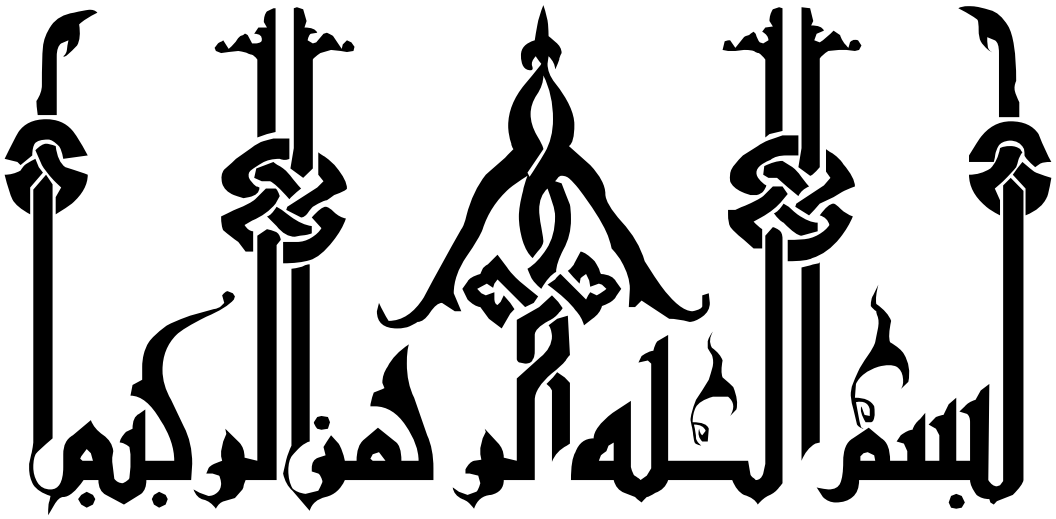

بسم الله الرحمن الرحيم

دانشکده کامپیوتر و فناوری اطلاعات

گروه کامپیوتر و هوش مصنوعی

استفاده از شبکه عصبی برای استخراج دانش در تحلیل داده‌های چند بعدی

دانشجو : مجتبی شهابی

استاد راهنما :

دکتر حمید حسن‌پور

استاد مشاور

دکتر هدی مشایخی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

اسفند ۱۳۹۴

تقدیم به

تقدیم با بوسه بر دستان پدرم:

به او که نمی دانم از بزرگی اش بگویم یا مردانگی، سخاوت، سکوت، مهربانی و . . .

پدرم راه تمام زندگی است

پدرم دلخوشی همیشگی است

تقدیم به مادر عزیزتر از جانم:

مادرم هستی من ز هستی توست، تا هستم و هستی دارمت دوست

گلسار جاودانی مادر است

چشمه سار مهربانی مادر است

تقدیر و تشکر

تحسین سپاس و ستایش از آن خداوندی است که بنده کوچکش را در دریای بیکران اندیشه، قطره‌ای ساخت تا وسعت آنرا از دریچه اندیشه‌های ناب آموزگارانی بزرگ به تماشا نشیند. لذا اکنون که در سایه سار بنده نوازی‌هایش پایان نامه حاضر به انجام رسیده است، بر خود لازم می‌دانم تا مراتب سپاس را از بزرگوارانی به جا آورم که اگر دست یاریگرشان نبود، هرگز این پایان نامه به انجام نمی‌رسید.

ابتدا از استاد گرانقدرم جناب آقای دکتر حمید حسن پور که زحمت راهنمایی این پایان نامه را بر عهده داشتند، کمال سپاس را دارم.

سپس از استاد عالی قدرم سرکار خانم دکتر مهدی مشایخی که زحمت مشاوره این پایان نامه را متحمل شدند صمیمانه تشکر می کنم.

همچنین از سرکار خانم نسترن دهقان که در طول این پروژه مرا یاری داده و با کمک ها و راهنمایی های خود موجب به ثمر رسیدن این پروژه شدند نهایت تشکر و قدردانی را دارم.

سپاس آخر را به مهربان ترین همراهان زندگی ام، به پدر و مادر عزیزم تقدیم می کنم که حضورشان در فضای زندگی ام مصداق بی ریای سخاوت بوده است.

مجتبی شهابی

اسفند ۱۳۹۴

تعهد نامه

اینجانب **مجتبی شهابی** دانشجوی دوره کارشناسی ارشد رشته **هوش مصنوعی** دانشکده **کامپیوتر و فناوری اطلاعات** دانشگاه صنعتی شاهرود نویسنده پایان نامه ***استفاده از شبکه‌های عصبی برای استخراج دانش در تحلیل داده‌های چند بعدی*** تحت راهنمائی **دکتر حمید حسن‌پور** متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توســـط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به د ست آمدن نتایج ا صلی پایان نامه تأثیرگذار بوده اند در مقالات م ستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) ا ستفاده شده ا ست ضوابط و ا صول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

*** متن این صفحه نیز باید در ابتدای نسخه های تکثیر شده پایان نامه وجود داشته باشد .**

چکیده

در دهه اخیر استفاده از تکنیک‌های هوش مصنوعی در حل مسائل علمی افزایش چشمگیری داشته است. شبکه‌های عصبی مصنوعی در مسائل دسته‌بندی و شناسایی الگو استفاده فراوانی دارند. یکی از بزرگ‌ترین مشکلات شبکه‌های عصبی مصنوعی ضعف در تفسیر نتایج خویش است. اگرچه شبکه‌های عصبی مصنوعی قادرند عملیات دسته‌بندی را با دقت بالایی انجام دهند، اما عدم شفافیت دانش ارائه شده توسط شبکه عصبی باعث می‌شود که نتوان نتایج آن را به راحتی در سایر سیستمها استفاده نمود. بدین منظور محققین سعی دارند تا با استخراج قوانین از شبکه عصبی این مشکل را رفع نمایند. امروزه روش‌های استخراج قانون متعددی وجود دارد که اغلب قادرند عملیات انجام شده توسط شبکه عصبی مصنوعی بر روی یک مجموعه داده ساده را در قالب یک مجموعه قانون ساده ارائه نمایند. اما این روش‌ها معمولاً در مقابل مسائل نسبتاً پیچیده از دقت پایینی برخوردار بوده و دقت بدست آمده از قوانین قابل مقایسه با نتایج شبکه عصبی نیست.

در این پایان‌نامه روش جدیدی به‌منظور استخراج قانون از شبکه عصبی مصنوعی تحت عنوان روش *استخراج قانون چهارمرحله‌ای* (FSRE) معرفی شده است. این روش قادر است دسته‌بندی انجام شده بر روی یک مجموعه داده را به مجموعه‌ای ساده، مفید و کارا از قوانین گزاره‌ای تبدیل کند. در استفاده از این روش، محدودیتی در پیچیدگی داده‌ها وجود ندارد. برای بررسی عملکرد و نتایج این روش، از چند مجموعه داده رایج در عملیات دسته‌بندی (برگرفته از پایگاه داده UCI) در این پایان‌نامه استفاده شده است. نتایج بدست آمده نشان داد که روش پیشنهادی نسبت به روش‌های موجود از دقت و کارایی بیشتری در استخراج قانون برخوردار است.

کلمات کلیدی: *شبکه عصبی مصنوعی، استخراج قانون، دسته‌بندی، داده کاوی*

لیست مقالات مستخرج از پایان نامه

- Mojtaba Shahabi, Hmid Hassanpour, and Hoda Mashayekhi. "*Rule Extraction for Fatty Liver Detection using Neural Networks*". Journal of Advances in Computer Research (JACR). (to be submitted)
- Mojtaba Shahabi & Hamid Hassanpour. "*Using the Artificial Intelligence Techniques for Diagnosing of Intensity of Non-Alcoholic Fatty Liver Disease by Clinical Parameters*". Journal of Knowledge and Health. Submitted in January 20, 2016
- Mojtaba Shahabi, Hamid Hassanpour, and Hoda Mashayekhi. "*Rule Extraction Using Neural Network for Data Classification*". Journal of Neurocomputing. (to be submitted)

فهرست مطالب

فصل اول: آشنایی با استخراج قانون

فصل دوم: روش‌های استخراج قانون به کمک شبکه عصبی

فصل سوم: روش استخراج قانون چهار مرحله‌ای

فصل چهارم: بررسی و تحلیل نتایج

فصل پنجم: کاربرد روش پیشنهادی در تشخیص بیماری کبد چرب

فصل ششم: جمع‌بندی و پیشنهادات

پیوست‌ها

مراجع

فهرست شکل‌ها

فصل اول: آشنایی با استخراج قانون

فصل دوم: روش‌های استخراج قانون به کمک شبکه عصبی

فصل سوم: روش استخراج قانون چهار مرحله‌ای

فصل چهارم: بررسی و تحلیل نتایج

فصل پنجم: کاربرد روش پیشنهادی در تشخیص بیماری کبد چرب

پیوست‌ها

فهرست جدول‌ها

فصل اول: آشنایی با استخراج قانون

فصل چهارم: بررسی و تحلیل نتایج

فصل پنجم: کاربرد روش پیشنهادی در تشخیص بیماری کبد چرب

فصل اول

آشنایی با استخراج قانون

۱-۱ مقدمه

امروزه استفاده از شبکه‌های عصبی مصنوعی[1] در عرصه‌های گوناگون مانند مسائل تجاری [۱]، صنعتی [۲] و پزشکی [۳] پیشرفت فراوانی داشته است. این پیشرفت به سبب وجود دقت بالای این ابزار در عملیاتی همچون دسته‌بندی، تخمین توابع و پیش‌بینی سری زمانی است؛ که در مقابل دیگر ابزارهای موجود مانند درخت تصمیم[2]، جدول تصمیم[3] و ماشین بردار پشتیبان[4] معمولاً نتایج ایده‌آل‌تری را تولید می‌نماید [۴،۵]. در کنار مزایای این روش، عواملی مانند درک پایین انسان نسبت به عملیات انجام شده درون شبکه عصبی و نحوه دست‌یابی به نتایج باعث می‌شوند که این ابزار در شرایطی که از حساسیت یا اهمیت بالایی برخوردار است، کاربرد چندانی نداشته باشد. بدین‌ترتیب روش‌های استخراج قانون جهت استفاده از دانش بدست آمده شبکه عصبی در ایجاد سیستم‌های خبره[5] ابداع گردیدند. در این فصل به معرفی مختصر شبکه‌های عصبی مصنوعی و آشنایی با مبحث استخراج قانون پرداخته می‌شود.

۱-۲ شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی از پرکاربردترین و عملی‌ترین روش‌های مدلسازی جهت مسائل بزرگ و پیچیده هستند که گاهی این مسائل شامل صدها پارامتر می‌باشند [۶،۷]. امروزه انواع گوناگونی از شبکه‌های عصبی مصنوعی وجود دارند که در کاربردهای مختلفی مورد استفاده قرار می‌گیرند. شبکه‌های عصبی چند لایه پرسپترون[6] رایج‌ترین نوع شبکه‌های عصبی بوده که در مسائلی مانند کلاس‌بندی (که در آن خروجی شبکه بیان‌گر شماره یک کلاس است) یا رگرسیون (که در آن خروجی شبکه بیان‌گر یک مقدار عددی است) کاربرد فراوانی دارد [۸].

---

[1] Artificial neural network (ANN)
[2] Decision tree
[3] Decision table
[4] Support vector machine
[5] Expert System
[6] Multi-layer perceptrop (MLP)

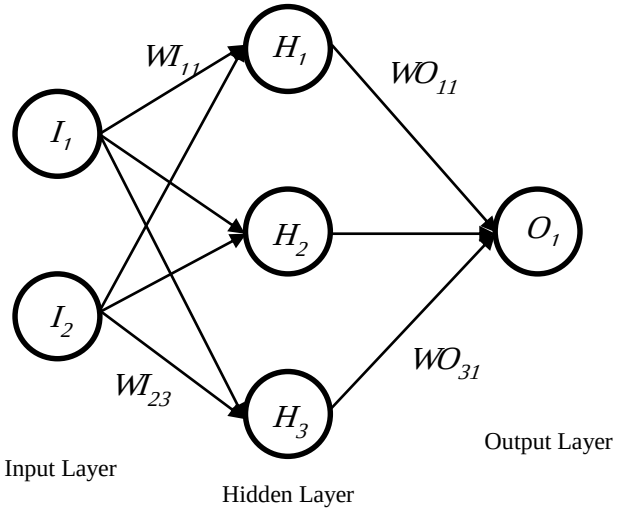
۱-۲-۱ ساختار

هر شبکه عصبی پرسپترون چندلایه شامل یک لایه ورودی، یک لایه خروجی و یک (یا چند) لایه مخفی است. هر لایه شامل تعدادی نرون[1] می‌باشد که وظایف این نرون‌ها بسته به لایه قرار گرفته در آن متفاوت است. هر نرون در لایه ورودی بیانگر یک ویژگی مسئله می‌باشد؛ که با توجه به مسئله موجود و تعداد ویژگی‌های آن، تعداد نرون‌های این لایه مشخص می‌شود. نرون‌های لایه خروجی معرف خروجی مسئله می‌باشند؛ در حالت عادی معمولاً برای عملیاتی مانند رگرسیون تعداد نرون‌های این لایه برابر ۱ عدد بوده و برای عملیاتی مانند دسته‌بندی، تعداد نرون به تعداد دسته‌های خروجی مسئله می‌باشد [۹].

لایه مخفی در این شبکه دارای اهمیت بالایی می‌باشد، بطوریکه تعیین تعداد این لایه و تعداد نرون‌های موجود در هر لایه، در پاسخ بدست آمده از شبکه تأثیر مستقیمی دارد. در مدل استاندارد شبکه عصبی چندلایه، تنها یک لایه مخفی وجود دارد (شکل (۱-۱)) ولی در حالت کلی تعداد لایه مخفی در شبکه عصبی متناسب با میزان پیچیدگی[2] مسئله می‌باشد.

نرون‌های موجود در لایه مخفی برای جداسازی داده‌ها در عملیات مورد نظر استفاده می‌شوند و تعداد این نرون‌ها بسته به میزان پیچیدگی مسئله، متغیر است [۱۰]. در واقع در هر لایه مخفی، یک پیش‌پردازش بر روی داده‌ها انجام شده تا بتوان در لایه بعدی عملیات مورد نظر را بهتر انجام داد. به‌غیر از لایه ورودی (که هر نرون تنها یک مقدار ورودی دارد) در دیگر لایه‌ها، پس از دریافت مقادیر از نرون‌های لایه قبل، یک تابع فعال‌ساز[3] اجرا شده و خروجی نرون توسط این تابع تعیین می‌گردد.

[1] Neuron
[2] Complexity
[3] Activation function



شکل (۱-۱) یک شبکه عصبی چند لایه نمونه با ساختار استاندارد سه لایه.

در یک شبکه عصبی چند لایه بین نرون‌های موجود در یک لایه هیچ ارتباط خاصی وجود ندارد، درحالیکه بین نرون‌های لایه‌های متوالی ارتباط کامل[۱] برقرار است. ارتباط بین هر دو نرون توسط یک یال مشخص می‌شود که پس از فاز آموزش، به هر یال یک وزن اختصاص می‌یابد.

یک شبکه عصبی پس از تعیین ساختار مورد نیاز، توسط مجموعه نمونه‌های آموزشی موجود و روش تعیین شده (مانند روش انتشار به عقب[۲])، آموزش دیده و کارایی آن ارزیابی می‌شود. برای تعیین کارایی شبکه از پارامترهایی مانند میانگین مربعات خطا[۳] یا مجموع مربعات خطا[۴] استفاده می‌شود. سپس می‌توان از این شبکه آموزش دیده برای نمونه‌های جدید استفاده کرده و پاسخ بهینه را برای این نمونه‌ها بدست آورد [۱۱].

[1] Full-joint
[2] Back-propagation
[3] Mean square error (MSE)
[4] Sum square error (SSE)

۱-۲-۲ مزایای استفاده از شبکه عصبی

این ابزار قادر است در شرایطی که هیچ راه‌حل شناخته شده‌ای برای مسئله وجود ندارد، پاسخ بهینه‌ای تولید نماید. همچنین استفاده از این ابزار در زمینه‌های مختلفی مانند دسته‌بندی، خوشه‌بندی، تخمین توابع، پیش‌بینی سری زمانی و بهینه‌سازی به‌صورت مکرر انجام می‌شود. یکی دیگر از مزیت‌های این ابزار استفاده از آن برای داده‌های حجیم و مجموعه داده‌های بزرگ می‌باشد. همچنین این ابزار بدلیل ساختار شبکه‌ای که دارد، به‌صورت توزیع شده بر روی چندین سیستم می‌تواند اجرا شود، و بدلیل انعطاف‌پذیری[1] که در این حالت وجود دارد درصورت خرابی یکی از سیستم‌ها (نرون‌ها)، شبکه عصبی باز هم قادر به انجام ادامه عملیات می‌باشد [۱۲].

۱-۲-۳ مسائل موجود در استفاده از این ابزار

برای استفاده از این ابزار، ابتدا به یک مجموعه داده جهت انجام فاز آموزش نیاز است. اندازه این مجموعه نسبت به مسئله موجود و سطح پیچیدگی آن می‌تواند متفاوت باشد، اما معمولاً وجود مجموعه بزرگ‌تر برای فاز آموزش، موجب آموزش بهتر شبکه عصبی می‌گردد. از طرفی حجم بیش از اندازه داده‌های آموزش منجر به طولانی شدن عملیات در این مرحله می‌شود. تعیین نحوه مناسب نمایش داده‌ها[2] و گسسته‌سازی[3] مقادیر (البته درصورت نیاز) تاثیر بسزایی در نتایج بدست آمده از این ابزار دارد. مرحله بعدی، تعیین ساختار مناسب شبکه برای مسئله مورد نظر است؛ البته قسمتی از این ساختار (ورودی‌ها و خروجی‌ها) با تعیین نحوه نمایش داده‌ها قابل تعیین هستند. اما تعیین تعداد لایه‌های مخفی و تعداد نرون‌های موجود در هر لایه کار دشواری می‌باشد. برای هر مسئله، با توجه به سطح پیچیدگی آن، ساختار شبکه مورد نیاز متفاوت بوده و معیار دقیقی برای تعیین آن وجود ندارد. دقت خروجی دریافت شده از شبکه به دقت ساختار آن وابسته است [۱۲]. یکی دیگر از مشکلات این ابزار ابهام در روش

[1] Flexibility
[2] Data representation
[3] Discretization level

دست‌یابی به نتایج و چگونگی تولید خروجی می‌باشد که باعث عدم امکان استناد به نتایج آن در برخی از ساختارها می‌شود [۹].

۱-۳ استخراج و ارائه قانون

همانطور که پیشتر بیان شد، یکی از اصلی‌ترین مشکلات شبکه‌های عصبی، مبهم بودن و غیر قابل فهم بودن نحوه دستیابی به نتایج می‌باشد. به همین دلیل در اکثر مسائل حساس و مهم مانند مسائل پزشکی، و تشخیص و پیش‌بینی بیماری، استفاده چندانی از این روش نمی‌شود. در واقع برای کاربران، این ابزار به‌صورت یک جعبه سیاه شناخته می‌شود که با دریافت یک مجموعه ورودی، خروجی مناسب را تولید می‌کند [۱۳].

این ابهام موجب بوجود آمدن خطر[1] در استفاده از این ابزار می‌شود که در موارد حساس، بسیاری از کاربران این خطر را نمی‌پذیرند. برای حل این مشکل روش‌هایی تحت عنوان استخراج قانون[2] از شبکه عصبی [۱۳،۹] ابداع شده است. در این گونه روش‌ها، سعی می‌شود تا با استفاده از یک سری قوانین عمدتاً ساده، چگونگی انجام عملیات توسط شبکه عصبی را تشریح کنند.

۱-۳-۱ قانون چیست؟

فرض کنید جدولی از نمونه‌ها با سه ویژگی در ورودی و یک کلاس خروجی به‌صورت جدول (۱-۱) وجود دارد. درصورتیکه بخواهیم این داده‌ها را در یک قالب مناسب و قابل فهم برای کاربردهای روزمره بیان کنیم؛ اولین روشی که به ذهن می‌رسد این است که هر نمونه را به‌صورت یک شرط بیان کنیم (مانند رابطه (۱-۱))

[1] Risk
[2] Rule extraction

جدول (۱-۱) مجموعه داده‌های آزمایشی جهت استخراج قانون

No	X	Y	Z	Class
1	α_1	β_2	γ_3	Class 1
2	α_3	β_5	γ_9	Class 2
3	α_4	β_7	γ_5	Class 1
4	α_8	β_{11}	γ_6	Class 3

$$(X = \alpha_i)\ \&\ (Y = \beta_j)\ \&\ (Z = \gamma_k) \rightarrow Class\ l \qquad (۱-۱)$$

اما مشکلی که در این روش وجود دارد این است که تعداد شرایط (قانون) به تعداد نمونه‌ها زیاد خواهد شد و این تعداد زیاد، باعث کاهش میزان درک از داده‌ها می‌شود. از طرفی این روش قادر به ارائه پاسخ مناسب برای نمونه جدید نمی‌باشد (نمونه‌ای که در جدول وجود ندارد).

پس روشی کارا برای بیان چنین داده‌هایی به‌صورت ساده و مفهومی مورد نیاز است؛ تا برای بازه‌ای از مقادیر (و نه یک مقدار خاص) پاسخ مورد نظر را تولید کند. اینکار توسط قوانین انجام می‌شود.

۱-۳-۲ انواع قانون

مسائل و مشکلات گوناگون معمولاً نیازمند قوانین متنوعی هستند؛ همچنین روش‌های مختلف قوانین متفاوتی را تولید می‌نمایند. برخی از متداول‌ترین انواع قوانین موجود عبارتنداز:

- *قوانین گزاره‌ای*[1] [۱۴]: این نوع قوانین متداول‌ترین نوع قوانین استفاده شده در کاربردهای مختلف بوده و به‌صورت $If \ldots Then \ldots Else \ldots$ نمایش داده می‌شود. رابطه (۱-۲) نمونه‌ای از این قانون را برای مسئله XOR دو بیتی نشان می‌دهد.

[1] Propositional rules

$$if\ (X_1 = 1\ and\ X_2 = 0)\ or\ (X_1 = 0\ and\ X_2 = 1)\ then\ Odd\ parity$$
$$else\ Even\ parity \qquad (۲-۱)$$

- *قوانین N از M* [۱۵]: در این نوع قانون که همانند قوانین گزاره‌ای معمولاً برای عملیات دسته‌بندی استفاده می‌شود، برای یک دسته خاص، ابتدا تعداد N شرط (بدون هیچ اولویت یا ترتیبی) در نظر گرفته می‌شود. سپس درصورتیکه Mتای آنها برای یک نمونه ورودی درست باشد، این نمونه در دسته مورد نظر قرار می‌گیرد. به‌عنوان مثال برای مسئله XOR دو بیتی این قانون به‌صورت رابطه (۱-۳) نمایش داده می‌شود.

$$if\ (exactly\ 1\ of\ 2\ inputs\ is\ true)\ then\ Odd\ parity$$
$$else\ Even\ parity \qquad (۳-۱)$$

- *قوانین فازی*[1] [۱۶]: در این نوع قانون از متغیرهای زبانی[2] مانند خیلی کم، کم، متوسط، کمی زیاد، و زیاد در گزاره‌های شرطی استفاده می‌شود؛ و معمولاً در مسائل دسته‌بندی استفاده می‌شوند. رابطه (۱-۴) نمونه‌ای از این قانون را نشان می‌دهد:

$$if\ (temp\ is\ high)\ then\ fan\ is\ turned\ on \qquad (۴-۱)$$

- *قوانین چند جمله‌ای*[3] [۱۷]: این نوع قوانین که بیشتر در مسائل ریاضی و تخمین توابع استفاده می‌شود، ضریبی از ورودی‌ها را با هم در نظر گرفته و حالتی مانند رابطه (۱-۵) دارد:

$$If\ X_1 + 2X_2 + 5X_3 \geq 100\ Then\ \ldots \qquad (۵-۱)$$

[1] Fuzzy rule
[2] Linguistic variable
[3] Oblique rule

- *قوانین سلسله مراتبی*: در این دسته از قوانین، برای هر قانون سطوح مختلفی وجود دارد و در هر سطح می‌توان از گزاره‌های شرطی مختلفی استفاده کرد. همچنین در این حالت می‌توان از ترکیبی از انواع قوانین ذکر شده استفاده نمود.

۱-۳-۳ مزایای استخراج قانون از شبکه عصبی

روش‌های مختلفی برای استخراج قانون وجود دارد که از آن جمله می‌توان به روشهای مبتنی بر شبکه-های عصبی مصنوعی، درخت تصمیم، جدول تصمیم، و ماشین بردار پشتیبان اشاره نمود. در بین روشهای موجود، شبکه‌های عصبی به سبب دارا بودن میزان دقت بالاتر [۱۳] متداول‌تر بوده و در این پایان‌نامه مورد توجه قرار گرفته است. درصورتیکه بتوان این قوانین را از شبکه‌های عصبی مصنوعی بدست آورد، می‌توان از مزایای این قوانین مانند موارد زیر استفاده نمود:

- کشف دانش موجود در مسئله مورد نظر برای حل آن؛
- استفاده از دانش بدست آمده، در زبان‌های برنامه نویسی؛
- ایجاد سیستم خبره برای حل مسئله مورد نظر؛

۱-۴ شبکه عصبی و استخراج قانون

برای انجام عملیات استخراج قانون از شبکه عصبی مصنوعی معمولاً مراحل زیر انجام می‌پذیرد:

۱-۴-۱ تعیین چگونگی ارائه داده به شبکه عصبی

این مرحله به‌عنوان حساس‌ترین مرحله از استخراج قانون شناخته می‌شود و در نتایج پایانی تاثیر بسزایی دارد. معمولاً داده‌ها به‌صورت خام به شبکه اعمال نمی‌شوند؛ بلکه عملیاتی مانند گسسته‌سازی و رقمی-کردن[1] بر روی آنها انجام می‌شود. چگونگی تعیین محدوده‌های گسسته‌ساز و تعداد این محدوده‌ها برای

[1] Digitalization

رقمی‌کردن از اهمیت بالایی برخوردار است. هرچه این محدوده‌ها بهتر بتوانند داده‌ها را از هم جدا کنند، قوانین بدست آمده نیز بهتر عمل می‌کنند [۱۸]. تعداد زیاد این بازه‌ها باعث افزایش تعداد قوانین می‌شود، از طرفی تعداد کم آنها باعث عدم دقت کافی و گاهی روی هم افتادن[1] بازه قابل اعمال یک قانون می‌شود. به همین دلیل سعی می‌شود تا تعداد و اندازه این بازه‌ها به‌صورت بهینه تعیین گردد. جدول (۱-۲) مثالی از تبدیل داده‌های ورودی به بازه‌های گسسته را نشان می‌دهد.

جدول (۱-۲) نمونه‌ای از عمل گسسته سازی داده‌ها با تعداد بیت‌های لازم

Sample No	**(X , Y)**	**X Bits**			**Y Bits**			**Class**	**Class Bits**		
1	(1 , 16)	1	0	0	0	1	0	1	1	0	0
2	(25 , 10)	0	1	0	1	0	0	2	0	1	0
3	(30 , 15)	0	0	1	1	0	0	2	0	1	0
4	(10 , 35)	0	1	0	0	0	1	3	0	0	1
5	(4 , 20)	1	0	0	0	1	0	1	1	0	0

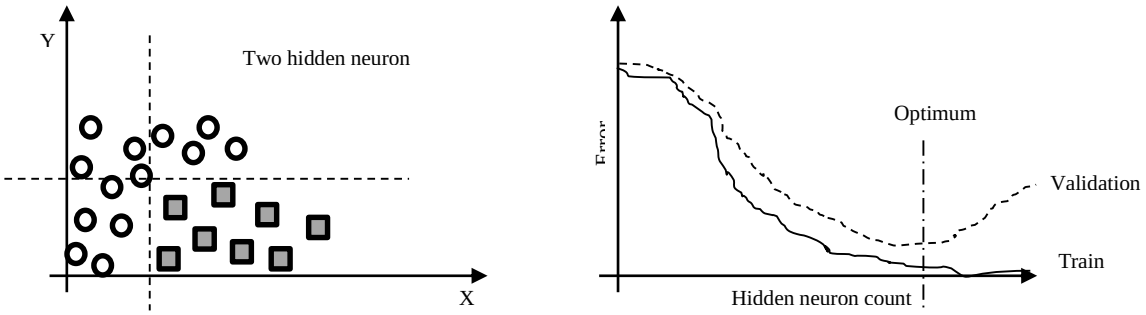
جدول (۱-۲) شامل پنج نمونه بوده که هر نمونه توسط دو ویژگی X و Y مشخص می‌شود. با توجه به مقادیر این ویژگی‌ها هر نمونه در یکی از سه کلاس مورد نظر قرار می‌گیرد. مقادیر X Bits و Y Bits نحوه کدگذاری مقادیر این ویژگی‌ها را (در سه بیت) برای انجام دسته‌بندی ایده‌آل نشان می‌دهد. مقدار Class Bits نیز مقادیر گسسته مجزا برای هر کلاس را مشخص می‌نماید.

۱-۴-۲ تعیین ساختار مناسب شبکه عصبی

این مرحله با توجه به داده‌های موجود و قوانین مورد نیاز انجام می‌شود، و همانند مرحله قبل از اهمیت بالایی برخوردار است. این ساختار باید به‌صورت مناسب و بهینه تعیین گردد تا نتایج تولید شده نیز کارایی قابل قبولی داشته باشند. تعداد نرون‌ها در لایه‌های ورودی و خروجی شبکه در مرحله قبل تعیین می‌گردند، ولی تعداد لایه‌های مخفی و تعداد نرون در این لایه‌ها با توجه به میزان پیچیدگی و پراکندگی داده‌ها تعیین می‌شوند. معمولاً شبکه عصبی با یک لایه مخفی می‌تواند اکثر مسائل را حل کند، و از

[1] Overlapping

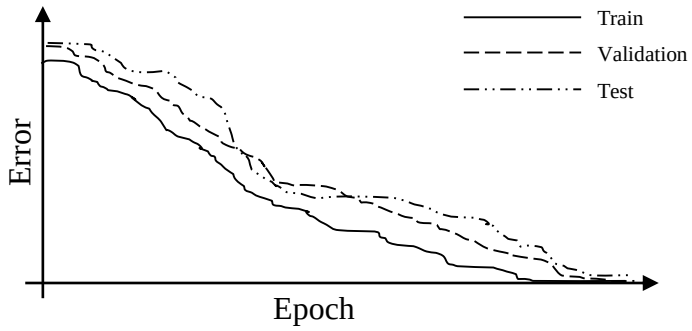
این‌رو روش‌های موجود برای استخراج قانون نیز بیش از یک لایه را پشتیبانی نمی‌کنند. اما برای تعیین تعداد نرون در این لایه می‌توان همانند شکل (۱-۲) به‌صورت دیداری و یا در حالت کلی از طریق آزمایش با داده‌های ارزیابی[1]، این تعداد را تعیین نمود [۱۱].



شکل (۱-۲) تعیین تعداد نرون در لایه مخفی

۱-۴-۳ آموزش شبکه

در این مرحله شبکه عصبی ایجاد شده با داده‌های موجود آموزش داده می‌شود تا به سطح قابل قبولی از کارایی برسد. این میزان کارایی همانند شکل (۱-۳) توسط نمودار خطای حاصل از آموزش تعیین می‌شود. این آموزش تا جایی ادامه پیدا می‌کند که بهترین نتایج بدست آیند [۱۱].

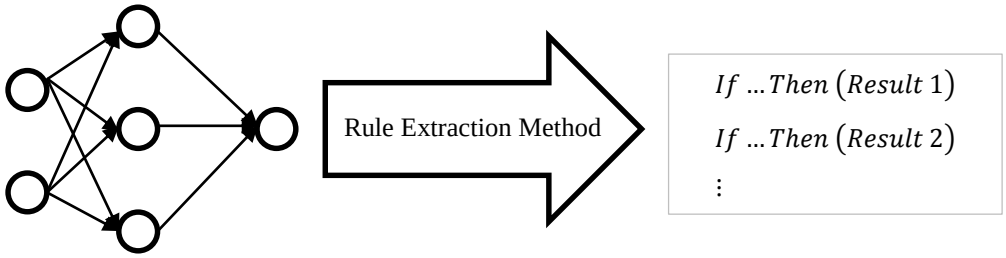


شکل (۱-۳) نمودار خطای حاصل از آموزش شبکه عصبی

[1] Validation dataset

۱-۴-۴ استخراج قوانین مورد نیاز

پس از آموزش شبکه عصبی، مقادیر وزن شبکه و همچنین خروجی نرون‌های لایه مخفی (مقادیر فعال شده[1]) استخراج شده و با استفاده از یک روش مناسب در این زمینه، قوانین مورد نیاز بدست می‌آیند [۹]. شکل (۱-۴) این حالت را نشان می‌دهد.



