

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، وجود دارد. 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. به‌منظور ارتباط مستقیم با نویسنده، نشانی faezirad@mail.um.ac.ir در دسترس است.

محمدعلی فائزی راد

زمستان ۱۴۰۱

فهرست مطالب

۱	مقدمه	۱
۶	۱-۱ ترکیب سیستم‌های عصبی و فازی	۶
۹	۲ سیستم‌های مشارکتی	۹
۱۰	۱-۲ سیستم‌های عصبی فازی مشارکتی	۱۰
۱۰	۱-۱-۲ ترکیب آبخاری	۱۰
۱۱	۲-۱-۲ نوع ابزار توسعه	۱۱
۱۶	۲-۲ سیستم‌های فازی عصبی مشارکتی	۱۶
۱۶	۱-۲-۲ شبکه عصبی به‌عنوان سازوکار اصلاحی	۱۶
۱۷	۲-۲-۲ شبکه عصبی در تعیین توابع عضویت (نوع ابزار توسعه)	۱۷
۲۴	۳-۲-۲ شبکه عصبی برای یادگیری قواعد فازی	۲۴
۳۸	۴-۲-۲ شبکه عصبی برای انتخاب ویژگی	۳۸
۳۹	۵-۲-۲ ترکیب آبخاری	۳۹
۳۹	۶-۲-۲ شبکه عصبی برای تعیین پارامتر	۳۹
۴۳	۳ سیستم‌های همزمان	۴۳

۴۷	سیستم‌های یکپارچه فازی عصبی
۴۹	۱-۴ شبکه‌های عصبی فازی با سیستم استنتاج فازی
۴۹	۱-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع ممدانی
۵۱	۲-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع تاکاگی-سوگنو
۵۴	۳-۱-۴ شبکه‌های مبتنی بر استنتاج فازی از نوع تسوکاموتو
۵۸	۲-۴ سیستم فازی مبتنی بر شبکه عصبی (شبکه Pi-Sigma)
۶۴	۳-۴ کنترل هوشمند مبتنی بر استدلال تقریبی
۶۴	۱-۳-۴ مدل ARIC
۶۹	۲-۳-۴ مدل GARIC
۷۷	۴-۴ شبکه‌های تابع پایه فازی (FBFN)
۸۰	۵-۴ شبکه فازی (FUN)
۸۳	۶-۴ ترکیب استنتاج فازی و شبکه‌های عصبی در نرم‌افزار استنتاج فازی (FINEST)
۸۶	۷-۴ کنترل‌کننده فازی عصبی (NEFCON)
۸۹	۱-۷-۴ طبقه‌بندی فازی عصبی (NEFCLASS)
۹۴	۲-۷-۴ تقریب تابع با رویکرد فازی عصبی (NEFPROX)
۹۶	۸-۴ سیستم فازی عصبی انطباقی
۹۶	۱-۸-۴ سیستم استنتاج فازی عصبی انطباقی (ANFIS)
۱۱۴	۲-۸-۴ سیستم استنتاج فازی عصبی هم‌کنش (CANFIS)
۱۱۸	۹-۴ سیستم‌های استنتاج فازی عصبی خودسازمان‌ده
۱۱۸	۱-۹-۴ شبکه کنترل یادگیری انطباقی فازی (FALCON)
۱۲۴	۲-۹-۴ شبکه استنتاج فازی عصبی خودساخت (SONFIN)
۱۲۹	۳-۹-۴ سیستم استنتاج فازی عصبی تکاملی پویا (DENFIS)
۱۳۴	۱۰-۴ بحث و بررسی

۱۴۱	۵ شبکه‌های عصبی مبتنی بر منطق فازی
۱۴۲	۱-۵ نورون‌های فازی
۱۴۳	۱-۱-۵ نورون فازی نوع یک
۱۴۸	۲-۱-۵ نورون فازی نوع دو
۱۵۰	۳-۱-۵ نورون فازی نوع سه
۱۵۳	۲-۵ فازی‌سازی مدل‌های شبکه عصبی
۱۵۳	۱-۲-۵ پرسپترون فازی
۱۶۰	۲-۲-۵ ART و ARTMAP فازی
۱۷۳	۳-۲-۵ شبکه خوشه‌بندی کوهنن فازی (FKCN)
۱۷۷	۳-۵ شبکه‌های عصبی با آموزش فازی
۱۷۷	۱-۳-۵ شبکه‌های عصبی با ورودی آموزشی فازی
۱۸۶	۲-۳-۵ شبکه‌های عصبی با پارامترهای فازی
۱۹۴	۳-۳-۵ شبکه پسانتشار فازی (Fuzzy BP)
۲۰۴	۴-۵ شبکه‌های عصبی فازی تکاملی
۲۰۹	۶ مدل‌های نوین
۲۰۹	۱-۶ ترکیب مجموعه‌های فازی نوع ۲ با شبکه‌های عصبی
۲۱۳	۲-۶ سیستم‌های عصبی فازی عمیق
۲۱۷	منابع

