامعین در هر طبقه نسبتاً غیر قابل تشخیص هستند. این همان جایی است که توابع عضویت طبقه فازی می‌تواند الگوریتم را بهبود بخشد. ایده اصلی پرسپترون فازی این است که با پایه‌گذاری مقدار تصحیح برای بردار وزن براساس عضویت‌های فازی، بردارهایی که طبقه آنها کمتر قابل تشخیص است (مقادیر عضویت نزدیک به ۰/۵) تأثیر کمتری بر تعیین بردار وزن داشته باشند. بنابراین برای مجموعه‌ای از بردارهای نمونه $\{x_1, x_2, \dots, x_p\}$ ، درجه عضویت هر بردار در هر یک از طبقه‌ها در یک افراز دو طبقه‌ای فازی با $\mu_i(x_k)$ نشان داده می‌شود که $i = 1, 2$ و $k = 1, 2, \dots, p$ است. در نتیجه ویژگی‌های زیر به دست می‌آید:

$$\sum_{i=1}^2 \mu_i(x_k) = 1, \quad 0 < \sum_{i=1}^p \mu_i(x_k) < p, \quad \mu_i(x_k) \in [0, 1] \quad (10-5)$$

به این ترتیب درجه تأثیر الگوی x_k بر به‌روزرسانی وزن را می‌توان با $\left| \mu_1(x_k) - \mu_2(x_k) \right|^m$ تعیین کرد که در آن m یک عدد ثابت است. طبیعتاً در الگوریتم پرسپترون فازی، قاعده به‌روزرسانی وزن به صورت زیر تبدیل می‌شود:

$$w_j(t+1) = w_j(t) + \eta \left| \mu_1(x_k(t)) - \mu_2(x_k(t)) \right|^m [y_k^d - y(t)] x_j^{(k)} \quad (11-5)$$

$$1 \leq j \leq n+1$$

1. Keller, J. M.
2. Hunt, D. J.
3. Fuzzy perceptron
4. Class

معادله فوق، هدف کاهش تأثیر بردارهای نامعین در تعیین بردار وزن را برآورده می‌کند. علاوه بر این مشابه پرسپترون قطعی، پرسپترون فازی می‌تواند یک ابرصفحه^۱ تفکیک‌کننده را طی تکرارهای محدودی در مسائل تفکیک‌پذیر خطی بیابد. از آنجایی که رابطه $1 = \left| \mu_1(x_k) - \mu_2(x_k) \right|^m$ با فرض $[\mu_1(x_k), \mu_2(x_k)] \in [0, 1]$ برقرار است، زمانی که مقادیر تابع عضویت قطعی باشند، پرسپترون فازی به پرسپترون قطعی تعدیل می‌شود.

انتخاب ثابت m در رابطه ۵-۱۱ نقش مهمی در همگرایی پرسپترون فازی ایفا می‌کند. به منظور بهترین استفاده از مقادیر عضویت فازی، الگوریتم یادگیری تعدیل‌های بزرگی را در خلال تکرارهای ابتدایی و در شرایطی که نقاط اشتباه طبقه‌بندی می‌شوند، انجام می‌دهد. در ادامه و طی تکرارهای پایانی تعدیل‌های کوچکی در بردار وزن ایجاد خواهد شد. بنابراین باید از انتخاب m کمتر از ۱ اجتناب نمود زیرا موجب تشدید تفاوت‌های ناچیز عضویت می‌شوند. مقدار m که برای دستیابی به نتایج خوب انتخاب می‌شود، به روش اختصاص عضویت‌های فازی به مجموعه‌های نمونه نیز بستگی دارد. قاعده کلی این است که اگر بردارهای نواحی هم‌پوشان توسط مقادیر عضویت نزدیک به ۰/۵ تخصیص یابند، تقریباً هر مقدار m که از ۱ بزرگتر باشد، نتایج خوبی تولید خواهد کرد. برعکس، اگر این بردارهای غیر نمونه‌ای مقادیر عضویت بالاتری داشته باشند، به توان بالاتری نیاز است تا نوسان‌های مرز تصمیم‌گیری تعدیل شوند. این موضوع در ادامه مورد بررسی قرار می‌گیرد.

با استفاده از قانون به‌روزرسانی وزن برای پرسپترون فازی در رابطه ۵-۱۱ باید روشی را برای تخصیص مقادیر عضویت فازی به مجموعه‌ای از بردارهای برچسب‌دار نمونه ارائه کرد. همان‌طور که در رابطه ۵-۱۰ نشان داده شد، رویکرد موردنظر تبدیل افزایش قطعی بردارها به افزایش فازی است.

روش تعیین مقدار عضویت یک بردار در طبقه‌ای که به آن متعلق است، به این ترتیب است: (آ) اگر بردار با میانگین طبقه‌اش برابر باشد، مقدار عضویت باید ۱ باشد؛ (ب) اگر بردار با میانگین طبقات دیگر برابر باشد، مقدار عضویت باید ۰/۵ باشد؛ (پ) اگر بردار از دو میانگین فاصله یکسانی داشته باشد، مقدار عضویت باید نزدیک به ۰/۵ باشد؛ (ت) مقدار عضویت هرگز نباید کمتر از ۰/۵ باشد؛ (ث) همان‌طور که یک بردار به میانگین خود نزدیک می‌شود و از میانگین دیگر دور می‌شود، مقدار عضویت باید به‌طور نمایی به ۱ میل کند؛ (ج) مقدار عضویت باید به فواصل نسبی از میانگین‌های طبقه‌ها وابسته باشد و نه

1. Hyperplane

به فواصل مطلق.

روش زیر برای تخصیص مقادیر عضویت فازی، توسط کلر^۱ و هانت^۲ (۱۹۸۵) پیشنهاد شده است و شرایط فوق را در نظر می‌گیرد.

برای x_k در طبقه ۱:

$$\mu_1(x_k) = 0.5 + \frac{e^{f(d_2-d_1)/d} - e^{-f}}{2(e^f - e^{-f})} \quad (12-5)$$

$$\mu_2(x_k) = 1 - \mu_1(x_k) \quad (13-5)$$

برای x_k در طبقه ۲:

$$\mu_2(x_k) = 0.5 + \frac{e^{f(d_1-d_2)/d} - e^{-f}}{2(e^f - e^{-f})} \quad (14-5)$$

$$\mu_1(x_k) = 1 - \mu_2(x_k) \quad (15-5)$$

در روابط فوق، d_1 فاصله بردار تا میانگین طبقه ۱، d_2 فاصله بردار تا میانگین طبقه ۲ و d فاصله بین دو میانگین است. مقدار ثابت f باید مثبت باشد. این مقدار، نرخ کاهش عضویت‌ها به سمت ۰/۵ را کنترل می‌کند. مشاهده شده است که انتخاب مقدار f به‌طور معکوس به m در رابطه ۵-۱۱ مرتبط است؛ یعنی زمانی که f کاهش می‌یابد، m باید افزایش پیدا کند تا با عضویت‌های بالاتر برای بردارهای نواحی هم‌پوشان مقابله کند. هر ترکیبی از m و f - اگر هر کدام بزرگتر از ۲ باشند - نتایج مطلوبی را در شبیه‌سازی‌ها از خود نشان داده‌اند.

پرسش دیگری که الگوریتم پرسپترون فازی باید به آن پاسخ بدهد این است که معیار توقف الگوریتم چیست؟ از آنجایی که افراز فازی جدید ممکن است تفکیک‌پذیر نباشد، به یک معیار توقف برای پرسپترون فازی نیاز است. دلیل تنظیم معیار توقف این است که بردارهایی که نامعین هستند - یعنی مقادیر عضویت در هر دو طبقه نزدیک به ۰/۵ است - نباید خودشان الگوریتم را به تکرار دیگری وادار کنند.

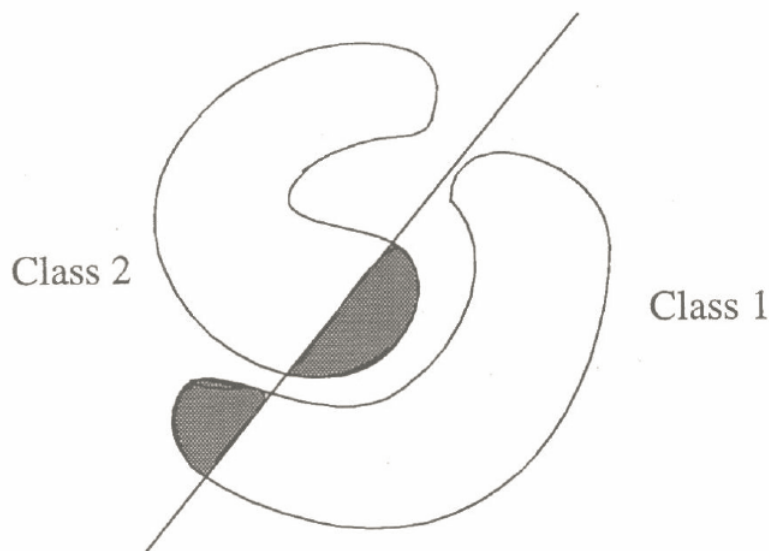
1. Keller, J. M.

2. Hunt, D. J.

بنابراین زمانی که الگوریتم یک تکرار کامل را برای همه بردارهای نمونه انجام می‌دهد، اگر همه تصحیح‌ها ناشی از بردارهای فازی باشند، آنگاه الگوریتم باید پایان یابد. به عبارت دیگر، زمانی که یک طبقه‌بندی اشتباه رخ می‌دهد، می‌توان با بررسی شرط زیر، معیار توقف را به کار گرفت:

$$\mu_1(\mathbf{x}_k) > 0.5 + \beta \quad \mu_1(\mathbf{x}_k) < 0.5 - \beta \quad (16-5)$$

β بازه‌ای را پیرامون ۰/۵ تعریف می‌کند که در آن بردارهای نامعین در نظر گرفته می‌شود. اگر شرط فوق برقرار شود، طبقه‌بندی نادرست متناظر نباید موجب تکرار دیگری گردد. لازم به ذکر است که شرط $\mu_1(\mathbf{x}_k) < 0.5 - \beta$ در رابطه ۱۶-۵ معادل شرط $\mu_2(\mathbf{x}_k) > 0.5 + \beta$ است زیرا فرض $\mu_1(\mathbf{x}_k) + \mu_2(\mathbf{x}_k) = 1$ در رابطه ۱۰-۵ وجود دارد. در ادامه، شیوه تعیین β در رابطه ۱۶-۵ توضیح داده می‌شود.



شکل ۵-۱۰: الگوریتم پرسپترون فازی

شرط لازم برای اینکه شرط توقف کار کند این است که بردارهای خارج از بازه $[0.5 - \beta, 0.5 + \beta]$ باید به طور خطی تفکیک‌پذیر باشند. شکل ۵-۱۰ را در نظر بگیرید. فرض کنید که خط تصمیم در شکل توسط الگوریتم پرسپترون فازی تعیین شده باشد. نواحی تیره داده‌هایی هستند که در معیار توقف صدق

نمی‌کنند. به منظور برآورده کردن شرط لازم، باید طوری β را انتخاب کرد که $\beta + 0.5$ بزرگتر یا مساوی مقدار عضویت بردارهایی باشد که از میانگین دو طبقه فاصله یکسانی دارند. لذا با قراردادن $d_1 = d_2$ در روابط ۱۲-۵ تا ۱۵-۵ خواهیم داشت:

$$\beta = \frac{1 - e^{-f}}{2(e^f - e^{-f})} + \epsilon \quad \epsilon \geq 0 \quad (17-5)$$

مقدار ϵ در این معادله اهمیت دارد. اگر ϵ خیلی کوچک باشد، ناحیه‌ای که ابرصفحه می‌تواند در آن واقع شود و معیار توقف را برآورده سازد، بسیار محدود خواهد بود. در نتیجه زمان زیادی توسط الگوریتم صرف خواهد شد تا به پایان برسد. از طرف دیگر، اگر ϵ خیلی بزرگ باشد، امکان دارد که الگوریتم خیلی سریع با تولید یک مرز تصمیم غیربهبینه متوقف شود. مطالعات تجربی نشان داده است که مقدار $\epsilon = 0.2$ نتایج خوبی را ایجاد می‌کند.

با جمع‌بندی توضیحات فوق، الگوریتم یادگیری پرسپترون فازی به صورت زیر قابل ارائه است.

FPLA: الگوریتم یادگیری پرسپترون فازی

گام ۱ (مقداردهی اولیه): $w_j(0)$ با $1 \leq j \leq n+1$ را اعداد تصادفی کوچک قرار دهید. $w_j(0)$ با $1 \leq j \leq n$ وزن اولیه‌ای است که به ورودی j ام وابسته است و $w_{n+1}(0)$ مقدار آستانه اولیه است.

گام ۲: درجه‌های عضویت همه الگوهای آموزشی را در هر طبقه با استفاده از روابط ۱۲-۵ تا ۱۵-۵ محاسبه کنید.

گام ۳: حالت به‌روزرسانی را برابر با FALSE قرار دهید.

گام ۴: یک الگوی ورودی جدید تکمیلی $\mathbf{x}_k = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}, -1]$ (که $x_{n+1}^{(k)} = -1$ است) را همراه با خروجی مطلوب $\{1, -1\}$ ارائه کرده و خروجی واقعی را محاسبه کنید:

$$y(t) = \operatorname{sgn} \left(\sum_{j=1}^{n+1} w_j(t) x_j^{(k)} \right) \quad (18-5)$$

اگر $y(t) = y_k^d$ باشد، آنگاه به گام ۷ بروید.

گام ۵: وزن‌ها را براساس قانون زیر تغییر دهید:

$$w_j(t+1) = w_j(t) + \eta \left| \mu_1(\mathbf{x}_k) - \mu_2(\mathbf{x}_k) \right|^m \left[y_k^d - y(t) \right] x_j^{(k)} \quad (5-19)$$

$$1 \leq j \leq n+1$$

گام ۶: اگر $\mu_1(\mathbf{x}_k) > 0.5 + \beta$ یا $\mu_1(\mathbf{x}_k) < 0.5 - \beta$ باشد، حالت به‌روزرسانی را به TRUE تغییر دهید.

گام ۷: اگر همه الگوهای ورودی ارائه شده باشند، به گام بعدی بروید و در غیراین صورت به گام ۴ بروید.

گام ۸: اگر حالت به‌روزرسانی برابر با TRUE باشد، به گام ۳ بروید و در غیراین صورت متوقف شوید و وزن‌هایی که مرز تصمیم آموخته شده را تعیین می‌کند، به‌عنوان خروجی در نظر بگیرید.

پایان FPLA

عملکرد پرسپترون فازی بر روی داده‌های Anderson's Iris با پرسپترون قطعی مقایسه شده است. در این مقایسه، الگوریتم یادگیری پرسپترون فازی مرز تصمیم بهتری را تولید کرده است و در تکرارهای کمتری نسبت به نقطه مقابل قطعی آن به پایان می‌رسد. حتی زمانی که از داده‌هایی که به‌سختی تفکیک‌پذیر هستند استفاده می‌شود، پرسپترون فازی در تعداد تکرارهای کمتری نسبت به پرسپترون قطعی همگرا می‌شود. علاوه‌براین پرسپترون فازی می‌تواند از درجه هر برداری که در طبقه خودش است، استفاده مؤثری کند.

اگرچه در این بخش، توابع عضویت به الگوریتم پرسپترون شناسانده شد، اما این روش را می‌توان با هر الگوریتم کاهش گرادیان برای تولید مرزهای تصمیم‌گیری خطی (یا غیرخطی تحت تبدیل مناسب) مورد استفاده قرار داد. مثلاً می‌توان مشابه روش ذکرشده، یک شبکه فازی Functional-Link ایجاد نمود. به‌طور خاص، می‌توان مفهوم کاهش تأثیر داده‌های نامعین در تعیین مرز تصمیم را در مسائل طبقه‌بندی‌ای

که مرزهای آن توسط الگوریتم پس‌انتشار تعیین می‌شود، پیاده‌سازی کرد.

۵-۲-۲ ART و ARTMAP فازی

ART فازی^۱ تعمیمی از نظریه تشدید/انطباقی (ART)^۲ است که می‌تواند ورودی‌های فازی را پردازش کند. ART فازی یک الگوریتم یادگیری بدون نظارت است. از سوی دیگر، ARTMAP فازی تعمیمی از ART فازی است که براساس یادگیری بانظارت عمل می‌کند. ARTMAP فازی درک ریاضیاتی دقیقی از مفهوم ART ارائه می‌کند که از نظر محاسباتی قدرتمند است و از نظر کارایی بر بسیاری از روش‌های مبتنی بر دانش دیگر غلبه می‌کند. تشریح این دو سیستم در ادامه این بخش ارائه شده است.

سیستم ART فازی

سیستم ART فازی در سال ۱۹۹۱ توسط کارپنتر^۳ و همکاران معرفی شده است. این مدل با گنجاندن نظریه مجموعه‌های فازی در شبکه عصبی ART 1 - که فقط قادر به دسته‌بندی ورودی‌های باینری است - موجب می‌شود که علاوه بر ورودی باینری، بتواند ورودی‌های آنالوگ را نیز در یادگیری لحاظ کند. این مهم به کمک جایگزینی عملگر اشتراک (\cap) در ART 1 با عملگر min فازی (\wedge) حاصل شده است. عملگر min فازی در مواجهه با ورودی‌های باینری به عملگر اشتراک تقلیل می‌یابد.

هر سیستم ART شامل سه دسته گره است که F_0 ، F_1 و F_2 نام‌گذاری می‌شوند. F_0 بردار ورودی را شامل می‌شود. F_1 دریافت‌کننده ورودی از هر دو لایه F_0 و F_2 است. F_2 نیز بیان‌گر کد فعال^۴ یا دسته^۵ است. F_0 توسط $I = (I_1, \dots, I_M)$ ، F_1 توسط $x = (x_1, \dots, x_M)$ و F_2 توسط $y = (y_1, \dots, y_N)$ نمادگذاری می‌شوند که در آن $I_i \in [0, 1]$ است.

متناظر با هر گره j با $1, 2, \dots, N$ در F_2 ، یک بردار وزن قابل به‌روزرسانی به فرم

1. Fuzzy ART
2. Adaptive Resonance Theory (ART)
3. Carpenter, G. A.
4. Active code
5. Category

مقدار اولیه وزن‌ها به صورت زیر تعیین می‌شود:

$$w_{j1} = w_{j2} = \dots = w_{jM} = 1 \quad (20-5)$$

پس از آنکه یک دسته برای کدگذاری انتخاب شود، آن را پذیرفته شده^۲ می‌نامند. مقدار وزن w_{ji} در طول زمان افزایش نمی‌یابد بلکه می‌تواند کاهش یافته و به یک مرکز مشخص همگرا شود. بردار وزن ART فازی (یعنی w_j) هر دو بردار وزن پایین به بالا و بالا به پایین در ART1 را دربرمی‌گیرد. پویایی ART فازی توسط سه پارامتر محقق می‌شود. پارامترهای این سیستم شامل $\alpha > 0$ ، $\beta \in [0, 1]$ و $\rho \in [0, 1]$ است که به ترتیب پارامتر انتخاب^۳، نرخ یادگیری^۴ و پارامتر هوشیاری^۵ (آستانه اطمینان) نامیده می‌شوند.

برای هر ورودی I و دسته j (گره j از F_2) می‌توان تابع انتخاب را به صورت زیر تعریف کرد:

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|} \quad (21-5)$$

در رابطه فوق، \wedge عملگر AND فازی است که بر پایه رابطه $p_i \wedge q_i = \min(p_i, q_i)$ عمل می‌کند. نرم $| \cdot |$ نیز به صورت $|p| = \sum_{i=1}^M |p_i|$ تعریف می‌شود. انتخاب دسته توسط سیستم وقتی انجام می‌شود که بتوان حداکثر یک گره از F_2 را در زمان مورد نظر فعال کرد. دسته انتخاب شده با اندیس j معرفی می‌شود:

$$T_J = \max\{T_j : j = 1, \dots, N\} \quad (22-5)$$

در صورتی که بیش از یک T_j حداکثر مقدار را دارا باشند، دسته j با کوچکترین اندیس انتخاب می‌شود؛ بنابراین گره‌ها به ترتیب $j = 1, 2, 3, \dots$ پذیرفته شده می‌شوند. وقتی دسته J انتخاب شود، $y_J = 1$

-
1. Uncommitted
 2. Committed
 3. Choice parameter
 4. Learning rate
 5. Vigilance parameter

و $y_j = 0$ برای سایر j ها (یعنی $j \neq J$) است. در یک سیستم انتخاب، تابع فعالیت x برای F_1 از رابطه زیر پیروی می‌کند:

$$x = \begin{cases} I & \text{اگر گره } F_2 \text{ فعال نباشد} \\ I \wedge w_J & \text{اگر گره } J \text{ از } F_2 \text{ انتخاب شود} \end{cases} \quad (23-5)$$

اگر دسته انتخاب‌شده (گره برنده) در شاخص هوشیاری بر پایه رابطه زیر صدق کند، تشدید^۱ رخ می‌دهد.

$$\frac{|I \wedge w_J|}{|I|} \geq \rho \quad (24-5)$$

به عبارت دیگر اگر دسته J براساس رابطه ۵-۲۱ انتخاب شود، تشدید با صدق در رابطه زیر اتفاق می‌افتد و مرتب‌سازی کامل می‌شود:

$$|x| = |I \wedge w_J| \geq \rho |I| \quad (25-5)$$

اگر تشدید نداشته باشیم، یعنی $|x| = |I \wedge w_J| < \rho |I|$ رخ داده است که به آن تنظیم مجدد عدم مطابقت^۲ گفته می‌شود. به بیان دیگر، هنوز بهترین نمونه دارای مطابقت پیدا نشده است و سیستم باید دسته جدیدی را بیابد. در همین راستا، مقدار تابع انتخاب T_J برابر با صفر قرار داده می‌شود تا طی ادامه الگوریتم، از انتخاب مجدد آن جلوگیری شود. در ادامه، اندیس J جدید که دسته فعال را نشان می‌دهد دوباره به کمک روابط ۵-۲۱ و ۵-۲۲ انتخاب می‌شود. فرآیند جستجو تا زمانی که J انتخاب‌شده در رابطه ۵-۲۴ صدق کند، ادامه می‌یابد.