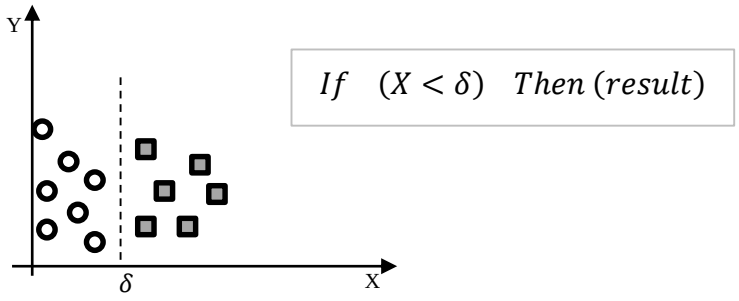
شکل (۱-۴)شمای کلی استخراج قانون از شبکه عصبی

۱-۵ رابطه ساختار شبکه و قوانین تولید شده

درصورتیکه ساختار قانون در حالت کلی مطابق رابطه (۱-۶) مورد نظر باشد:

$$If \quad (conditions) \quad Then \ (result) \tag{۱-۶}$$

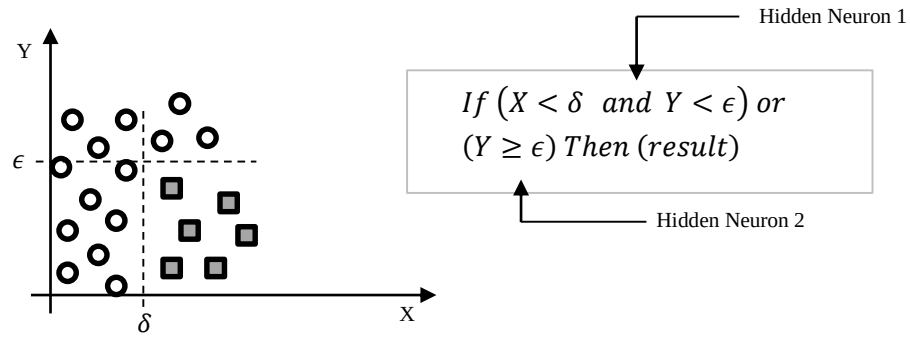
آنگاه هر ورودی شبکه عصبی به‌عنوان جزئی از گزاره شرطی بیان می‌شود (شکل (۱-۵)):



شکل (۱-۵) نقش ورودی در استخراج قالون توسط شبکه عصبی مصنوعی

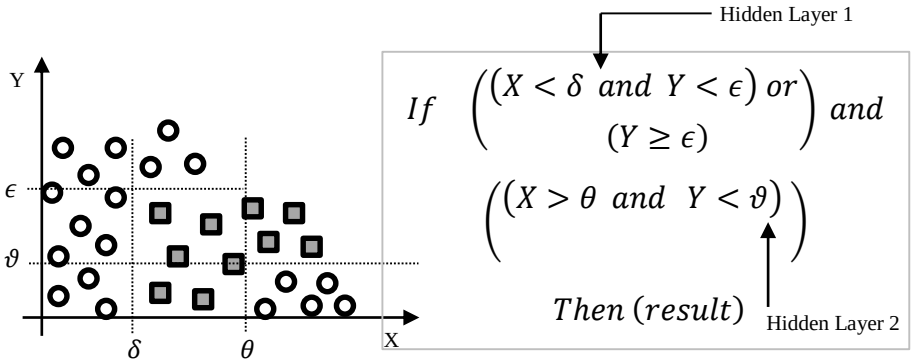
[1] Activation value

هر نرون لایه مخفی، یک گزاره شرطی (معادل یک خط جدا کننده) می‌باشد (شکل (۱-۶)):



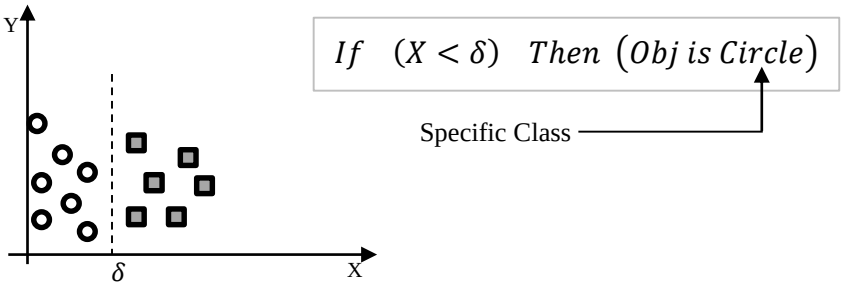
شکل (۱-۶) نقش هر نرون لایه مخفی در شبکه عصبی مصنوعی و استخراج قانون توسط آن

هر لایه مخفی برابر یک مجموعه گزاره شرطی برای یک محدوده داده می‌باشد (شکل (۱-۷)):



شکل (۱-۷) نقش هر لایه مخفی در شبکه عصبی مصنوعی و استخراج قانون توسط آن

و نهایتاً هر نرون خروجی بیانگر یک کلاس خاص از داده است (شکل (۱-۸)).



شکل (۱-۸) نقش هر نرون خروجی در استخراج قانون توسط شبکه عصبی مصنوعی

۱-۶ اهداف پایان‌نامه

با توجه به اینکه امروزه تعداد قابل توجهی از روش‌های استخراج قانون وجود دارند که بر روی شبکه‌های عصبی قابل استفاده هستند، اما همچنان اغلب روش‌های موجود تنها قادرند برای مسائل ساده قوانین مناسب را تولید نمایند. ضمنا، پیچیدگی بالای برخی از روش‌های استخراج قانون سبب کاهش استفاده از آنها در مسائل روزمره شده است.

هدف از انجام این پایان‌نامه، ارائه یک روش جهت استخراج دانش از شبکه عصبی، و ارائه آن در قالب یک مجموعه قانون ساده، مفید و کارا است که بتواند در مسائل با سطح پیچیدگی گوناگون مورد استفاده قرار گیرد. این روش با حذف پیچیدگی‌های اضافی و همچنین تولید نتایج ایده‌آل قادر خواهد بود تا در مسائل ساده کاربرد بیشتری داشته باشد؛ همچنین برای مسائل پیچیده نیز بتواند نتایج قابل قبولی را ارائه دهد.

در ادامه این پایان‌نامه، در فصل دوم با انواع دسته‌بندی‌های مختلف برای روش‌های استخراج قانون آشنا می‌شویم. روش پیشنهادی جهت انجام استخراج قانون در فصل سوم ارائه شده، و مراحل مختلف آن به‌صورت کامل تشریح می‌شود. در فصل چهارم نتایج بدست آمده از آزمایش روش ارائه شده بر روی برخی از مجموعه داده‌های استاندارد به همراه جزئیات آزمایش نشان داده می‌شود. فصل پنجم حاوی مثالی از کاربرد روش ارائه شده در تشخیص بیماری کبد چرب (به‌عنوان یک مثال پیچیده) بوده و در نهایت در فصل ششم جمع‌بندی کلی از پایان‌نامه به همراه پیشنهادات آتی اراده می‌گردد.

۱-۷ نتیجه گیری

در این فصل مروری بر مبحث شبکه‌های عصبی مصنوعی انجام شده و اهمیت و ارزش این ابزار در مسائل امروزی مطرح گردید. سپس منظور و هدف از استخراج قانون تشریح شده و برخی از انواع قوانین رایج‌تر معرفی گردیدند. همچنین عملیات کلی انجام شده در روش‌های استخراج قانون از شبکه عصبی

توضیح داده شد و رابطه هر جزء از ساختار شبکه عصبی با قانون استخراج شده نشان داده شد. در فصل بعد با انواع دسته‌بندی‌های مختلفی که برای روش‌های استخراج قانون وجود دارد، آشنا شده و نمونه‌های از هر دسته معرفی می‌گردد.

فصل دوم

روش‌های استخراج قانون به کمک شبکه‌های عصبی

۲-۱ مقدمه

روش‌های متعددی برای استخراج قانون به کمک شبکه‌های عصبی مصنوعی وجود دارد. روش‌های موجود از لحاظ ساختار شبکه‌های عصبی مورد استفاده و فرم قانونی که ارائه می‌دهند، دارای خصوصیات متفاوتی هستند، و باعث می‌شود در دسته‌های متفاوتی قرار گیرند. در این فصل، روش‌های استخراج قانون مبتنی بر شبکه‌های عصبی از پنج منظر مختلف دسته‌بندی می‌شوند، و در ادامه در مورد آنها بحث می‌شود.

۲-۲ دسته اول - بر اساس نوع شبکه

در این نوع دسته‌بندی، نوع شبکه مورد استفاده در عملیات مربوطه به‌عنوان پارامتر دسته‌بندی تعیین می‌گردد. شبکه‌هایی که امکان استخراج قانون از آنها وجود دارد عبارتنداز: شبکه‌های عصبی چند لایه (MLP)، شبکه‌های تابع پایه شعاعی (RBF[1]) ، و شبکه‌های نگاشت خود سازمان یافته (SOM[2]). در زیر، در مورد ویژگی‌های کلی هریک از این روشها توضیحاتی داده می‌شود.

۲-۲-۱ شبکه‌های عصبی چند لایه

این نوع شبکه‌های عصبی بدلیل آنکه برای عملیات دسته‌بندی و خوشه‌بندی رایج‌تر هستند، معمولاً بیشتر مورد استفاده قرار می‌گیرد. روش‌های استخراج قانون ارائه شده در [۱۹-۲۴،۱۴] جزء این دسته قرار می‌گیرند. به‌عنوان مثال، در [۲۴] روشی تحت عنوان NeuroRule معرفی شده است که قادر است طی مراحلی قوانین کلاس‌بندی در شبکه‌های عصبی MLP با ساختار سه لایه استاندارد (ورودی، مخفی و خروجی) را استخراج نماید. در این روش، ابتدا مقادیر خروجی نرون‌های لایه مخفی به محدوده‌های گسسته تقسیم می‌شوند. سپس روابط بین خروجی لایه مخفی با مقادیر گسسته و نتایج خروجی شبکه عصبی بدست می‌آیند. همچنین روابط بین داده‌های ورودی و مقادیر گسسته در لایه مخفی نیز به‌صورت

[1] Radial basis function
[2] Self-organizing map

مجزا استخراج می‌شوند. در نهایت، مجموعه روابط بدست آمده با هم ادغام شده و قوانین نهایی را تولید می‌نمایند.

کارایی این روش بر روی مسئله دیود LED در [۲۴] مورد بررسی قرار گرفت. در این مسئله، هدف شناسایی رقم نشان داده شده بر اساس حالات مختلف یک LED هفت بخشی[1] است که شامل $۱۲۸=۲^۷$ حالت مختلف می‌باشد.

نتایج بدست آمده از این روش بر روی سه مجموعه داده‌ی تست شده از مسئله LED نشان داد که قوانین بدست آمده دارای صحتی[2] ۱۰۰% می‌باشند. البته پیش از آن نیز محققان در [۲۵] روشی جهت استخراج قوانین فازی از شبکه‌های عصبی برای حل مسئله LED ارائه نموده بودند که قوانین بدست آمده از پیچیدگی بیشتری برخوردار بود.

۲-۲-۲ شبکه‌های تابع پایه شعاعی

شبکه‌های تابع پایه شعاعی نیز همانند شبکه‌های چند لایه عمدتاً در مسائل خوشه‌بندی مورد استفاده قرار می‌گیرد، اما کارایی و دقت این نوع شبکه‌های عصبی در مقایسه با شبکه‌های چند لایه در حد پایین‌تری قرار می‌گیرد. روش‌های ارائه شده در [۲۶،۲۷] نمونه‌هایی از این دسته می‌باشد. به‌عنوان مثال روش LREX که در [۲۷] مطرح شد، قادر است قوانین ساده‌ای را استخراج نماید که برای خوشه‌بندی داده‌ها به کمک شبکه‌های عصبی تابع شعاعی مورد استفاده قرار می‌گیرد.

در این روش نیاز است تا هر یک از نرون‌های لایه مخفی که به‌عنوان یک خوشه در عملیات دسته‌بندی استفاده می‌شود، تنها متعلق به یک کلاس باشد تا بتوان نتایج مناسبی را بدست آورد. پس از آموزش شبکه عصبی توسط نمونه‌های موجود، مقادیر مختصات مرکز خوشه‌ها (نرون‌های لایه مخفی) و وزن‌های لایه خروجی استخراج می‌شوند. وزن یال‌های ورودی برای هر خوشه، مرکز خوشه را تعیین می‌کنند.

[1] Seven segment
[2] Fidelity

بدین‌ترتیب برای هر خوشه یک بردار حاصل می‌شود که مراکز ویژگی‌های ورودی برای این خوشه را نشان می‌دهد. کلاس خروجی برای هر خوشه (نرون مخفی) نیز با توجه به بزرگترین وزن یال متصل به آن در لایه خروج مشخص می‌گردد. سپس "*نمونه‌های مؤثر*" برای هر نرون لایه مخفی مشخص می‌گردند. این نمونه‌های مؤثر در هنگام اعمال به شبکه عصبی منجر به تولید مقدار خروجی بیشتری برای نرون مخفی مورد نظر، نسبت به دیگر نرون‌های مخفی می‌شوند. مقدار بدست آمده از تابع شعاع گوسی، به‌عنوان انحراف معیار خوشه‌ها در نظر گرفته می‌شود. حد بالا و پایین ویژگی‌های هر خوشه با استفاده از مقادیر میانگین و انحراف معیار، توسط رابطه (۲-۱) بدست می‌آید.

$$X_{lower} = \mu_i - \sigma_i \times S \qquad X_{upper} = \mu_i + \sigma_i \times S \qquad (۱-۲)$$

در رابطه (۲-۱) σ_i بیانگر میزان انحراف معیار (واریانس) برای ویژگی iام و μ_i نیز بیانگر میانگین ویژگی iام از خوشه مورد نظر است. مقدار S توسط یکی از پارامترهای رابطه (۲-۲) برای نمونه‌های مؤثر خوشه مورد نظر بدست می‌آید.

$$S = [Min, Max, Mean, Std, Med] \qquad (۲-۲)$$

در نهایت با ترکیب مقادیر $[X_{lower_i}, X_{upper_i}]$ یک گزاره شرطی برای ویژگی iام در خوشه مورد نظر بدست می‌آید که با توکیب این ویژگی‌ها قانون خوشه مورد نظر بدست خواهد آمد.

این روش بر روی مجموعه داده‌های Monks 1، Diabetes و Credit (که در پیوست ۱ معرفی گردیده‌اند) مورد آزمایش قرار گرفته که به ترتیب نتایج ۸۳%، ۷۶% و ۹۳% برای این مجموعه داده‌ها بدست آمده است.

۲-۲-۳ شبکه‌های نگاشت خود سازمان یافته

از این نوع شبکه عصبی بیشتر برای نمایش همسایگی‌ها در مجموعه داده، و شباهت/تفاوت بین نمونه‌های مختلف استفاده می‌شود و برای عملیات دسته‌بندی کاربرد چندانی ندارند. بدین دلیل روش‌های استخراج

قانون موجود برای این نوع شبکه عصبی نسبت به دو نوع شبکه قبلی کمتر می‌باشد. روش‌های مطرح شده در [۲۸،۲۹] نمونه‌هایی از این دسته می‌باشند. به‌عنوان مثال در [۲۸] روشی جهت استخراج قوانین از شبکه‌های عصبی SOM مبتنی بر ماتریس فاصله یکپارچه[1] (U-matrix) ارائه شده است. این ماتریس شامل فاصله اقلیدسی وزن همه نرون‌های خروجی در شبکه عصبی SOM می‌باشد که به‌منظور تعیین محدوده خوشه‌ها مورد استفاده قرار می‌گیرد و حالتی مانند رابطه (۲-۳) دارد.

$$U = [U(1)\,, U(1\,,2)\,, U(2)\,, U(2\,,3)\,, U(3)\,, U(3\,,4)\,, U(4)] \qquad (۳-۲)$$

- $U(i\,,j)$: فاصله بین نرون‌های $i\,,j$
- $U(k)$: میانگین وزن بین نرون‌های مجاور

در این روش برای تولید قوانین، ابتدا نرون‌های لایه خروجی (که معمولاً در فضای دو بعدی قرار می‌گیرند) دسته‌بندی می‌شوند. بنابراین اولین نرون خروجی انتخاب می‌شود. سپس برای نرون انتخاب شده و نرون‌های مجاور با استفاده از رابطه (۲-۴) مقدار اختلاف مرز[2] تولید می‌شود.

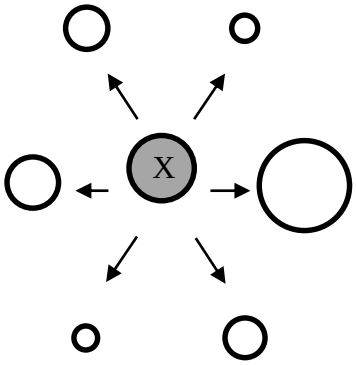
$$BDV = \frac{M_L - M_O}{R_O} \qquad (۴-۲)$$

در این رابطه:

- M_L: میانگین بین واحد کاندید برای مرز و واحد (های) انتخاب شده قبلی
- M_O: میانگین بین دیگر واحدهای مجاور باقیمانده
- R_O: محدوده دیگر واحدهای مجاور باقیمانده می‌باشد.

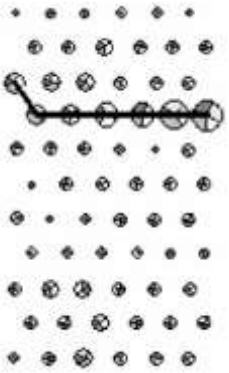
[1] Unified distance matrix
[2] Boundary difference value

نرونی که حاوی مقدار تولید شده بزرگتری باشد به‌عنوان نرون بعدی انتخاب می‌شود (شکل (۲-۱)). این عملیات تا زمانی ادامه می‌یابد که مقدار بدست آمده برای نرون‌های مجاور بیش از مقدار بدست آمده برای نرون فعلی باشد. بدین صورت یک مرز جداساز در نرون‌های لایه خروجی بدست می‌آید.



شکل (۲-۱) تولید قانون در شبکه‌های عصبی SOM با استفاده از مرزبندی نرون‌های لایه خروجی [۲۸]. از بین نرون‌های مجاور، نرونی انتخاب می‌شود که مقدار اختلاف مرز (BDV) بدست آمده بیشتری داشته باشد.

این عملیات برای همه واحدهای درون ماتریس U انجام می‌شود تا مجموعه‌ای از مرزها بدست آید. سپس مقدار BDV برای همه مرزهای بوجود آمده محاسبه شده و مرزهای با مقدار BDV بزرگتر، به‌عنوان مرزهای جداسازی خوشه‌ها تعیین می‌گردند. تعداد مرزهای انتخاب شده به تعداد مورد نیاز برای جداسازی خوشه‌های موجود است. شکل (۲-۲) جداسازی دو خوشه را توسط یک مرز جداساز نشان می‌دهد.



شکل (۲-۲) جداسازی نرون‌های لایه خروجی توسط مرز تعیین شده [۲۸]

پس از تعیین مرزها نیاز است تا ویژگی‌های کلیدی برای دسته‌بندی / خوشه‌بندی تعیین گردد. برای تعیین این ویژگی‌ها ابتدا مرزهای جداساز برای هر ویژگی ورودی محاسبه شده؛ درصورتیکه این مرزهای بدست آمده با مرزهای ماتریس U مطابقت داشت، ویژگی ورودی به‌عنوان ویژگی مناسب انتخاب می‌شود. در نهایت میانگین مقادیر مرز در ویژگی‌های انتخاب شده به‌عنوان گزاره‌های شرطی انتخاب می‌شوند.

هرچند شبکه‌های SOM مزیت بالایی در نمایش داده‌ها دارند ولی قوانین بدست آمده از این روش، ساده بوده و تنها برای مسائل ساده کاربرد دارد. نتایج بدست آمده از آزمایش این روش عبارت است از: ۹۵٫۳% برای مجموعه Iris، ۷۱٫۹% برای مجموعه Lung Cancer و ۶۱٫۳% برای مجموعه Monks1. این مجموعه داده‌ها در پیوست ۱ معرفی گردیده‌اند.

۲-۳ دسته دوم – براساس کاربرد

در این نوع دسته‌بندی اساساً نوع کاربرد شبکه‌عصبی مورد استفاده، مد نظر قرار گرفته و روش‌های موجود، بر این اساس دسته‌بندی می‌شوند. همانطور که قبلاً نیز بیان شد، یک شبکه عصبی می‌تواند برای کاربردهای متنوعی مورد استفاده قرار گیرد که برخی از آنها عبارتنداز: دسته بندی (classification) و تخمین توابع (Regression) .

۲-۳-۱ دسته‌بندی

در مسائل دسته‌بندی، هدف تعیین عضویت هر نمونه ورودی به یکی از دسته‌های خروجی موجود است. این دسته‌ها توسط مقادیر گسسته مشخص شده و بدون روی هم افتادگی هستند. تعداد نرون در لایه خروجی برای مسائل دسته‌بندی برابر با تعداد دسته‌های موجود در مسئله می‌باشد. روش‌های ارائه شده در [۱۴،۱۹-۲۲] نمونه‌هایی از تولید قوانین دسته‌بندی بر اساس شبکه عصبی هستند.

در [۱۹] روشی برای استخراج قوانین از شبکه عصبی ارائه شده است که در عملیات دسته‌بندی بر روی شبکه عصبی چند لایه مورد استفاده قرار می‌گیرد. در این روش ابتدا عملیات آموزش شبکه عصبی توسط مجموعه داده‌های آموزشی انجام گرفته؛ سپس از روش تخریبی[1] جهت هرس‌کردن شبکه استفاده می‌شود تا ساختار شبکه عصبی بهینه بدست آید. در این روش (تخریبی) برای هر کلاس خروجی، وزن‌های همه یال‌های شبکه با مقدار آستانه[2] تعیین شده، مقایسه شده و درصورتیکه کمتر از مقدار آستانه باشند از حذف می‌شوند. بدین‌ترتیب برای هر کلاس خروجی تنها یال‌های مؤثر باقی می‌ماند. مرحله استخراج قانون خود شامل دو زیر مرحله می‌شود:

- *مرحله اول استخراج قوانین با در نظر گرفتن هر ویژگی به‌صورت مجزا:* در این روش مقادیر هر ویژگی به تنهایی بر روی کلاس خروجی بررسی شده، درصورتیکه مقادیر یک ویژگی به تنهایی بتواند نمونه‌های کلاس مورد نظر را به میزان قابل قبولی از دیگر نمونه‌ها جدا کند، قوانین مربوط به آن استخراج می‌شوند.
- *مرحله دوم استخراج قوانین با در نظر گرفتن ترکیبی از ویژگی‌های ورودی:* در این حالت ترکیب‌های مختلف ورودی‌های مؤثر شبکه در نظر گرفته می‌شود. سپس ترکیباتی که برای یک کلاس خاص بیشتر تکرار می‌شوند و عملاً متفاوت از دیگر ترکیبات استفاده شده برای دیگر کلاس‌ها است، مشخص می‌گردند. در صورتیکه تعداد تکرار این ترکیبات بیش از مقدار آستانه در نظر گرفته شده باشد، این ترکیبات به قوانین جداساز برای این کلاس در نظر گرفته می‌شوند.

این روش برای شبکه‌های عصبی چند لایه مورد استفاده قرار می‌گیرد و تنها ساختار سه لایه استاندارد را پشتیبانی می‌کند. مقادیر آستانه تعیین شده در این روش برای کنترل میزان قوانین استخراج شده و پیچیدگی آنها می‌باشد که باید به‌صورت دقیق تعیین گردد. نتایج بدست آمده از این روش بر روی

[1] Destructive approach
[2] Threshold value

داده‌های استاندارد، نشان داده است که این روش توانایی تولید قوانین مناسب برای عملیات دسته‌بندی را دارد [۱۹].

میزان دقت بدست آمده از استخراج قوانین دسته‌بندی توسط این روش بر روی مجموعه‌های Monks 1 و Breast Cancer (که در پیوست ۱ معرفی گردیده‌اند) به ترتیب برابر ۱۰۰% و ۹۷٫۸۵% بوده است.

۲-۳-۲ تخمین توابع

منظور از مسائل تخمین تابع، تعیین مقدار (حقیقی) خروجی برای هر نمونه بر اساس پارامترهای ورودی آن نمونه است. لایه خروجی شبکه عصبی در این مسائل شامل تنها یک نرون بوده که به‌منظور تولید این مقدار استفاده می‌گردد. روش‌های ارائه شده در [۳۰-۳۳] نمونه‌های از این دسته می‌باشند.

روش REFANN[1] که توسط محققین در [۳۳] معرفی شده است، برای اینگونه مسائل کاربرد دارد. در این روش ابتدا برای مراحل آموزش و هرس‌سازی شبکه عصبی از روش N2PFA[2] [۳۴] استفاده می‌شود. در روش N2PFA پس از تقسیم مجموعه داده به سه بخش آموزش، اعتبارسنجی و تست، آموزش شبکه عصبی توسط تعداد مناسبی از نرون در لایه مخفی انجام می‌شود تا خطای آموزش و اعتبارسنجی (طبق رابطه (۲-۵)) به حداقل میزان موجود برسد.

$$ET = \frac{1}{|T|}\sum_{p \in T}\left|\overline{y_p} - y_p\right| \qquad EX = \frac{1}{|X|}\sum_{p \in X}\left|\overline{y_p} - y_p\right| \qquad (۲-۵)$$

که در این فرمول T مجموعه آموزش و ET میزان خطای بدست آمده شبکه عصبی بر روی این مجموعه می‌باشد. همچنین X نمایانگر مجموعه اعتبارسنجی و EX میزان خطای شبکه عصبی بر روی این مجموعه را نشان می‌دهد. مقادیر y_p و $\overline{y_p}$ نیز به ترتیب بیانگر خروجی حقیقی و خروجی تخمین‌زده شده می‌باشد.

[1] Rule extraction from artificial neural network
[2] Neural network pruning for function approximation

سپس جهت هرس کردن این شبکه عصبی، نرون‌های مخفی غیر ضروری و ورودی‌های نامرتبط به‌صورت آزمایش و خطا مشخص و حذف می‌گردند. این عملیات با توجه به مقادیر خطای بدست آمده از رابطه (۲-۵) تا زمانی ادامه می‌یاند که خطای حاصل از ساختار جدید (هرس شده) شبکه عصبی بیشتر از حد آستانه تعیین شده نباشد. در نهایت با اعمال داده‌های موجود به ساختار هرس شده شبکه عصبی، مقادیر خروجی هر نرون مخفی بدست آمده که به‌صورت یک منحنی می‌باشد. برای کشف معادله انجام شده در هر نورن مخفی، منحنی آن را به سه یا پنج قسمت تقسیم کرده و توسط توابع ریاضیاتی سه بخشی[1] و پنج بخشی[2] تابع مورد نظر برای منحنی تخمین زده می‌شود. سپس با توجه به توابع تخمین زده شده، قوانین مورد نظر برای داده‌ها تولید می‌گردد.

۲-۴ دسته سوم – بر اساس دیدگاه استخراج قانون

در حالت کلی سه دیدگاه در مبحث استخراج قانون از شبکه عصبی وجود دارد: دیدگاه تجزیه‌ای (De-compositional)، دیدگاه مبتنی بر آموزش (Pedagogical) و دیدگاه ترکیبی (Eclectic) [۱۳،۹].

۲-۴-۱ روش‌های مبتنی بر دیدگاه تجزیه‌ای

روش‌های مبتنی بر دیدگاه تجزیه‌ای، در اولین قدم ساختار شبکه عصبی را به اجزای تشکیل دهنده آن (لایه‌ها و نرون‌های هر لایه) تقسیم می‌نمایند. سپس با بررسی هر قسمت، رابطه جزئی بین این قسمت‌های متوالی را محاسبه و با ترکیب این روابط جزئی، رابطه کلی بین پارامترهای ورودی و مقدار / دسته‌های خروجی را در قالب قانون نمایش می‌دهند.

در [۲۰] روشی برای انجام عملیات استخراج قانون با نام REANN [3] معرفی شده است. این روش مبتنی بر ساختار تجزیه‌ای بوده و موارد استفاده از آن در شبکه‌های عصبی چند لایه با ساختار استاندارد (تنها

[1] Three pieces function
[2] Five pieces function
[3] Rule extraction from artificial neural network

یک لایه مخفی) می‌باشد. روش REANN از ابتدای ایجاد شبکه عصبی اعمال شده و در انتها با تعیین قوانین مورد نیاز برای دسته‌بندی نمونه‌های هر کلاس خاتمه می‌یابد.

در این روش پس از تعیین تعداد نرون‌های ورودی و خروجی، آموزش شبکه عصبی با یک نرون در لایه مخفی آغاز می‌گردد. سپس با افزایش تعداد نرون در لایه مخفی و آزمایش حالات مختلف آن، تعداد نرون مناسب در این لایه بدست می‌آید. سپس با استفاده از تابع پنالتی (رابطه (۲-۶)) شبکه عصبی آموزش دیده هرس می‌شود.

$$P(w,v) = \varepsilon_1 \left(\sum_{m=1}^{h} \sum_{l=1}^{n} \frac{\beta (w_l^m)^2}{1+\beta (w_l^m)^2} + \sum_{m=1}^{h} \sum_{p=1}^{o} \frac{\beta \left(v_p^m\right)^2}{1+\beta \left(v_p^m\right)^2} \right) + \varepsilon_2 \left(\sum_{m=1}^{h} \sum_{l=1}^{n} (w_l^m)^2 + \sum_{m=1}^{h} \sum_{p=1}^{o} \left(v_p^m\right)^2 \right) \quad (۲-۶)$$

- n: تعداد نرون‌های لایه ورودی
- h: تعداد نرون‌های لایه مخفی
- o: تعداد نرون‌های لایه خروجی
- w_l^m: وزن یال متصل به نرون ورودی l و نرون مخفی m
- v_p^m: وزن یال متصل به نرون مخفی m و نرون خروجی p
- $\beta, \varepsilon_2, \varepsilon_1$: نسبت به داده‌های موجود تعیین می‌شوند

پس از هرس‌سازی، مقادیر فعال شده هر نرون لایه مخفی توسط عملیات خوشه‌بندی به مقادیر گسسته تبدیل می‌شود و نهایتاً قوانین مورد نیاز توسط روش REx [۳۵] بدست می‌آیند. این روش به‌صورت تکراری مراحل استخراج قانون، خوشه‌بندی قانون، هرس کردن قانون را انجام می‌دهد تا همه نمونه‌ها پوشش داده شود.

میزان دقت بدست آمده از این روش در آزمایشات مختلف برابر بوده است با: ۹۶٫۲۸% برای مجموعه داده Breast Cancer، ۱۰۰% برای مجموعه داده Season و ۱۰۰% برای مجموعه Playing Golf می‌باشد (توضیحات لازم در مورد این مجموعه داده‌ها در پیوست ۱ آمده است). این روش برای مسائل ساده به خوبی پاسخگو بوده؛ اما در مقابل مجموعه داده‌های حجیم و مسائل پیچیده از پیچیدگی زمانی بالایی برخوردار است.

۲-۴-۲ روش‌های مبتنی بر دیدگاه آموزشی

در روش‌های مبتنی بر دیدگاه آموزشی، ساختار درونی شبکه عصبی مورد بررسی قرار نمی‌گیرد. در این حالت با در نظر گرفتن شبکه عصبی به‌عنوان یک جعبه سیاه، و تنها با استفاده از مقادیر ورودی و خروجی بدست آمده، رابطه (قوانین) مورد نظر بدست می‌آیند. محققین در [۳۰،۳۶،۳۷] نمونه‌هایی از اینگونه روش‌های استخراج قانون را معرفی نموده‌اند.

در [۳۶] روشی تحت عنوان ANN-DT برای استخراج قوانین از شبکه عصبی در قالب یک درخت تصمیم ارائه شده است. این روش توانایی استخراج قوانین از شبکه عصبی، بدون در نظر گرفتن ساختار شبکه و یا نوع ویژگی‌های ورودی را دارد؛ همچنین می‌توان از این روش در مواردی مانند رگرسیون، که خروجی شبکه عصبی مقادیر پیوسته‌ای می‌باشد نیز استفاده کرد. خروجی این روش یک درخت تصمیم تک متغیری است که در هر گره[1] تنها شرایط یک متغیر به همراه یک مقدار آستانه بررسی می‌شوند. هدف این روش تعیین ویژگی‌های مؤثر و مقادیر آستانه برای ویژگی است.

در فاز اول این روش، شبکه عصبی مورد استفاده توسط مجموعه داده موجود، آموزش دیده تا به سطح قابل قبولی از کارایی برسد. فاز دوم از عملیات شامل تعیین یک مجموعه داده برای استخراج قوانین است. در این روش به جای انتخاب همه نمونه‌ها که باعث افزایش شدید پیچیدگی زمانی و تعداد قوانین

[1] Node

حاصل شده می‌شود؛ تنها زیر مجموعه‌ای از نمونه‌ها (با در نظر گرفتن توزیع داده‌های اصلی) انتخاب می‌گردند که اصطلاحاً این عملیات را نمونه‌برداری[1] از فضای نمونه‌ها می‌نامند و باعث افزایش کارایی می‌گردد. فاز سوم مربوط به انتخاب ویژگی‌های مناسب و تعیین مقادیر آستانه برای تولید قوانین از روی نمونه‌های اتخاب شده می‌باشد. در این فاز به‌صورت بازگشتی در هر مرحله مجموعه نمونه‌های انتخاب شده را با استفاده از بهترین ویژگی به دو قسمت ایده‌آل تقسیم کرده و مقدار آستانه برای این تقسیم‌بندی را بدست می‌آورد. نهایتاً در فاز چهارم زمان توقف عملیات بازگشتی تعیین می‌گردد؛ که برای اینکار از حالتی مانند صفر شدن انحراف استاندارد یا واریانس در دسته‌ها استفاده می‌شود.

این روش به‌صورت بازگشتی عمل کرده و می‌تواند در شبکه‌های عصبی نوع چند لایه و تابع پایه شعاعی مورد استفاد قرار گیرد. در قسمت آزمایش و بررسی، روش مورد نظر بر روی سه مجموعه داده Sin-Cos، Abalone و Pine (پیوست ۱) آزمایش شده که نتایج بدست آمده از آن‌ها به ترتیب برابر با: ۸۴%، ۸۲٫۴۷% و ۹۲٫۳% بوده است.

همچنین در [۳۷] روشی برای استخراج قانون از شبکه عصبی به نام KDRuleEx مطرح شده است که این روش نیز مبتنی بر دیدگاه آموزشی می‌باشد. توانایی این روش در نمایش عملیات انجام شده توسط شبکه عصبی در قالب یک جدول تصمیم است. مزیت این روش به لطف استفاده از تکنیک آموزشی، پیچیدگی زمانی کمتر است. همچنین این روش در مقایسه با دیگر روش‌های مبتنی بر دیدگاه آموزشی موجود از مدیریت حافظه بهتری برخوردار است، زیرا از حالت بازگشتی استفاده نمی‌کند.