فصل ۱

مقدمه

منطق فازی و شبکه‌های عصبی مصنوعی ظاهراً دو حوزه مجزا از هوش محاسباتی هستند که طی سال‌های مختلف بسیار مورد توجه واقع شده‌اند. سیستم‌های مبتنی بر منطق فازی تلاش می‌کنند که با استفاده از عبارات زبانی^۱ از استدلال شبه انسانی پیروی کنند در حالی که 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 چگونه تصمیم‌گیری می‌کنند. از سوی دیگر، سیستم‌های فازی با اطلاعات محدود استنتاج کرده و به خوبی بیان می‌کنند که چگونه به این تصمیم رسیده‌اند اما نمی‌توانند قواعد مورد نیاز را به طور خودکار به دست بیاورند. 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 پردازد. 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 نقل) 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. Evolutionaty algorithms
 3. Defuzzification
 4. Siddique

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

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 ۲-۱۹ 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.

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

$\overline{u_s(\mathbf{x}_i)}$ مقدار استنتاج شده‌ای 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

فصل ۴

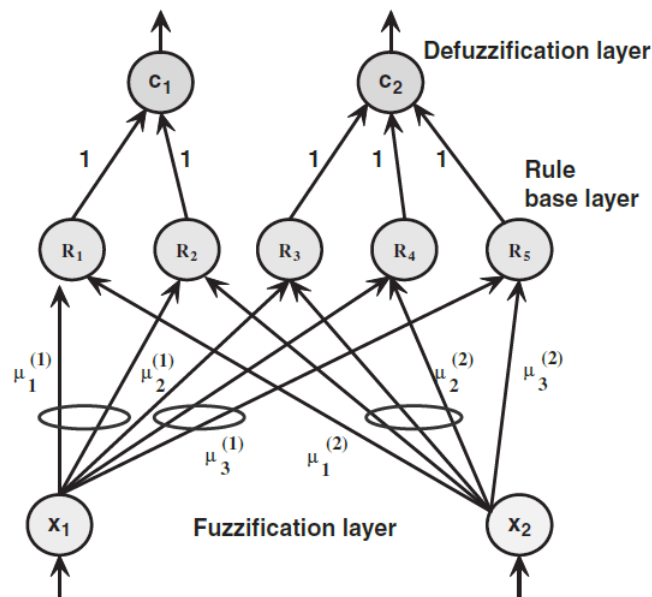
سیستم‌های یکپارچه فازی عصبی

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، به کارگیری 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 سیستم 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 می‌شوند. به منظور حداقل کردن زمان فرآیند، روش‌های پیش‌فازی‌زدایی^۴ مختلفی به کار گرفته

1. Hybrid Neuro-Fuzzy Systems
2. Integrated Neuro-Fuzzy Systems
3. Fused Neuro-Fuzzy Systems
4. Pre-defuzzification

یادگیری خیلی ساده‌تر بهره می‌گیرد که در آن از چندی‌سازی برداری^۱ در پیدا کردن قواعد (خوشه‌ها) استفاده نشده و همچنین در آن به اطلاعات گرادیان برای آموزش توابع عضویت نیازی نیست. مزیت‌های دیگری همچون قابلیت تفسیرپذیری، امکان مقداردهی اولیه (با دانش قبلی) و سهولت آن نیز وجود دارد.

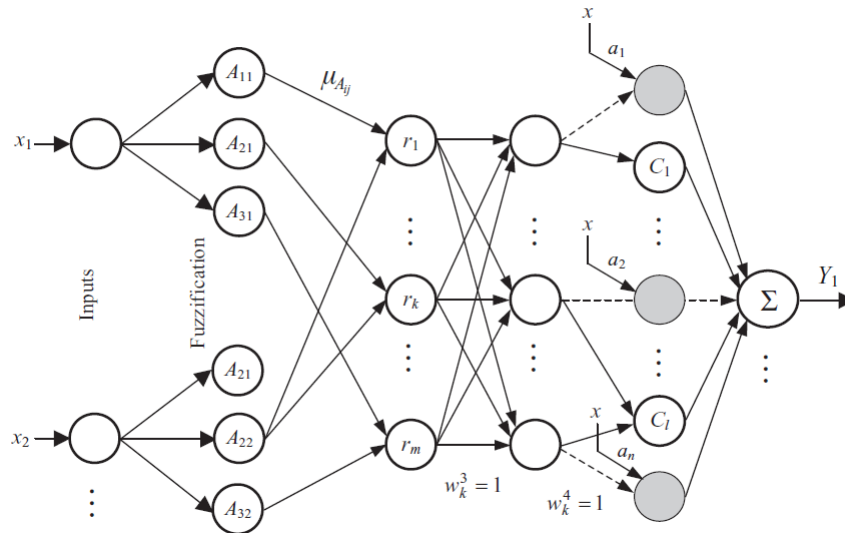


شکل ۴-۱۴: ساختار NEFCLASS

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.