هنگامی که جستجو به پایان رسید، بردار وزن w_J براساس رابطه زیر به‌روز می‌شود.

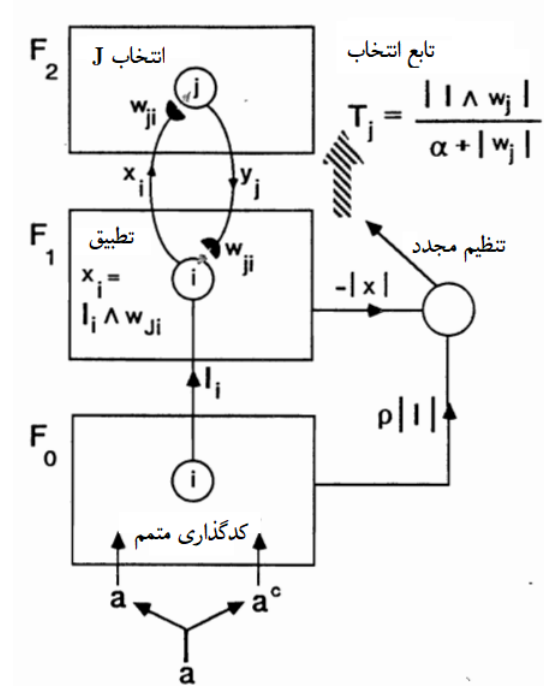
$$w_J^{(new)} = \beta \left(I \wedge w_J^{(old)} \right) + (1 - \beta) w_J^{(old)} \quad (26-5)$$

در صورتی که $\beta = 1$ باشد، یادگیری سریع رخ می‌دهد.

به‌منظور عملکرد مؤثر در مواجهه به داده‌های ورودی که نویز دارند، تا زمانی که J انتخاب نشده است از $\beta = 1$ استفاده کرده و زمانی که آن دسته انتخاب شد، $\beta < 1$ را در نظر می‌گیریم. بنابراین زمانی که دسته J برای اولین بار فعال شد، خواهیم داشت: $w_J^{(new)} = I$.

1. Resonance

2. Mismatch reset



شکل ۵-۱۱: مدل ART فازی

در برخی از سیستم‌های ART آنالوگ با مسأله ازدیاد دسته‌ها مواجه هستیم. این مسأله ناشی از کم‌اثر ساختن نُرم بردار وزن توسط مقادیر زیاد ورودی‌ها است. اگر ورودی‌ها نرمال شوند، از ازدیاد دسته‌ها در ART فازی جلوگیری می‌شود. به بیان دیگر، برای برخی $\gamma > 0$ و همه ورودی‌ها (I) داریم:

$$|I| = \gamma \quad (۲۷-۵)$$

نرمال کردن را می‌توان با پیش‌پردازش هر بردار ورودی a انجام داد؛ مثلاً می‌توان از رابطه زیر استفاده کرد.

$$I = \frac{a}{|a|} \quad (۲۸-۵)$$

یک قاعده دیگر برای نرمال‌سازی استفاده از کدگذاری مکمل^۱ است که دامنه اطلاعات را حفظ می‌کند. کدگذاری مکمل هر دو پاسخ ON و OFF به a را مطابق شکل ۵-۱۱ ارائه می‌کند. برای سهولت، فرض

1. Complement coding

می‌شود که a نشان‌دهنده پاسخ ON است و پاسخ OFF توسط مکمل a (یعنی $a_i^c = 1 - a_i$) بیان می‌شود. ورودی I به F_1 در قالب کدگذاری مکمل به صورت زیر تعریف می‌شود:

$$I = (a, a^c) = (a_1, \dots, a_M, a_1^c, \dots, a_M^c) \quad (29-5)$$

بنابراین خواهیم داشت:

$$|I| = |(a, a^c)| = \sum_{i=1}^M a_i + \left(M - \sum_{i=1}^M a_i \right) = M \quad (30-5)$$

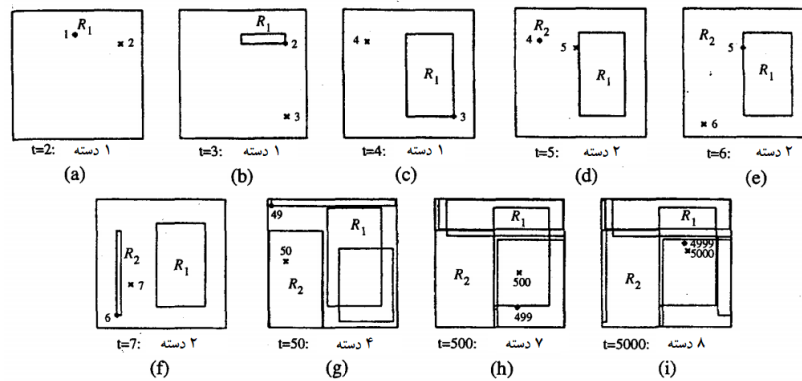
به این ترتیب پیش‌پردازش کدگذاری مکمل روی بردارهای ورودی باعث می‌شود که آنها به طور خودکار نرمال شوند. وقتی این کدگذاری انجام شود، شرایط اولیه وزن‌ها مطابق رابطه ۵-۲۰ درمی‌آید و هر دسته در حالت پذیرفته‌نشده است. از سوی دیگر، وزن‌های اولیه w_{ji} ممکن است بزرگتر از ۱ شوند. وزن‌های بزرگتر موجب جهت‌گیری سیستم در مقابل انتخاب گره‌های پذیرفته‌نشده می‌شود. به همین دلیل جستجوی عمیق‌تری در میان دسته‌های از پیش کدگذاری شده انجام می‌شود.

مثال ۵-۴. شبیه‌سازی کدگذاری مکمل در ART فازی

کارپنتر^۱ و همکاران (۱۹۹۱) در مثالی به شبیه‌سازی کدگذاری مکمل در ART فازی پرداختند. جزئیات این مثال در شکل ۵-۱۲ به نمایش درآمده است.

پارامترهای این سیستم، $\alpha \simeq 0$ ، $\beta = 1$ و $\rho = 0.4$ هستند و بردارهای ورودی $a^{(t)}$ به صورت تصادفی از یک مربع معین انتخاب شده‌اند. مستطیل‌های R_j طی یادگیری شبکه افزایش می‌یابند و دسته‌های جدید ایجاد می‌شوند تا زمانی که تمام مربع مذکور توسط آنها پوشانده شود. وقتی اولین دسته ایجاد شد، R_1 فقط شامل نقطه $a^{(1)}$ است. اگر $a^{(t)}$ در یک یا چند R_j موجود قرار گیرد، مستطیلی انتخاب می‌شود که کمترین مقدار $|R_j|$ را داشته باشد. در این حالت، تنظیم مجدد و تغییر وزن رخ نداده است. هر ورودی جدید که منجر به فعال شدن دسته j شود اما در مرزهای ایجاد شده قبلی آن قرار نگیرد، منجر به گسترش R_j می‌شود مگر اینکه چنین گسترشی موجب فراتر رفتن مقدار

1. Carpenter, G. A.



شکل ۵-۱۲: یک شبیه‌سازی از ART فازی

R_j از $1,2 = (1 - \rho) 2$ شود. در نهایت با در نظر گرفتن ورودی‌های بیشتر، کل مربع توسط هشت مستطیل R_j پوشانده شده است.

یادگیری در ART فازی وقتی تمام می‌شود که همه فضای ورودی پوشانده شود. با یادگیری سریع و داشتن مجموعه‌ای محدود از ورودی‌ها (با اندازه و ترکیب دلخواه)، یادگیری تنها پس از یک بار ارائه هر الگوی ورودی پایدار می‌شود. اتخاذ رویکرد پذیرش سریع - بازکدگذاری آهسته^۱ موجب می‌شود که یادگیری سریع با قاعده فراموشی ترکیب شده و اثرات نویز بر حافظه سیستم را کنترل کند. با استفاده از این رویکرد می‌توان رخداد‌های نادر را خیلی سریع یاد گرفت در حالی که یادگیری‌های قبلی را در مواجهه با تغییرات نامطمئن در ورودی‌ها به راحتی از دست ندهیم. برای جزئیات بیشتر در مورد الگوریتم یادگیری و توابع آنها می‌توانید به مقالات کارپنتر^۲ و همکاران (۱۹۹۱) و کارپنتر و گججا^۳ (۱۹۹۳) مراجعه کنید.

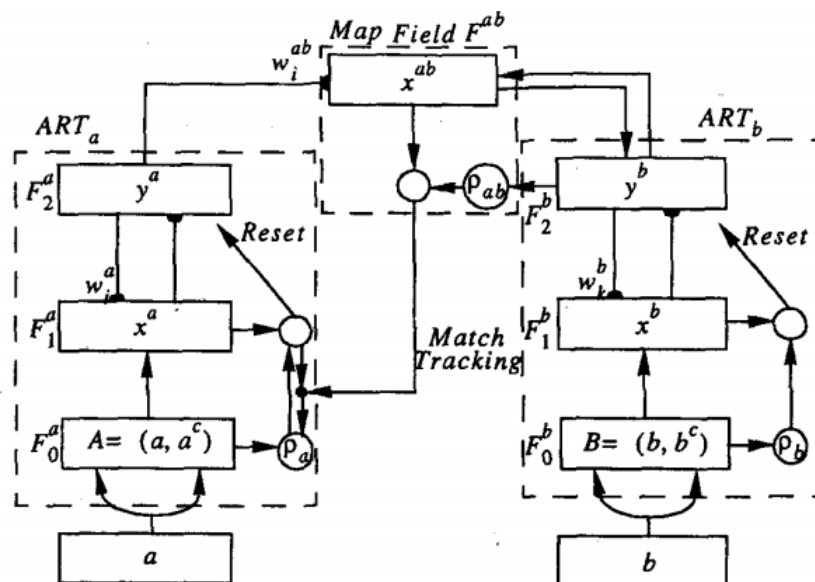
1. Fast-commit slow-recode

2. Carpenter, G. A.

3. Gjaja, M. N.

سیستم ARTMAP فازی

در این بخش، شبکه‌های ARTMAP فازی مورد بررسی قرار می‌گیرند که نمای کلی آن با کدگذاری مکمل^۱ براساس مدل کارپنتر^۲ و گروسبرگ^۳ (۱۹۹۲) در شکل ۵-۱۳ به نمایش درآمده است. هر سیستم ARTMAP از شبکه‌هایی از نوع ART با نام‌های ART_a و ART_b تشکیل می‌شود. طی یادگیری بانظارت در این شبکه‌ها، جریان‌های $\{a^{(p)}\}$ و $\{b^{(p)}\}$ از الگوهای ورودی به ترتیب توسط ART_a و ART_b دریافت می‌شوند. $\{b^{(p)}\}$ پیش‌بینی درست (خروجی مطلوب) براساس $\{a^{(p)}\}$ است.



شکل ۵-۱۳: معماری ARTMAP فازی

به‌طور معمول، ART_a و ART_b شبکه‌های ART1 از نوع یادگیری سریع با کدگذاری باینری بردارهای ورودی هستند. ART_a و ART_b توسط یک ناحیه نگاشت^۴ به هم متصل می‌شوند که به ART1 شباهت دارد. نام ARTMAP نیز از همین واقعیت آمده است که فرآیند تبدیل بردارهای

1. Complement coding
2. Carpenter, G. A.
3. Grossberg, S.
4. Map field

\mathbb{R}^2 به بردارهایی در \mathbb{R}^2 به یک نگاشت^۱ منتهی می‌شود که یادگیری آن براساس زوج نمونه‌های $\{a^{(p)}, b^{(p)}\}$ از بردارهای ورودی متوالی ($p = 1, 2, \dots$) انجام می‌شود. این ناحیه نگاشت، یادگیری نگاشت ارتباطی بین دسته‌های حاصل از ART_a و ART_b را کنترل می‌کند. این نگاشت مستقیماً نمونه‌های a و b را به هم مربوط نمی‌سازد بلکه خانواده‌های متراکم و نمادینی از نمونه‌های a و b را به هم ربط می‌دهد. ناحیه نگاشت، پیگیری تطابق پارامتر هوشیاری^۲ مربوط به ART_a را نیز کنترل می‌کند. عدم تطابق بین دسته فعال شده ART_a به وسیله ورودی a و دسته فعال شده ART_b به وسیله ورودی b موجب افزایش هوشیاری ART_a می‌شود. این افزایش، حداقل به میزان موردنیاز برای جستجو یا یادگیری دسته‌ای از ART_a است که پیش‌بینی آن مطابق دسته ART_b باشد. مورد اخیر یک ویژگی مهم در ARTMAP است که حداکثرسازی تعمیم‌پذیری و حداقل‌سازی خطای پیش‌بینی به صورت همزمان را هدف قرار می‌دهد. به عبارت دیگر، این سیستم می‌تواند یاد بگیرد که چگونه حداقل تعداد دسته‌های تشخیص داده شده به وسیله ART_a (گره‌های پنهان) که برای ارضای معیار دقت (حداقل خطای پیش‌بینی) موردنیاز است را ایجاد کند. لازم به ذکر است که مقادیر کم هوشیاری ART_a موجب ایجاد دسته‌های بزرگتر (تعداد دسته کمتر) شده که به تعمیم‌پذیری گسترده‌تر و کدگذاری فشرده‌تر و بیشتر منجر می‌شود.

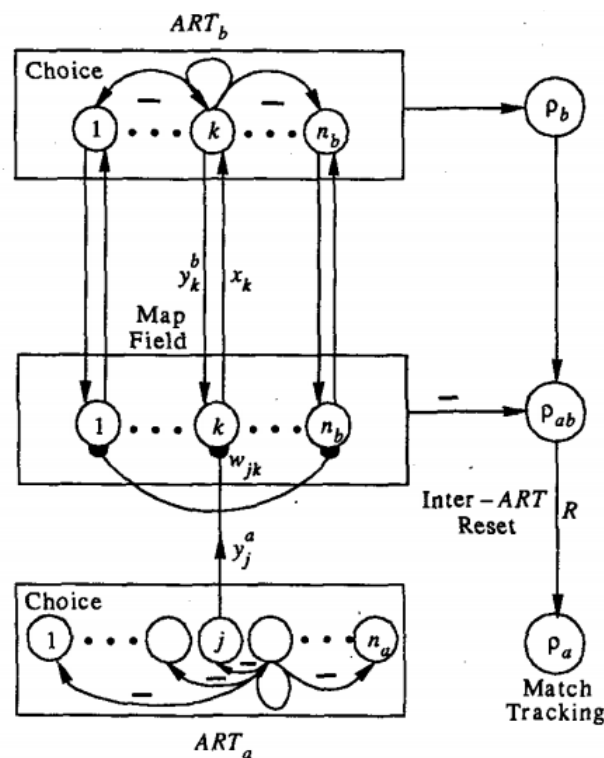
نظریه اینکه ARTMAP باینری از سیستم‌های ART1 برای دو بخش ART_a و ART_b استفاده می‌کند، در ARTMAP فازی از سیستم‌های ART فازی برای این دو بخش استفاده می‌شود. از آنجایی که ARTMAP یک مورد خاص از ARTMAP فازی است، از اینجا به بعد بر ARTMAP فازی متمرکز می‌شویم. پیش‌پردازش‌گر کدگذاری مکمل ART_a ، بردار m_a بعدی a را به بردار m_a بعدی^۲ می‌رساند. $A = (a, a^c)$ در لایه F_a^a از ART_a انتقال می‌دهد (شکل ۵-۱۳). A بردار ورودی به لایه F_a^a از ART_a است. به همین ترتیب ورودی به F_b^b نیز بردار m_b بعدی^۲ $B = (b, b^c)$ است. زمانی که پیش‌بینی حاصل از ART_a توسط ART_b تایید نشود، بازدارندگی از فعالیت ناحیه نگاشت موجب انجام فرآیند پیگیری تطابق^۳ خواهد شد. پیگیری تطابق، ضریب هوشیاری ART_a (ضریب ρ_a) را تا بالای

1. Map

2. Vigilance parameter

3. Match tracking

نسبت تطابق $|x^a|/|A|$ در F_1^a به F_0^a افزایش خواهد داد که در آن، بردار فعالیت (خروجی) F_1^a از ART_a را نشان می‌دهد (شکل ۵-۱۳). این روند، محرک جستجوی ART_a است که به فعال شدن دسته‌ای از ART_a که به درستی b را پیش‌بینی می‌کند یا یک گره مربوط به دسته‌ای از ART_a که از قبل پذیرفته شده، منجر می‌شود. پیگیری تطابق، ساختار دسته را شناسایی می‌کند؛ پس آن خطای پیش‌بینی طی ارائه دنباله‌های بعدی از ورودی‌ها تکرار نخواهد نشد. جزئیات معماری ناحیه نگاشت در شکل ۵-۱۴ نمایش داده شده است. ناحیه نگاشت به کمک مسیرهای یک‌به‌یک غیر تطابقی به F_1^b متصل می‌شود. هر گره F_1^a از طریق مسیرهای تطابقی به همه گره‌های ناحیه نگاشت متصل می‌شود. عدم تطابق بین دسته پیش‌بینی شده توسط a و دسته واقعی b موجب تنظیم مجدد F_1^a و افزایش هوشیاری (ρ_a) از طریق ناحیه نگاشت می‌شود.



شکل ۵-۱۴: اجزای اصلی ناحیه نگاشت

در ادامه، تشریح الگوریتم ARTMAP فازی (FARTMAP) طی گام‌های زیر ارائه می‌شود.

گام ۱ (فعال‌سازی ART_a و ART_b): ورودی‌های ART_a و ART_b در قالب کدگذاری مکمل هستند به طوری که برای ART_b و ART_a به ترتیب داریم: $I = B = (b, b^c)$ و $I = A = (a, a^c)$. هر یک از ART_b و ART_a به طور مستقل در قالب شبکه ART فازی فعالیت خود را انجام می‌دهند. متغیرهای ART_a و ART_b به کمک a و b طراحی می‌شوند. برای ART_a ، $x^a = (x_1^a, x_2^a, \dots, x_{m_a}^a)$ ، نمایش‌دهنده بردار خروجی F_1^a ، $y^a = (y_1^a, y_2^a, \dots, y_{n_a}^a)$ نمایش‌دهنده بردار خروجی F_2^a و $w_j^a = (w_{j,1}^a, w_{j,2}^a, \dots, w_{j,m_a}^a)$ نمایش‌دهنده بردار وزن j در ART_a است. برای ART_b ، $x^b = (x_1^b, x_2^b, \dots, x_{m_b}^b)$ ، نمایش‌دهنده بردار خروجی F_1^b ، $y^b = (y_1^b, y_2^b, \dots, y_{n_b}^b)$ نمایش‌دهنده بردار خروجی F_2^b و $w_k^b = (w_{k,1}^b, w_{k,2}^b, \dots, w_{k,m_b}^b)$ نمایش‌دهنده بردار وزن k در ART_b است. برای ناحیه نگاشت، $x^{ab} = (x_1^{ab}, x_2^{ab}, \dots, x_{n_b}^{ab})$ نمایش‌دهنده بردار خروجی F_{ab} و $w_j^{ab} = (w_{j,1}^{ab}, w_{j,2}^{ab}, \dots, w_{j,n_b}^{ab})$ نمایش‌دهنده بردار وزن j از F_2^a به F_{ab} است. در میان ارائه ورودی‌ها، مقادیر x^a ، y^a ، x^b و y^b برابر با \circ قرار داده می‌شود.