برای اجرای این روش ابتدا ویژگی‌های ورودی بررسی می‌شوند. درصورتیکه مقادیر این ویژگی‌ها پیوسته باشند، با استفاده از یکی از روش‌های گسسته‌سازی موجود این مقادیر پیوسته به محدوده‌های گسسته تبدیل می‌شوند. سپس بهترین ویژگی برای جداسازی داده‌های کلاس‌های مختلف تشخیص داده شده و مقادیر آن ویژگی به بخش‌هایی تقسیم می‌شوند تا بتوان جداسازی را بین داده‌ها بوجود آورد.

[1] Sampling

تقسیم‌بندی این ویژگی‌ها تا زمانی ادامه می‌یابد که هزینه محاسبات و قوانین بدست آمده از میزان تعیین شده بیشتر نشود. سپس تصمیمات حاصل شده در جدول مورد نظر وارد شده و بهترین ویژگی بعدی جهت انجام عملیات انتخاب می‌شود. اینکار تا زمانی انجام می‌شود که میزان درستی[1] تعیین شده بدست آید. این روش در مقایسه با دیگر روش‌ها، ساختار ساده و غیر بازگشتی دارد؛ به همین دلیل در مسائل ساده و نه چندان پیچیده کاربرد دارد.

۲-۴-۳ روش‌های مبتنی بر دیدگاه ترکیبی (Eclectic)

روش‌های مبتنی بر دیدگاه ترکیبی شامل روش‌هایی هستند که بخشی از مراحل استخراج قانون را مبتنی بر دیدگاه تجزیه‌ای و بخش دیگر را مبتنی بر دیدگاه آموزشی انجام می‌دهند. روش FERNN[2] نمونه‌ای از این دسته روش‌ها می‌باشد که در [۲۳] ارائه شده است. این روش بر روی شبکه عصبی با ساختار استاندارد سه لایه‌ای کاربرد دارد. یکی از مزیت‌های این روش نسبت به دیگر روش‌های موجود در این زمینه این است که در اکثر روش‌ها قبل از انجام عملیات استخراج قانون نیاز به هرس کردن شبکه عصبی آموزش دیده می‌باشد که اینکار خود باعث ارزیابی مجدد شبکه عصبی و افزایش هزینه عملیات می‌شود؛ اما در این روش نیاز به اینکار نبوده و موجب صرفه‌جویی در زمان و هزینه عملیات در مقابل دیگر روش‌ها می‌شود.

در این روش ابتدا شبکه عصبی با استفاده از داده‌های موجود آموزش می‌بینید و سپس یال‌های مفید و غیر مفید بر اساس بزرگی و ارزش وزن‌ها تعیین می‌شوند. در مرحله بعد برای تعیین اولویت نرون‌های لایه مخفی، و مفید و غیر مفید بودن آنها از پارامتر بهره اطلاعات[3] استفاده می‌شود. پس از آن برای هر نرون مخفی، یال‌های ورودی غیر مفید (با توجه به مقدار وزن آنها) حذف گردیده و با تقسیم‌بندی مقادیر خروجی هر نرون مخفی و ایجاد درخت تصمیم توسط روش C4.5 [۳۸]، نمونه‌های موجود در مجموعه

[1] Fidelity value
[2] Fast extraction of rules from artificial neural network
[3] Information Gain

داده‌ها دسته‌بندی می‌شوند. در انتها با استفاده از درخت تصمیم بدست آمده و پارامترهای ورودی مجموعه قوانین مورد نظر بدست می‌آیند.

در واقع در این روش ارتباط بین لایه ورودی و لایه مخفی شبکه عصبی با توجه به ساختار شبکه و مقادیر وزن یالها بدست می‌آید؛ و ارتباط بین لایه مخفی و لایه خروجی بدون توجه به ساختار شبکه و با استفاده از متد C4.5 بدست می‌آید. در نهایت این روش بر روی مجموعه داده Monks 1 به میزان دقت ۹۹٫۸۹% و بر روی مجموعه داده Breast Cancer به میزان دقت ۹۵٫۱% دست یافته است. مجموعه داده‌های استفاده شده برای بررسی این روش به‌صورت مختصر در پیوست ۱ معرفی گردیده‌اند.

۲-۵ دسته چهارم - بر اساس نوع قوانین استخراج شده

در این قسمت روش‌های استخراج قانون بر مبنای نتیجه خروجی که بدست می‌آورند، مورد دسته‌بندی قرار می‌گیرند. نتایج اینگونه روش‌ها عموماً یک مجموعه قانون می‌باشد که این قوانین براساس ساختاری که دارند، شامل انواع مختلفی هستند. متداول‌ترین قوانینی که از شبکه‌های عصبی مصنوعی استخراج می‌شند شامل قوانین گزاره‌ای، M-of-N و فازی می‌باشند.

۲-۵-۱ قوانین گزاره‌ای

روش‌های ارائه شده در [۲۲،۳۹-۱۴،۲۰] نمونه‌هایی از این دسته می‌باشند. به‌عنوان مثال روشی که در [۳۹] جهت استخراج قوانین گزاره‌ای یا $If \ldots Then$ ارائه شده است، بر روی شبکه‌های عصبی نوع چند لایه و برای مسائل دسته‌بندی کاربرد دارد. در این روش ابتدا شبکه عصبی مورد نظر توسط داده‌های موجود آموزش دیده، سپس از الگوریتم خوشه‌بندی ژنتیک (CGA[1] [۴۰]) برای خوشه‌بندی مقادیر خروجی در نرون‌های لایه مخفی استفاده می‌شود. مراکز خوشه‌های ایجاد شده برای هر نرون مخفی، به‌عنوان مقادیر گسسته برای آن نرون در نظر گرفته شده و در نهایت از روش RX [۴۱] برای تولید

[1] Clustering genetic algorithm

قوانین $If \ldots Then$ استفاده شد. در روش RX ابتدا رابطه بین مقادیر گسسته لایه مخفی و دسته‌های موجود در لایه خروجی تعیین می‌گردد. سپس رابطه بین نمونه‌های ورودی و مقادیر گسسته لایه مخفی محاسبه خواهد شد و در نهایت با ترکیب روابط بدست آمده در مراحل قبل، قوانین نهایی حاصل می‌شود.

این روش مبتنی بر دیدگاه ترکیبی می‌باشد که بخشی از مراحل تولید قوانین را از روی ساختار شبکه و بخش دیگر توسط روش خوشه‌بندی بدست می‌آید. نتایج آزمایشات انجام شده نشان می‌دهد که این روش قادر است قوانینی با میزان دقت ۹۶% برای مجموعه داده Iris و قوانینی با میزان دقت ۹۶٫۳۵% برای مجموعه داده Breast Cancer ایجاد نماید. مجموعه‌های Iris و Breast Cancer در پیوست ۱ معرفی شده‌اند.

۲-۵-۲ قوانین M-of-N

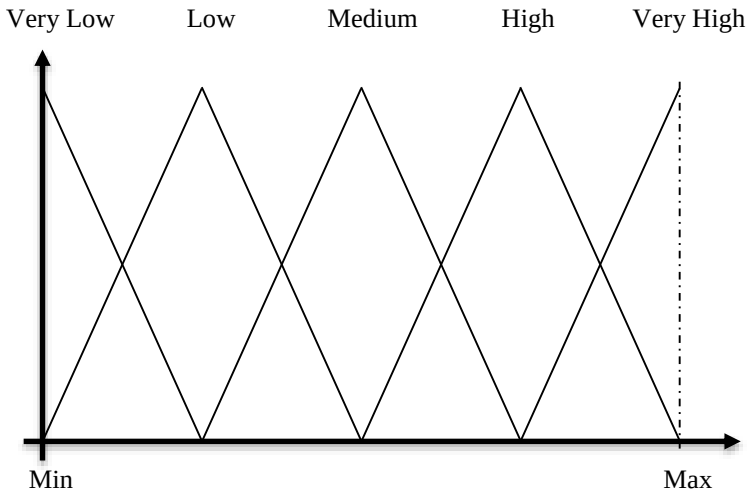
در [۱۵] روش استخراج قانونی معرفی شده است که نتیجه آن به‌صورت قوانین M-of-N می‌باشد. در این روش برای دستیابی به چنین نتیجه‌ای دو گام اصلی وجود دارد: در گام اول مقادیر نمونه‌های موجود قبل از اعمال به شبکه عصبی به مقادیر گسسته ۱- و ۱ تبدیل می‌شوند. این عمل باعث محدود شدن مقادیر ورودی شبکه عصبی می‌گردد. در گام دوم از تابع تانژانت هایپربولیک[1] بر روی همه اتصالات بین لایه ورودی و مخفی استفاده می‌شود؛ که موجب محدود شدن مقادیر ورودی به نرون‌های لایه مخفی می‌گردد. وجود این دو گام باعث ساده‌سازی استخراج قوانین به‌صورت M-of-N می‌شود. سپس با تعیین تابع فعال‌ساز مناسب برای لایه مخفی، شبکه عصبی مورد نظر آموزش می‌بینند. در نهایت با هرس کردن شبکه عصبی قوانین مورد نظر بدست می‌آیند.

[1] Tangent hyperbolic

در مرحله آزمایش این روش، از مجموعه داده‌های Monks 1، Mushroom و Breast Cancer (پیوست ۱) استفاده شده است که نتایج بدست آمده برای این مجموعه‌ها به ترتیب برابر ۱۰۰%، ۹۸٫۱۲% و ۹۴٫۰۴% بوده است.

۳-۵-۲ قوانین فازی

روش‌هایی که در [۱۶،۴۲،۴۳] معرفی شده‌اند، نمونه‌هایی از این گروه می‌باشند. به‌عنوان مثال روشی که در [۱۶] معرفی شده است قادر است قوانین فازی را برای مجموعه‌هایی شامل هر دو نوع ویژگی پیوسته و گسسته بدست آورد. مقادیر ویژگی‌های گسسته به مقادیر باینری در فرم ۱ از n تبدیل می‌شوند. همچنین مقادیر ویژگی‌های پیوسته با توجه به کمترین و بیشترین مقدار ویژگی و طبق تابع عضویت پنج متغیری شکل (۲-۳) به مقادیر فازی تبدیل می‌شوند.



شکل (۲-۳) تابع عضویت پنج متغیری نرمال [۱۶]

سپس شبکه عصبی مورد نظر توسط روش DE[1] [۴۴] که یک روش آموزش بهینه شده بر اساس جمعیت است، و با استفاده از داده‌های جدید بدست آمده آموزش می‌بینند. از روش TACO[2] [۴۵] نیز که یک روش استخراج قوانین فازی از شبکه عصبی است جهت استخراج قوانین مورد نظر استفاده شده است.

[1] Differential evolution
[2] Touring ant colony optimization

از این روش بر روی مجموعه داده‌های BUPA، Glass و Iris (پیوست ۱) استفاده شده است و نتایج حاصل از آن به ترتیب برابر بوده است با: ۸۵٫۶۰%، ۸۲٫۰۲% و ۹۷٫۷۸%.

۲-۵-۴ قوانین چند جمله‌ای (Oblique)

روش استخراج قانون ارائه شده در [۱۷] به‌منظور تولید قوانین چند جمله‌ای از شبکه عصبی مورد استفاده قرار می‌گیرد. در این روش که Neurolinear نامیده می‌شود، برای تولید شرایط درون قوانین از تقسیم فضای ویژگی‌ها به ابرصفحات[1] مختلف و برای گسسته‌سازی مقادیر نرون‌های لایه مخفی از روش Chi2 [۴۶] استفاده شده است. در نهایت با استخراج روابط ورودی - مخفی و مخفی – خروجی و همچنین ترکیب این روابط، قوانین نهایی حاصل می‌گردد.

نتایج بدست آمده از این روش برای مجموعه Credit برابر ۸۳٫۶۴% و برای مجموعه Boston Housing Data برابر ۸۰٫۶۰% بوده است. مجموعه داده‌های استفاده شده در پیوست ۱ معرفی گردیده‌اند.

۲-۶ دسته پنجم – بر اساس نوع ویژگی‌های موجود در پایگاه داده

در این نوع دسته‌بندی قابلیت اعمال روش‌های مختلف استخراج قانون بر روی انواع مجموعه‌های داده مد نظر قرار می‌گیرد. برخی از روش‌های موجود قادرند که بر روی مجموعه داده‌هایی با ویژگی‌های پیوسته، گسسته و یا هردو نوع ویژگی کاربرد داشته باشند.

۲-۶-۱ ویژگی‌های پیوسته

معمولاً روش‌های استخراج قانونی که بر روی مسائل تخمین توابع اعمال می‌شوند، تنها ورودی‌هایی از نوع مقادیر پیوسته را پشتیبانی می‌کنند. منظور از ویژگی‌های پیوسته، ویژگی‌هایی مانند دما، قد و وزن می‌باشد که غیر قابل شمارش هستند. روش‌های Neuro Linear [۱۷]، REFANN [۳۳] و RX [۴۷] از جمله روش‌های موجود در این دسته می‌باشند. به‌عنوان مثال روش REFANN که در [۳۳] ارائه

[1] Hyper plan

شده است و در بخش ۲-۳-۲ به‌صورت مختصر توضیح داده شد، برای اینگونه مسائل کاربرد داشته و ورودی‌ها را به‌صورت مقادیر پیوسته در نظر می‌گیرد.

ولی برخی از روش‌های استخراج قانون برای مسائل دسته‌بندی نیز وجود دارند که تنها مقادیر پیوسته را پشتیبانی می‌کنند. مانند روش استخراج قانون RX که توسط محققین در [۴۷] ابداع گردیده است. در این روش پس از آموزش و هرس کردن شبکه عصبی، مقادیر بدست آمده در لایه مخفی توسط عملیات خوشه‌بندی به مقادیر گسسته تبدیل می‌گردد. سپس رابطه بین مقادیر گسسته بدست آمده با خروجی‌های شبکه عصبی تعیین می‌شود. آنگاه برای هر نرون لایه مخفی به‌صورت زیر عمل می‌شود:

۱- تعداد اتصالات ورودی به هر نرون لایه مخفی پس از هرس سازی تعیین می‌شود.

۲- درصورتیکه این تعداد کمتر از حد آستانه تعیین شده باشد، قوانین مورد نظر برای دستیابی به مقادیر گسسته شده این نرون بر اساس ورودی‌ها بدست می‌آیند.

۳- در غیر اینصورت، نرون لایه مخفی و ورودی‌های متصل به آن به‌عنوان یک شبکه عصبی مجزا در نظر گرفته می‌شوند، و سپس

- هر مقدار گسسته بدست آمده از نرون لایه مخفی به‌عنوان یک خروجی شبکه عصبی جدید در نظر گرفته می‌شود.
- هر ورودی متصل به نرون لایه مخفی به‌عنوان ورودی شبکه عصبی جدید در نظر گرفته می‌شود.
- لایه مخفی جدیدی با تعداد مناسب نرون در این لایه ایجاد می‌گردد.
- این شبکه توسط داده‌های موجود آموزش دیده و مراحل استخراج قانون برای آن تکرار می‌شود.

۴- در انتها با ترکیب قوانین بدست آمده، مجموعه قوانین نهایی بدست می‌آید.

۲-۶-۲ ویژگی‌های گسسته

منظور از ویژگی‌های گسسته، ویژگی‌هایی مانند جنسیت، سطوح تحصیلی و مذهب می‌باشد که قابل شمارش هستند. محققین در [۱۴] روشی را برای استخراج قوانین از مجموعه داده‌های گسسته ابداع کردند که GLARE [1] نامیده می‌شود. این روش که مبتنی بر دیدگاه De-Compositional است، بر روی شبکه چند لایه با ساختار استاندارد اعمال شده و قادر است تا قوانین ساده‌ای که برای دسته‌بندی داده‌ها توسط این شبکه مورد استفاده قرار می‌گیرد را استخراج کند. در این روش نیاز است تا مقادیر ویژگی‌های ورودی و کلاس‌های خروجی گسسته‌سازی[2] شده و هر کدام با یک نرون مجزا نمایش داده شوند.

برای اعمال این روش ابتدا شبکه عصبی توسط نمونه‌های موجود آموزش دیده و کارایی آن توسط مجموعه داده‌های ارزیابی بررسی می‌شود. پس از مرحله آموزش و شکل‌گیری وزن‌ها، این روش که شامل هفت مرحله می‌باشد بر روی شبکه عصبی مورد نظر اعمال شده و قوانین مورد نظر بدست می‌آیند. این مراحل به‌صورت زیر است:

۱- مرتب‌سازی وزن‌های ورودی به هر نرون لایه مخفی به‌صورت جداگانه

۲- هرس کردن یال‌های با وزن کمتر از مقدار آستانه در نظر گرفته شده

۳- دست‌یابی به مقادیر فعال شده هر نرون لایه مخفی به ازای نمونه‌های کلاس مورد نظر

۴- مرتب‌سازی مقادیر بدست آمده در مرحله قبل

۵- ترکیب مقادیر بدست آمده در مرحله دوم و چهارم

۶- ایجاد جدول قوانین از روی نتایج مرحله پنجم

۷- تفسیر مقادیر جدول مرحله قبل در قالب فرمول‌های قابل فهم و کاربردی

[1] Generalized analytic rule extraction
[2] Discretize

نتایج و قوانین بدست آمده نشان داد که این روش برای دسته‌بندی داده‌های نه چندان پیچیده، قوانین مطلوب را تولید کرده و می‌توان از این قوانین در کاربردهای مربوطه استفاده کرد. در این روش هرچه میزان گسسته‌سازی انجام شده دقیق‌تر باشد، قوانین خروجی نیز دقیق‌تر هستند. نتایج بدست آمده از این روش بر روی مجموعه داده‌های BUPA، Glass و Iris (پیوست ۱) به ترتیب برابر ۶۶٫۶۷%، ۸۳٫۳۳% و ۹۳٫۷۵% می‌باشد.

۳-۶-۲ هر دو نوع ویژگی پیوسته و گسسته

روش‌های معرفی شده در [۱۶،۳۶،۴۸] قادرند بر روی هر دو نوع ویژگی پیوسته و گسسته اعمال شوند. در [۴۸] یک روش استخراج قانون مبتنی بر گرادیانت معرفی شده است که بیشتر برای عملیات کاهش ابعاد داده مورد استفاده قرار می‌گیرد. معمولاً در مجموعه داده‌های طبیعی بدلیل وجود برخی ویژگی‌های غیرمفید و ناکارآمد، انجام عملیات کاهش ابعاد داده قبل از عملیات اصلی (مانند دسته‌بندی و تخمین تابع) کاری ضروری است. در این روش از شبکه عصبی نوع RBF با تابع گوسی[1] در نرون‌های لایه مخفی استفاده شده است که می‌تواند ویژگی‌ها را بر اساس معیار همبستگی – تفکیک‌پذیری (SCM[2]) انتخاب می‌نماید. در این روش برای هر نرون لایه مخفی، یک قانون بدست می‌آید.

در این روش برای کمینه کردن میزان خطای قوانین بدست آمده، از تئوری گرادیانت کاهشی[3] استفاده می‌شود. همچنین وجود روی هم افتادگی در بین خوشه‌های متعلق به یک کلاس خاص باعث ایجاد قوانین کاملاً روی هم افتاده می‌شود که این قوانین در مرحله بعد حذف می‌گردند.

[1] Gaussian kernel function
[2] Separability-correlation measure
[3] Gradient descent theory

۲-۷ نتیجه گیری

روش‌های استخراج قانون معمولاً بر اساس ساختار و نتایج بدست آمده به حالت‌های گوناگونی دسته‌بندی می‌شوند. در این فصل پنج حالت مختلف جهت دسته‌بندی این روش‌ها توضیح داده شد. روش‌های موجود را می‌توان بر اساس هر یک از این حالات در دسته‌های مختلف دسته‌بندی نمود.

- *بر اساس نوع شبکه عصبی:* شبکه‌های عصبی چند لایه، شبکه‌های عصبی تابع شعاعی و شبکه‌های عصبی خود سازمان یافته؛
- *بر اساس نوع مسئله:* دسته‌بندی و تخمین توابع؛
- بر اساس نوع دیدگاه استخراج قانون: دیدگاه تجزیه‌ای، دیدگاه مبتنی بر آموزش و دیدگاه ترکیبی؛
- *بر اساس نوع قانون استخراج شده:* قوانین گزاره‌ای، قوانین M-of-N، قوانین فازی و قوانین چند جمله‌ای؛
- *بر اساس نوع ویژگی‌های ورودی:* ویژگی‌های پیوسته، ویژگی‌های گسسته و ویژگی‌های پیوسته و گسسته (هر دو)

در فصل بعد روش پیشنهادی جهت استخراج قانون معرفی شده و مراحل انجام عملیات در آن شرح داده می‌شود.

فصل سوم

روش استخراج قانون چهار مرحله‌ای

۳-۱ مقدمه

همانگونه که پیشتر بیان شد، استخراج قانون از شبکه عصبی روشی برای استدلال عملیات انجام شده توسط این ابزار است. در حالت کلی، برای استخراج قانون از یک شبکه عصبی، پس از انتخاب یک شبکه عصبی با ساختار مناسب برای مسئله مورد نظر و آموزش آن با داده‌های موجود، دانش مورد استفاده توسط شبکه عصبی در قالب یک مجموعه قانون بدست خواهد آمد. امروزه تعداد روش‌های بسیاری به‌منظور استخراج قانون از شبکه عصبی وجود دارند که در فصل دوم برخی از آنها معرفی گردید. اما روش‌های موجود معمولاً علاوه بر دارا بودن پیچیدگی نسبتاً بالا (در برخی از روش‌ها)، تنها بر روی مسائل ساده قابل استفاده هستند. یا با تولید تعداد قوانین بیش از حد، باعث کاهش درک انسان نسبت به نتایج بدست آمده می‌شوند. در واقع اینگونه مشکلات سبب عدم استفاده مناسب از این روش‌ها در مسائل نسبتاً پیچیده شده است.

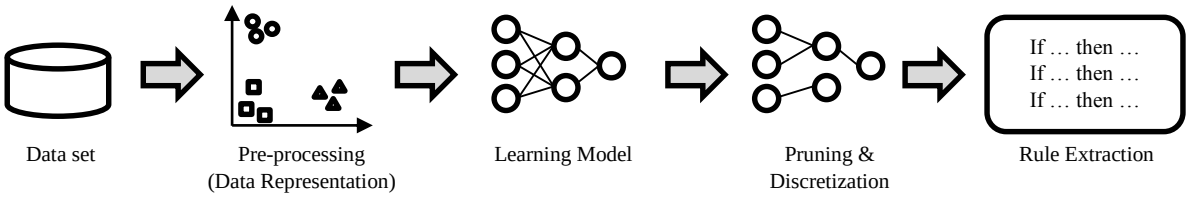
در این فصل روش جدیدی جهت استخراج قانون، به نام *استخراج قانون چهار مرحله‌ای* (FSRE[1]) معرفی می‌گردد. بر اساس دسته‌بندی‌هایی که در فصل قبل انجام شد، این روش مبتنی بر دیدگاه تجزیه‌ای (De-compositional) بوده و برای استفاده در شبکه‌های عصبی چند لایه (MLP) به‌منظور انجام عملیات دسته‌بندی مورد استفاده قرار می‌گیرد. این روش قادر است قوانین گزاره‌ای (IF…THEN) را برای این نوع مسائل بدست آورد. در ادامه این فصل مراحل مختلف این روش و چگونگی انجام عملیات توسط این روش بیان می‌گردد.

۳-۲ روش استخراج قانون FSRE

در این روش چهار فاز اصلی وجود دارد که در شکل (۳-۱) نشان داده شده است. فازهای اصلی در این روش عبارتنداز: فاز "*پیش پردازش یا نمایش داده‌ها*" که با انجام برخی تکنیک‌های داده کاوی، باعث

[1] Four step rule extraction

مرتب شدن و نرمال شدن داده‌های خام قبل از اعمال به ابزار دسته‌بند می‌شوند. فاز "*مدل یادگیری*" که عملیات اصلی شناسایی داده‌ها توسط یک شبکه عصبی چند لایه انجام می‌شود. فاز "*هرس کردن شبکه عصبی*" که به‌منظور انجام عملیات استخراج قانون از این ابزار، برخی اتصالات و گره‌های اضافی یا کم اهمیت‌تر حذف خواهند شد. فاز آخر مربوط به "*استخراج قانون*" می‌باشد که در این قسمت عملیات نهایی استخراج قانون از ساختار هرس شده شبکه عصبی انجام شده و نتایج نهایی بدست می‌آیند. این فرایند در ادامه توضیح داده خواهد شد.



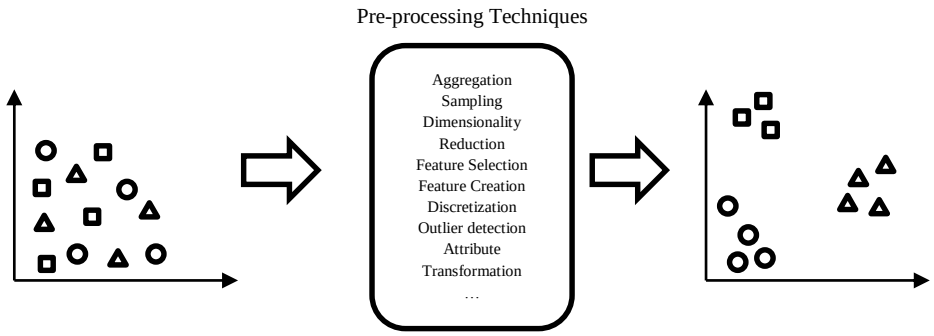
شکل (۳-۱) چهار فاز اصلی در عملیات استخراج قانون توسط روش پیشنهادی FSRE

این روش علاوه بر عملیاتی که در دیگر روش‌های استخراج قانون وجود دارد، شامل یک فاز پیش‌پردازش نیز می‌باشد که باعث انعطاف‌پذیری بیشتر روش در مقابل مسائل گوناگون با سطوح پیچیدگی مختلف می‌باشد. همچنین تنظیم پارامترهای شبکه عصبی به‌منظور آموزش آن در فاز مدل یادگیری به نحوی انجام می‌شود که نتایج بهتری را نسبت به دیگر روش‌های موجود بدست می‌آورد.

۳-۲-۱ پیش‌پردازش یا نمایش داده‌ها

معمولاً مجموعه داده‌ها پس از جمع‌آوری شدن، دارای مقادیر خام و بدون پردازشی هستند که این مجموعه داده‌های خام گستردگی زیادی دارند. ممکن است برای یک مسئله برخی از ویژگی‌ها غیر مؤثر و یا اضافی باشند و یا ویژگی‌های مختلف از یک مسئله انواع مختلفی داشته باشند. بدین‌ترتیب از یک مجموعه عملیات در قالب پیش‌پردازش قبل از انجام عملیات اصلی بر روی داده‌ها استفاده می‌شود. هدف از این پیش‌پردازش، نرمال کردن قالب داده‌ها جهت افزایش دقت نهایی و ساده‌سازی مسئله می‌باشد.

برخی از این تکنیک‌ها شامل تجمع[1]، نمونه برداری، کاهش بعد[2]، انتخاب زیرمجموعه‌ای از ویژگی‌ها[3]، ایجاد ویژگی[4] (شامل انواع استخراج، نگاشت و یا ساخت ویژگی)[5]، گسسته‌سازی[6]، انتقال ویژگی‌ها[7] و غیره می‌باشد.



شکل (۳-۲) هدف از انجام پیش‌پردازش بر روی داده‌های خام، بهینه‌سازی داده‌ها تا حد امکان برای عملیات بعدی است

همانطور که در شکل (۳-۲) مشاهده می‌کنید هدف از این فاز تبدیل مجموعه داده‌های خام به مجموعه داده‌های کارآمد و بهینه جهت انجام عملیات بعدی می‌باشد.

در روش پیشنهادی چگونگی انجام این عملیات و تعیین تکنیک‌های مورد نیاز بر اساس مجموعه داده مورد نظر تعیین می‌گردد. درصورتیکه نیاز به نرمال‌سازی مقادیر ویژگی‌ها باشد، از رابطه (۳-۱) برای اینکار استفاده می‌گردد.

$$NormData = \frac{(Data - \mu_{Data})}{\sigma_{Data}} \qquad (۳-۱)$$

[1] Aggregation
[2] Dimensionality reduction
[3] Feature subset selection
[4] Feature creation
[5] Extraction, mapping and construction
[6] Discretization
[7] Feature transformation

در این رابطه منظور از $Data$ مجموعه داده اولیه، μ_{Data} میانگین هر ویژگی از مجموعه داده و σ_{Data} انحراف استاندارد هر ویژگی می‌باشد.

همچنین از روش بهره اطلاعاتی [۴۹] برای ارزیابی ویژگی‌ها از نظر میزان کارایی و اهمیت هر کدام از ویژگی‌ها در عملیات مورد نظر استفاده می‌شود. بهره اطلاعاتی (رابطه (۳-۲)) یک خصوصیت آماری و مبتنی بر آنتروپی[1] (رابطه (۳-۳)) بوده؛ و عبارت است از میزان کاهش آنتروپی که به واسطه جداسازی نمونه‌ها از طریق یک ویژگی حاصل می‌شود [۴۹]. این مقدار برای هر ویژگی به‌صورت مجزا محاسبه می‌گردد.

$$I(Data|Attrib_x) = Entropy(Data) - Entropy(Data|Attrib_x) \qquad (۳-۲)$$

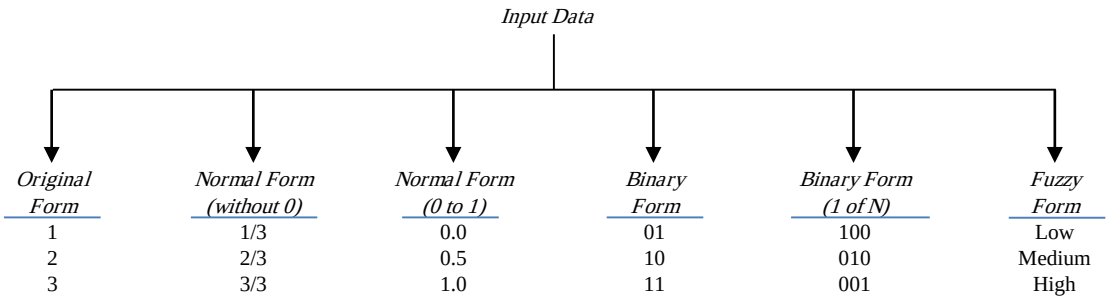
$$Entropy(Data) = -\sum_{i \in outcomes} p_i \times \log_2 p_i \qquad (۳-۳)$$

در روابط بالا منظور از p_i احتمال وجود نمونه‌های کلاس i در مجموعه داده $Data$ می‌باشد. روش FSRE در قسمت داده‌های ورودی قادر است که از انواع داده‌های گسسته و پیوسته پشتیبانی نماید. البته درصورتی که نیاز باشد می‌توان در قسمت پیش‌پردازش، مقادیر پیوسته را به مقادیر گسسته تبدیل نمود. در سال‌های اخیر، روش‌های زیادی برای انجام این عملیات ابداع و معرفی شده‌اند. روش‌هایی مانند: CAIM [۵۰]، Fast-CAIM [۵۱]، Modified CAIM [۵۲]، ur-CAIM [۵۳] از رایج‌ترین و بهترین روش‌های موجود در این زمینه هستند.

پس از انجام پیش‌پردازش، باید نحوه کدگذاری داده‌ها را برای انجام عملیات بعدی مشخص نماییم که اصطلاحاً این عمل را *نحوه نمایش داده‌ها* می‌نامند. تحقیقات انجام شده در این زمینه نشان می‌دهد که چگونگی کدگذاری و اعمال داده‌ها به شبکه‌های عصبی می‌تواند تاثیر بسزایی در نتایج بدست آمده

[1] Entropy

نهایی داشته باشد [۵۴]. شکل (۳-۳) نمونه‌ای از برخی روش‌های متداول‌تر برای نمایش یک مجموعه مقادیر ساختگی را نشان می‌دهد. نحوه نمایش داده باید به گونه‌ای انتخاب شود که بهترین نتیجه را در مسئله مورد نظر حاصل شود.



شکل (۳-۳) برخی از متداول‌ترین حالات نمایش داده برای یک ویژگی با سه مقدار متفاوت

به‌عنوان مثال روش‌های باینری[1] موجب افزایش ورودی‌های شبکه عصبی (بخصوص در حالت یک از N) شده اما موجب ساده‌تر شدن مسئله می‌گردد. در حالت‌های نرمال شده فضای تغییرات هر ویژگی در محدوده کوچک قرار گرفته که از ایجاد نوسان در مرحله آموزش شبکه عصبی جلوگیری می‌نماید. حالت فازی نیز برای مسائل فازی و تولید قوانین فازی در این مسائل استفاده می‌گردد.

در انتهای عملیات انجام شده در این فاز، مجموعه داده بدست آمده، به سه دسته مجزا جهت آموزش[2]، ارزیابی و تست[3] تقسیم می‌شوند. جهت تقسیم مجموعه داده به دسته‌های ذکر شده از نسبت ۷۰-۱۵-۱۵ درصد و یا ۵۰-۲۵-۲۵ درصد به ترتیب برای دسته‌های آموزش، ارزیابی و تست با توجه به تعداد نمونه‌ها و پیچیدگی آنها استفاده می‌گردد.