1. vector quantization



شکل ۴-۲۴: ساختار SONFIN

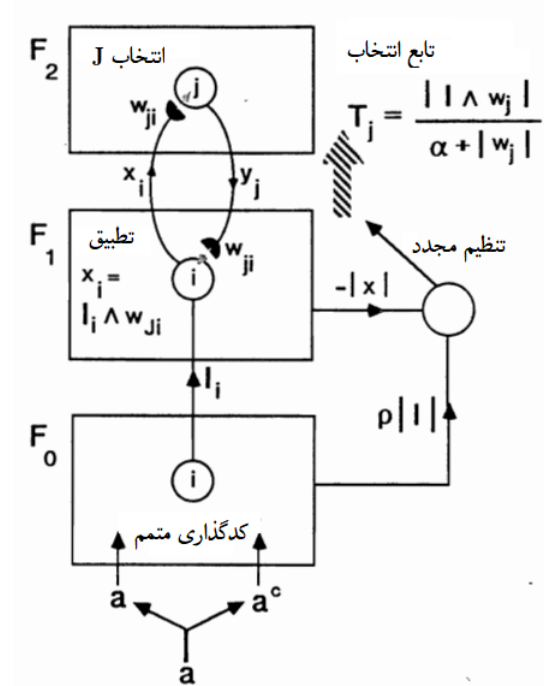
لایه ۱: هر گره در این لایه، متناظر یک متغیر ورودی است. این گره‌ها هیچ فرآیندی را دربردارند و فقط مقادیر ورودی را به لایه بعدی منتقل می‌کنند؛ یعنی:

$$x_i = f(x_i) \quad , \quad i = 1, 2, \dots, n \quad (۴-۱۲۸)$$

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:

$$\mu_{ji}(x_i) = f\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji}}\right) \quad (۴-۱۲۹)$$

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. تعداد توابع عضویت را برای ورودی x_i مشخص می‌کند. $f(\cdot)$ نشان‌گر تابع فعال‌سازی انتخاب شده برای گره‌ها است. نکته مهمی که باید یادآور شویم این است که ضرورتی ندارد توابع عضویت برای هر



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

در ادامه، تشریح الگوریتم ARTMAP فازی (FARTMAP) طی گام‌های زیر ارائه می‌شود.

الگوریتم FARTMAP

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: $I = A = (a, a^c)$ و $I = B = (b, b^c)$. هریک از ART_a و ART_b 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 $w_j^a = (w_{j1}^a, w_{j2}^a, \dots, w_{j,m_a}^a)$ 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 $w_k^b = (w_{k1}^b, w_{k2}^b, \dots, w_{k,m_b}^b)$ 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 $w_j^{ab} = (w_{j1}^{ab}, w_{j2}^{ab}, \dots, w_{j,n_b}^{ab})$ نمایش‌دهنده بردار وزن j از F_2^a به F_{ab} است. در میان ارائه ورودی‌ها، مقادیر x^a ، y^a ، x^b ، y^b و x^{ab} 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:

۲- اگر $w_j \geq 0$ و $w_{ji} < 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^U - \delta_p^U w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^L \quad (۶۷-۵)$$

۳- اگر $w_j < 0$ و $w_{ji} \geq 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^U - \delta_p^U w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^L \quad (۶۸-۵)$$

۴- اگر $w_j < 0$ و $w_{ji} < 0$ باشد، آنگاه:

$$\frac{\partial E_{ph}}{\partial w_{ji}} = -\delta_p^L w_j y_{pj}^U (1 - y_{pj}^U) y_{pi}^L - \delta_p^U w_j y_{pj}^L (1 - y_{pj}^L) y_{pi}^U \quad (۶۹-۵)$$

در توضیحات فوق، $t_p^L \triangleq [T_p]_h^L$ ، $t_p^U \triangleq [T_p]_h^U$ ، $y_p^L \triangleq [Y_p]_h^L$ و $y_p^U \triangleq [Y_p]_h^U$ است. همچنین، δ_p^L و δ_p^U در قالب $(t_p^L - y_p^L) y_p^L (1 - y_p^L)$ و $(t_p^U - y_p^U) y_p^U (1 - y_p^U)$ تعریف می‌شود. علاوه‌براین با استفاده از تابع هزینه زیر می‌توان شبکه را با مقادیر مختلفی از h آموزش داد.

$$E_p = \sum_h h \cdot E_{ph} \quad (۷۰-۵)$$

$$= \sum_h h \left[\frac{1}{\gamma} \left([T_p]_h^L - [Y_p]_h^L \right)^2 + \left(\frac{1}{\gamma} [T_p]_h^U - [Y_p]_h^U \right)^2 \right] \quad (۷۱-۵)$$

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.

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