گام ۲ (فعال‌سازی ناحیه نگاشت): ناحیه نگاشت F^{ab} زمانی فعال می‌شود که یکی از دسته‌های ART_a و ART_b فعال باشد. اگر گره J از F_2^a انتخاب شود، وزن‌های آن (w_J^{ab}) موجب فعال شدن F^{ab} می‌شود. اگر گره K در F_2^b فعال شود، گره K در F^{ab} نیز به وسیله مسیر یک‌به‌یک بین F_2^b و F^{ab} فعال می‌شود. اگر هر دو ART_a و ART_b ، F^{ab} در صورتی فعال می‌شود که ART_a همان دسته ART_b را از طریق وزن‌های w_J^{ab} پیش‌بینی کند. به عبارت دیگر بردار خروجی F^{ab} (x^{ab}) از رابطه زیر پیروی می‌کند:

$$x^{ab} = \begin{cases} y^b \wedge w_J^{ab} & \text{اگر گره } J \text{ از } F_2^a \text{ فعال باشد و } F_2^b \text{ فعال نباشد} \\ w_J^{ab} & \text{اگر گره } J \text{ از } F_2^a \text{ فعال نباشد و } F_2^b \text{ فعال باشد} \\ y^b & \text{اگر } F_2^a \text{ فعال نباشد و } F_2^b \text{ فعال باشد} \\ \circ & \text{اگر } F_2^a \text{ فعال نباشد و } F_2^b \text{ فعال نباشد} \end{cases} \quad (۳۱-۵)$$

براساس رابطه ۵-۳۱، اگر پیش‌بینی w_j^{ab} به‌وسیله y^b تایید نشود (یعنی $y^b \wedge w_j^{ab} = 0$) آنگاه $x^{ab} = 0$ خواهد بود. این عدم تطابق موجب می‌شود که جستجوی ART_a برای یک دسته بهتر براساس فرآیند پیگیری تطابق آغاز گردد.

گام ۳ (پیگیری تطابق): در ارائه ورودی، پارامتر هوشیاری ρ_a برابر با مقدار هوشیاری پایه ($\bar{\rho}_a$) قرار داده می‌شود. پارامتر هوشیاری ناحیه نگاشت برابر با ρ_{ab} مفروض است. اگر

$$|x^{ab}| < \rho_{ab} |y^b| \quad (32-5)$$

باشد، آنگاه ρ_a تا اندکی بیشتر از $|A \wedge w_j^a| |A|^{-1}$ افزایش می‌یابد که A ورودی F_1^a در قالب کدگذاری مکمل است. سپس رابطه زیر را خواهیم داشت که در آن J مربوط به گره فعال F_1^a است:

$$|x^a| = |A \wedge w_j^a| < \rho_a |A| \quad (33-5)$$

زمانی که این مورد رخ می‌دهد، جستجوی ART_a به فعال‌سازی یک گره دیگر از F_1^a (J') با کمک روابط زیر منجر می‌شود.

$$|x^a| = |A \wedge w_{j'}^a| \geq \rho_a |A| \quad (34-5)$$

$$|x^{ab}| = |y^b \wedge w_{j'}^{ab}| \geq \rho_{ab} |y^b| \quad (35-5)$$

اگر چنین گره‌ای در F_1^a وجود نداشت (یعنی $\rho_a > 1$)، جستجوی ART_a به متوقف کردن F_1^a برای باقی‌مانده ارائه ورودی‌ها منجر می‌شود. در این شرایط، ارتباط (a, b) توسط ARTMAP فازی رد می‌شود (یادگیری انجام نمی‌شود) تا زمانی که مجدداً به ARTMAP فازی ارائه شود.

گام ۴ (یادگیری ناحیه نگاشت): وزن‌های w_{jk}^{ab} در مسیرهای $F^a \rightarrow F^{ab}$ ابتدا در رابطه زیر صدق می‌کنند:

$$w_{jk}^{ab}(0) = 1 \quad (36-5)$$

سپس در ادامه از قانون یادگیری Outstar پیروی می‌کنند:

$$w_{jk}^{ab}(t+1) = w_{jk}^{ab}(t) + y_j^a (x_k - w_{jk}^{ab}) \quad (37-5)$$

که بر این دلالت دارد که طی تشدید با دسته J از ART_a ، w_j^{ab} به بردار x^{ab} از ناحیه نگاشت میل می‌کند. طی یادگیری سریع وقتی J یاد می‌گیرد که دسته K از ART_b را پیش‌بینی کند، ارتباط ماندگاری شکل می‌گیرد. به عبارت دیگر، همیشه $w_{JK}^{ab} = 1$ است.

موردکاوی

در حوزه مطالعات منابع طبیعی، بسیاری از پژوهش‌ها و گزارش‌ها از نقشه‌برداری زمین‌شناسی به منظور ایجاد نقشه‌های پوشش/کاربری اراضی استفاده می‌کنند. در سال‌های اخیر، استفاده از فناوری‌های سنجش از دور (Remote sensing) و به‌کارگیری داده‌های ماهواره‌ای برای مطالعات زمین‌شناسی گسترش یافته است. علاوه بر افزایش دقت و کاهش هزینه، مزایای دیگری مانند دسترسی به مناطق صعب‌العبور نیز از جمله ویژگی‌های استفاده از این فناوری‌ها است.

به منظور طبقه‌بندی پدیده‌ها و عوارض زمین به کمک داده‌های ماهواره‌ای روش‌های مختلفی توسعه داده شده‌اند که آرت‌مپ فازی (Fuzzy ARTMAP) یکی از آنها است. آرت‌مپ فازی در پژوهش‌های متعددی برای تهیه نقشه‌های پوشش/کاربری اراضی مورد استفاده قرار گرفته است زیرا طبقه‌بندی فازی به کمک ARTMAP، توانایی بالایی در تفکیک دسته‌های مختلف پوشش اراضی به‌ویژه با پیکسل‌های مختلط دارد.

آرخی و فتحی‌زاد (۲۰۱۵) در پژوهشی به بررسی عملکرد روش آرت‌مپ فازی در تهیه نقشه

پوشش/کاربری اراضی سه منطقه از استان ایلام با استفاده از تصاویر ماهواره‌ای ETM+ پرداختند. به‌همین منظور، پس از تصحیحات هندسی روی داده‌های سنجنده ETM+ و بازدید میدانی، نمونه‌های آموزشی به‌طور تصادفی با استفاده از تصاویر گوگل‌ارث و تصاویر رنگی مرکب (RGB) انتخاب شدند. طبقه‌های موردانتظار شامل جنگل، مرتع متوسط، مرتع فقیر، کشاورزی، اراضی مسکونی و اراضی بایر بوده است که طی اجرای آرت‌مپ فازی فضا تلاش می‌شود که بهترین طبقه‌بندی روی داده‌های آموزشی انجام شود. در ادامه این پژوهش، نتایج عملکرد آرت‌مپ فازی با شبکه‌های عصبی تابع پایه شعاعی، کوهونن و پرسپترون مقایسه شده است. نتایج آن نشان می‌دهد که در هر سه منطقه، آرت‌مپ فازی دقت بیشتری نسبت به سایر الگوریتم‌های طبقه‌بندی داشته است.

Arekhi, S. & Fathizad, H. (2015). Evaluating the Efficiency of Four Artificial Neural Network Methods in Preparing Land Cover/Land Use Map Using ETM+ Data Case study: Doiraje, Mehran and Sarableh. *Geography and Development Iranian Journal*, 12(37), 133-146.

الگوریتم ARTMAP قطعی مشابه الگوریتم ARTMAP فازی است با این تفاوت که عملگر AND فازی (\wedge) در الگوریتم فوق با یک عملگر اشتراک قطعی (\cap) جایگزین می‌شود و ورودی‌های ART_a و ART_b می‌تواند در قالب کدگذاری مکمل باشد یا نباشد. برای دسترسی به جزئیات بیشتر در مورد ARTMAP فازی به مقاله کارپنتر^۱ و همکاران (۱۹۹۲) مراجعه کنید.

۳-۲-۵ شبکه خوشه‌بندی کوهونن فازی (FKCN)

نگاشت خودسازمان‌ده (SOM) کوهونن^۲ یا شبکه خوشه‌بندی کوهونن (KCN)^۳ یک شبکه عصبی شناخته‌شده در حوزه تحلیل خوشه‌ای است. KCN شبکه بدون نظارت است که طی یک روش تکرارشونده

1. Carpenter, G. A.

2. Kohonen's Self Organizing Map

3. Kohonen Clustering Network (KCN)

و دنباله‌دار، بهترین وزن‌ها را برای خوشه‌های سخت^۱ پیدا می‌کند. ساختار KCN از دو لایه تشکیل شده که یکی از آنها لایه ورودی و دیگری لایه خروجی (رقابتی) است. با فرض یک بردار ورودی، گره‌های لایه خروجی با هم رقابت می‌کنند تا آن گره‌ای که وزن آن کمترین فاصله تا ورودی را دارد به‌عنوان برنده انتخاب شود. گره برنده، وزن‌های خود و برخی از مجموعه‌های همسایه را به‌روز می‌کند. این فرآیند تا زمانی که بردار وزن پایدار شود، ادامه پیدا می‌کند. در این طرح یادگیری، باید یک نرخ یادگیری تعریف کرد که با گذشت زمان به‌منظور توقف (پایان‌دهی) اجباری کاهش یابد. علاوه‌براین، باید یک تابع همسایگی تعریف شود که آن هم با گذشت زمان کاهش می‌یابد. KCN‌ها با چند مشکل مواجه هستند. اول اینکه KCN‌ها رویه‌های هیورستیک دارند و چنین توقفی (پایان‌دهی) مبتنی بر بهینه‌سازی هیچ مدلی از فرآیند یا داده‌های آن نیست. دوم، بردارهای وزن نهایی معمولاً به دنباله ورودی‌ها وابسته است. سوم، شرایط اولیه متفاوت معمولاً نتایج مختلفی را ایجاد می‌کند. چهارم، برخی از پارامترهای KCN - مثل نرخ یادگیری، اندازه تابع به‌روزرسانی همسایگی و استراتژی تغییرات این دو پارامتر در خلال یادگیری - باید از یک مجموعه داده به دیگری تغییر کند تا نتایج مطلوب حاصل شود.

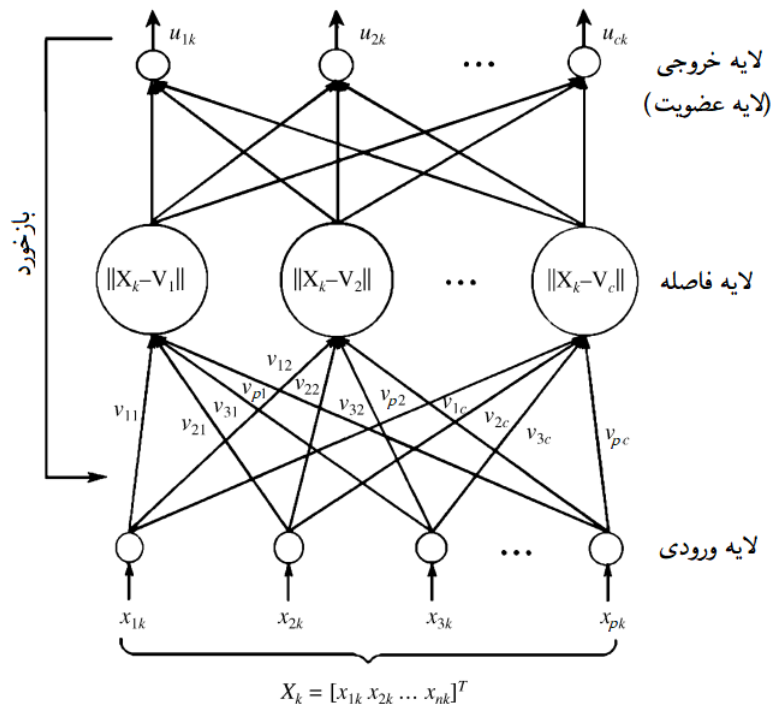
در این بخش، شبکه خوشه‌بندی کوهونن فازی (FKCN)^۲ تشریح می‌شود که مدل فازی FCM را با نرخ یادگیری و استراتژی‌های به‌روزرسانی KCN یکپارچه می‌کند. از آنجا که الگوریتم‌های FCM رویه‌های بهینه‌سازی هستند - درحالی‌که KCN نیستند - یکپارچه‌سازی FCM و KCN راهکاری برای حل برخی از مشکلات KCN است. این یکپارچه‌سازی، KCN را قادر می‌سازد که به‌جای خوشه‌بندی سخت، خروجی با مقادیر پیوسته را برای خوشه‌بندی فازی تولید کند. یکپارچه‌سازی FCM و KCN به کارایی محاسباتی نسبت به الگوریتم FCM دست می‌یابد که به‌دلیل استفاده از قاعده خطی به‌روزرسانی وزن در KCN است. یکی از نخستین تلاش‌ها برای این یکپارچه‌سازی توسط یزدک^۳ و همکاران (۱۹۹۲) توسعه داده شد که در ادامه تشریح می‌شود. در این تلاش‌ها عمدتاً بر پایه ترکیب ایده‌های مقادیر عضویت فازی برای نرخ یادگیری، توازی FCM و ساختار و قواعد به‌روزرسانی KCN است.

با فرض اینکه n بردار ورودی وجود داشته باشد که هرکدام دارای p بُعد باشد، وظیفه FKCN این

1. Hard clusters

2. Fuzzy Kohonen clustering network (FKCN)

3. Bezdek, J. C.



شکل ۵-۱۵: معماری شبکه خوشه‌بندی کوهونن فازی (FKCN)

است که با پیدا کردن c بردار، بتواند n بردار ورودی (الگوهای آموزشی) را به c خوشه دسته‌بندی کند. یک شبکه FKCN از سه لایه تشکیل می‌شود: لایه ورودی، لایه فاصله^۱ و لایه خروجی (لایه عضویت^۲) که به ترتیب دارای c و c و p گره هستند. ساختار این شبکه در شکل ۵-۱۵ به نمایش درآمده است. لایه ورودی، الگوهای آموزشی ورودی را دریافت کرده و آنها را به لایه فاصله انتقال می‌دهد. لایه فاصله، فواصل بین الگوهای ورودی و بردارهای خوشه‌ها را محاسبه کرده و مقادیر فاصله را به لایه خروجی می‌فرستد. لایه خروجی، درجه‌های عضویت محاسبه می‌کند تا تعلق هر الگوی ورودی به هر خوشه مشخص شود.

اتصال بین لایه ورودی و لایه فاصله توسط بردارهای وزن v_j انجام می‌شود ($j = 1, 2, \dots, c$)

1. Distance layer
2. Membership layer

که در آن، v_{ij} وزن اتصال بین گره ورودی i و گره فاصله j است. v_j براساس حداقل‌سازی تابع هدف زیر حاصل می‌شود.

$$\sum_k \sum_j (u_{kj})^m \|x_k - v_j\|^2 \quad (38-5)$$

که u_{ki} درجه عضویت تعلق x_k به خوشه j را نشان می‌دهد و m یک توان کنترلی است.

در خلال یادگیری، هر الگوی ورودی x_k به لایه ورودی ارائه می‌شود که $x_k = [x_{1k}, \dots, x_{pk}]$ و $1 \leq k \leq n$ است. هر گره j با $c = 1, 2, \dots, c$ در لایه فاصله، فاصله بین x_k و v_j را براساس رابطه زیر محاسبه می‌کند.

$$d_{kj} = \|x_k - v_j\|^2 = (x_k - v_j)^T (x_k - v_j) \quad (39-5)$$

سپس تمام وزن‌ها به کمک رابطه زیر به‌روز می‌شوند.

$$v_j(t) = v_j(t-1) + \left[\sum_{k=1}^N \alpha_{kj} (x_k - v_j(t-1)) \right] / \sum_{k=1}^N \alpha_{kj} \quad (40-5)$$

در رابطه فوق، α_{kj} نرخ یادگیری است که به‌صورت زیر محاسبه می‌شود.

$$\alpha_{kj} = (u_{kj})^m \quad (41-5)$$

$$m = m_0 - t \Delta m \quad (42-5)$$

$$\Delta m = \frac{m_0 - 1}{t_{\max}} \quad (43-5)$$

m_0 یک مقدار ثابت بزرگتر از ۱ است و t_{\max} به سقف تعداد دوره‌های آموزشی اشاره می‌کند. u_{kj} براساس رابطه زیر به دست می‌آید.

$$u_{kj} = \frac{1}{\sum_{i=1}^c \left(\frac{d_{kj}}{d_{ki}} \right)^{\frac{2}{m-1}}} \quad (44-5)$$

فرآیند آموزش تا زمانی ادامه پیدا می‌کند که یا مقدار وزن‌ها (v) در دو تکرار متوالی به قدر کافی نزدیک باشند یا تعداد تکرارهای آموزش به سقف تعیین‌شده (t_{\max}) برسد. الگوریتم FKCN را می‌توان در گام‌های زیر خلاصه کرد.

الگوریتم FKCN

- ۱: تعداد خوشه‌ها (c) را تعیین کنید و ϵ را برابر با یک مقدار مثبت کوچک قرار دهید.
- ۲: وزن‌ها (v_j) را به صورت تصادفی مقداردهی اولیه کرده و سقف تکرارهای آموزش (t_{\max}) و $m_0 > 1$ را تعیین کنید.
- ۳: برای $t = 1, 2, \dots, t_{\max}$ مراحل زیر را انجام دهید:
 - آ) همه نرخ‌های یادگیری $\alpha_{kj}^{(t)}$ را با در نظر گرفتن $c, j = 1, 2, \dots, n$ و $k = 1, 2, \dots, n$ براساس رابطه ۴۱-۵ محاسبه کنید.
 - ب) همه بردارهای وزن $v_j^{(t)}$ را با در نظر گرفتن $c, j = 1, 2, \dots, n$ براساس رابطه ۴۰-۵ محاسبه کنید.
 - پ) مقدار $E^{(t)}$ را براساس رابطه $E^{(t)} = \left\| v_j^{(t)} - v_j^{(t-1)} \right\|^2$ حاصل کرده و در صورتی که $E^{(t)} \leq \epsilon$ باشد، الگوریتم را متوقف کنید. در غیر این صورت، $t = t + 1$ شده و گام ۳ مجدداً تکرار شود.

۳-۵ شبکه‌های عصبی با آموزش فازی

در اغلب آموزش‌های شبکه‌های عصبی، فرض بر این است که همه داده‌های آموزشی، عددی هستند. باین حال در برخی از شرایط - نظیر یادگیری دانش خبرگان - امکان دارد فقط داده‌های آموزشی فازی کیفی وجود داشته باشد، مانند «اگر x_1 کوچک و x_2 بزرگ باشد، آنگاه y کوچک است». دستیابی به دانش در پژوهش‌های مبتنی بر سیستم‌های خبره اهمیت دارد. به‌طور کلی می‌توان گفت استخراج دانش از خبرگان و ساخت پایگاه دانش دشوار است. تحقیقات مختلف در زمینه شبکه‌های عصبی مصنوعی نشان می‌دهد که شبکه‌های عصبی می‌توانند دستیابی به دانش را تسهیل کرده و در عبور از تنگناهای آن مؤثر باشند. با استفاده از این شبکه‌ها می‌توان نگاشت‌هایی غیرخطی بین اطلاعات عددی ایجاد نمود. با این وجود، اطلاعات زبانی و مبهم در مسائل کاربردی پیش از به‌کارگیری در شبکه‌های عصبی باید در قالب عددی درآیند. در تبدیل و انتقال اطلاعات معمولاً اعوجاج و پیچیدگی فازی‌سازی رخ می‌دهد که به خروجی‌های نادقیق منجر می‌شود.

در ادامه، روش‌هایی برای ورود منطق فازی به فرآیند یادگیری شبکه‌های عصبی مورد بررسی قرار می‌گیرد. این روش‌ها شامل استفاده از سیگنال‌های آموزش فازی برای آموزش شبکه‌های عصبی با پارامترهای فازی یا قطعی است.

۵-۳-۱ شبکه‌های عصبی با ورودی آموزشی فازی

در این بخش، برخی از روش‌های یادگیری شبکه‌های عصبی در به‌کارگیری دانش خبرگان که توسط قواعد اگر-آنگاه فازی ارائه می‌شوند، بررسی خواهند شد. به‌طور خاص، شبکه پیش‌خور چندلایه با بردارهای ورودی از اعداد فازی در نظر گرفته می‌شود که توسط ایشیبوچی^۱ و همکاران (۱۹۹۳a) معرفی شده است. در بسیاری از موارد، این مدل در قالب یک تعمیم فازی از الگوریتم پس‌انتشار نیز دیده می‌شود. فرض می‌کنیم که S قاعده اگر-آنگاه فازی توسط خبرگان انسانی ارائه شده باشد:

$$(۴۵-۵) \quad \text{اگر } x_1 \text{ برابر } A_{p1} \text{ و } \dots \text{ و } x_n \text{ برابر } A_{pn} \text{ باشد، آنگاه } y \text{ برابر } T_p \text{ است.}$$

در قاعده فوق، $S, \dots, 2, 1, p$ است و A_{pi} و T_p اعداد فازی هستند. این قواعد فازی را می‌توان در قالب زوج‌های ورودی-خروجی فازی زیر دید:

$$(۴۶-۵) \quad (\mathbf{A}_p, T_p) \quad p = 1, 2, \dots, S$$

$\mathbf{A}_p = (A_{p1}, A_{p2}, \dots, A_{pn})$ یک بردار فازی است. مسأله این است که با استفاده از S زوج ورودی-خروجی فازی در رابطه ۴۶-۵ بتوان یک نگاشت غیرخطی فازی را تقریب زد. لازم به ذکر است که اگر مقدار عددی قطعی مثل $x_{p'i}$ وجود داشته باشد، می‌توان آن را به صورت فازی یگانه (تک مقدار)^۲ در نظر گرفت. به عبارت دیگر جفت‌های ورودی-خروجی قطعی $(x_{p'}, y_{p'})$ را می‌توان حالت خاصی از رابطه ۴۶-۵ محسوب کرد. از این جهت فرم کلی مجموعه داده‌های آموزشی که می‌تواند فازی یا قطعی یا ترکیبی از آنها باشد در رابطه ۴۶-۵ نمایش داده شده است. به این ترتیب می‌توان داده‌های زبانی و داده‌های عددی قطعی را در یک سیستم واحد پردازش اطلاعات یکپارچه نمود.