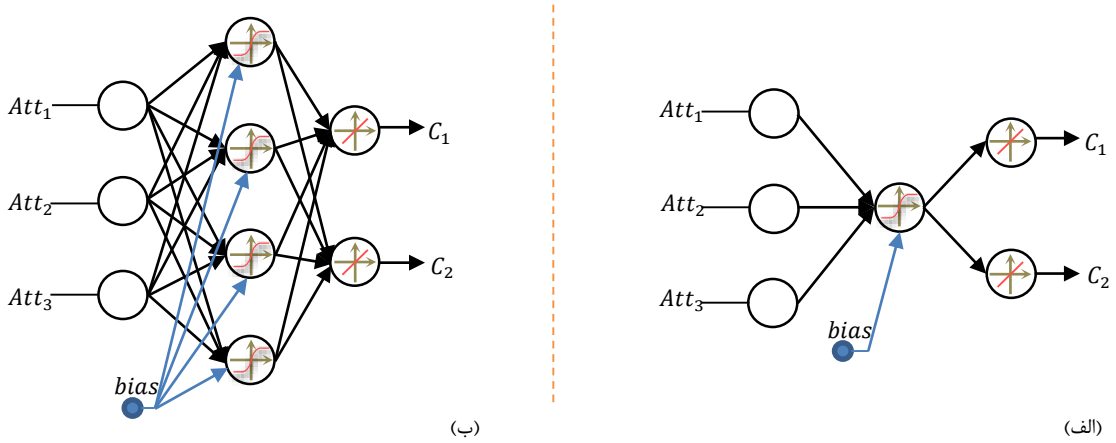
[1] Binary
[2] Train set
[3] Test set

۳-۲-۲ مدل یادگیری

در این قسمت نیاز است تا از یک ابزار هوشمند به‌عنوان یک مدل یادگیری در حل مسئله موجود استفاده شود. در روش FSRE از شبکه عصبی مصنوعی چند لایه با ساختار استاندارد به‌عنوان مدل یادگیری استفاده می‌شود.

شبکه عصبی که در این بخش در نظر گرفته می‌شود، در واقع مرجع قوانینی است که در روش استخراج قانون FSRE بدست می‌آید. در این شرایط چگونگی انجام مرحله آموزش شبکه عصبی و پارامترهای آن می‌تواند تعیین کننده میزان دقت در نتایج خروجی باشد. در فاز مدل یادگیری ابتدا یک شبکه عصبی با ساختار سه لایه استاندارد (ورودی، مخفی و خروجی) ایجاد می‌کنیم. تعداد نرون در لایه‌های مختلف بر اساس مجموعه داده موجود تعیین می‌گردد: تعداد نرون در لایه ورودی به تعداد ویژگی‌های موجود در مجموعه داده (یا بر اساس نحوه کدگذاری داده) و تعداد نرون در لایه خروجی به تعداد دسته‌های موجود در مسئله مورد نظر انتخاب می‌شود. تعداد نرون در لایه مخفی نیز در ابتدای آموزش برابر یک می‌باشد. پس از انجام آموزش در صورتیکه نتایج بدست آمده قابل قبول نباشد به لایه مخفی یک نرون اضافه شده و مرحله آموزش دوباره تکرار می‌شود. بدین صورت می‌توان تعداد نرون مناسب برای این لایه را بر اساس پیچیدگی موجود در مسئله و داده‌های موجود تعیین نمود. تمام وزن‌های درون شبکه عصبی به‌صورت تصادفی در یک محدوده کوچک مقداردهی اولیه می‌شوند. مقدار بایاس[1] برابر ۱ می‌باشد و این بایاس تنها به نرون‌های لایه مخفی متصل است. از تابع تانژانت هایپربولیک به‌عنوان تابع فعال‌ساز نرون‌های لایه مخفی و از تابع خطی[2] برای نرون‌های لایه خروجی استفاده می‌شود (پیوست ۲). این حالت، منجر به تولید نتایج خوبی در آزمایشات گوناگون شد. شکل (۳-۴) مثالی از شبکه عصبی با ساختار اولیه (الف) و ساختار نهایی (ب) برای یک مسئله دسته‌بندی دو کلاسه با سه ویژگی ورودی را نشان می‌دهد.

[1] Bias
[2] Linear function



شکل (۳-۴) مثالی از ایجاد یک شبکه عصبی جهت استخراج قانون از یک مسئله دسته‌بندی دو کلاسه با سه ویژگی در ورودی. الف) ساختار اولیه. ب) ساختار نهایی تایید شده بر اساس نتایج بدست آمده

اندازه ضریب یادگیری[1] که در عملیات آموزش شبکه جهت بروزرسانی مقادیر وزن یال‌های مورد استفاده قرار می‌گیرد، به‌صورت ثابت تعریف شده و در طول مراحل آموزش تغییر نمی‌کند. اما مقدار این ضریب یادگیری برای یال‌های بین لایه ورودی و مخفی با مقدار در نظر گرفته شده برای یال‌های مخفی خروجی متفاوت است. معمولاً این مقدار براساس پیچیدگی مسئله و با آزمایش و خطا مشخص می‌گردد.

در ادامه برای آموزش شبکه عصبی ایجاد شده، از مجموعه داده‌های آموزشی استفاده شده و کارایی آن براساس خطای بدست آمده از مجموعه ارزیابی محاسبه می‌گردد. شکل (۳-۵) فلوچارت مراحل ایجاد، آموزش و اعتبارسنجی شبکه عصبی را نمایش می‌دهد.

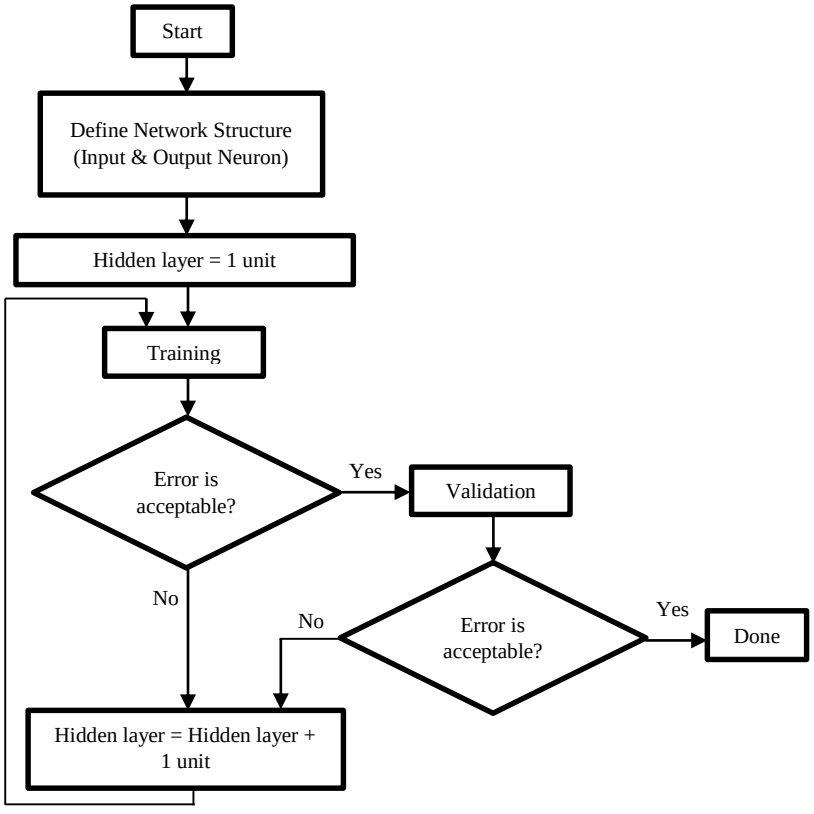
برای فاز آموزش از الگوریتم یادگیری پس انتشار خطا (پیوست ۳) استفاده می‌شود. در این شبکه عصبی، حداکثر تعداد تکرار[2] آموزش برابر ۱۰۰۰ بار در نظر گرفته می‌شود. همچنین از یک تابع هدف[3] با مقدار آستانه $\alpha = 0.001$ به‌منظور تعیین زمان مناسب برای پایان آموزش شبکه عصبی استفاده می‌گردد. در این تابع میزان خطای بدست آمده از اعمال مجموعه داده اعتبارسنجی به شبکه عصبی در هر تکرار بررسی می‌شود؛ در صورتیکه این میزان خطا کمتر از حد آستانه (α) تعیین شده باشد، مرحله آموزش

[1] Learning rate
[2] Iteration (or epoch)
[3] Goal function

شبکه عصبی پایان می‌یابد. برای محاسبه میزان خطای شبکه عصبی در این الگوریتم، از تابع میانگین مربعات خطا (MSE) بر روی مجموعه‌های آموزشی و ارزیابی استفاده می‌شود (رابطه ۳-۴).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2 \qquad (۴-۳)$$

در رابطه (۳-۵)، n تعداد نمونه در پایگاه داده، Y_i خروجی حقیقی برای iامین نمونه و $\widehat{Y}_i$ خروجی تخمین زده شده برای این نمونه در مجموعه داده مورد نظر می‌باشد.



شکل (۳-۵) مراحل ایجاد، آموزش و اعتبار سنجی شبکه عصبی ایجاد شده

پس از انجام مرحله آموزش، میزان کارایی شبکه عصبی برای مجموعه داده‌های آموزشی و ارزیابی محاسبه می‌گردد. جهت اندازه‌گیری میزان کارایی این شبکه عصبی، از پارامترهای دقت[1] (رابطه ۳-۵)،

[1] Accuracy

حساسیت[1] (رابطه ۳-۶)، و اختصاصی بودن[2] (رابطه ۳-۷) استفاده می‌شود. پارامترهای حساسیت و اختصاصی بودن تنها برای حالت دو کلاسه مورد استفاده قرار می‌گیرند.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} = \frac{Number\ of\ Correct\ Classified\ Instance}{Number\ of\ All\ Instance} \qquad (۵-۳)$$

$$Sensitivity = \frac{TP}{TP + FN} \qquad (۶-۳)$$

$$Specificity = \frac{TN}{TN + FP} \qquad (۷-۳)$$

که در این روابط:

- *صحیح مثبت (TP)*: نمونه کلاس A به درستی عضو کلاس A تشخیص داده شود.
- *مثبت کاذب (FP)*: نمونه کلاس A به اشتباه عضو کلاس B تشخیص داده شود.
- *منفی صحیح (TN)*: نمونه کلاس B بدرستی عضو کلاس B تشخیص داده شود.
- *منفی کاذب (FN)*: نمونه کلاس B به اشتباه عضو کلاس A تشخیص داده شود.

در صورتیکه میزان کارایی بدست آمده قابل قبول نباشد، یک نرون به لایه مخفی اضافه شده و مراحل آموزش و اعتبارسنجی دوباره تکرار می‌شوند. این عملیات تا زمانیکه کارایی شبکه عصبی آموزش دیده به میزان قابل قبولی برسد، به‌صورت چرخشی ادامه میابد.

در انتهای این فاز، پس از تایید نتایج بدست آمده از آموزش شبکه عصبی بر روی مجموعه داده‌های آموزش و ارزیابی، داده‌های مجموعه تست به شبکه عصبی اعمال می‌گردند. همچنین دقت شبکه عصبی آموزش دیده بر روی مجموعه تست نیز محاسبه می‌گردد.

[1] Sensitivity
[2] Specificity

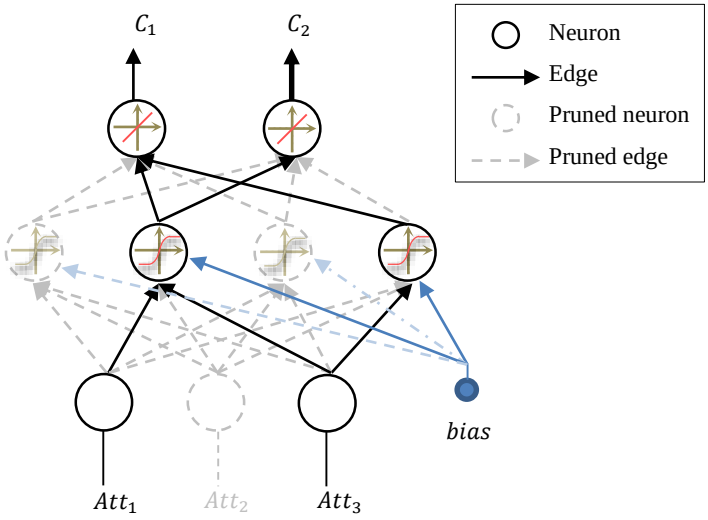
۳-۲-۳ هرس کردن شبکه عصبی

یک شبکه عصبی مصنوعی با اتصالات کامل[1] ایجاد شده و آموزش داده می‌شود. اما همگی این اتصالات و حتی همه پارامترهای ورودی، اهمیت یکسانی در تعیین خروجی ندارند. در واقع یکی از دلایل اصلی که باعث ایجاد ابهام در شبکه عصبی مصنوعی و چگونگی تولید خروجی در این ابزار می‌گردد، همین مسئله وجود اتصالات زیاد بین لایه‌های مختلف است. بنابراین می‌توان استخراج قانون در چنین حالتی را بسیار مشکل و در بیشتر موارد حتی غیر ممکن دانست.

هرس کردن شبکه عصبی یکی از راه حل‌های این مسئله می‌باشد [۳۴،۵۵]. در این روش اتصالات و نرون‌های غیرضروری در تعیین خروجی شبکه عصبی (پس از مرحله آموزش) شناسایی شده و حذف می‌گردند. بدین‌ترتیب می‌توان یالها و نرون‌های مؤثرتر در تعیین خروجی توسط شبکه عصبی را تشخیص داد. شکل (۳-۶) ساختار یک شبکه عصبی پس از هرس‌سازی را نشان می‌دهد.

یکی از دلایل وجود دقت بالا در انجام عملیات توسط شبکه عصبی وجود اتصالات کامل (و بعضاً زیاد) بین لایه‌های مختلف است. یعنی اینکه اگرچه هرس کردن اتصالات و نرون‌های اضافی در شبکه عصبی باعث کاهش ابهام در چگونگی عملیات درونی شبکه عصبی می‌گردد؛ اما این روش معمولاً با کاهش کارایی و دقت این ابزار همراه است. پس چگونگی هرس کردن و میزان هرس‌سازی یک شبکه عصبی باید به نحوی باشد که بتواند هم تا حد امکان باعث کاهش پیچیدگی شبکه عصبی (از طریق حذف اتصالات اضافی) گردد و هم از کاهش بیش از حد دقت شبکه عصبی در تعیین خروجی، جلوگیری کند.

[1] Fully-connected



شکل (۳-۶) مثالی از یک شبکه عصبی هرس شده

در روش FSRE برای هرس کردن شبکه عصبی از یک روش دو مرحله‌ای ساده و کارآمد استفاده می‌شود. در مرحله اول اتصالات مابین لایه مخفی و خروجی هرس شده، که باعث می‌شود نرون‌های موثر برای هر کلاس خروجی شناسایی شوند. برای انجام اینکار از الگوریتم شکل (۳-۷) استفاده می‌شود.

For $i = 1$ **to** *number of hidden neuron*
    If *all output weights have the same sign and* (*approximately*)*same value* **then**
        Remove all output edges (*or set weights to zero*)
    Else
        Keep the (*one or some*) *edge with largest absolute weight value*
        Remove the other edges (*or set weights to zero*)
    End IF
End FOR

شکل (۳-۷) پروسه هرس‌سازی برای حذف اتصالات بین لایه‌های مخفی و خروجی

پس از اجرای الگوریتم شکل (۳-۷) ممکن است همه اتصالات بین یک نرون خاص در لایه مخفی و همه نرون‌های لایه خروجی حذف گردند. در این حالت، این نرون همراه همه اتصالات آن از شبکه عصبی حذف می‌گردند. این حالت را می‌توان در شکل (۳-۶) مشاهده کرد.

این الگوریتم همه اتصالات مربوط به هر نرون لایه مخفی را مورد بررسی قرار می‌دهد. درصورتیکه همه اتصالات یک نرون مخفی دارای وزن‌های مشابهی باشند، آن نرون از ساختار شبکه عصبی حذف می‌گردد. زیرا این مسئله نشان می‌دهد که این نرون احتمال مشابهی را برای هر کلاس از مسئله موجود در نظر

می‌گیرد. در حالتی که وزن اتصالات مختلف مربوط به یک نرون خاص دارای مقادیر مختلفی باشند، این الگوریتم آن اتصالاتی که بزرگترین وزن را دارا هستند را حفظ کرده و باقی اتصالات را حذف می‌نماید.

در مرحله دوم از فاز هرس‌سازی، اتصالات مابین لایه‌های ورودی و مخفی هرس می‌شوند. در این قسمت، پارامترهای ورودی مؤثرتر برای انجام عملیات (که در روش FSRE یک عملیات دسته‌بندی مد نظر می‌باشد) شناسایی می‌شوند. برای انجام مرحله دوم عملیات هرس‌سازی از الگوریتم ذکر شده شکل (۳-۸) استفاده می‌گردد.

```
For k = 1 to Number of hidden neuron
        Flag = True
        While (Flag is True)
                Flag = False
                For i = 1 to Number of connection to the kth neuron
                        Remove the ith connection
                        Apply data and compute the performance
                        If the Performance is the acceptable then
                                Flag = True
                        Else
                                Return the ith connection
                        End if
                End for
        End while
End for
```

شکل (۳-۸) پروسه هرس‌سازی اتصالات بین لایه‌های ورودی و مخفی

در این مرحله با انجام عملیات هرس‌سازی، اتصالات اضافی بین دو لایه ورودی و مخفی حذف گردیده و بدین صورت پارامترهای ضروری برای هر نورن مخفی (باقی مانده از مرحله هرس سازی اول) به‌صورت مجزا شناسایی می‌شوند.

این الگوریتم به‌صورت تکرار شونده[1] عمل می‌نماید. متغیر $Flag$ کنترل کننده تغییرات بوجود آمده در اتصالات دو لایه ورودی و مخفی است. با ایجاد یک تغییر در این ساختار (نظیر حذف یکی از اتصالات)، مقدار $Flag$، True شده و الگوریتم را از تغییر بوجود آمده مطلع می‌سازد. بدین‌ترتیب، این روش همه

[1] Repeated

اتصالات را یکبار دیگر با هدف دستیابی به نتایج بهتر بررسی نموده تا در صورت امکان اتصالات بیشتری را از ساختار شبکه عصبی حذف کند. اساس این الگوریتم به‌صورت حریصانه[1] می‌باشد که متغیر $Flag$ تا حدودی باعث بهبود آن شده است. منظور از محاسبه کارایی در خط هفتم این الگوریتم، استفاده از پارامترهای کارایی معرفی شده در روابط (۳-۵)، (۳-۶) و (۳-۷) از این فصل می‌باشد.

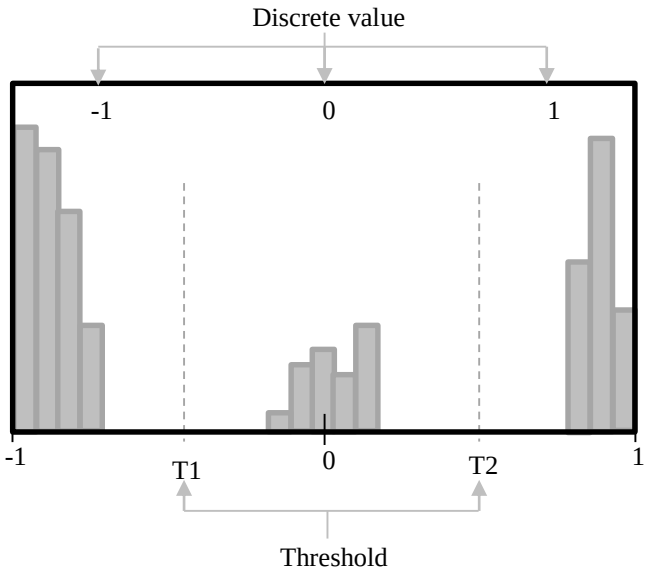
با انجام این مرحله از هرس‌سازی پارامترهای ضروری برای هر نرون لایه مخفی شناسایی شده و اتصالات غیر ضروری حذف می‌گردند. در صورتیکه همه اتصالات یک ورودی مشخص به همه نرون‌های لایه مخفی حذف گردد، بدین معنی است که این ویژگی در عملیات دسته‌بندی مورد نظر، ضروری نبوده و می‌توان آن را حذف کرد. این حالت را می‌توان در الگوریتم شکل (۳-۸) مشاهده نمود که باعث انتخاب ویژگی[2] از بین پارامترهای ورودی شده است. پس از هرس کردن، ساختار ساده شده‌ای از شبکه عصبی بدست می‌آید که می‌تواند مسئله موجود را با میزان کارایی و دقت قابل قبولی حل نماید.

یکی دیگر از مسائلی که باعث ایجاد ابهام در عملیات انجام شده توسط شبکه‌های عصبی می‌گردد، وجود مقادیر پیوسته در خروجی نرون‌های لایه مخفی است. برای رفع این مشکل، از روش‌های متنوع گسسته‌سازی استفاده می‌شود [۴۷]. در روش FSRE با توجه به توزیع مقادیر خروجی هر نرون و تعیین مقادیر آستانه برای آن، گسسته‌سازی مورد نظر را انجام می‌دهیم. در این روش ابتدا نمونه‌های موجود را به ساختار جدید شبکه عصبی اعمال کرده تا مقادیر خروجی برای هر نرون بدست آید. سپس با ترسیم این مقادیر (برای هر نرون به‌صورت مجزا) می‌توان مقادیر آستانه را با توجه به شرایط موجود تعیین نمود. شکل (۳-۹) نمونه‌ای از گسسته‌سازی این مقادیر را نشان می‌دهد.

این عملیات برای هر نرون از لایه مخفی به‌صورت مجزا انجام می‌شود و تعداد این مقادیر آستانه برای هر نرون می‌تواند متفاوت باشد. مقادیر خروجی بدست آمده برای هر نرون در لایه مخفی، بدلیل استفاده

[1] Greedy
[2] Feature selection

از تابع فعال‌ساز تانژانت هایپربولیک در محدوده ۱- تا ۱ قرار می‌گیرند که پس از گسسته‌سازی به محدوده‌های مشخص تقسیم می‌شوند. از این محدوده‌های گسسته بدست آمده در تعیین قوانین نهایی استفاده می‌شود. بنابراین در تعیین تعداد و محل این مقادیر آستانه باید دقت لازم بعمل آید.



شکل (۳-۹) گسسته‌سازی مقادیر خروجی یک نرون ساختگی نحوه تعیین تعداد و محل هر یک از مقادیر آستانه با توجه به توزیع داده‌ها انجام می‌شود.

با اقداماتی که تا این مرحله انجام می‌گیرد، ساختار شبکه عصبی تا حد امکان ساده گردیده و مقادیر خروجی هر نرون از لایه مخفی نیز در محدوده‌های گسسته‌ای شناسایی می‌شوند. در این شرایط درصورتیکه یک نمونه به شبکه عصبی موجود اعمال گردد، هر نرون در لایه مخفی یک مقدار گسسته (به جای مقادیر پیوسته) را برای این نمونه نشان می‌دهد. این مقادیر گسسته برای هر نرون در کنار هم "*الگوی*[1] *نمونه ورودی در لایه مخفی*" را نشان داده که برای استخراج قانون مورد استفاده قرار می‌گیرد.

۳-۲-۴ استخراج قانون

تا اینجا، ساختار شبکه عصبی پس از هرس شدن ساده گردیده و خروجی نرون‌های لایه مخفی گسسته‌سازی شده‌اند. آخرین فاز در روش FSRE انجام استخراج قوانین از ساختار نهایی شبکه عصبی

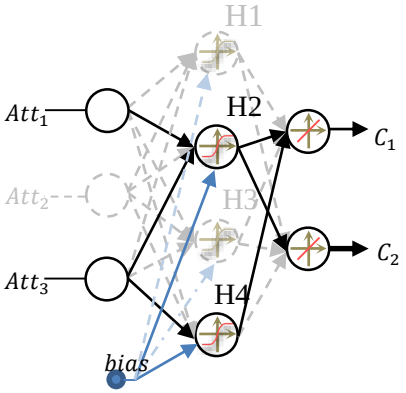
[1] Pattern

برای مسئله مورد نظر است. در روش پیشنهادی (که جزء روش‌های تجزیه‌ای می‌باشد) از الگوی نمونه‌های ورودی در لایه مخفی جهت تولید قوانین استفاده می‌گردد. در حالت کلی هر الگوی بدست آمده را می‌توان یک قانون در نظر گرفت، اما در اینصورت تعداد قوانین زیادی تولید گردیده و ترکیب این قوانین تولید شده نیز کاری دشوار و زمانبر خواهد بود. بنابراین در روش FSRE این الگوها پس از تولید، دسته‌بندی و مدیریت می‌شوند که باعث می‌شود مراحل تولید قانون به‌صورت بهینه‌تری انجام شود. این عملیات در چهار گام انجام می‌شود که در ادامه توضیح داده شده است.

گام اول) تعیین الگو برای نمونه‌های ورودی: در گام اول تمام نمونه‌های مجموعه آموزشی را به ترتیب به ساختار هرس شده شبکه عصبی اعمال می‌نماییم. سپس بررسی می‌شود که خروجی نرون‌های مخفی برای هر نمونه ورودی خاص در کدام محدوده گسسته قرار می‌گیرند. حال اگر اعداد گسسته بدست آمده برای هر نمونه ورودی را در کنار هم قرار دهیم، یک دنباله گسسته حاصل شده که الگوی آن نمونه ورودی نامیده می‌شود. این الگو برای هر نمونه با توجه به مقادیر ویژگی‌های موجود آن (به غیر از ویژگی‌های حذف شده) تولید می‌شود. شکل (۳-۱۰) این مرحله را نشان می‌دهد.

Input Sample			Output
1	5	7	C1
6	4	3	C2
5	1	3	C2
3	2	4	C2
2	7	1	C1
6	8	3	C2
2	6	7	C1

Hidden Neuron	Discrete Values
H1	pruned
H2	-1 1
H3	pruned
H4	-1 0 1



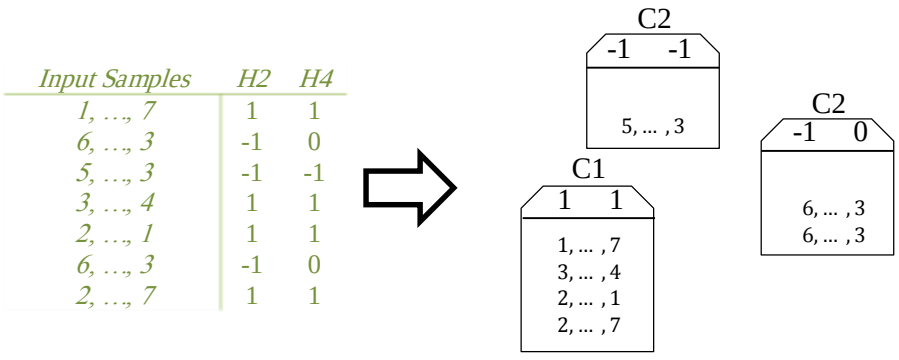
Input Samples	*H2*	*H4*
1, …, 7	1	1
6, …, 3	-1	0
5, …, 3	-1	-1
3, …, 4	1	1
2, …, 1	1	1
6, …, 3	-1	0
2, …, 7	1	1

شکل (۳-۱۰) تعیین الگوی لایه مخفی برای نمونه‌های ورودی. برای هر نرون لایه مخفی، مقادیر گسسته آن نشان داده شده است. ویژگی دوم از نمونه‌ها بدلیل حذف شدن از شبکه عصبی در نظر گرفته نشده است.

در مثال شکل (۳-۱۰) مجموعه داده ورودی دارای سه ویژگی متمایز و دو کلاس خروجی می‌باشد. شبکه عصبی آموزش دیده برای این مسئله، دارای چهار نرون در لایه مخفی بوده که پس از هرس‌سازی

شبکه عصبی تنها دو نرون آن (H2 و H4) باقی مانده‌اند. همچنین ویژگی ورودی دوم از داده‌ها نیز در زمان هرس‌سازی غیر ضروری تشخیص داده شده و حذف گردیده است. بنابراین از این ویژگی (و مقادیر آن) در مراحل بعد و استخراج قانون استفاده نمی‌شود. گسسته‌سازی نرون‌های لایه مخفی سبب ایجاد دو مقدار گسسته برای نرون H2 و سه مقدار گسسته برای نرون H4 گردیده است. با اعمال هر نمونه به این شبکه عصبی، الگوی آن نمونه در لایه مخفی بدست آمده است. در گام اول از استخراج قانون برای همه نمونه‌های موجود این الگو را تولید می‌نماییم. پس به ازای هر نمونه از مجموعه داده آموزشی، یک الگو بدست می‌آید.

گام دوم) دسته‌بندی نمونه‌های ورودی بر اساس الگو: برای برخی از نمونه‌های ورودی، در لایه مخفی مقادیر گسسته شده مشابهی بدست می‌آید. این مقادیر مشابه، الگوی یکسانی را در این لایه برای این نمونه‌ها تولید می‌نماید. هر کدام از این الگوهای بدست آمده بر اساس خروجی شبکه عصبی، متعلق به یک کلاس خروجی مشخص است. در این گام، برای هر الگوی بدست آمده از لایه مخفی، یک دسته جدا در نظر گرفته می‌شود. نمونه‌های ورودی نیز براساس الگوی لایه مخفی آنها در این دسته‌ها قرار می‌گیرند. بنابراین در هر دسته نمونه‌های ورودی مشابه قرار گرفته که اکثراً متعلق به یک کلاس خاص می‌باشند. شکل (۳-۱۱) این گام از استخراج قانون در روش FSRE را نشان می‌دهد.



شکل (۳-۱۱) دسته‌بندی نمونه‌ها بر اساس مدل بدست آمده از آنها

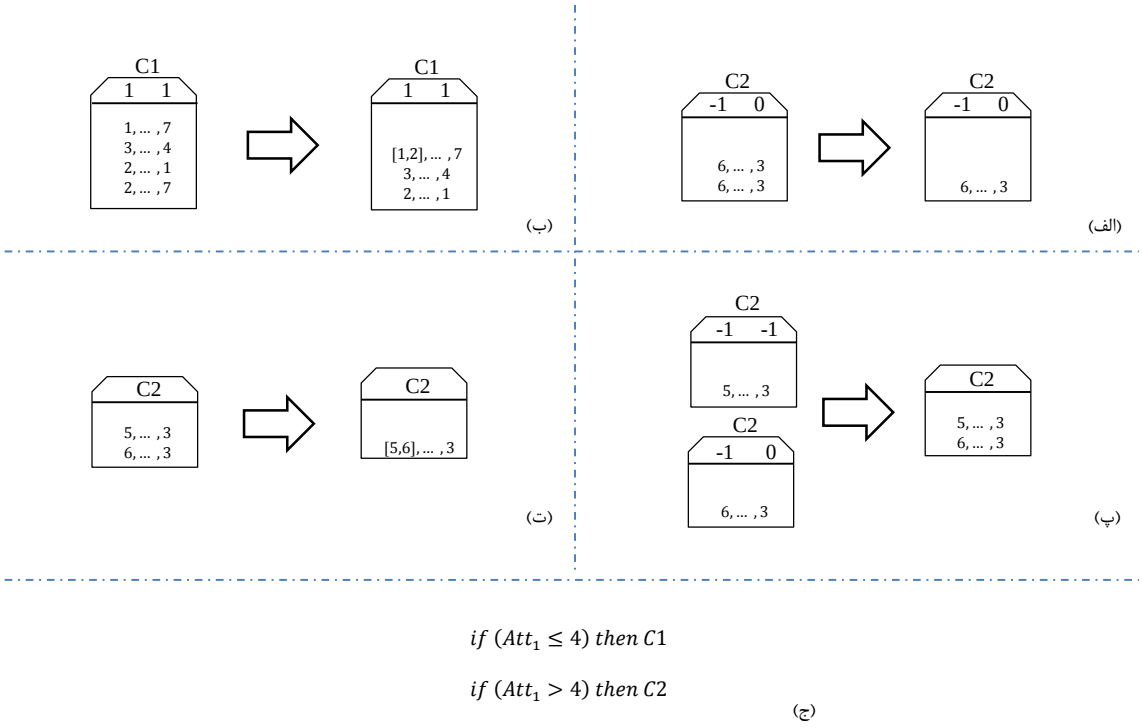
در ادامه مثال ارائه شده (شکل (۳-۱۱)) مشاهده می‌شود که سه نوع الگوی متفاوت برای نمونه‌ها ایجاد شده است. بنابراین سه دسته برای این نمونه‌ها ایجاد گردیده و هر نمونه بر اساس الگوی آن در یکی از این دسته‌ها قرار می‌گیرد. هر دسته تولید شده بر اساس خروجی شبکه عصبی برای نمونه‌های آن دسته، به یکی از کلاس‌های خروجی تعلق می‌یابد.

گام سوم) هرس‌سازی و ترکیب کردن: هر یک از دسته‌های ایجاد شده در لایه قبل معمولاً شامل نمونه‌هایی است که (اکثراً) متعلق به یک کلاس خاص هستند. ممکن است برای یک کلاس خروجی خاص، بیش از یک دسته در مرحله دوم تولید گردد. از طرفی شاید در برخی از مجموعه‌ها، نمونه‌های مشابهی (تکراری) وجود داشته باشند. این نمونه‌های تکراری بدلیل حذف برخی از پارامترهای ورودی در شبکه عصبی و مشابه بودن مقادیر پارامترهای باقیمانده در نمونه‌ها بوجود آمده است. بنابراین برای تولید مجموعه قوانین نهایی نیاز به مدیریت این مجموعه‌ها می‌باشد. در این گام:

- نمونه‌های تکراری در هر دسته حذف می‌شوند. درشکل (۳-۱۲) قسمت الف، نمونه‌های 6,8,3 و 6,4,3 از مثال ارائه شده بدلیل حذف ویژگی دوم در زمان هرس‌سازی شبکه عصبی، مشابه هم می‌باشند. در این حالت یکی از این نمونه‌ها حذف می‌گردند.
- نمونه‌های مشابه در هر دسته ترکیب شده و حالت (یا قانون) جامع‌تری را پدید می‌آورند. در شکل (۳-۱۲) قسمت ب، نمونه‌های 1,…,7 و 2,…,7 بدلیل دارا بودن مقدار یکسان در ویژگی سوم و مقادیر متوالی در ویژگی اول در این مثال، با هم ترکیب شده‌اند.
- دسته‌هایی که متعلق به کلاس خروجی یکسان هستند را با هم ترکیب می‌نماییم. در شکل (۳-۱۲) قسمت پ، دو دسته تولید شده برای کلاس C2 از این مثال با یکدیگر ترکیب شده‌اند.
- در صورت وجود حالت‌های (یا قوانین) مشابه در دسته‌های جدید، آنها را ترکیب/هرس کرده و حالت (قوانین) عمومی‌تری را تولید می‌نماییم. در شکل (۳-۱۲) قسمت ت، نمونه‌های 5,…,3

و [6,…,3] در دسته جدید تولید شده، بدلیل دارا بودن مقدار یکسان در ویژگی سوم و مقادیر متوالی در ویژگی اول در این مثال، با هم ترکیب شده‌اند.