1. Ishibuchi, H.

2. Singleton

ابتدا باید یک ساختار شبکه عصبی را معرفی کرد که بتواند بردارهای ورودی فازی را دربرگیرد به طوری که نگاهی از یک بردار ورودی فازی به یک عدد (بردار) خروجی فازی ایجاد شود. به همین منظور باید برخی از معادلات حساب بازه‌ای^۱ که مورد استفاده واقع خواهند شد را مرور کنیم. بازه‌های $X = [x^L, x^U]$ و $Y = [y^L, y^U]$ مفروض‌اند. L و U به ترتیب حد پایین و حد بالا را نشان می‌دهند. بنابراین خواهیم داشت:

$$X + Y = [x^L, x^U] + [y^L, y^U] = [x^L + y^L, x^U + y^U] \quad (۴۷-۵)$$

و

$$k \cdot X = k \cdot [x^L, x^U] = \begin{cases} \begin{bmatrix} kx^L & kx^U \\ kx^U & kx^L \end{bmatrix} & k \geq 0 \\ \begin{bmatrix} kx^U & kx^L \\ kx^L & kx^U \end{bmatrix} & k < 0 \end{cases} \quad (۴۸-۵)$$

در روابط فوق، k یک عدد حقیقی است. تابع فعال‌سازی شبکه عصبی نیز می‌تواند به یک رابطه ورودی-خروجی بازه‌ای بسط یابد:

$$f(\text{Net}) = f([net^L, net^U]) = [f(net^L), f(net^U)] \quad (۴۹-۵)$$

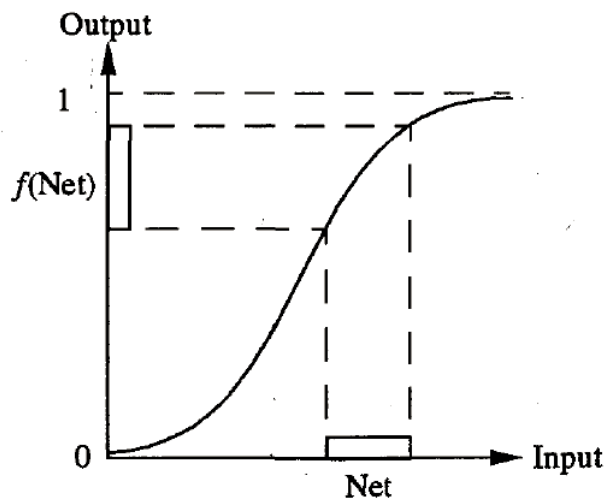
$\text{Net} = [net^L, net^U]$ یک ورودی بازه‌ای و $f(\cdot)$ یک تابع سیگموئید است. تابع فعال‌سازی بازه‌ای که توسط رابطه ۴۹-۵ تعریف شده است، در شکل ۵-۱۶ نمایان است.

توضیحات فوق مربوط به عملیات بر روی بازه‌ها است. از سوی دیگر عملیات بر روی اعداد فازی را می‌توان با استفاده از مفهوم مجموعه‌های برش به صورت زیر بیان کرد:

$$[A + B]_h = [A]_h + [B]_h \quad (۵۰-۵)$$

$$[kA]_h = k[A]_h \quad (۵۱-۵)$$

$$[f(A)]_h = f([A]_h) \quad (۵۲-۵)$$



شکل ۵-۱۶: تابع فعال‌سازی بازه‌ای هر واحد از یک شبکه عصبی (Net) و $f(\text{Net})$ به ترتیب ورودی بازه‌ای و خروجی بازه‌ای هستند)

A و B اعداد فازی هستند که به ترتیب با $\mu_A(x)$ و $\mu_B(x)$ تعریف می‌شوند. $[.]_h$ مجموعه‌های برش h از یک عدد فازی است. از آنجایی که مجموعه‌های برش h از اعداد فازی بازه‌ای هستند، عملیات بر روی مجموعه‌های برش معادل با عملیات بر روی بازه‌ها است.

برای تشریح فرآیند شبکه‌های عصبی که می‌توانند بردارهای ورودی فازی را دربرگیرند، از مجموعه برش h (روابط ۵-۵۰ تا ۵-۵۲) استفاده خواهیم نمود. از آنجایی که مجموعه‌های برش از اعداد فازی بازه‌ای هستند، محاسبات خروجی عدد فازی به حساب بازه‌ای نیاز دارد. در ادامه رابطه ورودی-خروجی را از شبکه عصبی برای بردار ورودی بازه‌ای مشتق خواهیم نمود. یک شبکه عصبی پیش‌خور چندلایه را با n گره ورودی، m گره پنهان و یک گره خروجی در نظر بگیرید. اگر بردار بازه‌ای $x_p = (X_{p1}, X_{p2}, \dots, X_{pn})$ به لایه ورودی شبکه عصبی ارائه شود، رابطه ورودی-خروجی هر واحد به صورت زیر محاسبه می‌شود:

گره‌های ورودی:

$$Y_{pi} = [y_{pi}^L, y_{pi}^U] = X_{pi} = [x_{pi}^L, x_{pi}^U] \quad i = 1, 2, \dots, n \quad (53-5)$$

گره‌های پنهان:

$$Y_{pj} = [y_{pj}^L, y_{pj}^U] = [f(\text{net}_{pj}^L), f(\text{net}_{pj}^U)] \quad j = 1, 2, \dots, m \quad (54-5)$$

$$\text{net}_{pj}^L = \sum_{\substack{i=1 \\ w_{ji} \geq 0}}^n w_{ji} y_{pi}^L + \sum_{\substack{i=1 \\ w_{ji} < 0}}^n w_{ji} y_{pi}^U + \theta_j \quad (55-5)$$

$$\text{net}_{pj}^U = \sum_{\substack{i=1 \\ w_{ji} \geq 0}}^n w_{ji} y_{pi}^U + \sum_{\substack{i=1 \\ w_{ji} < 0}}^n w_{ji} y_{pi}^L + \theta_j \quad (56-5)$$

گره خروجی:

$$Y_p = [y_p^L, y_p^U] = [f(\text{net}_p^L), f(\text{net}_p^U)] \quad (57-5)$$

$$\text{net}_p^L = \sum_{\substack{j=1 \\ w_j \geq 0}}^m w_j y_{pj}^L + \sum_{\substack{j=1 \\ w_j < 0}}^m w_j y_{pj}^U + \theta \quad (58-5)$$

$$\text{net}_p^U = \sum_{\substack{j=1 \\ w_j \geq 0}}^m w_j y_{pj}^U + \sum_{\substack{j=1 \\ w_j < 0}}^m w_j y_{pj}^L + \theta \quad (59-5)$$

وزن‌های w_{ji} و w_j و بایاس‌های θ_j و θ پارامترهای حقیقی هستند و خروجی‌های Y_{pi} ، Y_{pj} و Y_p بازه‌ای هستند.

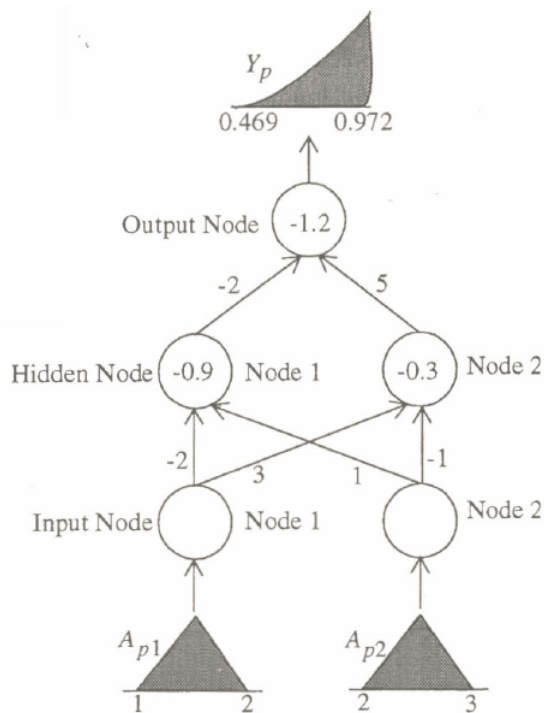
مثال ۵-۵. مثالی از روابط ورودی-خروجی فازی

شبکه عصبی نمایش داده شده در شکل ۵-۱۷ را در نظر بگیرید که در آن، وزن‌ها و بایاس‌ها به ترتیب در کنار اتصال‌ها و درون گره‌ها نشان داده شده است. بردار ورودی فازی در این شکل $\mathbf{A}_p = (A_{p1}, A_{p2}) = ((1/5, 0/5)_L, (2/5, 0/5)_L)$ است.

$A = (a, b)_L$ نشان‌گر یک عدد فازی مثلثی متقارن با مرکز a و پهنای b است که با تابع عضویت زیر تعریف می‌شود.

$$\mu_A(x) = \max \left\{ 1 - \frac{(x - a)}{b}, 0 \right\} \quad (۵-۶)$$

عدد فازی خروجی Y_p در شکل ۵-۱۷ نمایش داده شده است. Y_p با استفاده از روابط $h = ۰/۰۱, ۰/۰۲, \dots, ۱/۰۰$ مجموعه برش متناظر با ۱۰۰ براساس $۵-۵۹$ تا $۵-۵۳$ حاصل شده است.



شکل ۵-۱۷: مثالی از روابط ورودی-خروجی شبکه عصبی

با استفاده از خروجی فازی Y_p و خروجی هدف فازی T_p متناظر می‌توان تابع هزینه را برای

مجموعه‌های برش h از Y_p و T_p به صورت زیر تعریف نمود:

$$E_{ph} = \frac{([T_p]_h^L - [Y_p]_h^L)^2}{2} + \frac{([T_p]_h^U - [Y_p]_h^U)^2}{2} \quad (61-5)$$

در رابطه فوق، $[Y_p]_h = [Y_p]_h^L, [Y_p]_h^U$ و $[T_p]_h = [T_p]_h^L, [T_p]_h^U$ است. این تابع هزینه در قالب مجموع مربعات خطاها برای حدهای پایین و برای حدهای بالا تعریف می‌شود. لازم به ذکر است در شرایطی که بردار ورودی غیرفازی و خروجی هدف غیرفازی وجود داشته باشد، تابع هزینه فوق به مربع خطای به کارگرفته شده در الگوریتم پس‌انتشار ساده می‌شود.

یادگیری شبکه‌های عصبی شامل حداقل‌سازی تابع هزینه در رابطه ۵-۶۱ است. همانند الگوریتم

پس‌انتشار، وزن‌های w_j و w_{ji} مطابق قاعده زیر به‌روزرسانی می‌شود.

$$\Delta w_j(t+1) = -\eta \frac{\partial E_{ph}}{\partial w_j} + \alpha \Delta w_j(t) \quad (62-5)$$

$$\Delta w_{ji}(t+1) = -\eta \frac{\partial E_{ph}}{\partial w_{ji}} + \alpha \Delta w_{ji}(t) \quad (63-5)$$

براساس مدل ایشیبوچی و همکاران^۱ (۱۹۹۳a) $\partial E_{ph}/\partial w_j$ و $\partial E_{ph}/\partial w_{ji}$ را می‌توان با استفاده از قاعده زنجیره‌ای به صورت زیر محاسبه کرد.

برای $\partial E_{ph}/\partial w_j$:

۱- اگر $w_j \geq 0$ باشد، آنگاه:

1. Ishibuchi, H.

$$\begin{aligned}
\frac{\partial E_{ph}}{\partial w_j} &= \frac{\partial}{\partial w_j} \left(\frac{(t_p^L - y_p^L)^2}{2} \right) + \frac{\partial}{\partial w_j} \left(\frac{(t_p^U - y_p^U)^2}{2} \right) \quad (64-5) \\
&= \frac{\partial}{\partial y_p^L} \left(\frac{(t_p^L - y_p^L)^2}{2} \right) \frac{\partial y_p^L}{\partial \text{net}_p^L} \frac{\partial \text{net}_p^L}{\partial w_j} + \\
&\quad \frac{\partial}{\partial y_p^U} \left(\frac{(t_p^U - y_p^U)^2}{2} \right) \frac{\partial y_p^U}{\partial \text{net}_p^U} \frac{\partial \text{net}_p^U}{\partial w_j} \\
&= -(t_p^L - y_p^L) y_p^L (1 - y_p^L) y_{pj}^L - (t_p^U - y_p^U) y_p^U (1 - y_p^U) y_{pj}^U \\
&\triangleq -\delta_p^L y_{pj}^L - \delta_p^U y_{pj}^U
\end{aligned}$$

۲- اگر $w_j < 0$ باشد، آنگاه:

$$\begin{aligned}
\frac{\partial E_{ph}}{\partial w_j} &= -(t_p^L - y_p^L) y_p^L (1 - y_p^L) y_{pj}^U - (t_p^U - y_p^U) y_p^U (1 - y_p^U) y_{pj}^L \\
&\triangleq -\delta_p^L y_{pj}^U - \delta_p^U y_{pj}^L \quad (65-5)
\end{aligned}$$

برای $\partial E_{ph} / \partial w_j$:

۱- اگر $w_j \geq 0$ و $w_{ji} \geq 0$ باشد، آنگاه:

$$\begin{aligned}
\frac{\partial E_{ph}}{\partial w_{ji}} &= \frac{\partial}{\partial w_{ji}} \left(\frac{(t_p^L - y_p^L)^2}{2} \right) + \frac{\partial}{\partial w_{ji}} \left(\frac{(t_p^U - y_p^U)^2}{2} \right) \quad (66-5) \\
&= \frac{\partial}{\partial y_p^L} \left(\frac{(t_p^L - y_p^L)^2}{2} \right) \frac{\partial y_p^L}{\partial \text{net}_p^L} \frac{\partial \text{net}_p^L}{\partial y_{pj}^L} \frac{\partial y_{pj}^L}{\partial \text{net}_{pj}^L} \frac{\partial \text{net}_{pj}^L}{\partial w_{ji}} + \\
&\quad \frac{\partial}{\partial y_p^U} \left(\frac{(t_p^U - y_p^U)^2}{2} \right) \frac{\partial y_p^U}{\partial \text{net}_p^U} \frac{\partial \text{net}_p^U}{\partial y_{pj}^U} \frac{\partial y_{pj}^U}{\partial \text{net}_{pj}^U} \frac{\partial \text{net}_{pj}^U}{\partial w_{ji}} \\
&= -\delta_p^L w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^L - \delta_p^U w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^U
\end{aligned}$$

۲- اگر $w_j \geq 0$ و $w_{ji} < 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^U - \delta_p^U w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^L \quad (67-5)$$

۳- اگر $w_j < 0$ و $w_{ji} \geq 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^U - \delta_p^U w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^L \quad (68-5)$$

۴- اگر $w_j < 0$ و $w_{ji} < 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^L - \delta_p^U w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^U \quad (69-5)$$

در توضیحات فوق، $t_p^U \triangleq [T_p]_h^U$ ، $t_p^L \triangleq [T_p]_h^L$ ، $y_p^U \triangleq [Y_p]_h^U$ و $y_p^L \triangleq [Y_p]_h^L$ است. همچنین، δ_p^U و δ_p^L در قالب $(t_p^U - y_p^U) y_p^U (1 - y_p^U)$ و $(t_p^L - y_p^L) y_p^L (1 - y_p^L)$ تعریف می‌شود. علاوه‌براین با استفاده از تابع هزینه زیر می‌توان شبکه را با مقادیر مختلفی از h آموزش داد.

$$E_p = \sum_h h \cdot E_{ph} \quad (70-5)$$

$$= \sum_h h \left[\frac{1}{2} \left([T_p]_h^L - [Y_p]_h^L \right)^2 + \left(\frac{1}{2} [T_p]_h^U - [Y_p]_h^U \right)^2 \right] \quad (71-5)$$

مثال زیر، روش فوق را شفاف می‌سازد.

مثال ۵-۶.

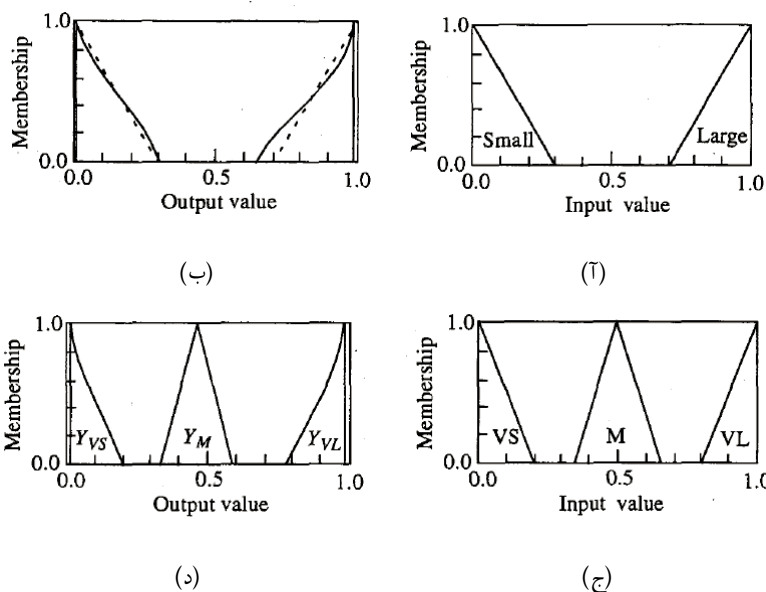
فضای ورودی و فضای خروجی را بازه واحد $[0, 1]$ قرار دهید. دو قاعده اگر-آنگاه فازی زیر مفروض است.

اگر x کوچک باشد، آنگاه y کوچک است.

اگر x بزرگ باشد، آنگاه y بزرگ است.

اعداد فازی «کوچک (S)» و «بزرگ (L)» در شکل ۵-۱۸ (آ) نمایش داده شده است. با استفاده از این دو قاعده، دو زوج ورودی-خروجی فازی (S, S) و (L, L) حاصل می‌شوند که به‌منظور آموزش یک شبکه عصبی با پنج گره پنهان مورد استفاده

قرار می‌گیرند. شکل ۵-۱۸ (ب) خروجی‌های فازی این شبکه آموزش‌دیده را نشان می‌دهد. این خروجی‌ها، نتایج شبیه‌سازی بعد از ۱۰,۰۰۰ تکرار با $\eta = 0.5$ و $\alpha = 0.9$ هستند که با استفاده از تابع هزینه معرفی شده در رابطه ۵-۷۱ با $h = 0.2, 0.4, 0.6, 0.8, 1$ حاصل شده‌اند.



شکل ۵-۱۸: (آ) توابع عضویت مقادیر زبانی «کوچک» و «بزرگ» (ب) خروجی‌های فازی حاصل از شبکه عصبی آموزش‌دیده در قیاس با خروجی‌های هدف فازی (ج) توابع عضویت مقادیر زبانی «خیلی کوچک»، «خیلی بزرگ» و «متوسط» (د) خروجی‌های فازی حاصل از شبکه عصبی آموزش‌دیده متناظر با بردارهای ورودی فازی

به منظور آزمون قابلیت تعمیم‌پذیری این شبکه عصبی آموزش‌دیده، سه عدد فازی جدید «خیلی کوچک» (VS)، «خیلی بزرگ» (VL) و «متوسط» (M) (شکل ۵-۱۸ (ج)) به عنوان ورودی به شبکه ارائه می‌شوند. خروجی‌های فازی متناظر آنها به ترتیب Y_{VS} ، Y_M و Y_{VL} هستند که درون‌یابی‌های خوبی را برای الگوهای آزمون بروز داده‌اند. این خروجی‌ها در شکل ۵-۱۸ (د) نمایش داده شده‌اند.

۵-۳-۲ شبکه‌های عصبی با پارامترهای فازی

در بخش قبل، شبکه‌های عصبی با وزن‌ها و بایاس‌های قطعی (عددی) را مورد بررسی قرار دادیم که می‌توانند داده‌های ورودی فازی را پردازش کنند. اکنون شبکه‌های عصبی پیش‌خور چندلایه‌ای را در نظر می‌گیریم که وزن‌ها و بایاس‌های آن اعداد فازی هستند. چنین شبکه‌های عصبی می‌توانند نگاهی از یک بردار ورودی شامل اعداد حقیقی را به یک خروجی فازی پدید آورند. به‌طور طبیعی، چنین شبکه‌هایی را می‌توان برای نگاشت بردار ورودی فازی به خروجی فازی نیز به‌کار گرفت. در اینجا، فقط خروجی‌های هدف فازی را در نظر خواهیم گرفت چراکه خروجی هدف قطعی را می‌توان حالت خاصی از خروجی هدف فازی در نظر گرفت. در ادامه، دو مدل (با تابع هزینه قطعی و با تابع هزینه فازی) را بررسی خواهیم کرد.