- مجموعه قوانین دسته‌بندی برای هر کلاس خروجی را تعیین می‌کنیم.



شکل (۳-۱۲) مرحله هرس‌سازی و ترکیب کردن برای تولید قوانین

در مثال ارائه شده با توجه به دسته بدست آمده برای کلاس C1 (شکل (۳-۱۲) قسمت ب) و دسته تولید شده برای کلاس C2 (شکل (۳-۱۲) قسمت پ) قوانین نهایی برای انجام دسته‌بندی نمونه‌ها تولید گردیده است (شکل (۳-۱۲) قسمت ج). در این مثال مقادیر ویژگی اول برای دو کلاس خروجی کاملاً از هم جدا بود و نیازی به استفاده از ویژگی سوم در تولید قوانین نیست.

گام چهارم) قانون پیش‌فرض[1]: در این گام یکی از دسته‌ها به‌عنوان پاسخ پیش‌فرض مسئله در نظر گرفته می‌شود. معمولاً برای تعیین قانون پیش‌فرض، دسته‌ای را که دارای حداقل یکی از پارامترهای: بیشترین تعداد نمونه، بیشترین میزان خطا در زمان شناسایی و یا بیشترین تعداد قانون تولید شده برای آن، باشد

[1] Default rule

را تعیین نموده و به‌عنوان پاسخ پیش‌فرض مسئله در نظر گرفته می‌شود. استفاده از قانون پیش‌فرض از یک سو باعث کاهش تعداد قوانین مورد نیاز برای مسئله موجود شده و از سوی دیگر برای نمونه‌هایی که قانونی تولید نشده است (مثلاً برای نمونه‌های پرت) حداقل یک پاسخ بدست می‌آید. به‌عنوان نمونه در مثال ارائه شده در این بخش، کلاس $C1$ بدلیل زیادتر بودن نمونه‌های آن به‌صورت پاسخ پیش‌فرض در نظر گرفته می‌شود. نتایج نهایی همانند رابطه (۳-۸) بدست می‌آید.

$$\begin{gathered} If\ (Att_1 > 4)\ Then\ C2 \\ Default\ is\ C1 \end{gathered} \qquad (۳-۸)$$

در نهایت مجموعه قوانین دسته‌بندی آماده می‌باشد و می‌تواند بر روی مجموعه داده دیده نشده (مجموعه تست) اعمال شود. درصورتیکه کارایی بدست آمده از اعمال این قوانین قابل قبول باشد، آنگاه می‌توان از این مجموعه قوانین، به جای شبکه عصبی، بر روی داده‌های جدید استفاده کرد.

۳-۳ نتیجه گیری

در این فصل، روش جدیدی جهت استخراج قانون از شبکه عصبی تحت عنوان *روش استخراج قانون چهار مرحله‌ای* (FSRE) معرفی شد. هدف از این روش، کاهش پیچیدگی به همراه افزایش دقت در عملیات استخراج قانون بوده که بتوان از این روش بر روی مسائل نسبتاً پیچیده نیز استفاده نمود. عملیات استخراج قانون در این روش طی چهار فاز کلی انجام می‌شود که شامل *پیش‌پردازش* برای آماده‌سازی داده‌ها، *مدل یادگیری* به‌منظور انجام عملیات دسته‌بندی، *هرس‌سازی* برای ساده‌سازی ساختار شبکه عصبی و *استخراج قانون* برای تولید نتایج نهایی می‌باشد. مرحله استخراج قانون خود شامل چهار گام بوده، ابتدا مدل نمونه‌های ورودی در لایه مخفی تولید می‌گردد. سپس نمونه‌های ورودی بر اساس مدل‌های تولید شده در لایه مخفی، دسته‌بندی می‌گردد. گام سوم شامل مدیریت و هرس‌سازی دسته‌ها بوده که نهایتاً قوانین مورد نظر برای هر دسته تولید می‌گردد. گام چهارم نیز مربوط به تعیین

قانون پیش‌فرض برای مسئله مورد نظر می‌باشد. در فصل‌های بعدی این پایان نامه، از روش FSRE برای تولید قوانین دسته‌بندی در برخی مجموعه داده‌های رایج استفاده شده و کارایی این روش مورد بررسی قرار می‌گیرد.

فصل چهارم

بررسی و تحلیل نتایج

۴-۱ مقدمه

در این فصل به بررسی و آزمایش روش FSRE در استخراج قانون از شبکه عصبی برای مجموعه داده‌های استاندارد می‌پردازیم. مسائل مورد بررسی در این فصل نسبتاً ساده بوده و بیشتر به جهت مقایسه نتایج این روش نسبت به روش‌های موجود می‌باشد. نتایج بدست آمده برای هر مجموعه داده به‌صورت مجزا و با ذکر جزئیات در این فصل ارائه شده و در پایان فصل نیز این نتایج با نتایج حاصل از دیگر روش‌های استخراج قانون مورد مقایسه قرار می‌گیرد.

نتایج حاصل از دسته‌بندی در این فصل توسط ماتریس درهم ریختگی[1] نمایش داده می‌شود. این ماتریس، یک ماتریس مربعی است که تعداد سطر و ستون آن به تعداد کلاس‌های موجود در مسئله مورد نظر است. توسط این ابزار به سادگی می‌توان میزان کارایی شبکه‌های عصبی و خطای موجود در دسته‌بندی نمونه‌های مختلف را بررسی کرد. جدول (۴-۱) مثالی از این ماتریس را نمایش می‌دهد.

جدول (۴-۱) نمونه‌ای از جدول در هم ریختگی (Confusion Matrix)

نام مجموعه داده		دسته تخمین زده شده		
		گربه	سگ	خرگوش
دسته واقعی	گربه	۵	۳	۰
	سگ	۲	۳	۱
	خرگوش	۰	۲	۱۱

در این ماتریس، مجموع مقادیر یک سطر خاص نشان دهنده تعداد نمونه‌های واقعی در کلاس مورد نظر از مجموعه داده می‌باشد. درحالیکه مجموع مقادیر هر ستون، بیانگر تعداد نمونه‌های تخمین زده شده توسط شناسایی کننده برای یک کلاس خاص می‌باشد.

[1] Confusion matrix

در این قسمت مجموعه داده‌های Simple Classes، Play Golf، Wine، Glass، Iris Flower و Wisconsin Breast Cancer برای ارزیابی میزان کارایی روش FSRE از مجموعه پایگاه‌های داده UCI [۵۶] تهیه و مورد استفاده قرار گرفتند. همچنین روش FSRE و آزمایشات انجام شده توسط نرم افزار Matlab شبیه سازی شده‌اند.

۴-۲ نتایج بدست آمده از مجموعه داده Simple Classes

مجموعه داده Simple Classes یک مجموعه داده مصنوعی[1] از داده‌ها است که به‌منظور انجام عملیات دسته‌بندی مورد استفاده قرار می‌گیرد. هدف از آزمایش این مجموعه داده، ارزیابی روش FSRE و ایجاد اطمینان در مورد کارایی و دقت آن است. این مجموعه داده به‌صورت پیش‌فرض در نرم‌افزار Matlab وجود دارد و توسط دستور زیر، بارگذاری شده و قابل دسترس می‌باشد.

```
load('simpleclass_dataset');
```
(۱-۴)

مجموعه Simple Classes دارای ۱۰۰۰ نمونه ساختگی با ۲ ویژگی ورودی است که در۴ کلاس خروجی دسته‌بندی می‌شوند (جدول (۴-۲)). توزیع نمونه‌های این مجموعه داده در کلاس‌های موجود تقریباً به‌صورت برابر انجام شده است و تعداد دقیق آن در جدول (۴-۲) مشاهده می‌شود.

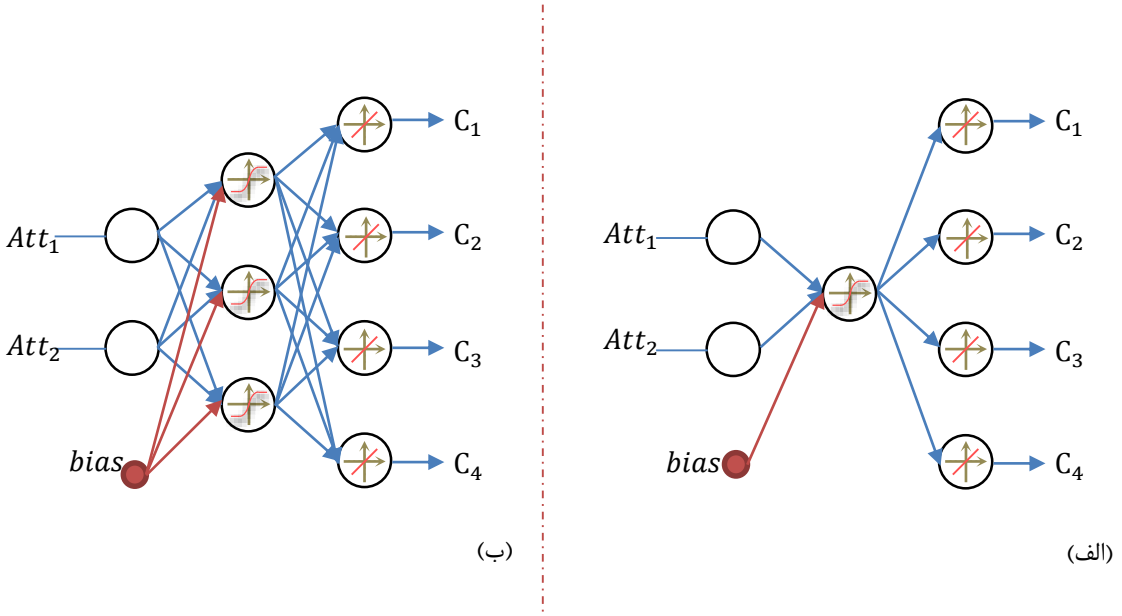
جدول (۴-۲) اطلاعات کلی در مورد مجموعه داده Simple Classes

نام مجموعه داده	**Simple Classes**			
تعداد کلاس‌های خروجی	4			
تعداد ویژگی‌های ورودی	2			
تعداد نمونه‌ها	1000			
تعداد نمونه در هر کلاس	C4 = 277	C3 = 233	C2 = 247	C1 = 243

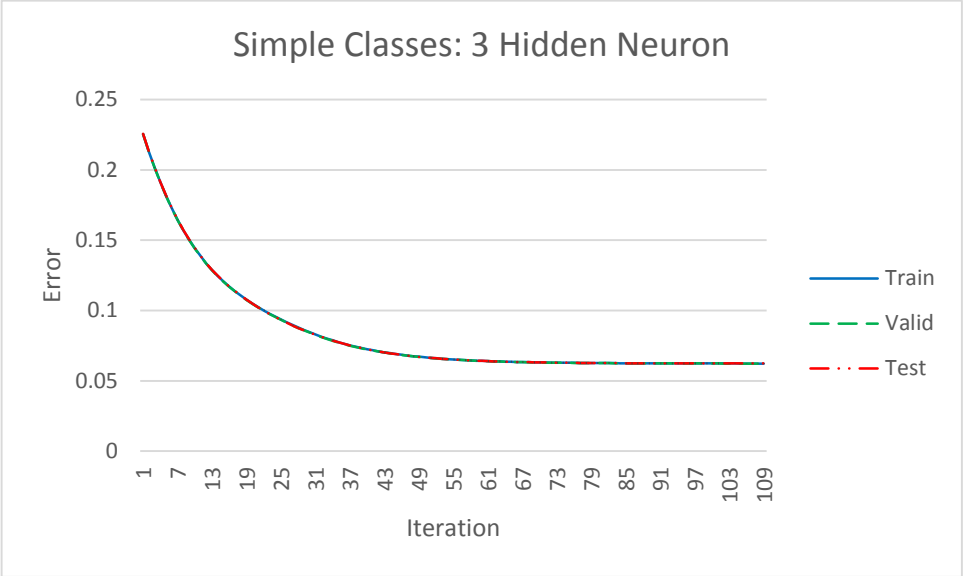
[1] Artificial dataset

نمونه‌های این مجموعه داده، در محدوده [0 1] با توزیع نرمال تولید شده‌اند و به همین دلیل نیازی به انجام پیش‌پردازش نمی‌باشد. از این مجموعه داده ۵۰% را به‌عنوان مجموعه آموزش، ۲۵% را به‌عنوان مجموعه ارزیابی و ۲۵% باقیمانده را به‌عنوان مجموعه تست (آزمایش) انتخاب می‌کنیم. نحوه انتخاب نمونه‌ها بگونه‌ای است که از هر کلاس ۵۰% در مجموعه آموزشی، ۲۵% در مجموعه ارزیابی و ۲۵% دیگر در مجموعه تست قرار گیرد.

در فاز مدل یادگیری ابتدا از یک شبکه عصبی با ساختار ۴-۱-۲ برای تعیین ساختار مناسب شبکه عصبی برای این مسئله استفاده گردید. پارامترهای این شبکه عصبی، نحوه اتصالات بایاس و مقادیر وزن‌ها همانند مقادیر و پارامترهای توضیح داده شده در بخش ۳-۲-۲ در نظر گرفته شد. نتایج بدست آمده نشان داد که این شبکه عصبی قادر است با ۳ نرون در لایه مخفی داده‌ها را به‌صورت کامل شناسایی کند. شکل (۴-۱) این ساختار را نشان می‌دهد.



شکل (۴-۱) ساختار شبکه عصبی مورد استفاده برای مجموعه داده Simple Classes الف) ساختار ابتدایی. ب) ساختار نهایی



شکل (۴-۲) نمودار خطای حاصل از آموزش شبکه عصبی به‌منظور حل مسئله Simple Classes

شکل (۴-۲) روند آموزش شبکه عصبی با ساختار نهایی و میزان خطای بدست آمده از آن را بر روی مجموعه داده آموزشی، ارزیابی و تست را نشان می‌دهد. همچنین نتایج بدست آمده از اعمال داده‌ها پس از انجام آموزش به شبکه عصبی مورد نظر نیز در جدول (۴-۳) آمده است. در این شبکه عصبی پس از اعمال یک نمونه ورودی، هر نرون لایه خروجی که مقدار خروجی بیشتری را تولید کند، نشان دهنده کلاس خروجی برای نمونه مورد نظر است. در واقع ماکزیمم مقداری که در لایه خروجی شبکه عصبی تولید شود کلاس خروجی را برای نمونه ورودی تعیین می‌کند. این نتایج نشان می‌دهد که این شبکه عصبی با تعداد سه نرون در لایه مخفی به خوبی در این مسئله عمل کرده و نتایج بدست آمده ایده‌آل می‌باشند.

نتایج بدست آمده در جدول (۴-۳) نشان می‌دهد که ساختار انتخاب شده برای شبکه عصبی قادر است مسئله مورد نظر را با میزان دقت ۱۰۰% و ۱۰۰% حل نماید. در مرحله بعد باید ساختار شبکه عصبی را تا حد امکان ساده کرده و هرگونه اتصال و نرون غیر ضروری را از ساختار این شبکه عصبی حذف نماییم. با اعمال الگوریتم‌های بیان شده در بخش ۳-۲-۳ جهت هرس‌سازی شبکه عصبی، ساختار هرس شده شبکه عصبی به‌صورت شکل (۴-۳) بدست می‌آید.

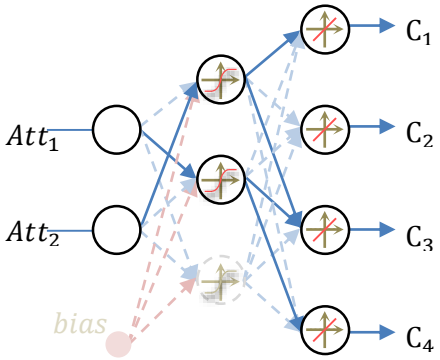
جدول (۴-۳) نتایج بدست آمده از اعمال مجموعه داده‌ها به شبکه عصبی آموزش دیده برای مسئله Simple Classes

Train	C1’	C2’	C3’	C4’
C1	121	0	0	0
C2	0	123	0	0
C3	0	0	117	0
C4	0	0	0	139

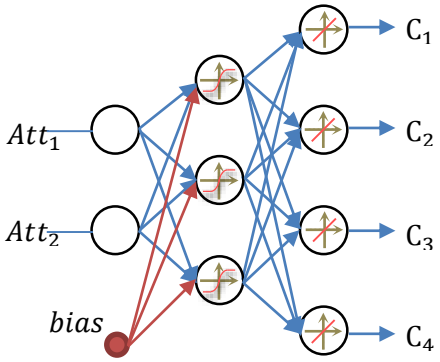
Valid	C1’	C2’	C3’	C4’
C1	61	0	0	0
C2	0	62	0	0
C3	0	0	58	0
C4	0	0	0	69

Test	C1’	C2’	C3’	C4’
C1	61	0	0	0
C2	0	62	0	0
C3	0	0	58	0
C4	0	0	0	69

All	C1’	C2’	C3’	C4’
C1	243	0	0	0
C2	0	247		0
C3	0	0	233	0
C4	0	0	0	277



(ب)



(الف)

شکل (۴-۳) ساختار شبکه عصبی مورد استفاده برای مسئله Simple Classes قبل (الف) و بعد (ب) از هرس سازی

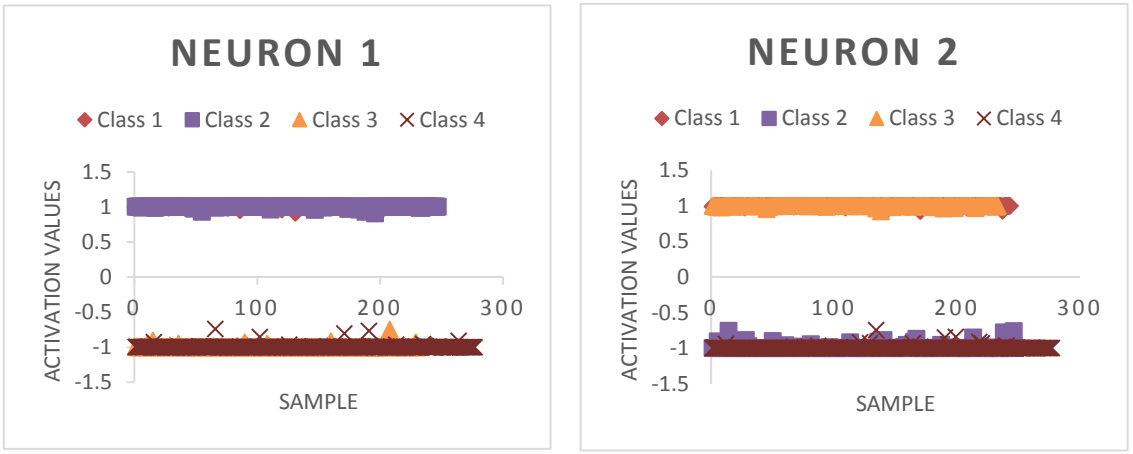
همانطور که در شکل (۴-۳) مشاهده می‌شود، یال‌های متصل به نرون خروجی کلاس C_2 همگی حذف شده‌اند. در این شرایط مقدار خروجی این نرون همواره صفر خواهد بود و درصورتیکه مقدار خروجی دیگر نرون‌های این لایه برای نمونه ورودی کمتر از صفر باشد، این نمونه به کلاس C_2 تعلق می‌گیرد. با اینحال، باز هم نتایج بدست آمده نشان داد که شبکه عصبی با ساختار هرس شده به‌صورت کاملاً ایده‌آل و همانند حالت اتصال کامل عمل می‌نماید. جدول (۴-۴) نتیجه اعمال همه نمونه‌ها را به شبکه عصبی هرس شده نشان می‌دهد.

جدول (۴-۴) نتایج حاص از اعمال همه نمونه‌های موجود به شبکه عصبی هرس شده برای مسئله Simple Classes

All	C1’	C2’	C3’	C4’
C1	243	0	0	0
C2	0	247	0	0
C3	0	0	233	0
C4	0	0	0	277

در این قسمت با گسسته‌سازی مقادیر خروجی در نرون‌های لایه مخفی می‌توان قوانین دسته‌بندی را برای این مجموعه داده بدست آورد.

شکل (۴-۴) مقادیر خروجی در این لایه را نشان می‌دهد.



شکل (۴-۴) مقادیر خروجی نرون‌های باقیمانده در لایه مخفی برای مسئله Simple Classes

البته بدلیل اینکه این مجموعه داده دارای نمونه‌ها و مقادیر مصنوعی می‌باشد، مشاهده می‌شود که خروجی نرون‌های لایه مخفی کاملاً مرتب و بدون پراکندگی است. بنابراین برای این مجموعه داده، مقادیر خروجی را به دو دسته کمتر از صفر و بیشتر از صفر تقسیم می‌شوند. در نهایت با استخراج قانون از این مجموعه داده نتایج بدست آمده به‌صورت رابطه (۴-۲) می‌باشند.

$$
\begin{aligned}
&if\ (0.1622 \le X_1 \le 1.6520)\ and\ (-1.7358 \le X_2 \le -0.2749)\ then\ C_3 \\
&if\ (-1.7196 \le X_2 \le -0.2783)\ then\ C_1 \\
&if\ (0.1579 \le X_1 \le 1.6335)\ then\ C_4 \\
&Default\ \rightarrow C_2
\end{aligned}
\qquad (۴-۲)
$$

این قوانین بدست آمده قادرند که تمامی نمونه‌ها را بدرستی شناسایی و دسته‌بندی نمایند. جدول (۴-۵) نتایج بدست آمده از این قوانین را بر روی مجموعه داده مورد نظر نشان می‌دهد.

جدول (۴-۵) نتایج حاصل از آزمایش قوانین بدست آمده بر روی مجموعه داده Simple Classes

All	C1’	C2’	C3’	C4’
C1	243	0	0	0
C2	0	247	0	0
C3	0	0	233	0
C4	0	0	0	277

مشاهده می‌شود که روش مذکور به خوبی قادر به تولید قوانین مناسب می‌باشد و می‌تواند نتایج مورد نظر را ایجاد نماید.

۴-۳ نتایج بدست آمده از مجموعه داده Play Golf

مجموعه داده Play Golf[1] که به نام Weather نیز شناخته می‌شود، یکی از مجموعه داده‌های رایج در مبحث دسته‌بندی است. ویژگی‌های این مجموعه داده که امروزه به‌منظور آزمایش و بررسی شناسایی کننده‌ها مورد استفاده فراوان قرار می‌گیرد، در جدول (۴-۶) آمده است. در این مجموعه داده، با تعیین پارامترهای متعدد آب و هوایی، انجام یا عدم انجام بازی گلف مورد بررسی قرار گرفته است. نمونه‌های این مجموعه در دو کلاس yes و no به معنای انجام و عدم انجام بازی گلف دسته‌بندی می‌شوند که تعداد نمونه‌ها در هر کلاس نیز در جدول (۴-۶) آمده است.

همانطور که در جدول (۴-۶) مشاهده می‌شود، این مجموعه داده شامل شش ویژگی بوده که دو ویژگی آن به نام‌های Temp و Humidity در دو نوع عددی[2] و اسمی[3] در مجموعه داده وجود دارند. در این

[1] http://gerardnico.com/wiki/data_mining/weather
[2] Numeric
[3] Nominal

آزمایش از مقادیر عددی برای این دو ویژگی‌ها استفاده شده است. بدین‌ترتیب ویژگی‌های استفاده شده در این عملیات به‌صورت رابطه (۴-۲) می‌باشد.

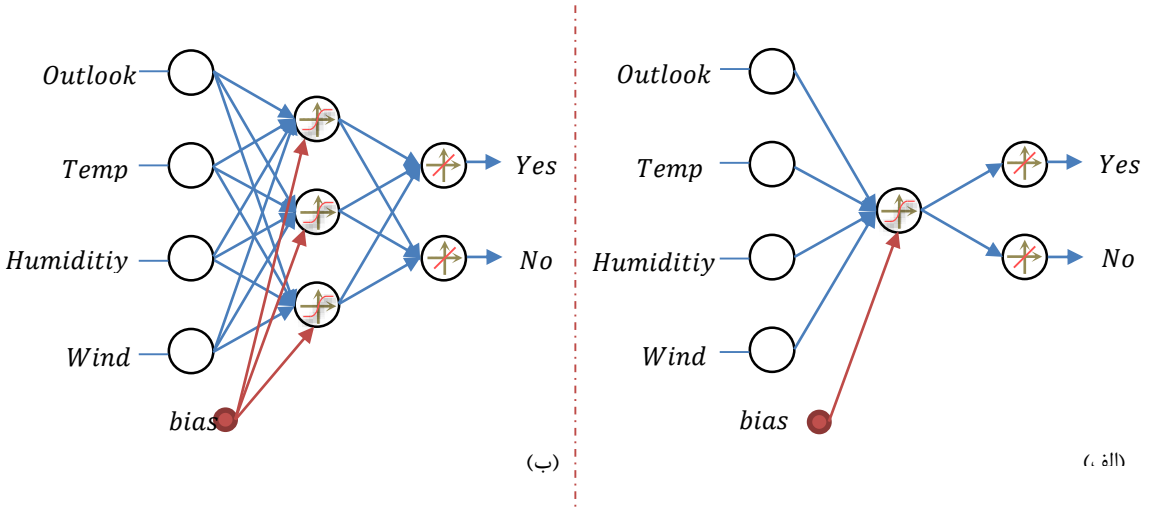
جدول (۴-۶) اطلاعات کلی در مورد مجموعه داده Play Golf

نام مجموعه داده	Play Golf (Weather)	
تعداد کلاس‌های خروجی	2	
تعداد ویژگی‌های ورودی	6	
تعداد نمونه‌ها	14	
تعداد نمونه در هر کلاس	No = 5	Yes = 9

$$Outlook, Temp_{(Numeric)}, Humidity_{(Numeric)}\ and\ Windy \rightarrow\ Play\ Golf\ ???$$ (۴-۲)

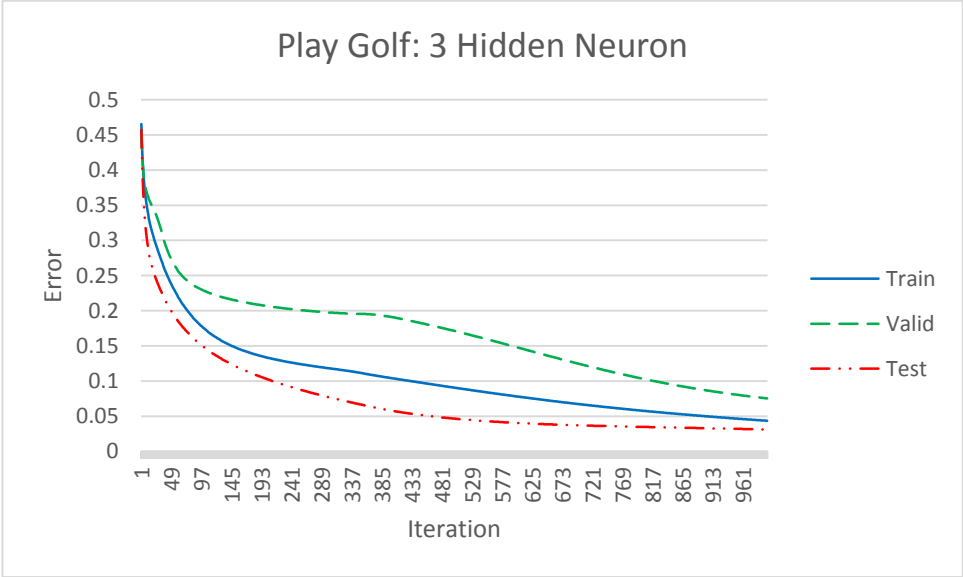
سپس از روابط معرفی شده در بخش ۳-۲-۱ به‌منظور نرمال‌سازی مقادیر این مجموعه داده استفاده شده است. در نهایت این مجموعه به سه بخش آموزش، ارزیابی و تست تقسیم می‌شوند و در این تقسیم‌بندی ۷۰% از داده‌ها به مجموعه آموزش و ۱۵% به مجموعه ارزیابی و ۱۵% باقیمانده را به مجموعه تست تخصیص می‌دهیم.

در فاز دوم از روش FSRE نیاز است تا یک شبکه عصبی مصنوعی به‌عنوان یک مدل شناسایی کننده برای دسته‌بندی نمونه‌های این مجموعه ایجاد شود. در این قسمت ابتدا یک شبکه عصبی با ساختار ۲-۴-۱ همانند شکل (۴-۴ الف) ایجاد گردیده و پارامترهای آن را بر اساس بخش ۳-۲-۲ تعیین می‌نماییم.



شکل (۴-۵) ساختار شبکه عصبی مورد استفاده برای مجموعه داده Play Golf. الف) ساختار ابتدایی. ب) ساختار نهایی

نتایج بدست آمده از آموزش این شبکه عصبی با ساختارهای متفاوت نشان داده است که با تعداد سه نرون در لایه مخفی (شکل (۴-۵) ب) می‌توان عملیات دسته‌بندی را با میزان دقت کاملاً ایده‌آل (۱۰۰%) انجام داد. در شکل (۴-۶) میزان خطای بدست آمده از شبکه عصبی در تکرارهای متوالی از مرحله آموزش این شبکه عصبی با سه نرون در لایه مخفی نشان داده شده است.



شکل (۴-۶) نمودار خطای حاصل از آموزش شبکه عصبی به‌منظور حل مسئله Play Golf

برای درک بهتر نمودار نشان داده شده در شکل (۴-۶) نتایج عددی بدست آمده از آموزش شبکه عصبی بر روی مجموعه داده‌های آموزش، ارزیابی و تست در جدول (۴-۷) نشان داده شده است.

جدول (۴-۷) نتایج بدست آمده از اعمال مجموعه داده‌ها به شبکه عصبی آموزش دیده برای مسئله Play Golf

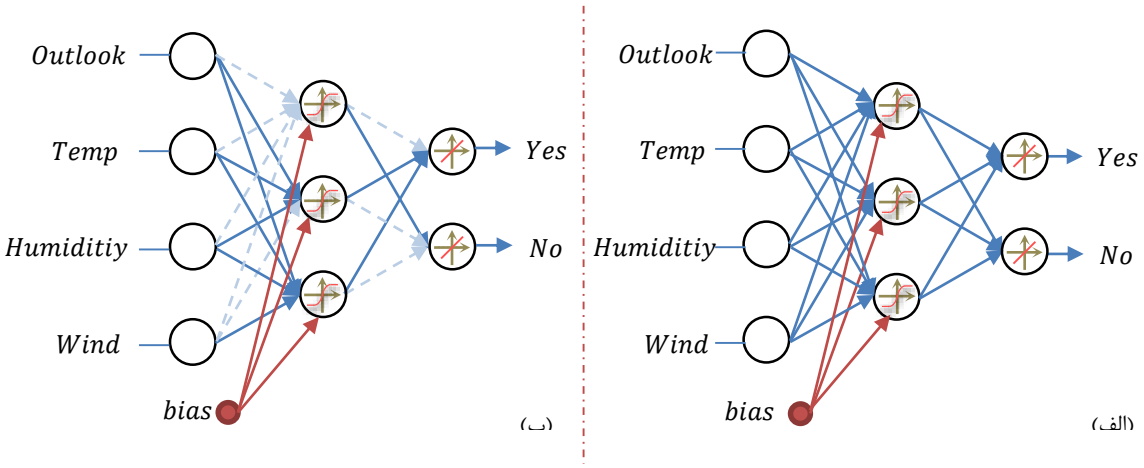
Train	Yes’	No’
Yes	7	0
No	0	3

Valid	Yes’	No’
Yes	1	0
No	0	1

Test	Yes’	No’
Yes	1	0
No	0	1

All	Yes’	No’
Yes	9	0
No	0	5

این نتایج حاکی از دقت خوب شبکه عصبی مورد نظر با تعداد سه نرون در لایه مخفی است. نمونه‌های هر دو دسته موجود (انجام بازی و عدم انجام بازی) به‌صورت کاملاً صحیح شناسایی شدند. بنابراین نتایج این آزمایش دارای میزان حساسیت و اختصاصی بودن ۱۰۰% در این مرحله بوده است. سپس به‌منظور ساده‌سازی این شبکه عصبی، اتصالات و نرون‌های غیر ضروری را از ساختار آن هرس می‌کنیم. شکل (۴-۷) ساختار شبکه عصبی را قبل و بعد از هرس‌سازی نشان می‌دهد.



شکل (۴-۷) ساختار شبکه عصبی برای مسئله Play Golf قبل (الف) و بعد (ب) از هرس‌سازی

همانطور که مشاهده می‌شود در این مسئله نیز همانند مسئله قبل (Simple Classes) یکی از نرون‌های لایه خروجی عملاً از ساختار شبکه عصبی کنار گذاشته می‌شود؛ اما در عملیات استخراج قانون خللی

ایجاد نمی‌کند. جدول (۴-۸) دقت شبکه عصبی در حالت هرس شده را با توجه به مجموعه داده موجود نشان می‌دهد.