تابع هزینه قطعی. ابتدا براساس مدل پیشنهادی ایشیبوچی^۱ و همکاران (۱۹۹۳b) مدل عصبی فازی با وزن‌های فازی و بایاس‌های فازی را مورد بررسی قرار می‌دهیم. ساختار پیشنهادی در شکل ۵-۱۹ نشان داده شده است که در آن، برای وزن‌های فازی و بایاس‌های فازی از اعداد فازی مثلثی متقارن استفاده شده است:

$$W_j = (w_j^L, w_j^C, w_j^U), \quad W_{ji} = (w_{ji}^L, w_{ji}^C, w_{ji}^U), \quad (72-5)$$

$$\Theta = (\theta^L, \theta^C, \theta^U), \quad \Theta_j = (\theta_j^L, \theta_j^C, \theta_j^U), \quad (73-5)$$

a^L ، a^C و a^U به ترتیب حد پایین، مرکز و حد بالای عدد فازی مثلثی $A = (a^L, a^C, a^U)$ هستند. از آنجایی که اعداد فازی مثلثی متقارن مورد استفاده قرار می‌گیرند، روابط زیر منعقد خواهند شد:

$$W_j^C = \frac{w_j^L + w_j^U}{2}, \quad W_{ji}^C = \frac{w_{ji}^L + w_{ji}^U}{2}, \quad (74-5)$$

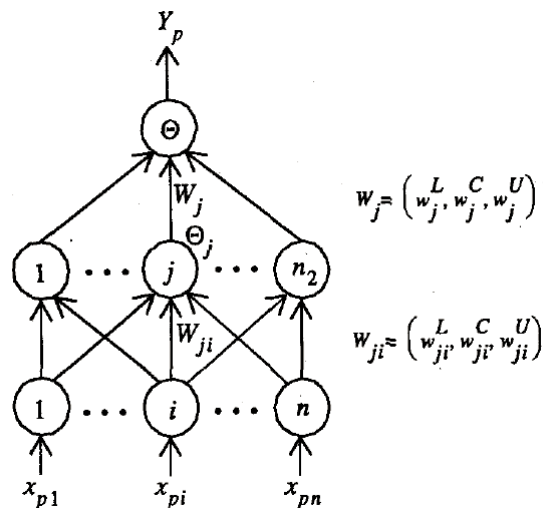
$$\theta^C = \frac{\theta^L + \theta^U}{2}, \quad \theta_j^C = \frac{\theta_j^L + \theta_j^U}{2}, \quad (75-5)$$

مجموعه‌های تحت برش h از وزن‌های فازی (و بایاس‌ها) را می‌توان به صورت زیر محاسبه کرد:

$$[W_j]_h^L = w_j^L \left(1 - \frac{h}{2}\right) + \frac{w_j^U h}{2}, \quad [W_j]_h^U = \frac{w_j^L h}{2} + w_j^U \left(1 - \frac{h}{2}\right) \quad (76-5)$$

1. Ishibuchi, H.

$$[W_{ji}]_h^L = w_{ji}^L \left(1 - \frac{h}{2}\right) + \frac{w_{ji}^U h}{2}, \quad [W_{ji}]_h^U = \frac{w_{ji}^L h}{2} + w_{ji}^U \left(1 - \frac{h}{2}\right) \quad (77-5)$$



شکل ۵-۱۹: ساختار شبکه‌های عصبی با وزن‌های فازی و بایاس‌های فازی

زمانی که بردار ورودی حقیقی n -بعدی $x_p = (x_{p1}, x_{p2}, \dots, x_{pn})^T$ به گره‌های ورودی

شبکه عصبی شکل ۵-۱۹ وارد شود، روابط ورودی-خروجی هر گره را می‌توان به فرم یک مجموعه برش

h مطابق زیر نوشت:

گره‌های ورودی:

$$y_{pi} = x_{pi} \quad i = 1, 2, \dots, n \quad (78-5)$$

گره‌های پنهان:

$$[Y_{pj}]_h = [f(\text{Net}_{pj})]_h = f([[\text{Net}_{pj}]_h]) \quad j = 1, 2, \dots, n_r \quad (79-5)$$

$$[\text{Net}_{pj}]_h = \sum_{i=1}^n [W_{ji}]_h y_{pi} + [\Theta_j]_h \quad j = 1, 2, \dots, n_r \quad (80-5)$$

گره‌های خروجی:

$$[Y_p]_h = [f(\text{Net}_p)]_h = f([\text{Net}_p]_h) \quad (۸۱-۵)$$

$$[\text{Net}_p]_h = \sum_{i=1}^{n_r} [W_j]_h [Y_{pj}]_h + [\Theta]_h \quad (۸۲-۵)$$

در ادامه، الگوریتم یادگیری شبکه فوق را بررسی می‌کنیم. تعداد S زوج ورودی قطعی - خروجی فازی (x_p, T_p) با $p = 1, 2, \dots, S$ مفروض است. فرض می‌شود که خروجی هدف T_p یک عدد فازی مثلثی متقارن یا نامتقارن به صورت $T_p = (t_p^L, t_p^C, t_p^U)$ است. تابع هزینه E_p و E_{ph} دقیقاً مشابه روابط ۵-۶۱ تا ۵-۷۱ تعریف می‌شود. بنابراین، قواعد یادگیری وزن‌های فازی (و بایاس‌های فازی) W_j و W_{ji} را می‌توان به صورت زیر نوشت:

$$\Delta w_j^L(t+1) = -\eta h \left(\frac{\partial E_{ph}}{\partial w_j^L} \right) + \alpha \Delta w_j^L(t) \quad (۸۳-۵)$$

$$\Delta w_j^U(t+1) = -\eta h \left(\frac{\partial E_{ph}}{\partial w_j^U} \right) + \alpha \Delta w_j^U(t) \quad (۸۴-۵)$$

$$\Delta w_{ji}^L(t+1) = -\eta h \left(\frac{\partial E_{ph}}{\partial w_{ji}^L} \right) + \alpha \Delta w_{ji}^L(t) \quad (۸۵-۵)$$

$$\Delta w_{ji}^U(t+1) = -\eta h \left(\frac{\partial E_{ph}}{\partial w_{ji}^U} \right) + \alpha \Delta w_{ji}^U(t) \quad (۸۶-۵)$$

مشتق‌های $\partial E_{ph} / \partial w_j^L$ ، $\partial E_{ph} / \partial w_j^U$ ، $\partial E_{ph} / \partial w_{ji}^L$ و $\partial E_{ph} / \partial w_{ji}^U$ از تابع هزینه رابطه ۵-۶۱ با استفاده از روابط ۵-۷۶ تا ۵-۸۲ حاصل می‌شود. برای مثال، $\partial E_{ph} / \partial w_j^L$ به صورت زیر درمی‌آید:

$$۱- اگر $[W_j]_h^L \leq [W_j]_h^U$ باشد، آنگاه:$$

$$\begin{aligned}
\frac{\partial E_{ph}}{\partial w_j^L} &= \frac{\partial E_{ph}^L}{\partial w_j^L} + \partial E_{ph}^U \partial w_j^L & (۸۷-۵) \\
&= \frac{\partial}{\partial [Y_p]_h^L} \left(\frac{([T_p]_h^L - [Y_p]_h^L)^2}{2} \right) \frac{\partial [Y_p]_h^L}{\partial [\text{Net}_p]_h^L} \frac{\partial [\text{Net}_p]_h^L}{\partial [W_j]_h^L} \frac{\partial [W_j]_h^L}{\partial w_j^L} + \\
&\quad \frac{\partial}{\partial [Y_p]_h^U} \left(\frac{([T_p]_h^U - [Y_p]_h^U)^2}{2} \right) \frac{\partial [Y_p]_h^U}{\partial [\text{Net}_p]_h^U} \frac{\partial [\text{Net}_p]_h^U}{\partial [W_j]_h^U} \frac{\partial [W_j]_h^U}{\partial w_j^L} \\
&= - \left([T_p]_h^L - [Y_p]_h^L \right) [Y_p]_h^L \left(1 - [Y_p]_h^L \right) [Y_{pj}]_h^L \left(1 - \frac{h}{2} \right) - \\
&\quad \left([T_p]_h^U - [Y_p]_h^U \right) [Y_p]_h^U \frac{\left(1 - [Y_p]_h^U \right) [Y_{pj}]_h^U h}{2} \\
&\triangleq -\delta_{ph}^L [Y_{pj}]_h^L \left(1 - \frac{h}{2} \right) - \frac{\delta_{ph}^U [Y_{pj}]_h^U h}{2}
\end{aligned}$$

که

$$\delta_{ph}^L = \left([T_p]_h^L - [Y_p]_h^L \right) [Y_p]_h^L \left(1 - [Y_p]_h^L \right)$$

و

$$\delta_{ph}^U = \left([T_p]_h^U - [Y_p]_h^U \right) [Y_p]_h^U \left(1 - [Y_p]_h^U \right)$$

هستند.

۲- اگر $0 < [W_j]_h^L \leq [W_j]_h^U$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_j^L} = -\delta_{ph}^L [Y_{pj}]_h^U \left(1 - \frac{h}{2} \right) - \frac{\delta_{ph}^U [Y_{pj}]_h^L h}{2} \quad (۸۸-۵)$$

۳- اگر $[W_j]_h^L < 0 \leq [W_j]_h^U$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_j^L} = -\delta_{ph}^L [Y_{pj}]_h^U \left(1 - \frac{h}{2} \right) - \frac{\delta_{ph}^U [Y_{pj}]_h^U h}{2} \quad (۸۹-۵)$$

بعد از تنظیم وزن‌های فازی با استفاده از روابط ۵-۸۳ تا ۵-۸۶، امکان دارد که حدود پایین از حدود بالا تجاوز کنند. برای مقابله با این شرایط نامطلوب، وزن‌های فازی جدید پس از تنظیم، به صورت زیر تعریف می‌شود:

$$w_j^L = \min \{w_j^L(t+1), w_j^U(t+1)\} \quad (90-5)$$

$$w_{ji}^L = \min \{w_{ji}^L(t+1), w_{ji}^U(t+1)\}$$

$$w_j^U = \max \{w_j^L(t+1), w_j^U(t+1)\} \quad (91-5)$$

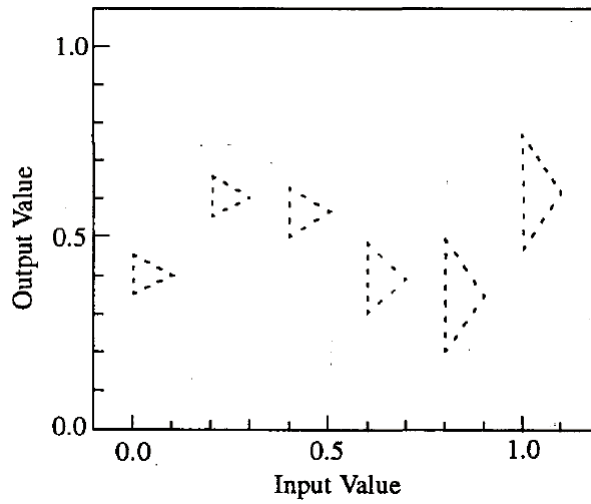
$$w_{ji}^U = \max \{w_{ji}^L(t+1), w_{ji}^U(t+1)\}$$

مرکزهای w_{ji}^c و w_j^c توسط ۵-۷۴ محاسبه می‌شوند. به‌روزرسانی بایاس‌های فازی نیز به شیوه‌ای مشابه وزن‌های فازی قابل انجام است.

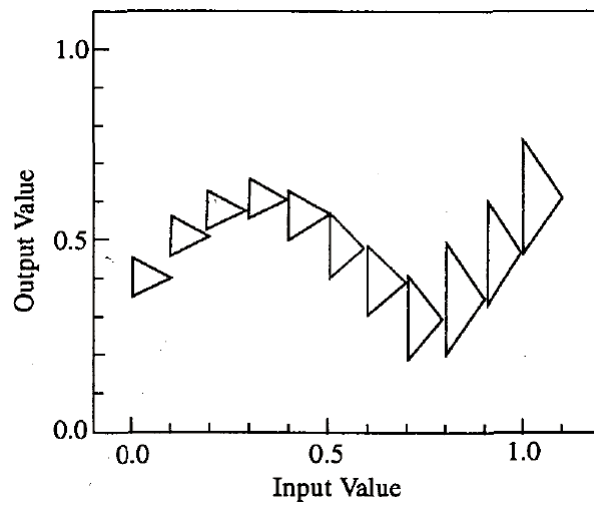
مثال ۵-۷.

داده‌های آموزشی در شکل ۵-۲۰ (آ) را در نظر بگیرید. خروجی‌های هدف فازی متناظر با شش مقدار ورودی $x = 0, 0.2, \dots, 1$ در قالب اعداد فازی مثلثی هستند. شبکه عصبی پیش‌خور با یک گره ورودی، ۵ گره پنهان و یک گره خروجی طی ۱۰۰۰۰ تکرار با استفاده از قواعد یادگیری و با در نظر گرفتن $\alpha = 0.9$ ، $\eta = 0.5$ و $h = 0.2, 0.4, 0.6, 0.8, 1.0$ آموزش می‌بیند. خروجی فازی حاصل از شبکه عصبی آموزش‌یافته متناظر با ۱۱ مقدار ورودی $(x = 0, 0.1, \dots, 1)$ در شکل ۵-۲۰ (ب) نشان داده شده است. این نتایج حاکی از آن است که برازش خوبی روی شش خروجی هدف فازی به همراه درونیابی مطلوب برای ۵ ورودی جدید حاصل شده است.

تابع هزینه فازی. الگوریتم یادگیری که پیش‌تر مورد بحث قرار گرفت، از یک تابع هزینه غیرفازی (رابطه ۵-۶۱ یا ۵-۷۱) برای شبکه‌های عصبی پیش‌خور با پارامترهای فازی استفاده می‌کرد. در ادامه،



(ا)



(ب)

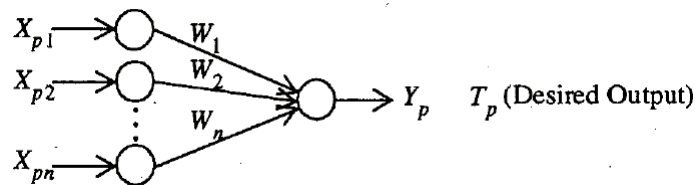
شکل ۵-۲۰: (آ) داده‌های آموزشی برای یادگیری شبکه عصبی (ب) خروجی‌های فازی حاصل از شبکه عصبی آموزش‌یافته

رویکرد دیگری را در نظر می‌گیریم که فازی‌سازی الگوریتم پسانتشار آن، مستقیماً براساس یک تابع هزینه فازی مقدار انجام می‌شود. یک شبکه دو لایه ساده را مطابق شکل ۵-۲۱ در نظر بگیرید. تمامی ورودی‌های فازی (X_{pi}) و خروجی‌های فازی (Y_p) ، اعداد فازی در بازه $[0, 1]$ هستند اما امکان دارد که وزن‌ها (W_i) مجموعه‌های فازی یا اعداد حقیقی باشند. رابطه ورودی-خروجی این شبکه در اینجا تشریح می‌شود:

$$Y_p = f(\text{Net}_p) \quad (92-5)$$

$$\text{Net}_p = \sum_{i=1}^n W_i X_{pi} + \Theta \quad (93-5)$$

در روابط فوق، f تابع سیگموئید و Θ بایاس فازی است. به منظور محاسبه Net_p در رابطه ۵-۹۳، حساب فازی استاندارد و برای محاسبه Y_p در رابطه ۵-۹۲ نیز اصل توسعه^۱ مورد استفاده قرار می‌گیرد. به منظور سهولت، الگوریتم یادگیری را براساس یک تابع هزینه فازی برای این شبکه عصبی ساده فراهم می‌آوریم. به طور طبیعی، چنین الگوریتم یادگیری را می‌توان به سادگی برای شبکه‌های عصبی چندلایه با وزن‌ها و بایاس‌های فازی (اعداد فازی دلخواه) تعمیم داد.



شکل ۵-۲۱: شبکه عصبی با ورودی‌های فازی، خروجی فازی و وزن‌های فازی

با فرض زوج‌های ورودی-خروجی فازی (X_p, T_p) با $p = 1, 2, \dots, S$ ، فازی‌سازی تابع هزینه

استفاده شده مطابق زیر است:

$$E = \frac{1}{2} \sum_{p=1}^S (Y_p - T_p)^2 \quad (94-5)$$

1. Extension Principle: $\mu_{Y_p}(z) = \max \left\{ \mu_{\text{Net}_p}(x) \mid z = f(x) \right\}$

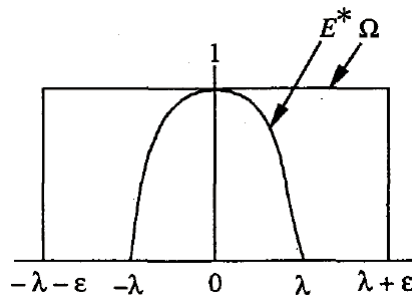
به منظور حاصل کردن E - که یک عدد فازی است - از حساب فازی استفاده می‌شود. اکنون الگوریتم یادگیری را خواهیم یافت که E را به صفر متمایل کند. با این حال به دلیل تفریق فازی، E برابر با صفر نخواهد شد حتی اگر برای همه p ها، رابطه $Y_p = T_p$ برقرار باشد. بنابراین باید یک شرط توقف تعریف شود. بازه $[t_{p1} \quad t_{p2}]$ با $1 \leq p \leq S$ را پشتیبان T_p در نظر می‌گیریم. اگر برای همه p ها، رابطه $Y_p = T_p$ برقرار باشد، بازه $[-\lambda \quad \lambda]$ پشتیبان T_p است به طوری که:

$$\lambda = \frac{1}{2} \sum_{p=1}^S (t_{p2} - t_{p1})^2 \quad (95-5)$$

وقتی برای همه p ها، رابطه $Y_p = T_p$ برقرار باشد، $\epsilon > 0$ نشانگر انحراف‌های قابل قبول از مقدار E خواهد بود. به این ترتیب شرط توقف برای پایان تکرار به روزرسانی وزن را می‌توان به صورت زیر تعریف کرد که در شکل ۵-۲۲ واضح شده است.

$$E \subset \Omega = [-\lambda - \epsilon \quad \lambda + \epsilon] \times [0 \quad 1] \quad (96-5)$$

بر اساس رابطه فوق و شکل مذکور، شرط توقف وقتی صدق می‌کند که خطای E در مستطیل Ω باشد.



شکل ۵-۲۲: شرط توقف در رابطه ۵-۹۶

E^* نشان دهنده مقدار E در شرایطی است که رابطه $Y_p = T_p$ برای همه p ها برقرار باشد. با این شرط توقف، الگوریتم یادگیری تا زمانی که شرط توقف صدق کند، اجرا می‌شود:

$$W_i(t+1) = W_i(t) - \eta \frac{\partial E}{\partial W_i} \quad i = 1, 2, \dots, n \quad (97-5)$$

η یک عدد حقیقی به عنوان نرخ یادگیری است. برای یافتن مشتق در رابطه ۵-۹۷، کافی است طوری مشتق بگیریم گویی که E و W اعداد حقیقی هستند و سپس این اعداد حقیقی را با مجموعه‌های فازی E, W_i, Y_p, T_p و X_{pi} جایگزین کنیم.

$$\frac{\partial E}{\partial W_i} = \sum_{p=1}^S (Y_p - T_p) (Y_p) (1 - Y_p) X_{pi} \quad (98-5)$$

رابطه‌های ۵-۹۷ و ۵-۹۸ قاعده دلتای فازی شده را به دست می‌دهند. این یک مشتق کاملاً صوری است زیرا مشتق فازی نشان داده شده در رابطه ۵-۹۷ در واقع انجام نمی‌شود.

۵-۳-۳ شبکه پس‌انتشار فازی (Fuzzy BP)

در بسیاری از کاربردها - نظیر سیستم کنترل، شناسایی الگو و تصمیم‌گیری - ورودی‌ها عبارات فازی بوده و خروجی‌ها مقادیر قطعی هستند. در همین راستا، لی^۱ و لیو^۲ (۱۹۹۴) یک مدل شبکه عصبی ایجاد کردند که می‌تواند نگاهی را از ورودی‌های فازی به خروجی‌های قطعی ایجاد کند. این نسخه فازی از مدل شبکه عصبی با نام پس‌انتشار فازی (Fuzzy BP)^۳ ارائه شده است. عنصر اساسی این مدل، یک نورون فازی با قابلیت پردازش اعداد فازی است که به آن اشاره خواهد شد. به منظور آموزش پس‌انتشار فازی از یک الگوریتم یادگیری شبیه به پس‌انتشار استفاده می‌شود. برای سادگی ساختار و کاهش بار محاسباتی، عدد فازی مثلثی LR مورد استفاده قرار می‌گیرد.

نورون فازی

نورون فازی، عنصر اساسی پس‌انتشار فازی است. این نورون می‌تواند نگاهی غیرخطی بین مجموع موزون بردارهای ورودی فازی و خروجی‌های قطعی ایجاد کند. پیش‌تر در بخش ۵-۱ به تشریح ساختار این نورون‌ها پرداخته‌ایم. در اینجا ساختاری شبیه به نورون‌های نوع دو را شاهد هستیم که در شکل ۵-۲۳

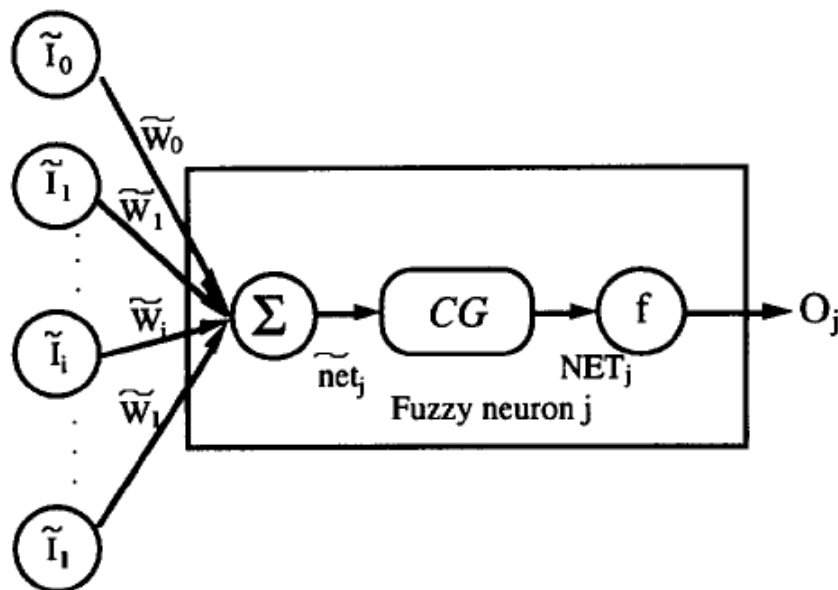
1. Lee, H. M.

2. Lu, B. H.

3. Fuzzy Backpropagation

نمایش داده شده است. مقدار خروجی (O) را می‌توان به صورت زیر بیان کرد:

$$O = f(NE T) = f\left(CE\left(\sum_i \widetilde{W}_i \widetilde{I}_i\right)\right), \quad \widetilde{I}_0 = (1, 0, 0) \quad (99-5)$$



شکل ۵-۲۳: ساختار نرون فازی

عدد فازی $\widetilde{I} = (\widetilde{I}_0, \widetilde{I}_1, \dots, \widetilde{I}_l)$ بردار ورودی است و $\widetilde{W} = (\widetilde{W}_0, \widetilde{W}_1, \dots, \widetilde{W}_l)$ بردار وزن فازی است؛ \widetilde{I}_0 نیز نقش بایاس را ایفا می‌کند. مجموع موزون فازی $(\widetilde{net} = \sum_j \widetilde{W}_j \widetilde{I}_j)$ و نتیجه استنتاج $(NE T = CE(\widetilde{net}))$ در شکل ۵-۲۳ نمایش داده شده است. تابع CE یک عمل‌گر مرکزوار^۱ در عدد فازی مثلثی است که می‌توان آن را به‌عنوان یک عمل‌گر فازی‌زدایی برش‌مرد که نگاهیست مقادیر مجموع وزنی فازی به یک مقدار قطعی را انجام می‌دهد. با فرض اینکه مجموع موزون فازی $\widetilde{net} = (net_m, net_\alpha, net_\beta)$ باشد، تابع CE به فرم زیر درمی‌آید:

$$CE(\widetilde{net}) = net_m + \frac{1}{3}(net_\beta - net_\alpha) = NET \quad (100-5)$$

1. Centroid

که NET نتیجه فازی‌زدایی عدد فازی مثلثی \widetilde{net} است. اعداد فازی مثلثی هم در بردارهای وزن فازی و هم در بردارهای ورودی فازی مورد استفاده قرار می‌گیرد. تابع f یک تابع سیگموئید است که نگاشت غیرخطی بین ورودی و خروجی‌ها را انجام می‌دهد. این تابع به صورت زیر تعریف می‌شود:

$$f(NET) = \frac{1}{1 + \exp(-NET)} \quad (101-5)$$

ساختار پس‌انتشار فازی

ساختار پس‌انتشار فازی با یک شبکه پیش‌خور سه لایه در شکل ۵-۲۴ نشان داده شده است. زمانی که بردار $\widetilde{I}_p = (\widetilde{I}_{p1}, \widetilde{I}_{p2}, \dots, \widetilde{I}_{pl})$ به لایه ورودی ارائه می‌شود، فرآیند هر لایه مجزا به صورت زیر بیان می‌گردد. نورون‌های ورودی در رابطه ۵-۱۰۲، نورون‌های پنهان در رابطه‌های ۵-۱۰۳ و ۵-۱۰۴ و نورون‌های خروجی در رابطه‌های ۵-۱۰۵ و ۵-۱۰۶ تشریح شده‌اند:

$$\widetilde{O}_{pi} = \widetilde{I}_{pi}, \quad i = 1, 2, \dots, l; \quad \widetilde{O}_{p0} = (1, 0, 0) \quad (102-5)$$

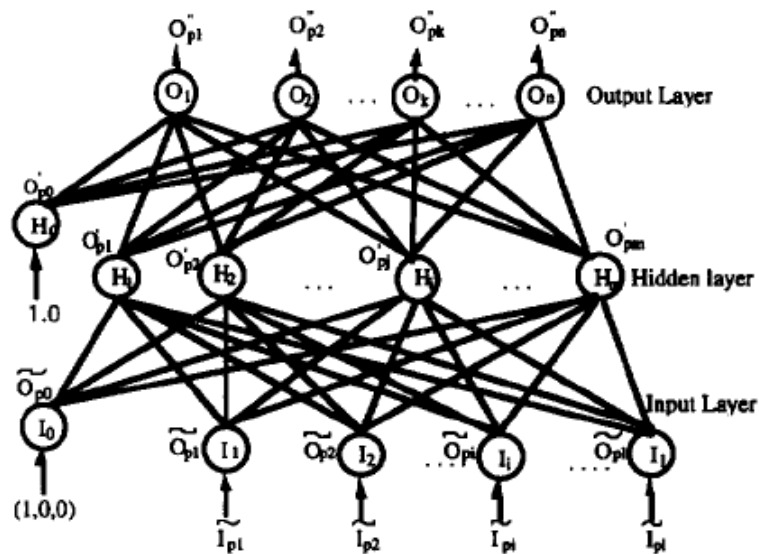
$$O'_{pj} = f(NE_{T_{pj}}), \quad j = 1, 2, \dots, m; \quad O'_{p0} = 1 \quad (103-5)$$

$$NE_{T_{pj}} = CE \left(\sum_{i=0}^l \widetilde{W}_{ji} \widetilde{O}_{pi} \right) \quad (104-5)$$

$$O''_{pk} = f(NE'_{T_{pk}}), \quad k = 0, 1, \dots, n \quad (105-5)$$

$$NE'_{T_{pk}} = CE \left(\sum_{j=0}^m \widetilde{W}'_{kj} \widetilde{O}'_{pj} \right) \quad (106-5)$$

در روابط فوق، \tilde{I}_{pi} عنصر ورودی نام از الگوهای ورودی است. \tilde{O}_{pi} مقدار خروجی از گره ورودی نام را نشان می‌دهد. O'_{pj} و O''_{pk} به ترتیب خروجی‌های قطعی فازی زدایی نام و نام از گره‌های پنهان و گره‌های خروجی هستند. \tilde{W}'_{kj} وزن‌های اتصال فازی بین گره پنهان نام و گره خروجی نام را نشان می‌دهد. علاوه بر این، f و CE توابع سیگموئید و مرکزوار هستند که در بخش قبل به آنها اشاره شد.



شکل ۵-۲۴: ساختار پس‌انتشار فازی

فرآیند آموزش

در این بخش، روند یادگیری مورد بررسی قرار می‌گیرد. ابتدا، تابع میانگین مربعات خطا برای الگوی p به صورت زیر تعریف می‌شود:

$$E_p = \sum_i \frac{1}{2} (D_{pi} - O''_{pi})^2 \quad (۱۰۷-۵)$$

D_{pi} مقدار خروجی مطلوب از گره خروجی نام بوده و O''_{pi} مقدار خروجی واقعی از آن گره خروجی است. خطای کلی حاصل از الگوی آموزشی توسط $E = \sum_p E_p$ بیان می‌شود. در مرحله یادگیری، به منظور

حداقل سازی E باید مقادیر وزن‌ها را تنظیم نمود. در تکرار t ، تغییر وزن به صورت زیر بیان می‌شود:

$$\Delta \widetilde{W}(t) = -\eta \nabla E_p(t) + \alpha \Delta \widetilde{W}(t-1) \quad (108-5)$$

η نرخ یادگیری و α یک مقدار ثابت است. عبارت $\alpha \Delta \widetilde{W}(t-1)$ نقش تکانه^۱ را ایفا می‌کند که به منظور بهبود سرعت همگرایی اضافه می‌شود. عبارت $\nabla E_p(t)$ را می‌توان به صورت زیر نوشت:

$$\nabla E_p(t) = \frac{\partial E_p}{\partial \widetilde{W}(t)} = \left(\frac{\partial E_p}{\partial \widetilde{W}_m(t)}, \frac{\partial E_p}{\partial \widetilde{W}_\alpha(t)}, \frac{\partial E_p}{\partial \widetilde{W}_\beta(t)} \right) \quad (109-5)$$

که $\widetilde{W}(t) = (\widetilde{W}_m(t), \widetilde{W}_\alpha(t), \widetilde{W}_\beta(t))$ است. براساس آنچه در ادامه می‌آید، $\frac{\partial E_p}{\partial \widetilde{W}_m}$ ، $\frac{\partial E_p}{\partial \widetilde{W}_\beta}$ و $\frac{\partial E_p}{\partial \widetilde{W}_\alpha}$ تشریح خواهند شد. همان‌طور که در شکل ۵-۲۴ نشان داده شده است، فرض می‌کنیم که $\widetilde{W}_{ji} = (W_{m_{ji}}, W_{\alpha_{ji}}, W_{\beta_{ji}})$ وزن‌های اتصال بین گره ورودی m و گره پنهان j ام و $\widetilde{W}'_{kj} = (W'_{m_{kj}}, W'_{\alpha_{kj}}, W'_{\beta_{kj}})$ وزن‌های اتصال بین گره پنهان j ام و گره خروجی k ام هستند. با پیاده‌سازی قواعد زنجیره‌ای خواهیم داشت:

$$\frac{\partial E_p}{\partial W'_{m_{kj}}} = \frac{\partial E_p}{\partial net'_{m_k}} \frac{\partial net'_{m_k}}{\partial W'_{m_{kj}}} \quad (110-5)$$

عبارت اول به صورت زیر به دست می‌آید:

$$\begin{aligned} \frac{\partial E_p}{\partial net'_{m_k}} &= \frac{\partial E_p}{\partial O''_k} \frac{\partial O''_k}{\partial net'_{m_k}} \quad (111-5) \\ &= -(D_{pk} - O''_{pk}) \frac{\partial O''_k}{\partial NET'_k} \frac{\partial NET'_k}{\partial net'_{m_k}} \\ &= -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) \quad (1) \end{aligned}$$

عبارت دوم را نیز می‌توان به این صورت حاصل کرد:

$$\frac{\partial net'_{m_k}}{\partial W'_{m_{kj}}} = O'_{pj} \quad (112-5)$$

1. Momentum

پس براساس روابط ۵-۱۱۰ تا ۵-۱۱۲ می‌توان نوشت:

$$\begin{aligned} \frac{\partial E_p}{\partial W'_{m_{kj}}} &= \frac{\partial E_p}{\partial O''_k} \frac{\partial O''_k}{\partial NET'_k} \frac{\partial NET'_k}{\partial net'_{m_k}} \frac{\partial net'_{m_k}}{\partial W'_{m_{kj}}} \quad (113-5) \\ &= -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) (1) O'_{pj} \end{aligned}$$

به‌طور مشابه، خواهیم داشت:

$$\begin{aligned} \frac{\partial E_p}{\partial W'_{\alpha_{kj}}} &= \frac{\partial E_p}{\partial O''_k} \frac{\partial O''_k}{\partial NET'_k} \frac{\partial NET'_k}{\partial net'_{\alpha_k}} \frac{\partial net'_{\alpha_k}}{\partial W'_{\alpha_{kj}}} \quad (114-5) \\ &= -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) \left(-\frac{1}{3}\right) O'_{pj} \end{aligned}$$

$$\begin{aligned} \frac{\partial E_p}{\partial W'_{\beta_{kj}}} &= \frac{\partial E_p}{\partial O''_k} \frac{\partial O''_k}{\partial NET'_k} \frac{\partial NET'_k}{\partial net'_{\beta_k}} \frac{\partial net'_{\beta_k}}{\partial W'_{\beta_{kj}}} \quad (115-5) \\ &= -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) \left(\frac{1}{3}\right) O'_{pj} \end{aligned}$$

در ادامه، مقادیر خطاهای δ_{m_k} ، δ_{α_k} و δ_{β_k} تعریف می‌شود:

$$\delta_{m_k} = \frac{\partial E_p}{\partial net'_{m_k}} = -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) (1) \quad (116-5)$$

$$\delta_{\alpha_k} = \frac{\partial E_p}{\partial net'_{\alpha_k}} = -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) \left(-\frac{1}{3}\right) \quad (117-5)$$

$$\delta_{\beta_k} = \frac{\partial E_p}{\partial net'_{\beta_k}} = -(D_{pk} - O''_{pk}) O''_{pk} (1 - O''_{pk}) \left(\frac{1}{3}\right) \quad (118-5)$$

براساس نتایج قبلی، وزن‌های اتصال بین لایه‌های ورودی و پنهان را می‌توان به‌صورت زیر محاسبه

نمود:

$$\frac{\partial E_p}{\partial W_{m_{ji}}} = \frac{\partial E_p}{\partial O'_j} \frac{\partial O'_j}{\partial W_{m_{ji}}} = \left(\sum_k \delta_{m_k} W'_{m_{kj}} \right) O'_{pj} (1 - O'_{pj}) (1) O_{pi} \quad (119-5)$$

$$\frac{\partial E_p}{\partial W_{\alpha_{ji}}} = \left(\sum_k \delta_{\alpha_k} W'_{\alpha_{kj}} \right) O'_{pj} (1 - O'_{pj}) \left(-\frac{1}{3}\right) O_{pi} \quad (120-5)$$

$$\frac{\partial E_p}{\partial W_{\beta_{ji}}} = \left(\sum_k \delta_{\beta_k} W'_{\beta_{kj}} \right) O'_{pj} (1 - O'_{pj}) \left(\frac{1}{3}\right) O_{pi} \quad (121-5)$$

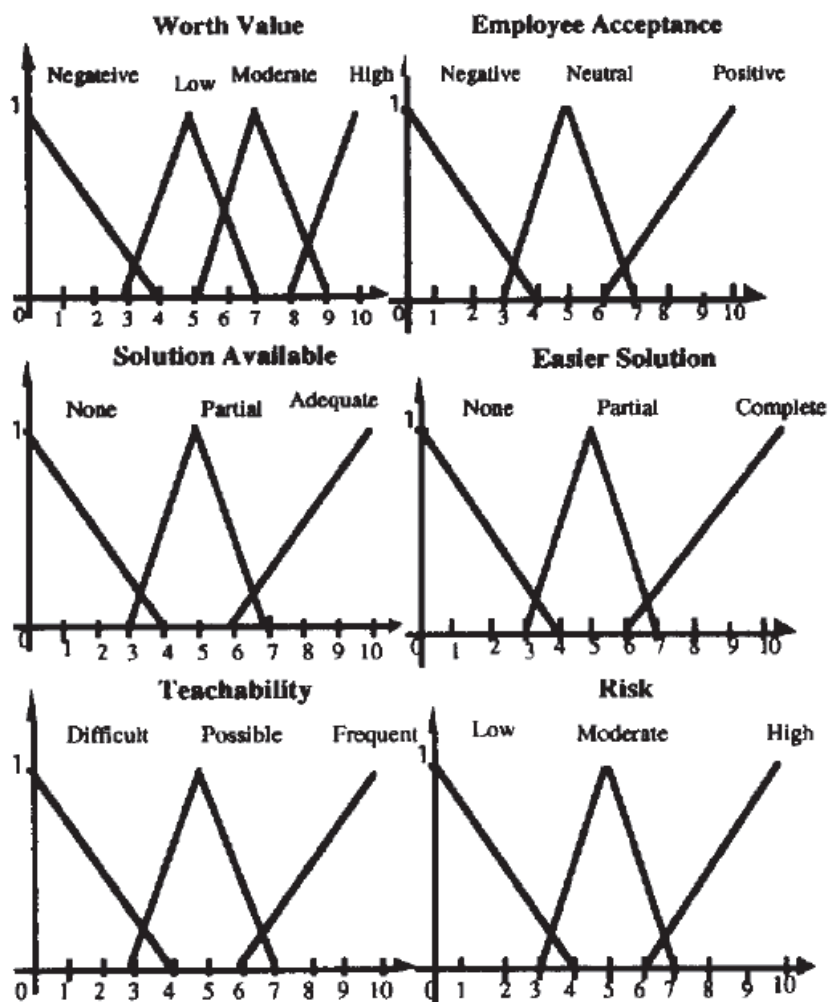
در نتایج فوق، روابط ۵-۱۱۳ تا ۵-۱۱۵ و ۵-۱۱۹ تا ۵-۱۲۱ به عنوان فرمول‌های تغییر وزن‌ها به کار گرفته می‌شوند. از آنجایی که W_m نسبت به W_α و W_β تأثیر بیشتری بر تغییر مرکزوار دارد، می‌توان تصور کرد که بازه تنظیم و تعدیل W_m نسبت به W_α و W_β بزرگتر است. علاوه بر این، تغییر جهت W_α و W_β مخالف است. براساس نتایج حاصل‌شده، تفاوت ناچیز بین روابط ۵-۱۱۳ تا ۵-۱۱۵ و ۵-۱۱۹ تا ۵-۱۲۱ به معنای این است که بار محاسباتی پس‌انتشار فازی سنگین نیست.

مثال ۵-۸. تجربه‌ای از ارزیابی پایگاه دانش با استفاده از پس‌انتشار فازی

به منظور شفافیت بیشتر در کار با پس‌انتشار فازی، نمونه‌ای از ارزیابی‌کننده پایگاه دانش (KBE)^۱ در اینجا شبیه‌سازی می‌شود. KBE سیستم خبره‌ای است که سازگاری به‌کارگیری یک سیستم خبره را در یک حوزه مشخص ارزیابی می‌کند. پایگاه دانش در این مثال، دارای ۱۸ نمونه است. هر شاخص ورودی یک عبارت زبانی است و سازگاری یک سیستم خبره تحت این ارزیابی توسط خروجی نشان داده شده است. شاخص‌های موردنظر در این مطالعه عبارتند از: ارزشمندی، پذیرش کارمندان، وجود راه‌حل، راه‌حل ساده‌تر، قابلیت آموزش و ریسک.

شکل ۵-۲۵ توابع عضویت هر شاخص KBE را مشخص می‌کند. همان‌طور که در شکل مشخص است، هر متغیر زبانی دارای سه یا چهار مجموعه فازی است. این مجموعه‌های فازی از نوع LR هستند. شاخص‌های KBE و نمونه‌های آموزشی در جدول ۵-۲ نشان داده شده است. علامت * در جدول به معنای بی‌تفاوتی است یعنی هر عبارت امکان‌پذیری را می‌توان به جای آن قرار داد. به این ترتیب برای عبارات بی‌تفاوتی در نمونه‌های ۲ و ۳ می‌توان جایگشت همه عبارت‌های ممکن را در نظر گرفت؛ پس از این ۱۸ نمونه، ۳۴۰ نمونه تولید می‌شود. در این ۳۴۰ نمونه، ۲۹۰ نمونه به‌طور تصادفی به عنوان نمونه‌های آموزشی انتخاب شده و سایر نمونه‌ها به عنوان نمونه‌های آزمون تلقی می‌شوند.

در این مطالعه، از یک پس‌انتشار فازی سه لایه‌ای با شش گره ورودی، شش گره پنهان و



شکل ۵-۲۵: توابع عضویت هر شاخص در مثال KBE

یک گره خروجی استفاده شده است. مقادیر α و η به ترتیب برابر با $0/9$ و $0/1$ در نظر گرفته شده و زمان یادگیری متوسط، 483 تکرار در 10 آزمایش بوده است.