جدول (۴-۸) دقت شبکه عصبی هرس شده برای مسئله Play Golf

All	Yes’	No’
Yes	9	0
No	0	5

این نتایج نشان می‌دهد که این شبکه عصبی پس از هرس‌سازی نیز قادر است که همه نمونه‌ها را به‌صورت درست شناسایی و دسته‌بندی کند. در نهایت با انجام استخراج قانون از ساختار هرس شده شبکه عصبی و با توجه به مجموعه داده مورد نظر، قوانین بدست آمده به‌صورت رابطه (۴-۳) می‌باشد.

$$
\begin{aligned}
&if\ (X_1 = 3)\ and\ (X_3 \geq 85)\ then\ NO_{(don't\ play)} \\
&if\ (X1 = 2)\ and\ (X_4 = 1)\ then\ NO_{(don't\ play)} \\
&Default\ \rightarrow YES_{(play)}
\end{aligned}
\qquad (۴-۳)
$$

با بررسی نتایج بدست آمده بر روی مجموعه داده موجود مشخص می‌شود که این قوانین به خوبی قادرند نمونه‌ها را شناسایی نمایند. نتیجه این آزمایش را در جدول (۴-۹) مشاهده می‌کنید.

جدول (۴-۹) نتایج حاصل از آزمایش قوانین بدست آمده برای مجموعه Play Golf

All	Yes’	No’
Yes	9	0
No	0	5

بدین‌ترتیب مشخص می‌شود که در این مسئله قوانین بدست آمده با میزان دقت ۱۰۰% می‌توانند به جای شبکه عصبی بر روی این داده‌ها اعمال شوند.

۴-۴ نتایج بدست آمده از مجموعه داده Wine

مجموعه داده Wine[1] یکی دیگر از مجموعه داده‌های دسته‌بندی می‌باشد که به‌منظور آزمایش شناسایی کننده‌های مختلف، به‌صورت فراوان استفاده می‌گردد. این مجموعه داده برای شناسایی ۳ نوع نوشیدنی مختلف، از ۱۳ ویژگی پیوسته در ورودی استفاده می‌کند. این مجموعه داده در نرم افزار Matlab به‌صورت پیش‌فرض وجود دارد و برای بارگذاری آن تنها لازم است که از دستور رابطه (۴-۴) استفاده شود.

$$load('wine\_dataset'); \quad (۴-۴)$$

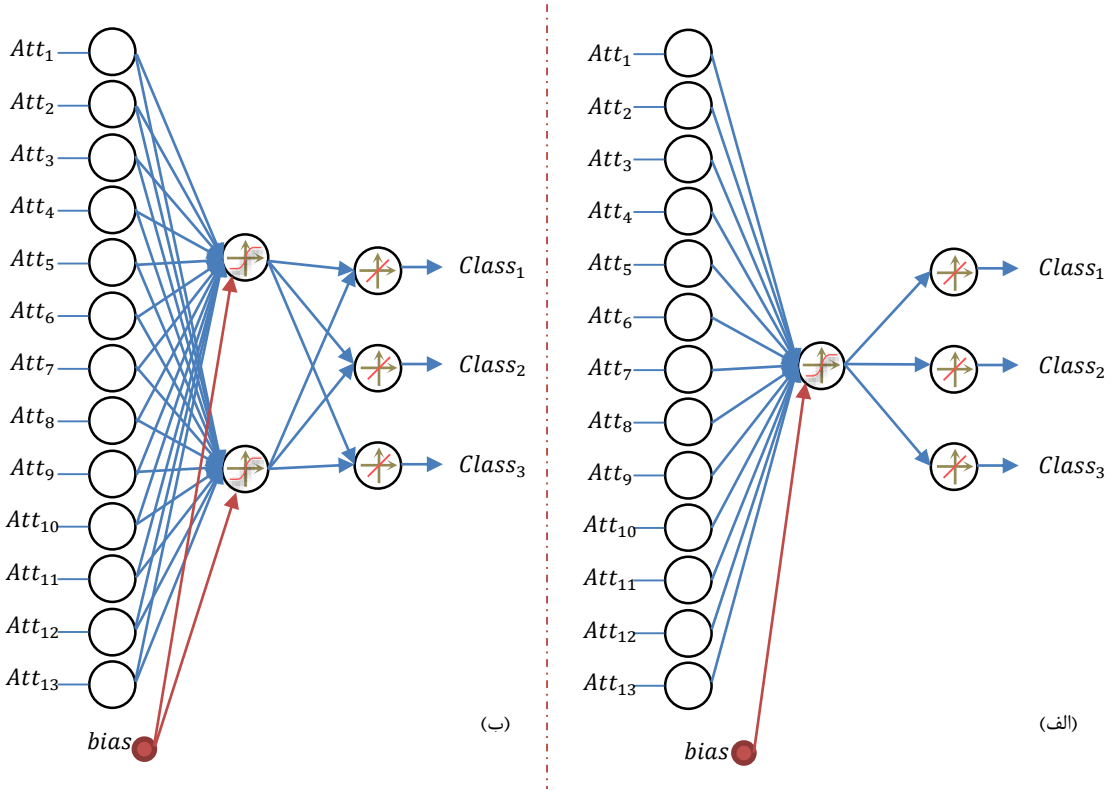
اطلاعات کلی از این مجموعه داده در جدول (۴-۷) آمده است. این مجموعه داده پس از نرمال‌سازی به سه مجموعه آموزش (۵۰%)، ارزیابی (۲۵%) و تست (۲۵%) تقسیم شد. سپس برای اجرای فاز دوم از عملیات استخراج قانون، یک شبکه عصبی با ساختار ابتدایی ۳-۱-۱۳ ایجاد می‌شود. شکل (۴-۸ الف) این شبکه عصبی را نشان می‌دهد. از مجموعه آموزش و ارزیابی به ترتیب برای آموزش شبکه عصبی و تعیین ساختار مناسب آن استفاده می‌گردد.

جدول (۴-۱۰) اطلاعات کلی در مورد مجموعه داده Wine

نام مجموعه داده	Wine		
تعداد کلاس‌های خروجی	3		
تعداد ویژگی‌های ورودی	13		
تعداد نمونه‌ها	178		
تعداد نمونه در هر کلاس	C3 = 48	C2 = 71	C1 = 59

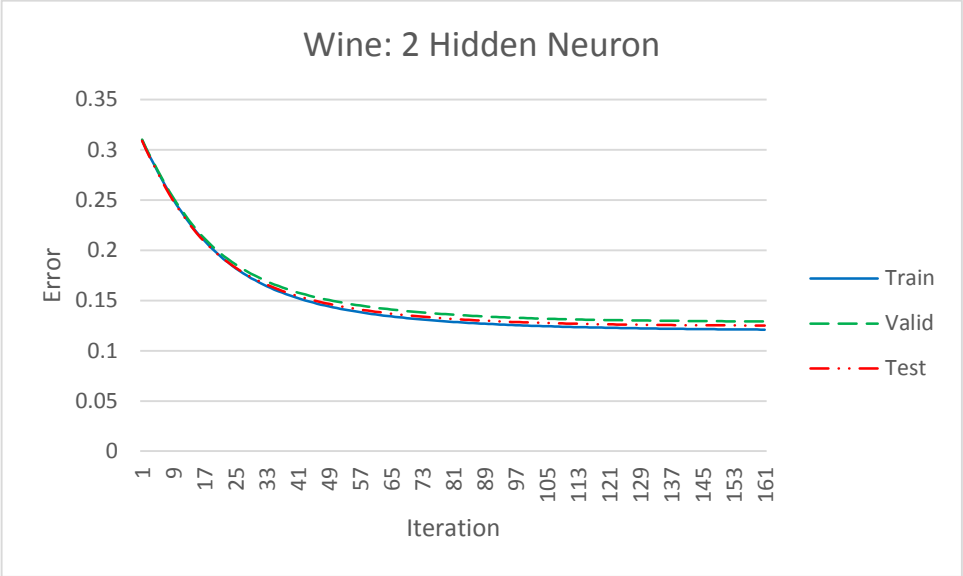
[1] https://archive.ics.uci.edu/ml/datasets/Wine

توسعه ساختار شبکه عصبی که با اضافه کردن نرون به لایه مخفی انجام می‌شود، باعث افزایش دقت شبکه عصبی در دسته‌بندی نمونه‌ها می‌شد. نتایج بدست آمده نشان داد که با قرار دادن تعداد ۲ نرون در لایه مخفی، این شبکه عصبی می‌تواند نمونه‌ها را با دقت ۱۰۰% به‌صورت کامل شناسایی کند. بنابراین این ساختار که در شکل (۴-۸ ب) مشاهده می‌کنید به‌عنوان ساختار نهایی در ادامه مورد استفاده قرار گرفت.



شکل (۴-۸) ساختار شبکه عصبی مورد استفاده برای مجموعه داده Wine. الف) ساختار ابتدایی. ب) ساختار نهایی تایید شده

همچنین میزان خطای بدست آمده از مرحله آموزش شبکه عصبی برای مجموعه داده‌های آموزش، ارزیابی و تست را در شکل (۴-۹) مشاهده می‌شود.



شکل (۴-۹) نمودار خطای بدست آمده از فاز آموزش شبکه عصبی برای مسئله Wine

نتایج این شبکه عصبی پس از مرحله آموزش به‌صورت جدول (۴-۱۱) می‌باشد. در این جدول نتایج بدست آمده از مجموعه داده‌های آموزش و تست به‌صورت مجزا قرار گرفته است که نشان می‌دهد، مرحله آموزش شبکه عصبی به خوبی انجام شده و شبکه عصبی می‌تواند به‌صورت کامل نمونه‌ها را دسته‌بندی نماید.

جدول (۴-۱۱) نتایج بدست آمده از اعمال مجموعه داده‌ها به شبکه عصبی آموزش دیده برای مسئله Wine

Train	C1’	C2’	C3’
C1	29	0	0
C2	0	35	0
C3	0	0	24

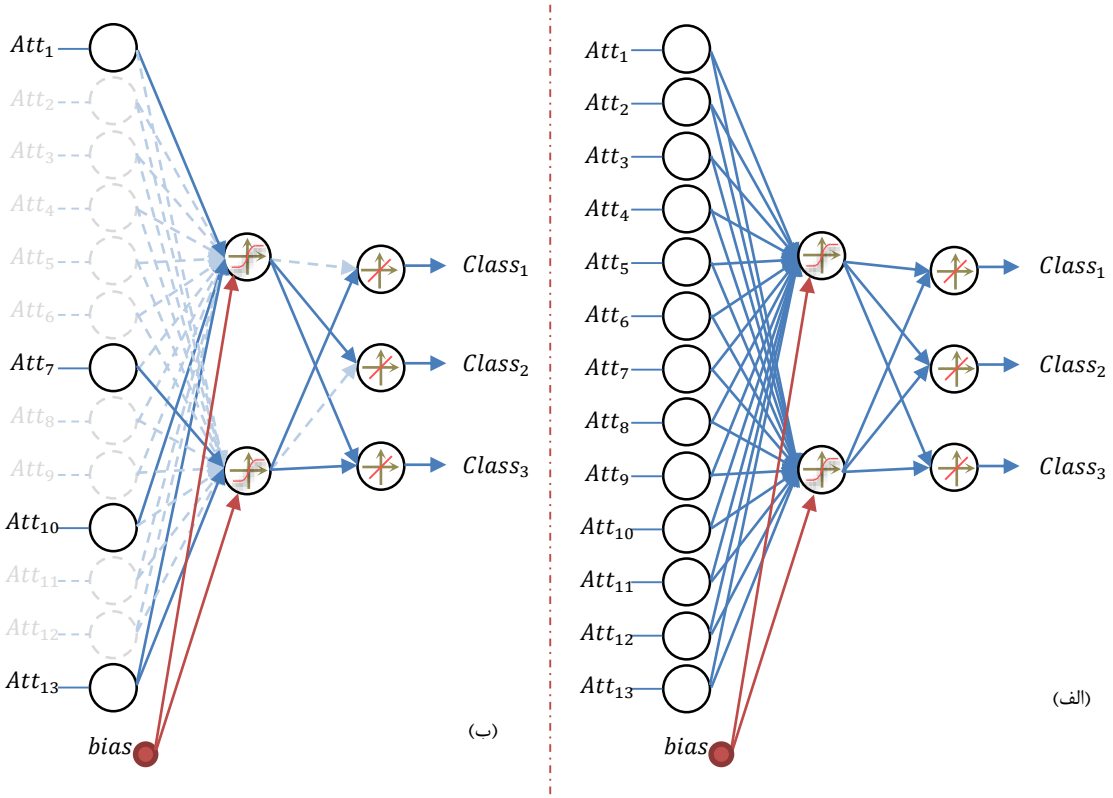
Valid	C1’	C2’	C3’
C1	15	0	0
C2	0	18	0
C3	0	0	12

Test	C1’	C2’	C3’
C1	15	0	0
C2	0	18	0
C3	0	0	12

All	C1’	C2’	C3’
C1	59	0	0
C2	0	71	0
C3	0	0	48

نتایج بدست آمده نشان داد که توسط ۲ نرون در ساختار لایه مخفی از شبکه عصبی می‌توان به میزان دقت ۱۰۰% و ۱۰۰% به ترتیب برای مجموعه‌های آموزش و تست دست یافت. حال می‌توان ساختار

این شبکه عصبی را با استفاده از الگوریتم‌های معرفی شده در بخش ۳-۲-۳ هرس نمود تا ویژگی‌ها و نرون‌های مؤثرتر این ساختار شناسایی شوند. ساختار هرس شده این شبکه عصبی به‌صورت شکل (۴-۱۰) می‌باشد.



شکل (۴-۱۰) ساختار شبکه عصبی مورد استفاده برای مسئله Wine قبل (الف) و بعد (ب) از هرس‌سازی

در شکل (۴-۱۰) مشاهده می‌شود که از بین ۱۳ پارامتر ورودی تنها ۴ پارامتر باقی مانده و بقیه حذف شده‌اند. این ساده سازی باعث کاهش دقت در عملیات دسته‌بندی توسط شبکه عصبی می‌گردد. البته نتایج بدست آمده همچنان قابل قبول می‌باشند. جدول (۴-۱۲) نتایج بدست آمده از دقت شبکه عصبی هرس شده برای این مسئله را نشان می‌دهد.

جدول (۴-۱۲) نتایج بدست آمده از اعمال همه نمونه‌ها به ساختار هرس شده شبکه عصبی برای مسئله Wine

All	C1’	C2’	C3’
C1	59	0	0
C2	2	69	0
C3	0	8	40

پس از تایید نتایج بدست آمده از این مرحله، مجموعه قوانین دسته‌بندی با استفاده از ساختار هرس شده شبکه عصبی و داده‌های موجود بدست می‌آیند. این قوانین در رابطه (۴-۵) نشان داده شده است. نتایج حاصل از اعمال این قوانین بر روی مجموعه داده مورد نظر نشان می‌دهد که قوانین بدست آمده قادرند (تقریباً) مشابه شبکه عصبی هرس شده، نمونه‌ها را دسته‌بندی نمایند. البته این نتایج برای کلاس C3 همراه با بهبود و برای کلاس C2 با کاهش دقت همراه بوده است. جدول (۴-۱۳) این نتایج را نشان می‌دهد.

$$\begin{gathered} if\ (X_7 \geq 2)\ and\ (X_{13} \geq 725)\ then\ Class_1 \\ if\ (X_7 < 1.6)\ and\ (X_{10} > 3.5)\ then\ Class_3 \\ Default\ \rightarrow Class_2 \end{gathered} \qquad (۴-۵)$$

جدول (۴-۱۳) نتایج حاصل از آزمایش قوانین بدست آمده برای مجموعه Wine

All	C1’	C2’	C3’
C1	58	1	0
C2	3	64	4
C3	0	0	48

بنابراین می‌توان از این قوانین به جای شبکه عصبی هرس شده استفاده نمود و نمونه‌های این مجموعه داده را توسط این قوانین دسته‌بندی کرد.

۴-۵ نتایج بدست آمده از مجموعه داده Glasses

این مجموعه داده یکی دیگر از مجموعه داده‌های دسته‌بندی در UCI بوده که شامل ۲۱۴ نمونه در ۲ کلاس متفاوت می‌باشد. برای بارگذاری این مجموعه داده در نرم افزار Matlab، از دستور رابطه (۴-۶) استفاده می‌شود.

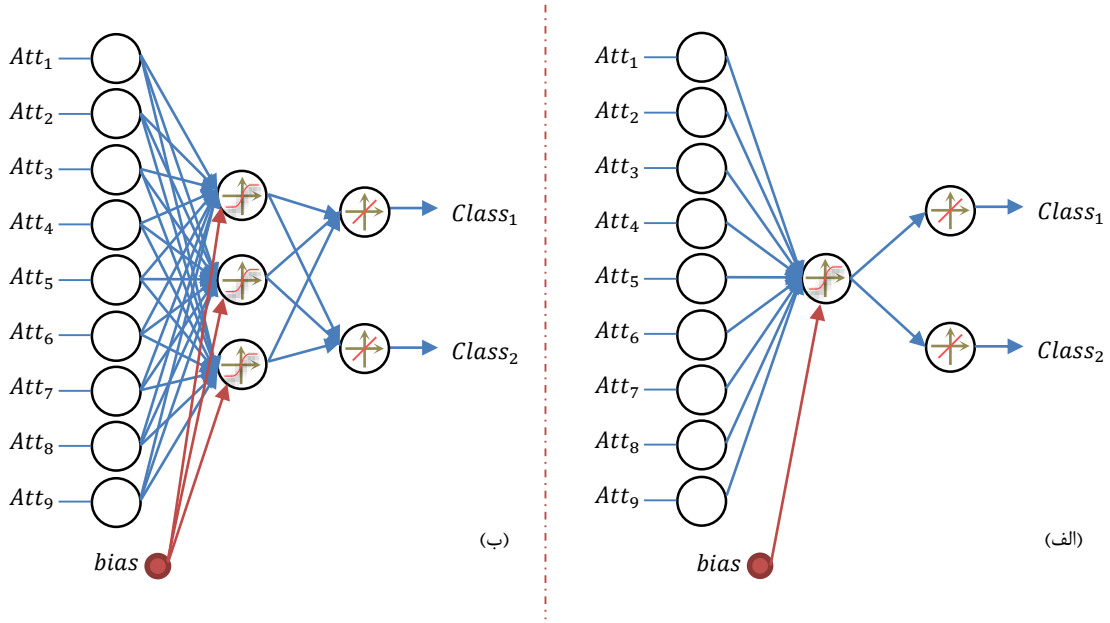
$$load('glass\_dataset'); \qquad (۴-۶)$$

در این مجموعه داده به‌منظور دسته‌بندی نمونه‌ها از ۹ ویژگی ورودی استفاده می‌شود. جدول (۴-۱۴) مشخصات مجموعه داده Glass را نشان می‌دهد. ویژگی‌های موجود در این مجموعه داده دارای مقادیر پیوسته می‌باشد. در فاز پیش‌پردازش ابتدا این مقادیر نرمال‌سازی شده و سپس نمونه‌های موجود را به دو مجموعه آموزش و تست نسبت می‌دهیم. در این تقسیم‌بندی ۵۰% نمونه‌ها در مجموعه آموزش، ۲۵% در مجموعه ارزیابی و ۲۵% باقیمانده نیز به مجموعه تست نسبت داده شد و سعی شد که از نمونه‌های هر کلاس به همان نسبت در مجموعه‌های ذکر شده وجود داشته باشند.

جدول (۴-۱۴) اطلاعات کلی در مورد مجموعه داده Glass

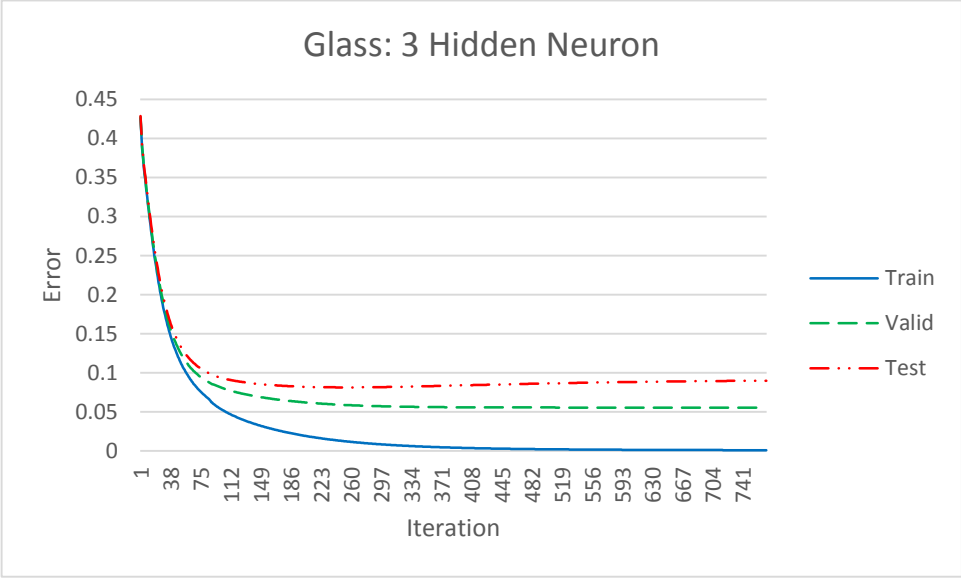
نام مجموعه داده	Glass	
تعداد کلاس‌های خروجی	2	
تعداد ویژگی‌های ورودی	9	
تعداد نمونه‌ها	214	
تعداد نمونه در هر کلاس	Glass2 = 163	Glass1 = 51

پس از آماده‌سازی داده‌ها، در فاز دوم یک شبکه عصبی با ۹ نرون در لایه ورودی (به تعداد ویژگی‌ها) و ۲ نرون در لایه خروجی (به تعداد کلاس‌های موجود) تولید گشته و با آزمایش حالات مختلف، مشخص شد که تعداد ۳ نرون در لایه مخفی برای انجام عملیات شناسایی مناسب می‌باشد. شکل (۴-۱۱) ساختار اولیه و ساختار نهایی شبکه عصبی برای مسئله Glass را نشان می‌دهد.



شکل (۴-۱۱) ساختار شبکه عصبی مورد استفاده برای مسئله Glass. الف) ساختار ابتدایی. ب) ساختار نهایی

نمودار خطای بدست آمده از مرحله آموزش این شبکه عصبی به‌صورت شکل (۴-۱۲) می‌باشد.



شکل (۴-۱۲) نمودار خطای حاصل از مرحله آموزش شبکه عصبی برای مسئله Glass

نمودار خطای شکل (۴-۱۲) نشان می‌دهد که استفاده از ۳ نرون در لایه مخفی از شبکه عصبی باعث کاهش خطا به میزان چشمگیر بوده و روند آموزش شبکه عصبی به خوبی انجام شده است. برای درک

بهتر نتایج حاصل از این شبکه عصبی، جدول (۴-۱۵) میزان درستی و نادرستی نمونه‌های دسته‌بندی شده توسط این شبکه عصبی آموزش دیده را نشان می‌دهد.

جدول (۴-۱۵) نتایج بدست آمده از اعمال مجموعه داده‌ها به شبکه آموزش دیده برای مسئله Glass

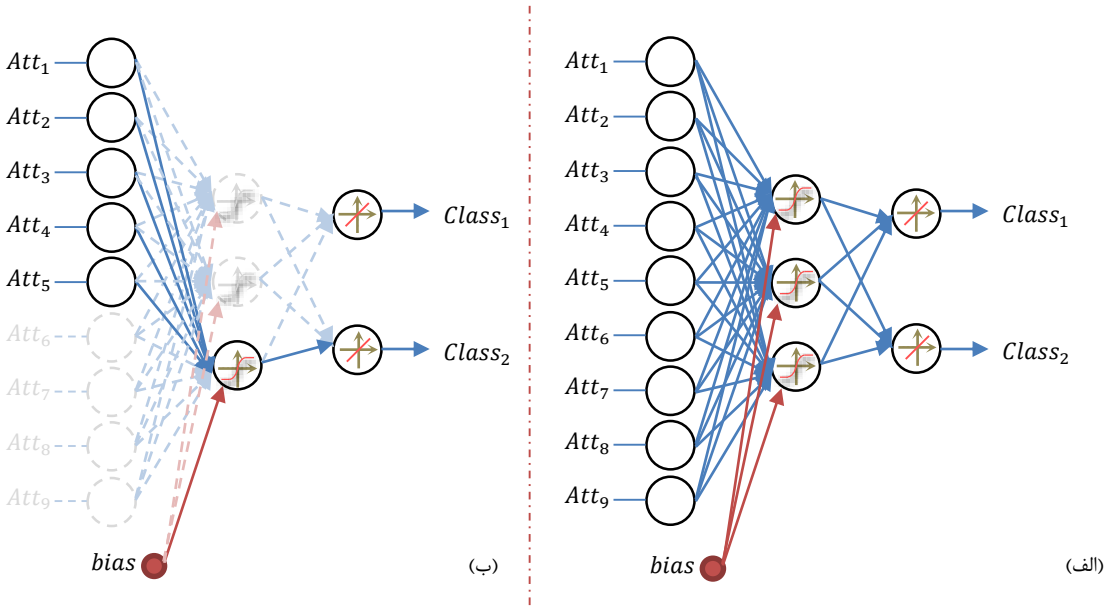
Train	C1’	C2’
C1	25	0
C2	0	81

Valid	C1’	C2’
C1	11	2
C2	1	40

Test	C1’	C2’
C1	10	3
C2	2	39

All	C1’	C2’
C1	46	5
C2	3	160

همانگونه که مشاهده می‌شود، اگرچه شبکه عصبی آموزش دیده همه نمونه‌ها را به‌صورت درست دسته‌بندی نکرده است، اما قادر است با همین ساختار ساده، مجموعه‌های آموزش، ارزیابی و تست را به ترتیب با میزان دقت ۱۰۰%، ۹۴٫۴۴% و ۹۰٫۷۴% دسته‌بندی نماید که این مقادیر قابل قبول می‌باشد. سپس مرحله بعدی یعنی هرس‌سازی شبکه عصبی انجام می‌شود. در این فاز ساختار شبکه عصبی به‌صورت شکل (۴-۱۳ ب) تغییر می‌کند.



شکل (۴-۱۳) ساختار شبکه عصبی مورد استفاده برای مسئله Glass قبل (الف) و بعد (ب) از هرس‌سازی

نتایج حاصل از دقت شبکه عصبی پس از عملیات هرس‌سازی توسط اعمال نمونه‌های موجود به شبکه عصبی و همچنین دسته‌بندی آنها، بدست آمده که این نتایج در جدول (۴-۱۶) قابل مشاهده هستند.

جدول (۴-۱۶) نتایج حاصل از اعمال نمونه‌ها به شبکه عصبی هرس شده برای مسئله Glass

All	C1’	C2’
C1	45	6
C2	4	159

نتایج بدست آمده نشان می‌دهد که ساختار هرس شده شبکه عصبی قادر است نمونه‌ها را به خوبی دسته‌بندی نماید و در مقایسه با نتایج بدست آمده از ساختار اتصال کامل (جدول (۴-۱۵)) شبکه عصبی، میزان خطا در دسته‌بندی نمونه‌ها به مقدار ۲ نمونه کاهش یافته است. با توجه به این شرایط، حال می‌توان قوانین دسته‌بندی لازم را از ساختار جدید شبکه عصبی و با در نظر گرفتن نمونه‌های موجود، شناسایی نمود. این قوانین بدست آمده به‌صورت رابطه (۴-۷) می‌باشد.

$$if\ (X_1 > 1.5237)\ or\ (X_3 \geq 2.7)\ then\ Class_2$$
$$Default\ \rightarrow Class_1 \qquad (۴-۷)$$

جهت بررسی نتایج بدست آمده از اعمال این قوانین بر روی مجموعه داده موجود استفاده شد که نتایج آن در جدول (۴-۱۷) قابل مشاهده می‌باشد.

جدول (۴-۱۷) نتایج حاصل از آزمایش قوانین بدست آمده بر روی مجموعه داده Glasses

All	C1’	C2’
C1	48	3
C2	6	157

این نتایج نشان‌دهنده عملکرد مشابه قوانین بدست آمده نسبت به شبکه عصبی هرس‌شده می‌باشد که می‌توان از این قوانین با دقت ٪۹۵٫۷۹ به جای این شبکه عصبی جهت دسته‌بندی نمونه‌ها استفاده کرد.

۴-۶ نتایج بدست آمده از مجموعه داده Iris Flower

یکی دیگر از مجموعه داده‌هایی که استفاده زیادی در بررسی نحوه کارایی سیستم‌های شناسایی‌گر دارد، مجموعه داده Iris[1] می‌باشد. این مجموعه شامل نمونه‌های مختلف گل زنبق می‌باشد که در ۳ دسته متفاوت تقسیم‌بندی می‌شوند. مشخصات این مجموعه داده در جدول (۴-۱۸) قرار دارد. این مجموعه در نرم افزار Matlab نیز به‌عنوان یکی از مجموعه داده‌های آزمایشی برای مسائل دسته‌بندی وجود داشته و می‌توان توسط دستور رابطه (۴-۸) از آن استفاده کرد.

```
load('iris dataset');
```
(۴-۸)

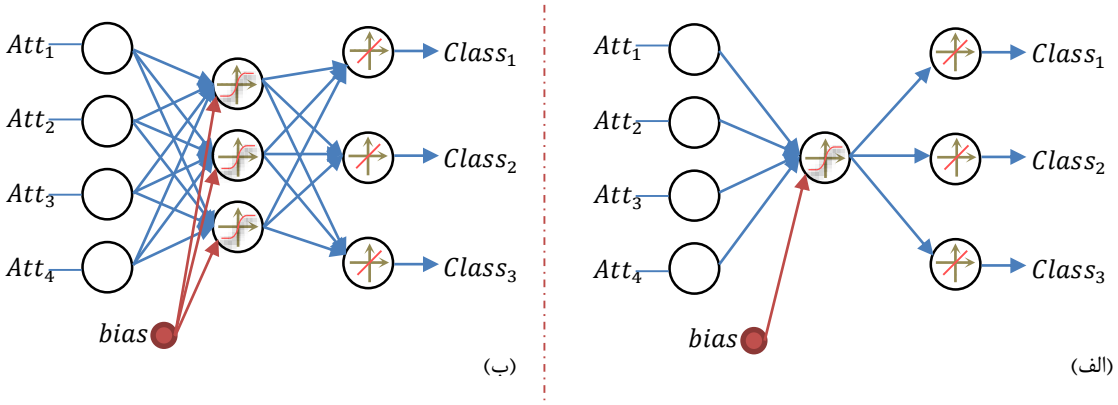
برای آماده‌سازی این داده‌ها قبل از اعمال به شبکه عصبی مصنوعی، عملیات نرمال‌سازی بر روی آن انجام می‌شود. در این مرحله از رابطه‌های معرفی شده در بخش ۳-۲-۱ به‌منظور نرمال‌سازی این مقادیر استفاده شده است. سپس ۵۰% از این داده‌ها را برای آموزش شبکه عصبی تحت عنوان مجموعه آموزش، ۲۵% را برای اعتبارسنجی شبکه عصبی تحت عنوان مجموعه ارزیابی و ۲۵% باقیمانده نیز به‌عنوان مجموعه تست در نظر گرفته شد.

جدول (۴-۱۸) اطلاعات کلی در مورد مجموعه داده Iris Flower

نام مجموعه داده	**Iris Flower**		
تعداد کلاس‌های خروجی	3		
تعداد ویژگی‌های ورودی	4		
تعداد نمونه‌ها	150		
تعداد نمونه در هر کلاس	Versicolor = 50	Virginica = 50	Setosa = 50

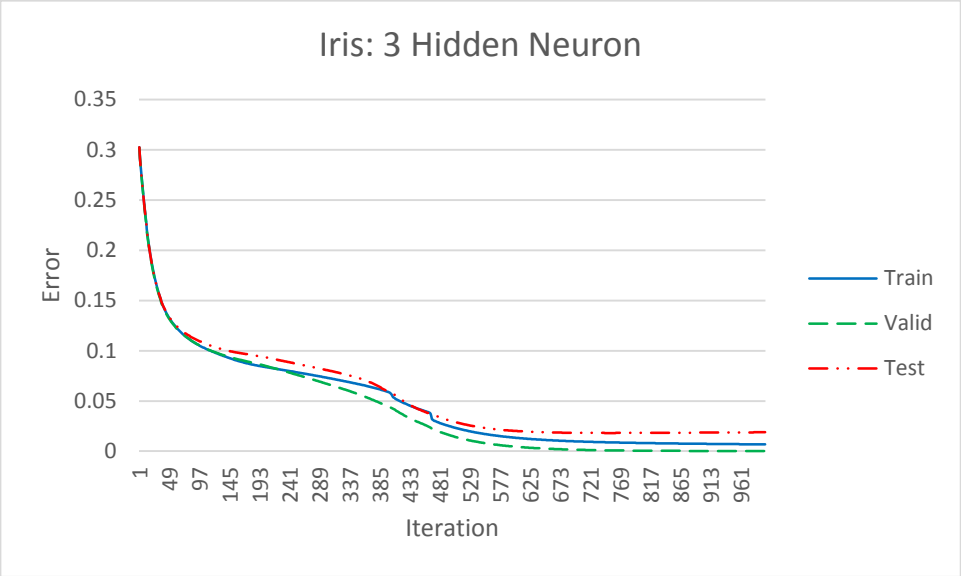
[1] https://archive.ics.uci.edu/ml/datasets/Iris

سپس یک شبکه عصبی مصنوعی با ساختار ابتدایی ۳-۱-۴ برای انجام دسته‌بندی نمونه‌های این مجموعه ایجاد گردید. پارامترهای این شبکه عصبی همانند پارامترهای تعیین شده در فصل سوم در نظر گرفته شد. توسعه ساختار شبکه عصبی و اضافه کردن نرون در لایه مخفی باعث افزایش کارایی و بهبود نتایج بدست آمده شد. شکل (۴-۱۴) ساختار شبکه عصبی مورد استفاده برای مسئله مورد نظر را نشان می‌دهد.



شکل (۴-۱۴) ساختار شبکه عصبی مورد استفاده برای مسئله Iris الف) ساختار ابتدایی ب) ساختار نهایی

آزمایش حالات مختلف شبکه عصبی (با ساختارهای مختلف) نشان داد که شبکه عصبی با ۳ نرون در لایه مخفی قادر است این نمونه‌ها را به میزان قابل قبولی شناسایی کند. نمودار خطای حاصل از آموزش شبکه عصبی با ساختار ذکر شده در شکل (۴-۱۵) نمایش داده شده است.



شکل (۴-۱۵) نمودار خطای حاصل از مرحله آموزش شبکه عصبی

همچنین نتایج عددی حاصل از اعمال نمونه‌های موجود به شبکه آموزش دیده در جدول (۴-۱۹) آمده است.

جدول (۴-۱۹) نتایج حاصل از اعمال مجموعه داده‌ها به شبکه عصبی آموزش دیده برای حل مسئله Iris

Train	C1’	C2’	C3’
C1	24	0	0
C2	0	24	0
C3	0	0	24

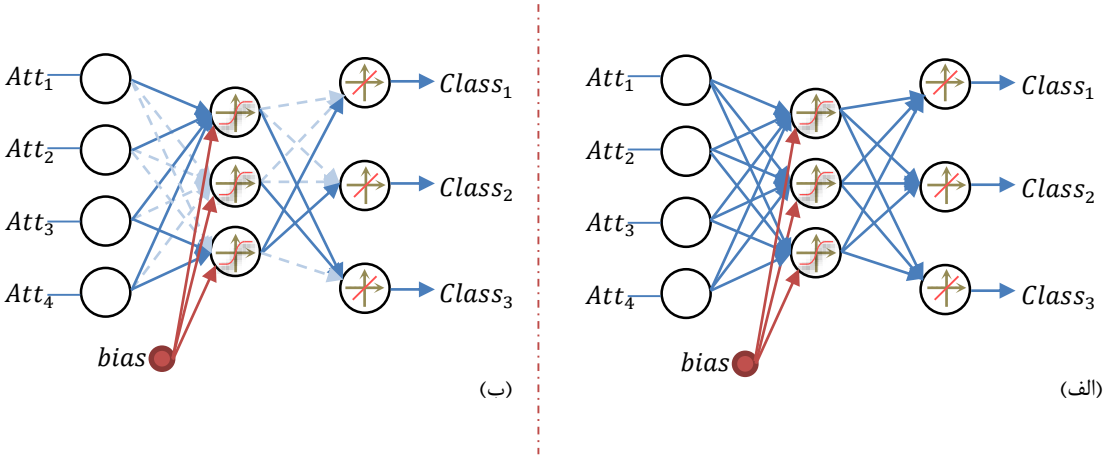
Valid	C1’	C2’	C3’
C1	13	0	0
C2	0	13	0
C3	0	0	13

Test	C1’	C2’	C3’
C1	13	0	0
C2	0	12	1
C3	0	0	13

All	C1’	C2’	C3’
C1	50	0	0
C2	0	49	1
C3	0	0	50

این نتایج نشان می‌دهد که فاز آموزش شبکه عصبی به خوبی انجام شده و شبکه عصبی قادر به شناسایی نمونه‌های مختلف این مجموعه داده با دقت ۱۰۰%، ۱۰۰% و ۹۷٫۴۳% به ترتیب برای مجموعه‌های آموزش، ارزیابی و تست می‌باشد. حال می‌توان به‌منظور انجام استخراج قانون از شبکه عصبی، ساختار آن را تا حد امکان ساده نمود. این عملیات با حذف کردن و هرس کردن یال‌های غیر ضروری شبکه

عصبی توسط الگوریتم‌های مطرح شده می‌باشد. شکل (۴-۱۶) ساختار شبکه عصبی مصنوعی را قبل و بعد از هرس‌سازی را نشان می‌دهد.



شکل (۴-۱۶) ساختار شبکه عصبی مورد استفاده برای مسئله Iris Flower قبل (الف) و بعد (ب) از هرس‌سازی

پس از هرس نمودن ساختار شبکه عصبی، با اعمال نمونه‌ها به ساختار جدید شبکه عصبی، میزان دقت آن محاسبه می‌شود. نتایج حاصل از آن در جدول (۴-۲۰) نشان داده شده است.

جدول (۴-۲۰) نتایج بدست آمده از اعمال نمونه‌ها به ساختار جدید شبکه عصبی برای مسئله Iris Flower

All	C1’	C2’	C3’
C1	50	0	0
C2	0	49	1
C3	0	0	50

نتایج بدست آمده نشان می‌دهد که دقت شبکه عصبی پس از هرس‌سازی نیز همانند حالت اتصال کامل بوده و این شبکه عصبی قادر است تا با این تعداد اتصالات کمتر نیز نتایج مشابه تولید نماید.

سپس قوانین دسته‌بندی لازم برای این مسئله با توجه به ساختار جدید شبکه عصبی و نمونه‌های موجود، تولید می‌گردند. این نتایج در رابطه (۴-۹) نشان داده شده است. حال می‌توان از این قوانین به جای شبکه عصبی مصنوعی در عملیات دسته‌بندی و شناسایی نمونه‌های این مجموعه داده استفاده

نمود. دقت بدست آمده از این قوانین در جدول (۴-۲۱) ارائه شده است که تقریباً نزدیک به میزان دقت بدست آمده از شبکه عصبی می‌باشد.

$$if\ (X_3 \leq 1.9)\ then\ C1_{Setosa}$$
$$if\ (X_3 \leq 5)\ and\ (X_4 \leq 1.7)\ then\ C2_{Versicolor} \quad (۹-۴)$$
$$Default\ \rightarrow C3_{Virginica}$$

جدول (۴-۲۱) نتایج بدست آمده از دسته‌بندی نمونه‌های مجموعه Iris Flower توسط قوانین بدست آمده

All	C1’	C2’	C3’
C1	50	0	0
C2	0	48	2
C3	0	0	50

قوانین بدست آمده قادر است که نمونه‌ها را با دقت ۹۸٫۶۷% شناسایی کرده و دسته‌بندی نماید.

۴-۷ نتایج بدست آمده از مجموعه داده Wisconsin Breast Cancer

مجموعه داده سرطان سینه یا Wisconsin Breast cancer[1] یکی از مجموعه داده‌های دسته‌بندی موجود در زمینه پزشکی و تشخیص بیماری می‌باشد. در این مجموعه داده دو نوع سرطان سینه خوش‌خیم[2] و بدخیم[3] وجود دارد که با استفاده از ۹ ویژگی موجود در این مجموعه داده، عملیات دسته‌بندی داده‌ها انجام می‌شود. این مجموعه داده نیز یکی دیگر از مجموعه داده‌های موجود در نرم‌افزار Matlab برای انجام آزمایشات دسته‌بندی و شناسایی است که توسط دستور رابطه (۴-۱۰) قابل استفاده می‌باشد.

$$\text{load('cancer\_ dataset');} \quad (۱۰-۴)$$

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)
[2] Benign
[3] Malignant

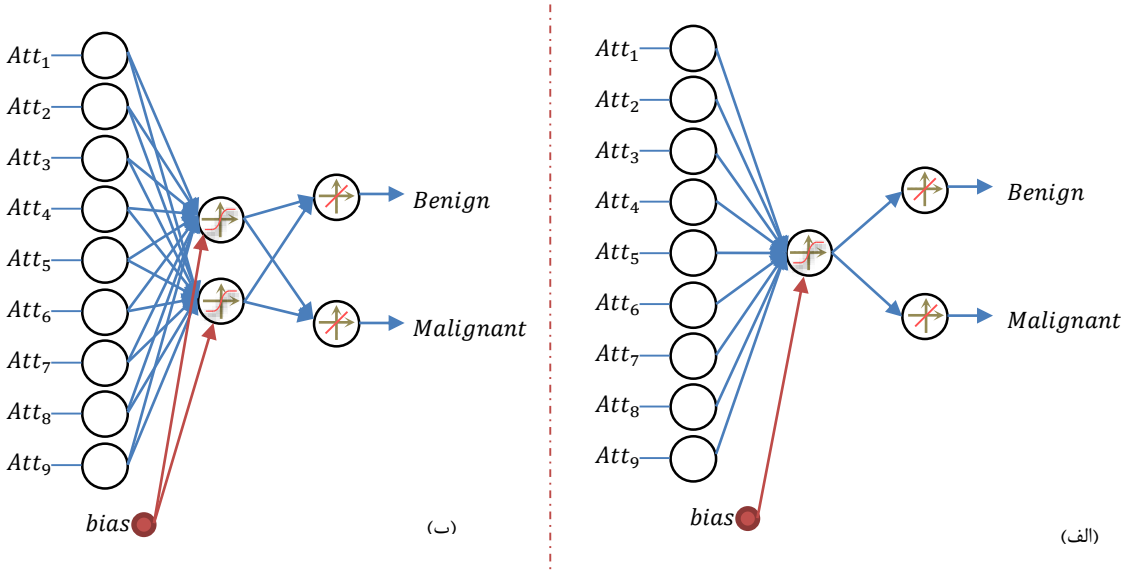
جدول (۴-۱۳) اطلاعات کلی را در رابطه با این مجموعه داده نشان می‌دهند. همانطور که در مشاهده می‌شود، تعداد نمونه‌های دو دسته موجود اختلاف زیادی دارند، بطوریکه تعداد نمونه‌های خوش‌خیم حدود ۶۵٫۵% از این مجموعه داده را تشکیل داده و ۳۴٫۵% نمونه‌های موجود نیز متعلق به حالت بدخیم هستند.

جدول (۴-۲۲) اطلاعات کلی در رابطه با مجموعه داده Wisconsin Breast Cancer

نام مجموعه داده	Wisconsin Breast Cancer	
تعداد کلاس‌های خروجی	2	
تعداد ویژگی‌های ورودی	9	
تعداد نمونه‌ها	699	
تعداد نمونه در هر کلاس	Malignant = 241	Benign = 458

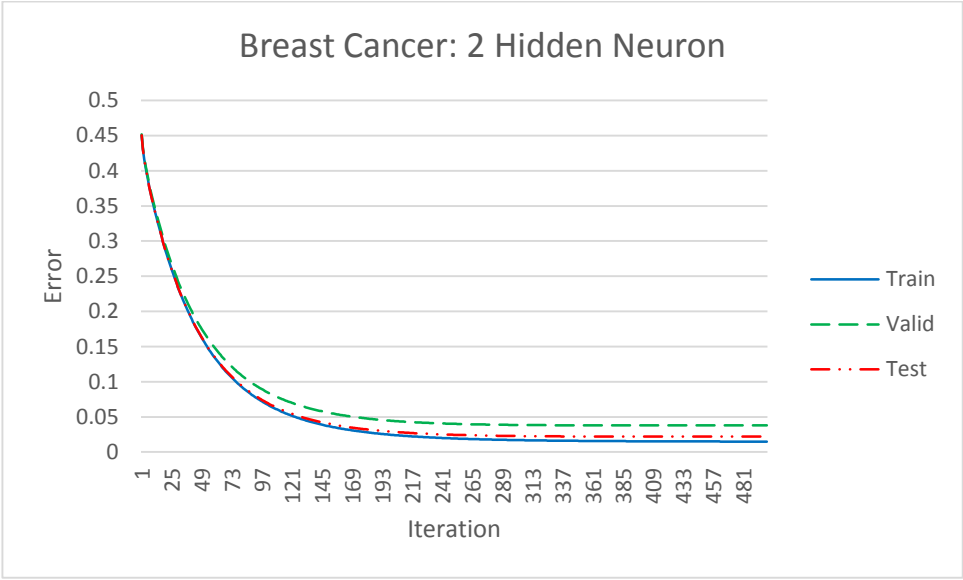
این مجموعه داده شامل ویژگی‌ها با مقادیر گسسته بوده که نیاز است تا قبل از اعمال به شبکه عصبی، نرمال‌سازی شوند. برای انجام اینکار از رابطه‌های نرمال‌سازی تعیین شده در فصل سوم استفاده می‌گردد. در نهایت این مجموعه داده به سه مجموعه تحت عناوین آموزش (۵۰%)، ارزیابی (۲۵%) و تست (۲۵%) برای آمزش و ارزیابی شبکه عصبی تقسیم‌بندی می‌شود.

سپس یک شبکه عصبی مصنوعی با ساختار ۲-۱-۹ به‌منظور دسته‌بندی این داده‌ها ایجاد گردیده و پارامترهای این شبکه عصبی همانند پارامترهای در نظر گرفته شده برای روش FSRE تعیین می‌شود. سپس با تغییر تعداد نرون لایه مخفی در این شبکه عصبی و همچنین آزمایش حالات مختلف مشخص شد که این شبکه عصبی با ۲ نرون در لایه مخفی قادر خواهد بود نمونه‌ها را به میزان قابل قبولی دسته‌بندی نماید. ساختار شبکه عصبی مورد استفاده برای این مسئله را در شکل (۴-۱۲) مشاهده می‌کنید.



شکل (۴-۱۷) ساختار شبکه عصبی مورد استفاده جهت حل مسئله Wisconsin Breast Cancer. الف) ساختار ابتدایی. ب) ساختار نهایی

همچنین نمودار خطای بدست آمده از آموزش شبکه عصبی با ساختار نهایی نیز در جدول (۴-۲۲) نمایش داده شده است.



شکل (۴-۱۸) نمودار خطای بدست آمده از آموزش شبکه عصبی برای حل مسئله Wisconsin Breast Cancer

با توجه به نمودار خطای بدست آمده از آموزش مشخص می‌شود که مجموعه داده آموزش، ارزیابی و تست رفتار تقریباً مشابهی دارند و میزان دقت سه مجموعه داده تقریباً برابر می‌باشد. برای درک بهتر از

کارایی شبکه آموزش دیده، نتایج بدست آمده از آن برای هر مجموعه به‌صورت مجزا در جدول (۴-۲۳) آمده است.

جدول (۴-۲۳) نتایج بدست آمده از اعمال مجموعه داده‌ها به شبکه عصبی آموزش دیده برای حل مسئله Wisconsin Breast Cancer

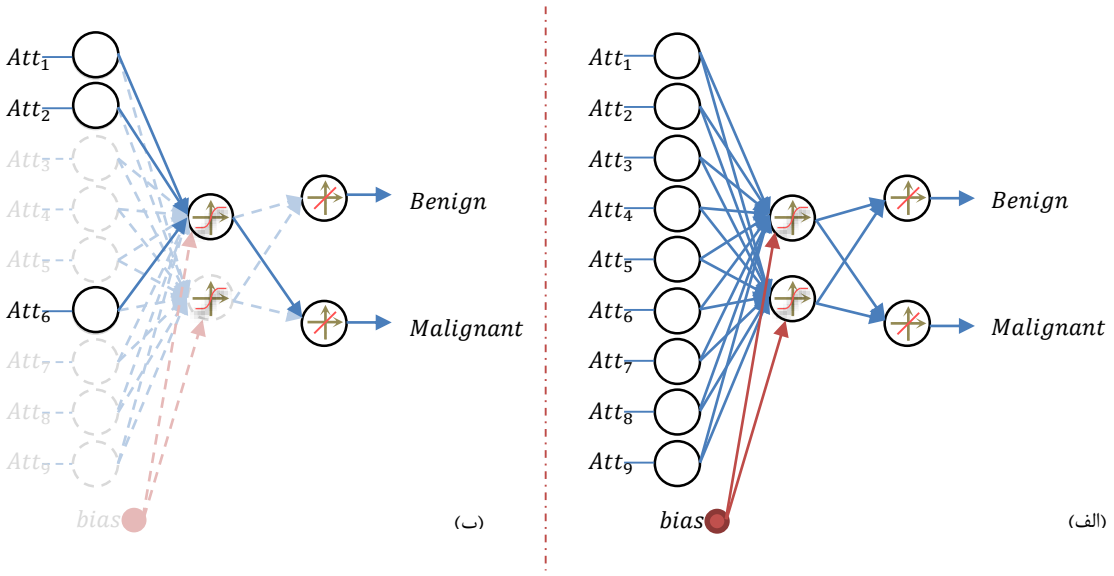
Train	C1’	C2’
C1	120	1
C2	4	224

Valid	C1’	C2’
C1	59	1
C2	6	109

Test	C1’	C2’
C1	59	1
C2	4	111

All	C1’	C2’
C1	238	3
C2	14	444

نتایج بدست آمده نشان از کارایی مناسب و قابل قبول شبکه عصبی در شناسایی و دسته‌بندی نمونه‌های این مجموعه داده دارد. بدین‌ترتیب مقادیر ۹۸٫۵۶%، ۹۶٫۰% و ۹۷٫۱۴% به ترتیب به‌عنوان دقت بدست آمده از شبکه عصبی برای مجموعه‌های آموزش، ارزیابی و تست محاسبه گردید. حال ساختار شبکه عصبی را توسط الگوریتم‌های بیان شده ساده هرس کرده تا اتصالات و نرون‌های غیر ضروری از ساختار این شبکه عصبی حذف شود. شکل (۴-۱۹) ساختار شبکه عصبی مورد استفاده را قبل و بعد از هرس‌سازی نشان می‌دهد.



شکل (۴-۱۹) ساختار شبکه عصبی مورد استفاده برای مسئله Wisconsin Breast Cancer. الف) ساختار با اتصالات کامل. ب) ساختار هرس شده

همانطور که در شکل (۴-۱۹) مشاهده می‌شود، اتصالات زیادی از شبکه عصبی حذف گردیده و ساختار نهایی پس از هرس‌سازی، کاملاً ساده شده به نحوی که تنها ۳ ویژگی ورودی در شبکه عصبی باقی مانده است. برای بررسی میزان کارایی این ساختار از شبکه عصبی، کل مجموعه داده را به شبکه عصبی هرس شده اعمال کرده که نتایج آن را در جدول (۴-۲۴) مشاهده می‌کنید.

جدول (۴-۲۴) نتایج بدست آمده از اعمال نمونه‌ها به ساختار شبکه عصبی هرس شده

All	C1’	C2’
C1	237	4
C2	17	441

مشاهده می‌شود که شبکه عصبی پس از هرس‌سازی نیز به خوبی می‌تواند نمونه‌ها را شناسایی کند. میزان دقت شبکه عصبی در حالت هرس شده نیز برابر ۹۶٫۹۹% می‌باشد. پس از تایید نتایج بدست آمده از ساختار هرس شده شبکه عصبی، مجموعه قوانین دسته‌بندی برای مسئله مورد نظر بدست می‌آید. این قوانین در رابطه (۴-۱۱) نمایش داده شده است.

$$if\ (X_1 \geq 0.7)\ or\ (X_2 \geq 0.7)\ or\ (X_6 \geq 0.6)\ then\ Class_2$$
$$Default\ \rightarrow Class_1 \qquad (۱۱-۴)$$

پس از آزمایش قوانین بدست آمده بر روی مجموعه داده مورد نظر مشخص شد که این قوانین قادرند نمونه‌ها را با دقت ۹۶٫۵۶% شناسایی و دسته‌بندی نمایند. جدول (۴-۲۵) جزئیات نتایج بدست آمده از آزمایش قوانین بدست آمده را نشان می‌دهند.

جدول (۴-۲۵) نتایج حاصل از آزمایش قوانین بدست آمده بر روی مجموعه داده Wisconsin Breast Cancer

All	C1’	C2’
C1	228	13
C2	11	447

۴-۸ تحلیل و مقایسه نتایج بدست آمده

در آزمایشات انجام شده در این فصل، از مجموعه داده‌های استانداردی جهت بررسی میزان کارایی روش مورد نظر استفاده شده است. این آزمایشات که شامل مجموعه‌های Simple Classes، Play Golf، Wine، Glass، Iris flower و Wisconsin Breast Cancer می‌باشد، اکثراً از پایگاه داده UCI جمع‌آوری شده و مورد استفاده قرار گرفته‌اند. همچنین روش‌هایی را که به‌منظور مقایسه در این بخش مورد بررسی قرار گرفته‌اند شامل روش‌های REANN [۲۲]، RGANN [۵۷]، ESRNN [۵۸]، RULES [۵۹]، RULES-2 [۶۰]، DIFACONN [۱۶]، Rex [۳۵]، Rex-P [۴۲]، Rex-M [۴۲]، SV-DT [۶۱]، GARulExNN [۶۲]، RBFGD [۲۶]، GRBF [۶۳]، ERBF [۶۳]، TB-RBF [۶۴]، MMDT [۶۵] و MDTF [۶۶] هستند.

❖ *مجموعه داده Simple Classes*

مجموعه داده Simple Classes که یک مجموعه داده مصنوعی می‌باشد، به‌صورت پیش‌فرض در نرم افزار Matlab وجود دارد. نتایج بدست آمده از اعمال روش FSRE بر روی این مجموعه داده در جدول (۴-۲۶) آمده است.

جدول (۴-۲۶) تحلیل و بررسی نتایج بدست آمده برای مجموعه داده Simple Classes

نام مجموعه داده	ویژگی‌ها	FSRE
Simple Classes	تعداد قوانین	۴
	میانگین تعداد شرایط در قوانین	۲
	دقت قوانین	%۱۰۰

همانطور که در جدول (۴-۲۶) مشاهده می‌شود، قوانین بدست آمده برای این مجموعه داده قادر است که تمام نمونه‌ها را به‌صورت کاملاً صحیح شناسایی نماید.

❖ مجموعه داده Play Golf

نتایج بدست آمده از آزمایش روش FSRE بر روی مجموعه داده Play Golf نیز در جدول (۴-۲۷) آمده است. همچنین نتایج دیگر روش‌های اخیر نیز به‌منظور مقایسه در این جدول قرار دارد.

جدول (۴-۲۷) تحلیل و بررسی نتایج بدست آمده برای مجموعه داده Play Golf

نام مجموعه داده	ویژگی‌ها	FSRE	REANN	RGANN	Rex	ESRNN	RULES	RULES-2
Play Golf	تعداد قوانین	۳	۳	۳	۳	۳	۸	۱۴
	میانگین تعداد شرایط در قوانین	۲	۲	۲	۲	۲	۲	۲
	دقت قوانین	%۱۰۰	%۱۰۰	%۱۰۰	%۱۰۰	%۱۰۰	%۱۰۰	%۱۰۰

از نتایج بدست آمده در جدول (۴-۲۷) می‌توان نتیجه گرفت که روش روش FSRE قادر است همانند روش‌های REANN، RGANN، Rex و ESRNN نتایج بهینه‌ای را برای این مجموعه داده تولید نماید.

درحالیکه روش‌های RULES و RULES-2 تعداد قوانین بیشتری را برای این مجموعه داده تولید می‌نمایند.

❖ *مجموعه داده Wine*

جدول (۴-۲۸) نتایج بدست آمده از روش‌های مختلف استخراج قانون را بر روی مجموعه داده Wine نشان می‌دهد.

جدول (۴-۲۸) تحلیل و بررسی نتایج بدست آمده از مجموعه Wine

نام مجموعه داده	ویژگی‌ها	FSRE	RGANN	ESRNN	Rex-P	Rex-M	SV-DT	GARulExNN
Wine	تعداد قوانین	۳	۳	۳	۴	۱۳٫۸	۱۸	۱۱٫۳
	میانگین تعداد شرایط در قوانین	۲	۳	۳	---	---	۶٫۲۸	---
	دقت قوانین	%۹۵٫۵۰	%۹۱٫۰۱	%۹۱٫۰۱	%۹۰٫۹۸	%۹۲٫۲۲	%۸۱٫۳۶	%۸۸٫۲

در این نتیج مشاهده می‌شود که روش FSRE بهینه‌ترین پاسخ را برای این مجموعه داده فراهم نموده است. پس از آن روش‌های RGANN، ESRNN و Rex-P نتایج تقریباً مناسبی تولید کرده و در آخر روش‌های Rex-M، SV-DT و GARulExNN بدلیل تولید تعداد قوانین زیاد و یا دقت پایین، از قابلیت درک کمتری برخوردارند.

❖ *مجموعه داده Glass*

نتایج بدست آمده از آزمایش روش FSRE بر روی مجموعه داده Glass در جدول (۴-۲۹) آورده شده است.

همچنین به‌منظور مقایسه روش FSRE با دیگر روش‌های استخراج قانون، نتایج برخی از روش‌های اخیر بر روی این مجموعه داده نیز ذکر شده است. این نتایج نشان می‌دهد که روش FSRE با فاصله زیادی نسبت به دیگر روش‌ها بهینه‌تر عمل می‌کند. پس از آن روش RBF-GD با تولید شش قانون برای این مسئله نتیجه تقریباً مناسبی را ارائه کرده است. اما روش‌های DIFACONN، GRBF، ERBF، TB-RBF و MMDT بدلیل تولید تعداد قوانین زیاد و همچنین دقات پایین در رده بعدی قرار می‌گیرند.

جدول (۴-۲۹) بررسی و تحلیل نتایج بدست آمده از مجموعه داده Glass

نام مجموعه داده	ویژگی‌ها	FSRE	DIFACONN	RBF-GD	GRBF	ERBF	TB-RBF	MMDT
Glass	تعداد قوانین	۲	۱۷٫۶۳	۶	۱۶	۱۶	۱۶٫۴	۱۳٫۲۷
	میانگین تعداد شرایط در قوانین	۳	---	۳٫۳۳	---	---	---	۶۵
	دقت قوانین	%۹۵٫۷۹	%۸۴٫۱۳	%۸۶٫۲۱	%۶۴٫۷۶	%۶۸٫۵۷	%۶۲٫۸	%۸۶٫۵۸

❖ مجموعه داده Iris Flower

نتایج بدست امده از آزمایش روش FSRE بر روی مجموعه Iris در جدول (۴-۳۰) نشان داده شده است.

جدول (۴-۳۰) بررسی و تحلیل نتایج بدست آمده از مجموعه داده Iris Flower

نام مجموعه داده	ویژگی‌ها	FSRE	DIFACONN	ESRNN	SV-DT	Rex-P	Rex-M	MDTF
Iris	تعداد قوانین	۳	۳٫۲۷	۳	۷	۳٫۸	۸٫۸	۵
	میانگین تعداد شرایط در قوانین	۱	---	۱	۳٫۵۷	---	---	---
	دقت قوانین	%۹۸٫۶۷	%۹۸٫۰۷	%۹۸٫۶۷	%۸۰	%۹۸٫۶۷	%۹۷٫۶	%۹۷٫۳۳

نتایج جدول (۴-۳۰) نشان می‌دهد که روش FSRE همانند روش‌های ESRNN و Rex-P به‌صورت بهینه عمل کرده اس ا‌ت. اما میزان دقت نتایج بدست آمده از روش‌های DIFACONN، SV-DT، Rex-M و MDTF کمتر بوده و تعداد قوانین تولید شده در این روش‌ها بیشتر بوده است.

❖ *مجموعه داده Wisconsin Breast Cancer:*

نتایج حاصل از آزمایش روش FSRE بر روی مجموعه داده Wisconsin Breast Cancer در جدول (۴-۳۱) قرار دارد.

جدول (۴-۳۱) بررسی و تحلیل نتایج حاصل از مجموعه داده Wisconsin Breast Cancer

نام مجموعه داده	ویژگی‌ها	FSRE	REANN	RGANN	Rex	SV-DT	Rex-P	Rex-M
Breast Cancer	تعداد قوانین	۲	۲	۲	۲	۴	۳	۱۵٫۶
	میانگین تعداد شرایط در قوانین	۳	۳	۳	۳	۲٫۲۵	---	---
	دقت قوانین	%۹۶٫۵۶	%۹۶٫۲۸	%۹۶٫۲۸	%۹۶٫۲۸	%۹۲٫۹۸	%۹۵٫۹۷	%۹۳٫۴۳

همانطور که از نتایج بدست آمده مشخص می‌شود، در این آزمایش نیز روش FSRE نتایج ایده‌آل‌تری را تولید نموده است. البته روش‌های REANN، RGANN و Rex نتایج تقریباً مشابهی را تولید نموده‌اند. اما میزان دقت بدست آمده از قوانین تولید شده توسط روش‌های SV-DT، Rex-P و Rex-M به نسبت کمتر از دیگر روش‌ها بوده است.

۹-۴ نتیجه گیری

در این فصل از روش پیشنهادی برای تولید قوانین در چند مسئله دسته‌بندی استفاده شد. مجموعه داده‌های مورد استفاده در این فصل شامل Simple Classes، Play Golf، Wine، Glasses، Iris Flower و Wisconsin Breast Cancer بودند. با معرفی هر یک از این مجموعه داده‌ها، مراحل انجام روش FSRE برای تولید قوانین و همچنین نتایج بدست آمده از هر مجموعه با ذکر جزئیات ارائه شد. همچنین بررسی و تحلیل نتایج بدست آمده از این روش نشان داد که روش پیشنهادی FSRE قادر است در مسائل این چنینی همانند دیگر روش‌های موجود و یا بهتر از آنها عمل نماید. همچنین تعداد قوانین و شرایط درون قوانین تولید شده در این روش معمولاً برابر دیگر روش‌ها و یا کمتر است که این ویژگی باعث افزایش قابلیت فهم قوانین تولید شده می‌گردد. در فصل بعدی این پایان‌نامه، از روش FSRE در یک مسئله متفاوت و پیچیده‌تر استفاده می‌شود.

فصل پنجم

کاربرد روش پیشنهادی در تشخیص بیماری کبد چرب

۵-۱ مقدمه

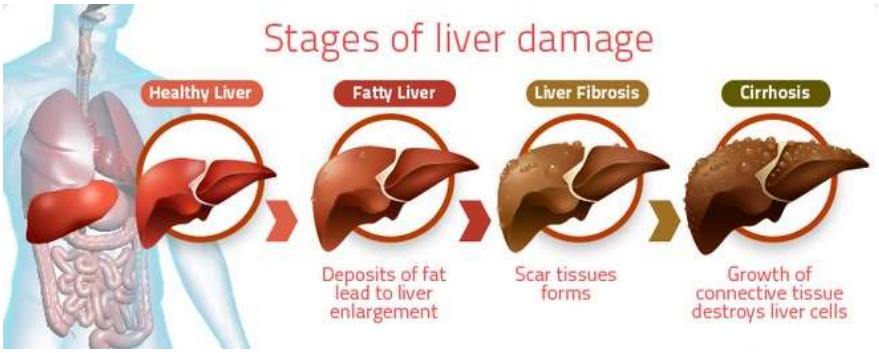
کبد، بزرگترین عضو داخلی بدن شمرده شده که عملیات حیاتی زیادی را در بدن انجام می‌دهد. امروزه بیماری‌های کبدی به‌عنوان شایع‌ترین بیماری‌ها در بین افراد جامعه شناخته می‌شوند. بیماری کبد چرب غیر الکلی (NAFLD[1]) یکی از خطرناک‌ترین گونه‌های بیماری کبدی است که تشخیص سریع و بموقع این بیماری می‌تواند در درمان آن مفید و مؤثر باشد [۶۷،۶۸]. در حال حاضر روش‌های گوناگونی برای شناسایی این بیماری و میزان پیشروی (یا سطح شدت) آن وجود دارد که دقیق‌ترین آنها روش فیبرواسکن[2] می‌باشد [۶۹]. اما همچنان نیاز به یک روش کم هزینه‌تر و در دسترس‌تر برای شناسایی این بیماری احساس می‌شود [۶۷،۷۰-۷۲].

در این فصل از روش FSRE برای شناسایی این بیماری و میزان پیشروی آن استفاده شده است. برای انجام اینکار از یک مجموعه داده استفاده گردیده که توسط محققین ایرانی در سال‌های اخیر جمع‌آوری شده است [۷۳]. این مجموعه داده شامل مقادیر داده‌ای خام بوده و نیازمند مدیریت و پیش‌پردازش برای دست‌یابی به نتایج بهینه‌تر می‌باشد. همچنین وجود پیچیدگی زیاد مسئله مورد نیازمند استفاده از ساختار پیچیده‌تری به‌منظور دسته‌بندی می‌باشد.

۵-۲ معرفی بیماری

کبد چرب یک التهاب کبدی است که در اثر تجمع بیش از حد چربی (تری گلیسرید) در بافت کبد ایجاد می‌شود که در این حالت بیش از ۵% از بافت کبد را چربی فرا می‌گیرد [۷۴]. این میزان چربی می‌تواند باعث اختلال در عملکرد طبیعی کبد گردیده و با پیشروی آن باعث نارسایی کبدی گردد. شکل (۵-۱) مراحل پیشروی این بیماری را نشان می‌دهد.

[1] Non-alcoholic fatty liver disease
[2] FibroScan



شکل (۵-۱) مراحل پیشروی بیماری کبد چرب و اهمیت تشخیص زود هنگام بیماری [۶۹]

بیماری کبد چرب به دو گروه عمده تقسیم می‌شود: کبد چرب الکلی که ناشی از مصرف طولانی مدت الکل می‌باشد و کبد چرب غیر الکلی که بیشتر از طریق عوامل زمینه‌ساز دیابت، بیماری‌های قلبی و متابولیسمی بوجود می‌آید. بیماری NAFLD امروزه در جوامع بشری از شیوع بیشتری برخوردار است و دارای چندین سطح شدت مختلف می‌باشد: از *استاتوز ساده*[۱] شروع شده و تا *سیروز*[۲] کبدی افزایش می‌یابد [۷۵]. این سطوح در بین پزشکان توسط علائم F0، F1، F2، F3 و F4 شناخته شده که معانی هر یک در جدول (۵-۱) آمده است [۷۵،۷۴،۷۲،۶۸].

جدول (۵-۱) سطوح مختلف شدت بیماری NAFLD

کلاس‌ها	توضیحات
F0	Absence of fibrosis (No fibrosis)
F1	Perisinusoidal or portal involvement (Portal fibrosis without septa)
F2	Perisinusoidal and portal/periportal involvement (Portal fibrosis and few septa)
F3	Septal or bridging fibrosis (Numerous septa without cirrhosis)
F4	Cirrhosis

۵-۳ مجموعه داده مورد استفاده

طی یک مطالعه تشخیصی که در سال ۱۳۹۰ از ساکنان شهر گند کاووس به عمل آمد، یک مجموعه داده شامل نتایج آزمایشات خون[۳]، سنوگرافی[۴] و فیبرواسکن جمع‌آوری شد [۷۳]. این مجموعه که در

[1] Simple steatosis
[2] Cirrhosis
[3] Complete blood count (CBC)
[4] Ultrasonography

فاصله بین شهریور ماه تا بهمن ماه طی شش ماه بدست آمد، شامل تعداد ۷۲۶ نمونه و ۱۸ ویژگی برای هر نمونه می‌باشد. این ویژگی‌ها برگرفته از پارامترهای موجود در نتایج آزمایش‌های ذکر شده می‌باشد که جدول (۵-۲) این ویژگی‌ها را تشریح می‌نماید.