پس از هر یادگیری، از نمونه‌های آزمون برای سنجش قابلیت تعمیم پس‌انتشار فازی مذکور استفاده شده است. مقدار خروجی این‌گونه تعیین شد: «اگر خروجی بزرگتر یا مساوی $0/5$ باشد، سازگاری خوب است و اگر کوچکتر از $0/5$ باشد، سازگاری ضعیف

جدول ۵-۲: نمونه‌های مثال KBE

نمونه	ارزشمندی	پذیرش کارمندان	وجود راه‌حل	راه‌حل ساده‌تر	قابلیت آموزش	ریسک	خروجی
۱	بالا	مثبت	هیچ	هیچ	مکرر	پایین	خوب
۲	منفی	*	*	*	*	*	ضعیف
۳	پایین	*	*	*	*	بالا	ضعیف
۴	متوسط	خنثی	کافی	کامل	دشوار	بالا	ضعیف
۵	پایین	منفی	هیچ	نسبی	مکرر	پایین	ضعیف
۶	بالا	منفی	نسبی	هیچ	دشوار	متوسط	خوب
۷	بالا	مثبت	نسبی	کامل	مکرر	بالا	ضعیف
۸	بالا	مثبت	نسبی	نسبی	ممکن	پایین	ضعیف
۹	پایین	مثبت	کافی	هیچ	مکرر	پایین	خوب
۱۰	بالا	منفی	نسبی	هیچ	مکرر	بالا	خوب
۱۱	پایین	مثبت	هیچ	کامل	دشوار	متوسط	ضعیف
۱۲	پایین	خنثی	کافی	کامل	مکرر	پایین	خوب
۱۳	پایین	خنثی	هیچ	هیچ	دشوار	پایین	خوب
۱۴	متوسط	مثبت	کافی	هیچ	دشوار	بالا	ضعیف
۱۵	بالا	منفی	کافی	نسبی	مکرر	بالا	ضعیف
۱۶	بالا	منفی	نسبی	کامل	ممکن	پایین	خوب
۱۷	متوسط	منفی	هیچ	نسبی	دشوار	بالا	خوب
۱۸	متوسط	خنثی	کافی	نسبی	دشوار	پایین	ضعیف

است». در نتیجه، هر نمونه آزمون را می‌توان به درستی طبقه‌بندی کرد.

موردکاوی

خصوصی‌سازی صنعت برق در بسیاری از نقاط جهان، اهمیت پیش‌بینی مقدار بار را افزایش داده است؛ به‌ویژه اینکه در این‌باره نیاز جدی به درک و پیش‌بینی دقیق‌تر از میزان تقاضا وجود دارد. در همین زمینه، سیها و ساراوانان (۲۰۰۷) با استفاده از شبکه پس‌انتشار فازی اقدام به پیش‌بینی کوتاه‌مدت بار کردند.

مدل این پژوهش از دو ورودی (بار و دما) و یک خروجی (بار) متناظر تشکیل شده است. بار هریک از روزهای هفته با همان روز در هفته‌های پیشین مشابهت داشته است؛ یعنی مثلاً بار روزهای یکشنبه شبیه به یکدیگر بوده و این موضوع کار تحلیل را ساده‌تر می‌کند. ورودی‌های شبکه برای هر روز، مقدار بار روز قبل (۲۴ ورودی)، میزان دمای روز قبل (۲۴ ورودی) و حداکثر، حداقل و متوسط دمای پیش‌بینی‌شده برای آن روز (۳ ورودی) است که مجموعاً ۵۱ مقدار را تشکیل می‌دهد. خروجی نیز پیش‌بینی مقدار بار برای ۲۴ ساعت آن روز است که عملاً ۲۴ خروجی خواهیم داشت. بنابراین با استفاده از داده‌های گذشته شبکه پس‌انتشار فازی (Fuzzy BP) و نیز یک شبکه عصبی پرسپترون چندلایه (MLP) تحت آموزش قرار گرفتند. خروجی‌ها مقدار قطعی داشته و ورودی‌ها پس از نرمال‌سازی در قالب اعداد فازی LR درآمدند؛ یعنی مقدار بار و دما در بازه یکسانی قرار گرفتند. نرمال‌سازی یکی از مراحل مهم است زیرا موجب کاهش زمان یادگیری در هر دو نوع شبکه می‌شود.

در فرآیند آموزش شبکه، وزن اولیه بین لایه‌های مختلف برابر با $(0/5, 0/5, 0/5)$ قرار داده شد. همچنین نرخ یادگیری (η) و تکانه (α) به ترتیب برابر با $0/79$ و $0/59$ در نظر گرفته شدند. مقدار خروجی مطلوب شبکه به تعداد داده‌های مورد استفاده در فرآیند آموزش بستگی دارد و اگر تعداد داده‌ها ناکافی باشد، آنگاه خروجی نمی‌تواند نماینده خوبی برای ورودی‌های درست باشد. از سوی دیگر، اگر تعداد داده‌های ورودی زیاد باشد، زمان آموزش طولانی خواهد شد. از این روی، در این مطالعه به منظور یافتن تعداد داده‌های مناسب برای آموزش ابتدا با ۱۵، ۲۰ و ۲۵

الگو آموزش داده شد تا بتوان آنها را مقایسه نمود. در میان آنها، تعداد ۲۰ الگوی آموزشی برای هر دو شبکه پس‌انتشار فازی و MLP کمترین میزان خطا را داشته است و عملاً داده‌های مربوط به ۲۰ هفته گذشته مورد بررسی قرار گرفت.

در نهایت با آزمون نتایج حاصل از هر دو شبکه، نشان داده شد که شبکه پس‌انتشار فازی در اغلب روزها عملکرد بهتری نسبت به MLP دارد. این عملکرد با استفاده از مقایسه میانگین درصد خطای مطلق (MAPE) و درصد خطای حداکثر بار سنجیده شده است.

Seetha, H. & Saravanan, R. (2007) Short Term Electric Load Prediction Using Fuzzy BP. Journal of Computing and Information Technology, 15(3), 267-282.

برای مطالعه و دسترسی به جزئیات بیشتر می‌توانید به مقاله لی^۱ و لیو^۲ (۱۹۹۴) یا کتاب راجاسکاران^۳ و ویجی آلاکشمی پای^۴ (۲۰۰۳) مراجعه کنید.

۴-۵ شبکه‌های عصبی فازی تکاملی

شبکه‌های عصبی فازی تکاملی (EFuNN)^۵ و شبکه‌های عصبی فازی تکاملی پویا (dmEFuNN)^۶ براساس چارچوب سیستم‌های پیوندگرای تکاملی (ECOS)^۷ برای سیستم‌های هوشمند انطباقی بنا شده‌اند که به سبب تکامل و یادگیری افزایشی، هیبریدی (با نظارت و بدون نظارت) و آنلاین شکل گرفته‌اند. آنها می‌توانند داده‌های ورودی جدید شامل ویژگی‌های جدید، طبقه‌های جدید و غیره را از طریق تنظیم مؤلفه‌ها سازگار سازند.

در EFuNNها، همه گره‌ها در خلال یادگیری ایجاد می‌شوند. EFuNN یک ساختار پنج لایه‌ای دارد

1. Lee, H. M.
2. Lu, B. H.
3. Rajasekaran, S
4. Vijayalakshmi Pai, G. A.
5. Evolving Fuzzy Neural Networks
6. Dynamic Evolving Fuzzy Neural Networks
7. Evolving COnnectionist Systems

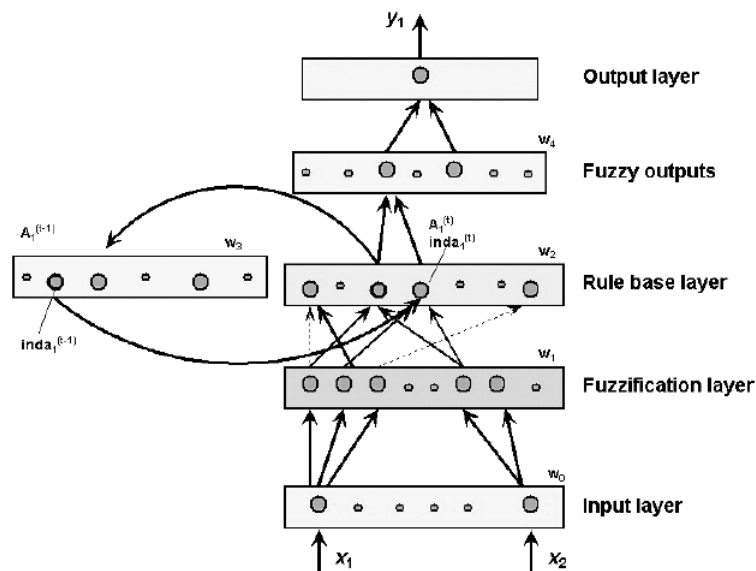
که در شکل ۵-۲۶ به نمایش درآمده است. لایه ورودی یک لایه بافر است که متغیرهای ورودی را ارائه می‌کند. در لایه دوم، توصیف فازی ویژگی‌ها^۱ برای هر فضای متغیر ورودی بیان می‌شود. هریک از متغیرهای ورودی توسط گروهی از نوروهای مرتب‌شده فضایی^۲ که به منظور ارائه یک چندی‌سازی فازی^۳ از این متغیرها ایجاد شده‌اند، نمایان می‌گردد. گره‌هایی که توابع عضویت (مثلثی، گوسی و غیره) را بیان می‌کنند، می‌توانند در حین یادگیری اصلاح و تعدیل شوند. لایه سوم شامل گره‌های قواعد هستند که از طریق یادگیری با نظارت / بدون نظارت مرکب تکامل می‌یابند. گره‌های قواعد، نمونه‌های اولیه از داده‌های هم‌پیوند ورودی-خروجی را ارائه می‌کنند که به صورت تصویری در قالب پیوند ابرگره‌ها از فضای فازی ورودی و خروجی بیان می‌شود. هر گره قاعده r توسط دو بردار از وزن‌های اتصال تعریف می‌گردد: $W_1(r)$ و $W_2(r)$. اولی از طریق یادگیری با نظارت براساس خطای خروجی تعدیل می‌گردد و دومی از طریق یادگیری بدون نظارت براساس میزان تشابه در یک ناحیه محلی از فضای مسأله ورودی تعدیل می‌شود. لایه چهارم، توصیف ویژگی‌های فازی را برای متغیرهای ورودی ارائه می‌دهد. لایه پنجم نیز مقادیر حقیقی متغیرهای ورودی را بیان می‌کند. در حالت one-of-n از EFN، بیشترین مقدار فعال‌سازی گره‌های قواعد به لایه بعد انتشار می‌یابد. در حالت many-of-n نیز همه مقادیر فعال‌سازی گره‌های قواعد که بالاتر از یک حد آستانه فعال‌سازی باشند، در این ساختار پیوندگرا منتشر می‌شود.

جزئیات بیشتر درباره شبکه‌های عصبی فازی تکاملی در پژوهش کاسابوف^۴ (۱۹۹۸) قابل دسترسی است.

شبکه‌های عصبی فازی تکاملی پویا (dmEFuNN)

مدل شبکه‌های عصبی فازی تکاملی پویا (dmEFuNN) بر اساس این ایده توسعه یافته است که نه تنها فعال‌سازی گره قاعده برنده انتشار یابد بلکه گروهی از گره‌های قواعد به طور پویا برای هر بردار ورودی جدید انتخاب می‌شود و مقادیر فعال‌سازی آنها به منظور محاسبه پارامترهای پویایی تابع خروجی مورد استفاده قرار می‌گیرد. با آنکه EFN از قواعد فازی موزون ممدانی استفاده می‌کند، dmEFuNN از

1. Fuzzy quantification
2. Spatially arranged neurons
3. Fuzzy quantization
4. Kasabov, N.



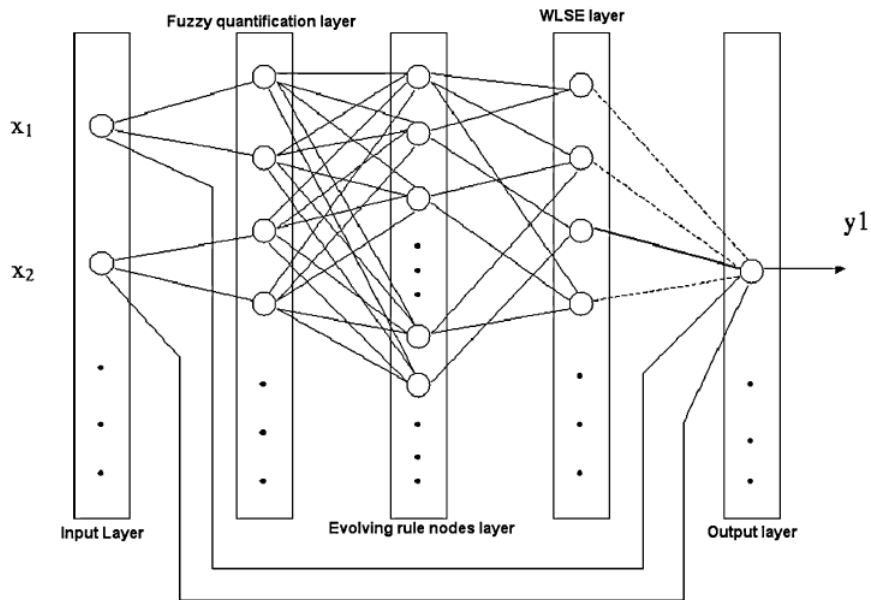
شکل ۵-۲۶: ساختار EFuNN

قواعد فازی تاکاگی-سوگنو بهره می‌گیرد. این ساختار در شکل ۵-۲۷ نشان داده شده است.

لایه‌های اول، دوم و سوم از dmEFuNN دقیقاً ساختارها و توابعی همانند EFuNN دارند. لایه چهارم - لایه استنتاج فازی - m گره قاعده از لایه سوم که نزدیک‌ترین فاصله محلی نرمالایز شده فازی را از بردار ورودی فازی دارند انتخاب می‌کند و سپس یک قاعده فازی تاکاگی-سوگنو را با استفاده از تخمین کمترین مربعات موزون ساختار شکل می‌دهد. لایه آخر نیز خروجی dmEFuNN را محاسبه می‌کند.

تعداد گره‌های فعال (m) که برای محاسبه مقادیر خروجی dmEFuNN مورد استفاده قرار می‌گیرد، کمتر از تعداد گره‌های ورودی به علاوه یک نیست. مشابه EFuNNها، dmEFuNNها را نیز می‌توان در هر دو نوع یادگیری آفلاین و آنلاین به کار برد چنان‌که بهینه‌سازی خطای تعمیم سراسری یا خطای تعمیم محلی نیز ممکن می‌شود. در EFuNNها برای یک بردار ورودی جدید (که بردار خروجی آن نامعلوم است) زیرفضایی شامل m گره قاعده یافته شده و یک قاعده به منظور محاسبه مقدار خروجی dmEFuNN به کار گرفته می‌شود. به این ترتیب، یک dmEFuNN به عنوان یک تقریب‌گر تابع فراگیر^۱ عمل می‌کند که از

1. Universal function approximator



شکل ۵-۲۷: ساختار dmEFuNN

m تابع خطی در یک زیرفضای کوچک m بعدی استفاده می‌کند. دقت این تخمین به اندازه زیرفضاهای مذکور بستگی دارد؛ هرچه زیرفضا کوچک‌تر باشد، دقت بیشتری ایجاد خواهد شد. این یعنی اگر بردارهای داده آموزشی و گره‌های قواعد به میزان کافی ایجاد شوند، می‌توان دقت مطلوبی را حاصل کرد. تشریح بیشتر این روش در پژوهش کاسابوف^۱ و سانگ^۲ (۱۹۹۹) قابل مطالعه است.

1. Kasabov, N. K.

2. Song, Q.

فصل ۶

مدل‌های نوین

در فصل‌های قبل، مفاهیم و برخی از ساختارهای متداول حاصل از ترکیب شبکه‌های عصبی مصنوعی و منطق فازی معرفی شدند. این ساختارها ریشه در شبکه‌های عصبی کم‌عمق و منطق فازی نوع ۱ داشتند. این در حالی است که یادگیری ماشین با شتاب در حال حرکت است و هر روز بر مقالات و یافته‌های این دانش افزوده می‌شود. شبکه‌های عصبی مصنوعی و منطق فازی نیز مدام در حال تکامل هستند و با ساختارها و الگوریتم‌های نو توسعه می‌یابند. در سال‌های اخیر، شبکه‌های عصبی مصنوعی با یادگیری عمیق و منطق فازی با مجموعه‌های فازی نوع ۲ بیشتر مورد توجه قرار گرفته‌اند و در مسیر توسعه هستند. در این فصل، سیستم‌های ترکیبی فازی و عصبی مبتنی بر هر یک از این دو حوزه مورد اشاره واقع می‌شوند تا مقدمه‌ای برای پژوهشگران و دانشجویان فراهم آید.

۶-۱ ترکیب مجموعه‌های فازی نوع ۲ با شبکه‌های عصبی

همان‌طور که در فصول قبل این کتاب تشریح شد، سیستم‌های عصبی فازی به‌طور گسترده در محیط‌های بسیار پویا و غیرقطعی در حل مسائل کنترل، برآورد و طبقه‌بندی مورد استفاده قرار می‌گیرند. این سیستم‌ها نوعاً براساس توابع عضویت نوع ۱ ایجاد شده‌اند، که توابع عضویت به‌کاررفته در بخش‌های مقدم و تالی قواعد فازی آنها از فازی نوع ۱ پیروی می‌کنند. فازی نوع ۱ توانایی اداره مؤثر عدم قطعیت‌های

شناختی (یعنی عدم قطعیت‌هایی درباره صحت نتایج حاصل از پردازش اطلاعات توسط افراد) را همراه با فرآیندهای پیچیده ندارند. به منظور فعال‌سازی یک سیستم که بتواند به روشی مشابه آنچه انسان‌ها عمل می‌کنند به عدم قطعیت‌های شناختی بپردازد، می‌توان مفهوم منطق فازی نوع ۲ را با شبکه‌های عصبی ترکیب کرد. مجموعه‌های فازی نوع ۲ به منظور حداقل کردن اثرات عدم قطعیت در پایگاه قواعد مورد استفاده قرار می‌گیرد.

سیستم‌های فازی نوع ۲ پارامترهای بیشتری نسبت به فازی نوع ۱ دارند و به همین جهت، توانایی بیشتری در مواجهه با عدم قطعیت در مسائل مختلف پیدا می‌کنند. این مجموعه‌ها به دو نوع بازه‌ای^۱ و کلی (عمومی)^۲ تقسیم می‌شوند. سیستم‌های فازی نوع ۲ بازه‌ای پرکاربردتر هستند زیرا بار محاسباتی کمتری داشته و کاربری ساده‌تری دارند. در این مجموعه‌ها بعد سوم برابر با ۱ در نظر گرفته می‌شود پس نمایش آنها دو بُعدی است؛ اما در سیستم‌های فازی نوع ۲ عمومی بُعد سوم یک مجموعه فازی بوده که نمایش سه بُعدی ایجاد می‌کند.

یک سیستم عصبی فازی نوع ۲ شبیه به سیستم‌های نوع ۱ ساخته می‌شود. چنین سیستم‌هایی با یکپارچه‌سازی مجموعه‌های فازی نوع ۲، سازوکار استنتاج سیستم فازی نوع ۲ و یک ساختار شبکه عصبی ایجاد می‌شود. این سیستم از قواعد اگر-آنگاه استفاده می‌کند تا بتواند نگاهی از مجموعه‌های فازی نوع ۲ ورودی به مجموعه‌های فازی نوع ۲ خروجی فراهم آورد.

در سال‌ها اخیر، سیستم‌های متنوعی توسط پژوهشگران ارائه شده است که استنتاج فازی نوع ۲ را با شبکه‌های عصبی مصنوعی ترکیب کرده و به سیستم‌های ترکیبی عصبی و فازی نوع ۲ منجر شده است. در ادامه، به برخی از متداول‌ترین سیستم‌های استنتاج فازی عصبی موجود در ادبیات اشاره می‌شود. تابع عضویت، معیار یادگیری در بخش مقدم و بخش تالی، حوزه کاربردی سیستم و استنتاج فازی مورد استفاده برای هریک از این سیستم‌ها در جدول زیر ارائه شده است. فهرست اولیه این سیستم‌ها از مقاله مروری شهاب‌الدین^۳ و پیلای^۴ (۲۰۱۸) استخراج شده که پس از بررسی تعداد ارجاعات هر مقاله طی سال‌های مختلف، مقالاتی که بیشتر مورد توجه بوده‌اند، در فهرست زیر خلاصه شده است.

1. Interval

2. General

3. Shihabudheen, K. V.

4. Pillai, G. N.

جدول ۶-۱: ساختار برخی سیستم‌های استنتاج فازی عصبی مبتنی بر فازی نوع ۲

روش	سال	تابع عضویت	یادگیری مقدم	یادگیری تالی	کاربرد	استنتاج فازی
T2FNN	۲۰۰۴	زنگوله‌ای نوع ۲ بازه‌ای	GA	پس‌انتشار	کنترل و شناسایی سیستم	نوع ۲ بازه‌ای
SEIT2FNN	۲۰۰۸	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	فیلتر کالمن	کاهش نویز، پیش‌بینی سیگنال تحت آشوب	TSK نوع ۲ بازه‌ای
IT2-TSK-FIS	۲۰۰۹	زنگوله‌ای نوع ۲ بازه‌ای	پس‌انتشار	پس‌انتشار با نرخ یادگیری انطباقی	شناسایی آنلاین، پیش‌بینی سری زمانی	TSK
RSEIT2FNN	۲۰۰۹	زنگوله‌ای نوع ۲	کاهش گرادیان	فیلتر کالمن	شناسایی سیستم پویا، پیش‌بینی سری زمانی	TSK نوع ۲ بازه‌ای
IT2FNN-SVR	۲۰۱۰	زنگوله‌ای نوع ۲ بازه‌ای	خوشه‌بندی	SVM	رگرسیون، کنترل و پیش‌بینی سری زمانی	TSK

T2FINN-DE	۲۰۱۱	مثلی نوع ۲	DE	DE	شناسایی سیستم	TSK
DIT2NFS-IP	۲۰۱۲	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	حداقل مربعات بازگشتی با اولویت قواعد	شناسایی سیستم با نویز، پیش‌بینی سری زمانی	TSK مرتب صفر
eT2FIS	۲۰۱۳	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	کاهش گرادیان	شناسایی آنلاین و پیش‌بینی سری زمانی	ممدانی
TSCIT2FNN	۲۰۱۳	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	فیلتر کالمن متغیر گسترده	شناسایی سیستم، حذف نویز، پیش‌بینی سری زمانی	TSK
SIT2FNN	۲۰۱۳	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	کاهش گرادیان	شناسایی و پیش‌بینی سری زمانی	TSK
IT2NFS-SIFE	۲۰۱۴	زنگوله‌ای نوع ۲ بازه‌ای	کاهش گرادیان	حداقل مربعات بازگشتی با اولویت قواعد	شناسایی سیستم و رگرسیون	TSK

McIT2FIS	۲۰۱۴	زنگوله‌ای نوع ۲ بازه‌ای	الگوریتم یادگیری نگاشتی	الگوریتم یادگیری نگاشتی	طبقه‌بندی	TSK
McIT2FIS	۲۰۱۵	زنگوله‌ای نوع ۲ بازه‌ای	فیلتر کالمن توسعه‌یافته	فیلتر کالمن توسعه‌یافته	پیش‌بینی و برآورد	TSK
eT2ELM	۲۰۱۶	زنگوله‌ای چن نوع ۲ بازه‌ای	تصادفی	حداقل مربعات بازگشتی تعمیم‌یافته	طبقه‌بندی	TSK

۲-۶ سیستم‌های عصبی فازی عمیق

در سال‌های اخیر، پژوهش‌های مبتنی بر یادگیری عمیق نظیر شبکه‌های عصبی عمیق DNN^۱ به شدت مورد توجه قرار گرفته است. DNN یک کلاس از مدل‌های قدرتمند یادگیری ماشین است که از لایه‌های متوالی واحدهای پردازش غیرخطی استفاده می‌کنند. این شبکه‌ها می‌تواند در تعامل با داده‌های گسترده و پیچیده به خوبی رفتار کنند. به سبب ساختار عمیق در این شبکه‌ها، بهینه‌سازی و محاسبات هزاران یا میلیون‌ها پارامتر در فرآیند آموزش آنها طی می‌شود. بنابراین فرآیند این شبکه‌ها از لحاظ محاسباتی کاملاً پیچیده و فشرده است. مدل‌های DNN به دلیل عدم شفافیت ناشی از ذات جعبه سیاه^۲ خود مورد انتقاد واقع می‌شوند. همچنین، در آموزش شبکه‌های عمیق معمولاً از روش‌های بهینه‌سازی نظیر کاهش گرادیان استفاده می‌شود که دست‌یابی به عملکرد بهینه را تضمین نمی‌کنند. از سوی دیگر، روش‌های یادگیری عمیق اغلب به نویز در داده‌ها حساس هستند و در شرایطی که داده‌ها ناقص هستند به خوبی عمل نمی‌کنند.

1. Deep Neural Networks

2. Black-Box

یک روش جایگزین و در عین حال کمتر مورد بررسی قرار گرفته در بهبود عملکرد یادگیری عمیق، استفاده از سیستم‌های فازی است. سیستم‌های فازی به ما این امکان را می‌دهد که با عدم قطعیت در داده‌ها کار کنیم. همان‌طور که در فصل‌های قبل مشاهده کردید، تنوع زیادی از ترکیب سیستم‌های فازی و شبکه‌های عصبی وجود دارد که به مدل‌های گسترده‌ای نیز منجر شده است. حال با برجسته‌تر شدن یادگیری عمیق در سال‌های اخیر، ظرفیت خوبی برای پژوهشگران ایجاد شده تا ساختارهای قاعده‌محور و تفسیرپذیر سیستم‌های استنتاج فازی و منطق فازی را با شبکه‌های عصبی عمیق ترکیب کرده و مدل‌هایی موسوم به سیستم‌های عصبی فازی عمیق یا شبکه‌های عصبی فازی را توسعه دهند.

داس^۱ و همکاران (۲۰۲۰) و تالپور^۲ و همکاران (۲۰۲۲a، ۲۰۲۲b) در مقاله‌های مروری خود به بررسی ظرفیت، نوع و کاربرد ترکیب‌های سیستم‌های فازی و شبکه‌های عصبی عمیق پرداخته‌اند.

براساس تقسیم‌بندی داس و همکاران (۲۰۲۰)، ترکیب شبکه‌های عصبی عمیق و منطق فازی به دو دسته اصلی تقسیم می‌شوند. در دسته اول، منطق فازی به عنوان یک بخش از مدل‌های یادگیری عمیق یکپارچه می‌شوند و در دسته دوم، یادگیری عمیق و منطق فازی در قالب یادگیری جمعی^۳ ظاهر می‌شوند:

۱. مدل‌های یکپارچه

- ماشین بولتزن محدود شده فازی
- ماشین بولتزن عمیق فازی فیثاغورثی
- خودرمنگار انباشته آموزش دیده با منطق فازی
- شبکه فازی عمیق تاکاگی-سوگنو

۲. مدل‌های جمعی

- مدل‌های متوالی

– مدل‌هایی با ورودی فازی

1. Das, R.

2. Talpur, N.

3. Ensemble learning

* یادگیری عمیق با ویژگی‌های فازی

* شبکه‌های عصبی کانولوشنی عمیق

* ماشین بولتزمن محدودشده با ورودی‌های فازی

* شبکه باور عمیق با ورودی‌های فازی دانه‌ای

- مدل‌هایی با خروجی فازی

* شبکه باور عمیق با خروجی‌های فازی

• مدل‌های موازی

- شبکه عصبی عمیق فازی سلسله‌مراتبی

در مطالعه دیگری که توسط تالپور و همکاران (۲۰۲۲a) بر روی ترکیب‌های جاری شبکه‌های عصبی عمیق و سیستم‌های استنتاج فازی صورت گرفت، پس از بررسی مدل‌های موجود در ادبیات موضوع، سه نوع ساختار معرفی شده است: متوالی، موازی و مشارکتی:

- در مدل متوالی، یک جریان خطی را شاهد هستیم که در آن هر بخش از اطلاعات خروجی بخش پیش از خود استفاده می‌کند. در این ساختار، سیستم استنتاج فازی می‌تواند پیش از شبکه عصبی قرار گیرد تا تبدیل ورودی‌های قطعی به متغیرهای زبانی فازی را بر عهده داشته باشند و بتواند ورودی شبکه عصبی عمیق را تأمین کند. از طرف دیگر، سیستم استنتاج فازی می‌تواند بعد از شبکه عصبی عمیق به کار گرفته شود تا خروجی تفسیرپذیر نهایی را تولید کند.
- مدل موازی از سازوکار یادگیری مشترک^۱ استفاده می‌کند که در آن اطلاعات به‌طور مجزا توسط سیستم استنتاج فازی و شبکه عصبی عمیق پردازش شده و خروجی نهایی به کمک یکپارچه کردن خروجی این دو سیستم حاصل می‌شود.
- در مدل مشارکتی از سیستم استنتاج فازی برای پردازش ورودی‌ها و خروجی‌های مدل استفاده می‌شود. مقادیر ورودی به مقادیر فازی تبدیل شده و سپس توسط شبکه عصبی عمیق مورد پردازش قرار می‌گیرد. در نهایت، خروجی‌های قطعی به کمک نافازی‌سازی توسط سیستم استنتاج فازی خروجی حاصل می‌شوند. در ساختاری دیگر از مدل مشارکتی، پردازش اطلاعات ورودی

1. Joint learning

ابتدا توسط شبکه عصبی عمیق چند لایه انجام شده که منجر به ساخت یک سیستم استنتاج فازی می‌شود و خروجی سیستم فازی نیز به اصلاح پارامترهای شبکه عصبی می‌انجامد.

دسته‌بندی داس^۱ و همکاران (۲۰۲۰) و تالپور^۲ و همکاران (۲۰۲۲a) در مدل‌های متوالی و موازی به یکدیگر شباهت دارد. براساس تقسیم‌بندی موجود در این کتاب، این مدل‌ها را می‌توان در مدل‌های مشارکتی و همزمان جای داد. از سوی دیگر، مدل‌های یکپارچه معرفی شده در بالا نیز مشابه مدل‌های هیبریدی هستند. جزئیات بیشتر درباره ساختار این مدل‌ها و نیز روند یادگیری و الگوریتم‌های آموزش متناسب با آنها را می‌توانید در مقالات پایه معرفی شده در مقالات مروری فوق بیابید.

1. Das, R.

2. Talpur, N

منابع

- [1] Abraham, A. (2005). Adaptation of fuzzy inference system using neural learning. In *Fuzzy Systems Engineering: Theory and Practice* (pp. 53-83). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [2] Abraham, A. & Nath, B. (2000). Designing optimal neuro-fuzzy architectures for intelligent control. In *6th International Conference on Control, Automation, Robotics and Vision*.
- [3] Adeli, H. & Hung, S. L. (1994). *Machine learning: neural networks, genetic algorithms, and fuzzy systems*. John Wiley and Sons, Inc.
- [4] Arnould, T. & Tano, S. I. (1994, June). Definition and formulation of backward-reasoning with fuzzy If... Then... rules. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference* (pp. 864-869). IEEE.
- [5] Barto, A. G., Sutton, R. S. & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5), 834-846.
- [6] Berenji, H. R. (1992). A reinforcement learning—based architecture for fuzzy logic control. *International Journal of Approximate Reasoning*, 6(2), 267-292.
- [7] Berenji, H. R. & Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements, *IEEE Transactions on Neural Networks*, 3(5), 724-740.
- [8] Berenji, H. R. & Khedkar, P. (1993, March). Clustering in product space for fuzzy inference. In *[Proceedings 1993] Second IEEE International Conference on Fuzzy Systems* (pp. 1402-1407). IEEE.
- [9] Bezdek, J. C., Tsao, E. K. & Pal, N. R. (1992, March). Fuzzy Kohonen clustering networks. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems* (pp. 1035-1043). IEEE.
- [10] Carpenter, G. A. & Gjaja, M. N. (1993). *Fuzzy ART choice functions* (pp. 713-722). Boston, MA, USA: Boston University, Center for Adaptive Systems and Department of Cognitive and Neural Systems.
- [11] Carpenter, G. A. & Grossberg, S. (1992). A self-organizing neural network for supervised learning, recognition, and prediction. *IEEE Communications Magazine*, 30(9), 38-49.
- [12] Carpenter, G. A., Grossberg, S. & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural networks*, 4(6), 759-771.

- [13] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H. & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on neural networks*, 3(5), 698–713.
- [14] Cordon, O. & Herrera, F. (2001). Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems. *Fuzzy sets and systems*, 118(2), 235–255.
- [15] Das, R., Sen, S. & Maulik, U. (2020). A survey on fuzzy deep neural networks. *ACM Computing Surveys (CSUR)*, 53(3), 1–25.
- [16] González Muñoz, A. & Herrera Triguero, F. (1997). Multi-stage genetic fuzzy systems based on the iterative rule learning approach. *Mathware and soft computing*. 1997 Vol. 4 Núm. 3.
- [17] Gupta, M. M. & Qi, J. (1991, July). On fuzzy neuron models. In *International Joint Conference on Neural Networks*. (Vol. 2, pp. 431–436).
- [18] Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- [19] Hu, Q. & Hertz, D. B. (1994). Fuzzy logic controlled neural network learning. *Information Sciences–Applications*, 2(1), 15–33.
- [20] Ishibuchi, H., Fujioka, R. & Tanaka, H. (1993a). Neural networks that learn from fuzzy if-then rules. *IEEE Transactions on Fuzzy Systems*, 1(2), 85–97.
- [21] Ishibuchi, H., Morioka, K. & Turksen, I. B. (1995). Learning by fuzzified neural networks. *International Journal of Approximate Reasoning*, 13(4), 327–358.
- [22] Ishibuchi, H., Tanaka, H. & Okada, H. (1993b). Fuzzy neural networks with fuzzy weights and fuzzy biases. In *IEEE international conference on neural networks* (pp. 1650–1655). IEEE.
- [23] Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4), 295–307.
- [24] Jang, J. S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665–685.
- [25] Jang, J. S. R. & Sun, C. T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE transactions on Neural Networks*, 4(1), 156–159.
- [26] Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). *Neuro-fuzzy and soft computing—a computational approach to learning and machine intelligence* [Book Review]. *IEEE Transactions on automatic control*, 42(10), 1482–1484.
- [27] Jiang, H., Kwong, C. K., Kremer, G. O. & Park, W. Y. (2019). Dynamic modelling of customer preferences for product design using DENFIS and opinion mining. *Advanced Engineering Informatics*, 42, 100969.
- [28] Jin, Y. & Jiang, J. (1999). Techniques in Neural-network-based Fuzzy System Identification and Their Application to control of complex systems. In *Fuzzy Theory Systems* (pp. 111–128). Academic Press.
- [29] Jin, Y., Jiang, J. & Zhu, J. (1995). Neural network based fuzzy identification and its application to modeling and control of complex systems. *IEEE transactions on systems, man, and cybernetics*, 25(6), 990–997.

- [30] Juang, C. F. & Lin, C. T. (1998). An online self-constructing neural fuzzy inference network and its applications. *IEEE transactions on Fuzzy Systems*, 6(1), 12-32.
- [31] Kasabov, N. (1998). Evolving fuzzy neural networks-algorithms, applications and biological motivation. *Methodologies for the conception, design and application of soft computing*, World Scientific, 1, 271-274.
- [32] Kasabov, N. K. (2013). *Evolving connectionist systems: Methods and applications in bioinformatics, brain study and intelligent machines*. Springer Science and Business Media.
- [33] Kasabov, N. K. & Song, Q. (1999). Dynamic Evolving Fuzzy Neural Networks with "m-out-of-n" Activation Nodes for On-line Adaptive Systems. Department of Information Science, University of Otago.
- [34] Kasabov, N. K. & Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE transactions on Fuzzy Systems*, 10(2), 144-154.
- [35] Keller, J. M. & Hunt, D. J. (1985). Incorporating fuzzy membership functions into the perceptron algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 693-699.
- [36] Klassen, M. S. & Pao, Y. H. (1988). Characteristics of the functional link net: a higher order delta rule net. *IEEE Proceedings of the International Conference on Neural Networks*, San Diego, 507-513.
- [37] Kosko, B. (1992) *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Learning*, Prentice-Hall, Englewood Cliffs, NJ.
- [38] Lee, H. M. & Lu, B. H. (1994, June). FUZZY BP: a neural network model with fuzzy inference. In *Proceedings of 1994 IEEE international conference on neural networks (ICNN'94)* (Vol. 3, pp. 1583-1588). IEEE.
- [39] Lin, C. T. & Lee, C. G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on computers*, 40(12), 1320-1336.
- [40] Lin, C. T. & Lee, C. G. (1994). *Neural fuzzy control systems with structure and parameter learning*. World Scientific, Singapore.
- [41] Lin, C. T. & Lee, C. G. (1996). *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Prentice-Hall, Inc.
- [42] Mizutani, E. & Jang, J. S. R. (1995, November). Coactive neural fuzzy modeling. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 2, pp. 760-765). IEEE.
- [43] Mizutani, E., Jang, J. S. R., Nishio, K. & Auslande, D. M. (1994, August). Coactive neural networks with adjustable fuzzy membership functions and their applications. In *Iizuka'94: proceedings the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing* (pp. 581-582). Fuzzy Logic Systems Institute.
- [44] Nauck, D. & Kruse, R. (1995, February). NEFCLASS - a neuro-fuzzy approach for the classification of data. In *Proceedings of the 1995 ACM symposium on applied computing* (pp. 461-465).

- [45] Nauck, D. & Kruse, R. (1996). Neuro-fuzzy classification with NEFCLASS. In Operations Research Proceedings 1995: Selected Papers of the Symposium on Operations Research (SOR'95), Passau, September 13–September 15, 1995 (pp. 294–299). Springer Berlin Heidelberg.
- [46] Nauck, D. & Kruse, R. (1999). Neuro-fuzzy systems for function approximation. *Fuzzy sets and systems*, 101(2), 261–271.
- [47] Nauck, D., Klawonn, F. & Kruse, R. (1997). *Foundations of neuro-fuzzy systems*. John Wiley and Sons, Inc.
- [48] Nayak, P. C., Sudheer, K. P., Rangan, D. M. & Ramasastri, K. S. (2004). A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, 291(1–2), 52–66.
- [49] Oyama, T., Tano, S. & Arnould, T. (1994, June). A tuning method for fuzzy inference with fuzzy input and fuzzy output. In Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference (pp. 876–881). IEEE.
- [50] Pedrycz, W. & Card, H. C. (1992, March). Linguistic interpretation of self-organizing maps. In [1992 Proceedings] IEEE International Conference on Fuzzy Systems (pp. 371–378). IEEE.
- [51] Pedrycz, W. & Rocha, A. F. (1993). Fuzzy-set based models of neurons and knowledge-based networks. *IEEE Transactions on Fuzzy Systems*, 1(4), 254–266.
- [52] Rajasekaran, S. & Vijayalakshmi Pai, G. A. (2003). *Neural networks, fuzzy logic and genetic algorithm: synthesis and applications (with cd)*. PHI Learning Pvt. Ltd.
- [53] Shihabudheen, K. V. & Pillai, G. N. (2018). Recent advances in neuro-fuzzy system: A survey. *Knowledge-Based Systems*, 152, 136–162.
- [54] Shin, Y. C. & Xu, C. (2017). *Intelligent systems: modeling, optimization, and control*. CRC press, Boca Raton, FL.
- [55] Siddique, N. & Adeli, H. (2013). *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley and Sons.
- [56] Siddique, N.H. (2002). *Intelligent control of flexible-link manipulator system*, PhD Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.
- [57] Sulzberger, S. M., Tschichold-Gurman, N. & Vestli, S. J. (1993, March). FUN: Optimization of fuzzy rule based systems using neural networks. In IEEE International Conference on Neural Networks (pp. 312–316). IEEE.
- [58] Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning*. PhD Thesis, University of Massachusetts.
- [59] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3, 9–44.
- [60] Takagi, H. (1992). Applications of neural networks and fuzzy logic to consumer products. The First International Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems, Texas, 1629–1633.

- [61] Takagi, H. (1995). Applications of neural networks and fuzzy logic to consumer products. *Industrial Applications of Fuzzy Control and Intelligent Systems*, J. Yen, R. Langari and L. Zadeh (eds), IEEE Press, Piscataway, NJ, 93–106.
- [62] Takagi, H. (1997). Introduction to fuzzy systems, neural networks, and genetic algorithms. *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*, 3–33.
- [63] Takagi, H. & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International journal of approximate reasoning*, 5(3), 191–212.
- [64] Talpur, N., Abdulkadir, S. J., Alhussian, H., Aziz, N. & Bamhdi, A. (2022a). A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods. *Neural Computing and Applications*, 1–39.
- [65] Talpur, N., Abdulkadir, S. J., Alhussian, H., Hasan, M. H., Aziz, N. & Bamhdi, A. (2022b). Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: a systematic survey. *Artificial Intelligence Review*, 1–49.
- [66] Tano, S. I., Oyama, T. & Arnould, T. (1996). Deep combination of fuzzy inference and neural network in fuzzy inference software—FINEST. *Fuzzy Sets and Systems*, 82(2), 151–160.
- [67] Tano, S. I., Oyama, T., Arnould, T., Bastian, A. & Umamo, M. (1994, June). Definition and tuning of unit-based fuzzy systems in FINEST. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference* (pp. 436–441). IEEE.
- [68] Tsoukalas, L. H. & Uhrig, R. E. (1997). *Fuzzy and neural approaches in engineering*. John Wiley and Sons, Inc.
- [69] Viharos, Z. J. & Kis, K. B. (2015). Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement. *Measurement*, 67, 126–136.
- [70] Wang, L. X. & Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks*, 3(5), 807–814.
- [71] Yager, R. R. (1994). Modeling and formulating fuzzy knowledge bases using neural networks. *Neural Networks*, 7(8), 1273–1283.
- [72] Yea, B., Konishi, R., Osaki, T. & Sugahara, K. (1994). The discrimination of many kinds of odor species using fuzzy reasoning and neural networks. *Sensors and Actuators A: Physical*, 45(2), 159–165.
- [73] Zhang, Y. Q. & Kandel, A. (1998). Compensatory neurofuzzy systems with fast learning algorithms. *IEEE Transactions on Neural Networks*, 9(1), 83–105.