از بین این ویژگی‌ها، ۱۶ ویژگی اول از طریق نتایج آزمایش کامل خون و ویژگی Ultra-Score نیز از نتیجه آزمایش سنوگرافی بدست آمده است. این ویژگی‌ها به‌عنوان مقادیر موجود و "ورودی‌ها"ی مسئله در نظر گرفته شد.

جدول (۵-۲) ویژگی‌ها و اطلاعات موجود در مجموعه داده مورد استفاده

ردیف	نام کامل ویژگی	علامت اختصاری
۱	جنسیت	Sex
۲	اندازه قد	Height
۳	اندازه وزن	Weight
۴	سن	Age
۵	میزان کلسترول خون	T-CHOL
۶	میزان کلسترول خوب خون	HDL
۷	میزان کلسترول بد خون	LDL
۸	سطح تری گلیسرید خون	TG
۹	میزان کراتینین خون	CRE
۱۰	میزان تصفیه گلومرولی	GFR
۱۱	میزان اوره خون	BU
۱۲	یکی از آنزیم‌های موجود در کبد	ALP
۱۳	سطح گاماگلوتامیل ترانسفراز خون	GGT
۱۴	میزان قند خون	FBS
۱۵	میزان گلوبول‌های سفید خون	WBC
۱۶	میزان پلاکت خون	PLT
۱۷	درجه چربی بدست آمده از آزمایش سنوگرافی	Ultra-Score
۱۸	نتیجه آزمایش فیبرواسکن	Fibrosis

آخرین ویژگی در جدول (۵-۲) مربوط به نتیجه آزمایش فیبرواسکن بوده که برای هر فرد انجام شده و به‌عنوان "خروجی" مورد انتظار در مسئله موجود در نظر گرفته می‌شود. مقادیر این ویژگی به‌صورت

اعداد حقیقی و اعشاری بوده که در نگاه اول به‌صورت یک سری اعداد پیوسته دیده می‌شوند. اما برای تعیین سطح بیماری از روی این مقادیر، متخصصین این مقادیر را به گونه‌های مختلف دسته‌بندی می‌کنند [۶۷،۶۸،۷۵]. در این پایان‌نامه نیز از دسته‌بندی انجام شده در [۷۵] به‌منظور گسسته‌سازی این مقادیر و تعیین سطح بیماری هر فرد با توجه به مقدار بدست آمده از آزمایش فیبرواسکن، استفاده شده است. جدول (۵-۳) تعداد نمونه‌های موجود در هر کلاس پس از تعیین سطح بیماری را نشان می‌دهد.

جدول (۵-۳) تعداد نمونه‌های موجود در هر کلاس پس از انجام تعیین سطح

کلاس‌ها خروجی	**کلاس F0**	**کلاس F1**	**کلاس F2**	**کلاس F3**	**کلاس F4**
تعداد نمونه	415	151	132	23	5

همانطور که در جدول (۵-۳) مشاهده می‌شود، تفاوت زیادی در تعداد نمونه‌های دسته‌های مختلف وجود دارد. این اختلاف بدلیل تصادفی بودن نمونه گیری در زمان مجع‌آوری این مجموعه بوده و به نحوی باعث افزایش پیچیدگی در مسئله مورد نظر می‌شود.

۵-۴ استخراج قانون

حال به‌منظور دست‌یابی به قوانین دسته‌بندی برای تشخیص بیماری NAFLD و شدت آن، روش FSRE که در بخش ۳-۲ معرفی شد را بر روی مجموعه داده ذکر شده اعمال می‌کنیم.

۵-۴-۱ پیش پردازش

مجموعه داده‌ای که در این پژوهش مورد استفاده قرار می‌گیرد، از داده‌های خام تشکیل شده است. اولین قدم برای آماده‌سازی داده‌ها قبل از اعمال به دسته‌بند مورد نظر این است که ویژگی‌های موجود در این مجموعه را بررسی نماییم. این مجموعه دارای تعداد ویژگی‌های زیاد (۱۷ عدد) و بعضاً نامرتبط می‌باشد. برای بررسی و تعیین میزان اهمیت ویژگی‌ها، از روش بهره اطلاعاتی [۴۹] استفاده شده است. جدول (۵-۴) نتایج بدست آمده از بهره اطلاعاتی را برای ویژگی‌های برتر نشان می‌دهد.

جدول (۵-۴) مقادیر بهره اطلاعاتی بدست آمده برای هر ویژگی به‌منظور شناسایی هر کلاس

ردیف	کلاس F1	کلاس F2	کلاس F3	کلاس F4
۱	LDL (۰/۰۰۷)	FBS (۰/۰۲۵)	HDL (۰/۰۱۲)	HDL (۰/۰۰۸)
۲	Weight (۰/۰۰۵)	Ultra-Score(۰/۰۲۴)	T-CHOL (۰/۰۰۹)	Ultra-Score(۰/۰۰۶)
۳	HDL (۰/۰۰۴)	HDL (۰/۰۱۶)	Weight (۰/۰۰۶)	Weight (۰/۰۰۴)
۴	Ultra-Score(۰/۰۰۳)	Weight (۰/۰۱۲)	LDL (۰/۰۰۶)	LDL (۰/۰۰۴)
۵	Height (۰/۰۰۲)	T-CHOL (۰/۰۱۱)	Ultra-Score(۰/۰۰۳)	T-CHOL (۰/۰۰۴)
⋮	⋮	⋮	⋮	⋮

در جدول (۵-۴) مشاهده می‌شود که ۵ ویژگی برتر برای شناسایی سطوح مختلف بیماری لیست شده و مقادیر بهره اطلاعاتی آنها نیز در جلوی هر ویژگی ذکر شده است. مشاهده می‌شود که معیار بهره اطلاعاتی برای بهترین ویژگی‌ها کمتر از ۰/۰۵ و نزدیک به صفر می‌باشد. این نتایج بدین معنی است که این ویژگی‌ها به تنهایی برای شناسایی سطوح بیماری مناسب نبوده و به ویژگی‌های موثرتری نیاز است. در روشی که توسط Forns و همکارانش [۷۱] معرفی شد، از فرمول زیر به‌منظور شناسایی بیماری NAFLD از طریق پارامترهای بالینی استفاده گردیده بود:

$$Score = 7.811 - 3.131 \times ln(PLT) + 0.781 \times ln(GGT) + 3.467 \times ln(Age) - 0.014 \times (TCHOL) \quad (۱-۵)$$

طبق نتایج بدست آمده توسط محققین نامبرده، این روش توانسته است ۷۸% از بیماران را در سطوح مؤثرتر بیماری (F2 تا F4) به‌صورت صحیح شناسایی کند [۷۱]. این روش قابلیت شناسایی حالت ابتدایی بیماری (F1) را ندارد. در این پایان‌نامه، از معادله (۵-۱) به‌منظور ایجاد یک ویژگی جدید به نام "Forns-Score" جهت توسعه و بهبود نتایج در تشخیص این بیماری استفاده شده است. علاوه بر این ویژگی BMI[1] نیز توسط معادله (۵-۲) تولید شده و هر دو ویژگی جدید به مجموعه داده اضافه شدند.

$$BMI = \frac{Weight_{kg}}{Height_{m}^{2}} \quad (۲-۵)$$

[1] Body mass index

مسئله مورد نظر، شناسایی نمونه بین ۵ کلاس موجود است؛ بنابراین توسط ۴ شناسایی کننده می‌توان این عملیات را انجام داد. کلاس F0 از ابتدای عملیات به‌عنوان پاسخ پیش‌فرض کنار گذاشته شد. منظور از پاسخ پیش‌فرض این است که ابتدا فرض می‌شود نمونه ورودی جزء این کلاس است؛ درصورتیکه قوانین موجود برای این نمونه قابل اعمال نباشند این نمونه جزء همین کلاس باقی می‌ماند و در غیر اینصورت تعلق این نمونه به یکی از کلاس‌های موجود تغییر می‌یابد. بنابراین برای چهار حالت بیماری باقیمانده (بر اساس جدول (۵-۱))، چهار سیستم شناسایی کننده مجزا در نظر گرفته شده است. هفت ویژگی برتری که توسط معیار بهره اطلاعاتی برای شناسایی هر کدام از این کلاس‌های خروجی بدست آمده‌اند را جدا کرده و به‌عنوان ویژگی‌های ورودی این سیستم‌های شناسایی کننده در نظر می‌گیریم. لیست ویژگی‌های انتخاب شده برای شناسایی کننده‌ها در جدول (۵-۵) نشان داده شده است.

جدول (۵-۵) لیست ویژگی‌های انتخاب شده برای هر شناسایی کننده

ردیف	ویژگی‌های انتخاب شده	نام کامل ویژگی
1	Ultra_Score	نتیجه آزمایش سنوگرافی
2	HDL	کلسترول خوب خون
3	LDL	کلسترول بد خون
4	TG	میزان تری‌گلیسرید خون
5	FBS	میزان قند خون
6	BMI	نسبت قد به وزن
7	Frons_Score	روش مورد استفاده توسط Forns [۷۱] جهت تشخیص بیماری

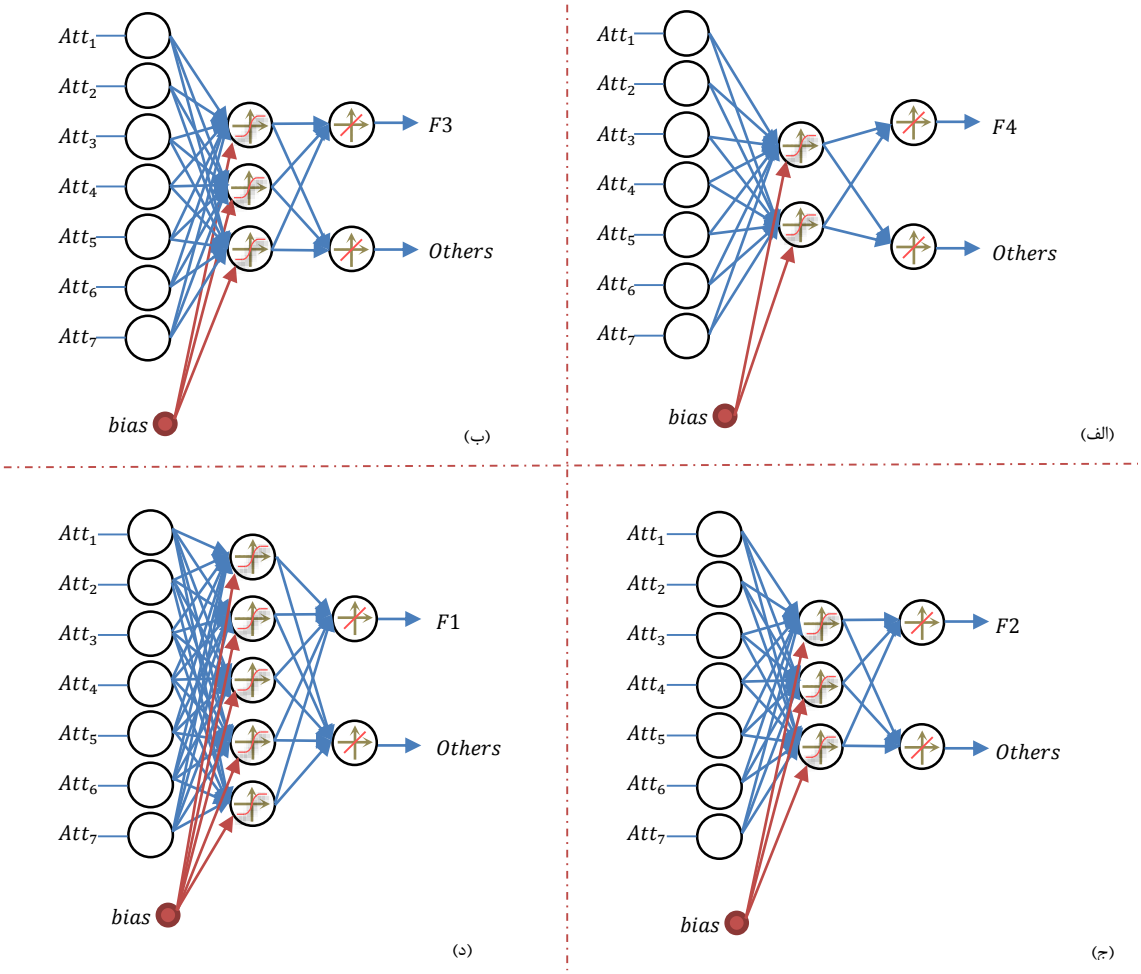
این ویژگی‌های انتخاب شده دارای مقادیر پیوسته عددی هستند که به‌منظور افزایش دقت نتایج بدست آمده و ساده‌سازی عملیات به مقادیر گسسته تبدیل شوند. در این تحقیق، از روش ur-CAIM [۵۳] بدلیل به‌روزتر بودن و بهینه‌تر بودن نتایج آن استفاده شده است. سپس مقادیر بدست آمده از گسسته‌سازی، جهت اعمال به شناسایی کننده‌ها (شبکه‌های عصبی) نرمال‌سازی شدند.

پس از تکمیل این مراحل داده‌های بدست آمده، برای اعمال به شناسایی کننده‌ها آماده می‌باشند. از این مجموعه داده ۷۰% برای آموزش شبکه عصبی (مجموعه آموزش)، ۱۵% برای اعتبارسنجی شبکه آموزش دیده (مجموعه ارزیابی) و ۱۵% باقیمانده به‌عنوان مجموعه تست استفاده می‌شود.

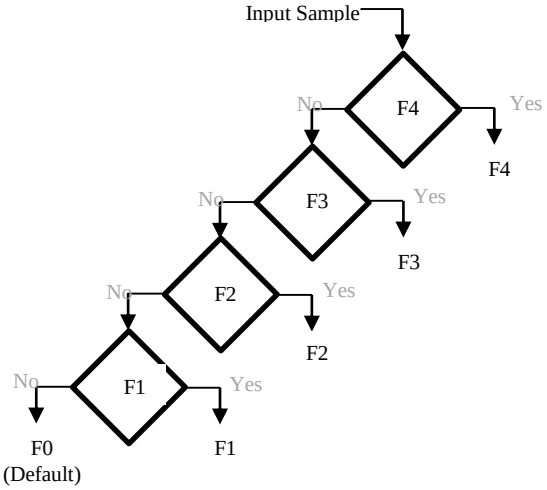
۵-۴-۲ مدل یادگیری

در این قسمت ابتدا چهار سیستم شناسایی کننده برای چهار سطح بیماری موجود (به غیر از حالت F0) در نظر گرفته می‌شود و ساختار آنها با توجه به بخش ۳-۲-۲ تعیین می‌شود. برای آموزش این شناسایی کننده‌ها از روش یکی بر علیه همه[1] استفاده می‌شود. بدین‌ترتیب هر شناسایی کننده فقط این تصمیم را می‌گیرد که "آیا نمونه ورودی جزء این دسته هست یا خیر؟". همانند آزمایشات انجام شده در فصل چهارم، برای این مسئله نیز ابتدا از ساختار ابتدایی ۲-۱-۷ برای هر چهار شناسایی کننده استفاده می‌شود، اما ساختار نهایی هر شناسایی کننده متفاوت می‌باشد. شکل (۵-۲) ساختار نهایی این چهار شناسایی کننده را نشان می‌دهد. پس از انجام آموزش این شناسایی کننده‌ها بر اساس احتمال هر دسته و تعداد نمونه‌های آن دسته، به‌صورت سلسله مراتبی قرار می‌گیرند. شکل (۵-۳) نحوه قرار گرفتن چهار شناسایی کننده در این مسئله را نشان می‌دهد.

[1] One vs. all



شکل (۵-۲) ساختار نهایی شبکه‌های عصبی مورد استفاده برای دسته‌بندی نمونه‌های کلاس‌های مختلف. الف) شناسایی کننده دسته F4. ب) شناسایی کننده دسته F3. ج) شناسایی کننده دسته F2. د) شناسایی کننده دسته F1



شکل (۵-۳) نحوه قرار گرفتن شناسایی کننده‌ها به‌صورت سلسله مراتبی برای کاهش ابهامات احتمالی

سپس نمونه ورودی به‌صورت سلسله مراتبی (و نه همزمان) به این شناسایی کننده‌ها اعمال می‌شوند. مزیت این حالت نسبت به حالت همزمان این است که ابهام و اشتراکی در تصمیم‌گیری شناسایی کننده‌ها بوجود نمی‌آورد. از طرفی بدلیل اینکه هر نمونه قطعاً عضو یک کلاس مشخص می‌باشد، حالت سلسله مراتبی باعث کاهش میزان خطای شناسایی کننده‌ها می‌شود. در این حالت میزان کارایی این مجموعه شناسایی کننده برای هر دسته به‌صورت مجزا محاسبه می‌شود.

۳-۴-۵ هرس‌سازی

سپس عملیات هرس‌سازی و حذف یال‌های اضافی در هر یک از شناسایی کننده‌ها انجام شده و کارایی شناسایی کننده‌ها پس از هرس‌سازی نیز محاسبه می‌گردد. جدول (۵-۶) تعداد نرون‌های مخفی بکار گرفته شده در هر یک از شناسایی کننده‌ها را به همراه نتایج بدست آمده از آموزش و هرس سازی آنها نشان می‌دهد.

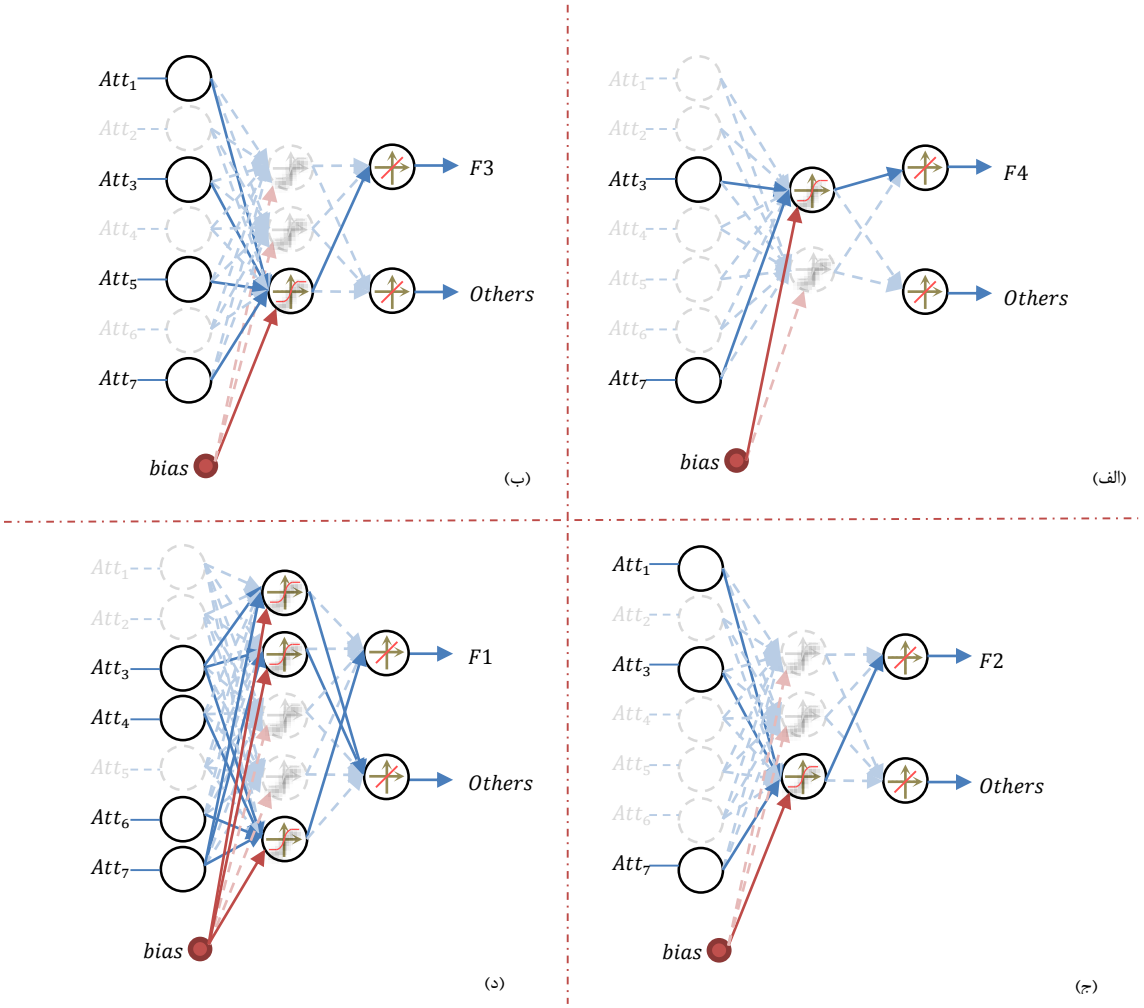
جدول (۵-۶) تعداد نرون‌های لایه مخفی و نتایج بدست آمده از آموزش و هرس‌سازی هر شناسایی کننده

کلاس‌های خروجی	تعداد نرون‌های لایه مخفی	شبکه عصبی آموزش دیده		شبکه عصبی هرس شده	
		حساسیت	اختصاصی بودن	حساسیت	اختصاصی بودن
کلاس F1	۵	٪۸۰٫۷۹	٪۸۲٫۷۸	٪۸۴٫۷۶	٪۸۱٫۲۱
کلاس F2	۳	٪۹۴٫۹۶	٪۹۳٫۶۰	٪۹۳٫۹۳	٪۹۳٫۷۷
کلاس F3	۳	٪۱۰۰	٪۹۹٫۲۸	٪۱۰۰	٪۹۹٫۲۸
کلاس F4	۲	٪۱۰۰	٪۱۰۰	٪۱۰۰	٪۱۰۰

نتایج جدول (۵-۶) نشان می‌دهد که شبکه‌های عصبی به‌عنوان شناسایی کننده‌های در نظر گرفته شده پس از مرحله آموزش نتایج خوبی را بدست می‌آورند و می‌توانند به خوبی نمونه‌های موجود را دسته‌بندی کنند. این شبکه‌های عصبی برای حالات خطرناک‌تر بیماری (یعنی F3 و F4)، با ساختار ساده‌تر نتایج ایده‌آل‌تری را بدست می‌آورند و برای دسته‌های F1 و F2 نیز نتایج بیش از ۸۰ درصد بدست آمده که قابل قبول می‌باشد. پس از انجام هرس‌سازی این شبکه‌های عصبی نیز حساسیت شناسایی کننده‌ها

افزایش می‌یابد؛ و با توجه به حذف برخی اتصالات نتایج بدست آمده همواره ایده‌آل و قابل قبول می‌باشد.

شکل (۵-۴) ساختار هرس شده این چهار شناسایی کننده را نشان می‌دهد.



شکل (۵-۴) ساختار هرس شده شبکه‌های عصبی. الف) برای کلاس F4. ب) برای کلاس F3. ج) برای کلاس F2. د) برای کلاس F1

۵-۴-۴ استخراج قانون

پس از آن با اعمال نمونه‌ها به این شناسایی کننده‌ها، خروجی نرون‌های لایه مخفی به مقادیر گسسته تبدیل می‌شوند که با تعیین مدل گسسته برای هر نمونه در این لایه، قوانین شناسایی برای هر کلاس از بیماری بدست می‌آید. نتایج استخراج قانون برای هر کلاس بیماری نیز در جدول (۵-۷) ارائه شده است.

جدول (۵-۷) نتایج بدست آمده از استخراج قانون برای تشخیص بیماری NAFLD توسط روش FSRE

کلاس‌ها	تعداد قوانین تولید شده	حساسیت	اختصاصی بودن
کلاس F1	۵	٪۹۹٫۳۴	٪۷۵٫۶۵
کلاس F2	۳	٪۹۳٫۹۳	٪۹۳٫۷۷
کلاس F3	۳	٪۱۰۰	٪۹۹٫۲۸
کلاس F4	۱	٪۱۰۰	٪۱۰۰

این نتایج نشان می‌دهد که شبکه‌های عصبی با ساختار ساده‌تر، منجر به تولید تعداد قوانین کمتری شده است و نتایج حاصل شده از این قوانین تقریباً نزدیک به نتایج حاصل شده از فاز هرس‌سازی (جدول (۵-۶)) می‌باشد. این نتایج برای سطوح موثر بیماری (F2، F3 و F4) به‌صورت ایده‌آل و برای حالت ابتدایی بیماری (F1) قابل قبول می‌باشد.

در نهایت مجموعه قوانینی که به‌منظور شناسایی بیماری NAFLD و میزان پیشروی آن بدست آمده است در زیر نشان داده می‌شود. این قوانین پس از حاصل شدن از شناسایی کننده‌ها به همان ترتیب سلسله مراتبی در شکل (۵-۳) قرار می‌گیرند.

- Class F4:

$$if\ (LDL < 216)\ and\ (FornsScore \geq 115.01)\ then\ ClassF4$$

- Class F3:

$$if\ (UltraScore = [1\ or\ 7])\ and\ (LDL < 178.5)\ and\ (FronsScore \geq 109.035)\ \ then\ ClassF3$$

$$if\ (UltraScore = [1\ or\ 3\ or\ 5\ or\ 7])\ and\ (LDL \geq 178.5)\ and\ (69.8 \leq FBS < 220.7)\ and\ (FornsScore \geq 109.035)\ then\ ClassF3$$

$$if\ (2 \leq UltraScore \leq 6)\ and\ (LDL < 178.5)\ and\ (69.8 \leq FBS < 220.7)\ and\ (FornsScore \geq 109.035)\ \ then\ ClassF3$$

- Class F2:

$$if\ (LDL \geq 71.5)\ and\ (FornsScore \geq 106.485)\ then\ ClassF2$$

$$if\ (3 \leq UltraScore \leq 5)\ and\ (LDL < 65.5)\ and\ (FornsScore \geq 106.485)\ then\ ClassF2$$

$$if\ (UltraScore = 1)\ and\ (65.5 \leq LDL < 71.5)\ and\ (FornsScore \geq 106.485)\ then\ ClassF2$$

- Class F1:

$$if\ (LDL \geq 105.5)\ and\ (TRIG < 275.5)\ and\ (BMI < 32.108)\ and\ (105.085 \leq FronsScore < 106.725)\ then\ ClassF1$$

$$if\ (LDL < 104.5)\ and\ (TRIG < 263)\ and\ (BMI < 32.108)\ and\ (105.085 \leq FronsScore < 106.725)\ then\ ClassF1$$

$$if\ (LDL < 104.5)\ and\ (TRIG \geq 275.5)\ and\ (BMI < 32.108)\ and\ (105.085 \leq FronsScore < 106.725)\ then\ ClassF1$$

$$if\ (LDL < 104.5)\ and\ (TRIG < 241.5)\ and\ (BMI \geq 32.667)\ and\ (105.085 \leq FronsScore < 106.725)\ then\ ClassF1$$

- Class F0:

$$Default\ is\ ClassF0$$

۵-۵ نتیجه گیری

در این فصل، از روش پیشنهادی برای تشخیص میزان شدت بیماری NAFLD از طریق یک مجموعه داده جمع‌آوری شده، به‌عنوان یک مسئله پیچیده استفاده شد. همچنین تمامی عملیات انجام شده بر روی این مجموعه داده خام جهت آماده‌سازی هرچه بهتر آنها قبل از اعمال به شبکه عصبی تشریح شد. نتایج بدست آمده از این آزمایش نشان داد که روش FSRE به خوبی قادر است قوانین شناسایی را برای این مسئله تولید نماید. همچنین نتایج تولید شده از نظر تعداد قانون و شرایط درون قوانین به میزان قابل درک و فهم می‌باشد. این قوانین قادر است نمونه‌های دسته‌های مختلف را به دقت بیش از ۸۰% به‌صورت صحیح شناسایی نماید.

فصل ششم

جمع‌بندی و پیشنهادات

۶-۱ مقدمه

امروزه استفاده از شبکه عصبی مصنوعی به‌عنوان یک ابزار داده کاوی در زمینه‌های گوناگونی توسعه یاقته است. آزمایشات انجام شده نشان داده است که این ابزار معمولاً دارای دقت بالاتری در عملیات مورد نظر نسبت به دیگر ابزارات مشابه بوده است. اما غیر قابل استناد بودن نتایج این روش به دلیل ابهامی که در چگونگی تولید نتایج خروجی وجود دارد، سبب کاهش استفاده از آن شده است.

استخراج قانون از شبکه عصبی یکی از تکنیک‌هایی می‌باشد که جهت حل این مشکل مورد استفاده قرار می‌گیرد. اینگونه تکینک‌ها قادرند تا عملیات انجام شده درون شبکه عصبی را به مجموعه‌ای از قوانین ساده، مفید و کارا تبدیل نمایند که در سیستم‌های خبره و عملیات بعدی بتواند به جای شبکه عصبی مصنوعی عملیات مورد نظر را انجام دهند. امروزه روش‌های گوناگونی جهت استخراج دانش و تولید قوانین از شبکه عصبی وجود داشته که نسبت به معیارهای مختلف در دسته‌های متنوعی قرار می‌گیرند. هدف همه این روش‌ها تولید یک مجموعه قانون قابل درک و فهم است که دقت نتایج آن مشابه دقت خود شبکه عصبی باشد.

اما با وجود تعداد روش‌های فراوان جهت استخراج قانون مبتنی بر شبکه عصبی و تنوع کاربرد آنها، همواره این روش‌ها در مسائل محدود و ساده‌ای مورد استفاده قرار می‌گیرند. علت اصلی را می‌توان میزان دقت بدست آمده از نتایج عملیات مربوطه توسط قوانین، نسبت به خود شبکه عصبی مصنوعی دانست. زیرا روش‌های موجود با کاهش نتایج بدست آمده نسبت به شبکه عصبی همراه بوده و یا با تولید تعداد قوانین بسیار زیاد سبب کاهش درک و فهم نسبت به نتایج بدست آمده می‌شوند. در برخی از روش‌ها نیز که برای مسائل پیچیده مورد استفاده قرار می‌گیرد، از پیچیدگی محاسباتی و زمانی بالایی برخوردارند که در معمولاً با تولید تعداد قوانین زیاد خود باعث کاهش استفاده از آنها می‌شوند.

بدین دلیل محققان همچنان در تلاش‌اند تا روش‌هایی را ابداع نمایند که قادر باشد مسائل پیچیده‌تر را با تولید قوانین ساده‌تر و دستیابی به دقت بالاتر حل نمایند.

۶-۲ نوآوری تحقیق

در این پایان‌نامه یک روش جدید به‌منظور استخراج قانون از شبکه عصبی چندلایه معرفی شد. هدف از این روش تولید قوانین بهینه‌تر و افزایش نتایج بدست آمده از استخراج قانون به‌منظور استفاده از این روش در عملیات نسبتا پیچیده بوده است. روش پیشنهادی که FSRE نامیده می‌شود، شامل چهار فاز اصلی است که امکان استفاده از این روش بر روی مسائل ساده و پیچیده را فراهم می‌کند. روش FSRE قادر است در عملیات دسته‌بندی انجام گرفته توسط شبکه عصبی چند لایه، دانش مورد استفاده توسط شبکه عصبی را استخراج نموده و این دانش در قالب یک مجموعه قانون گزاره‌ای ارائه دهد.

نتایج بدست آمده از آزمایش روش FSRE بر روی برخی مسائل استاندارد در زمینه دسته‌بندی منجر به تولید قوانین مفیدتر و با دقت بالاتری نسبت به دیگر روش‌های موجود شد. نتایج این روش در مسائل مورد بررسی نشان داد که روش FSRE از نظر تعداد قوانین، شرایط درون قوانین و همچنین دقت قوانین بدست آمده به‌صورت بهینه عمل می‌کند.

همچنین با توجه به نتایج بدست آمده از روش FSRE برای مسئله شناسایی شدت بیماری کبد چرب، می‌توان نتیجه گرفت که این روش بر روی مسائل پیچیده نیز قابل استفاده می‌باشد.

۶-۳ پیشنهادات

کارهایی که در ادامه این روش می‌توان انجام داد عبارتنداز:

روش FSRE قادر است قوانین گزاره‌ای را برای مجموعه داده‌های گسسته و پیوسته، با استفاده از شبکه عصبی چند لایه تولید نماید. این در حالی است که امروزه استفاده از تئوری فازی در بسیاری از مسائل باعث افزایش دقت آن شده است. درصورتیکه با توسعه این روش، امکان استفاده از آن برای مسائل فازی نیز بوجود آید، می‌توان قابلیت درک و فهم قوانین تولید شده و دقت حاصل شده از قوانی را افزایش داد.

در این حالت امکان ایجاد سیتم‌های خبره از نوع فازی برای مسائل مختلف با میزان دقت مناسب حاصل می‌گردد.

روش پیشنهادی تنها بر روی شبکه عصبی نوع MLP دارای ساختار استاندارد سه لایه قابل اعمال می‌باشد. در یکی از کارهای پیش‌رو می‌توان روش معرفی شده را برای دیگر انواع شبکه عصی مانند RBF نیز مورد استفاده قرار داد. محققین می‌توانند با اعمال تغییرات جزئی، این روش را برای استفاده در دیگر انواع شبکه عصبی مناسب گردانند. بدین ترتیب می‌توان استخراج قانون از شبکه‌های عصبی دیگر را نیز با استفاده از این روش با دقت مناسب انجام داد.

روش FSRE از دیدگاه تجزیه‌ای برای استخراج قانون از شبکه عصبی استفاده می‌نماید. این حالت باعث می‌شود که قوانین بدست آمده مبتنی بر ساختار درونی شبکه عصبی باشند. در این حالت نمی‌توان از این روش بر روی ابزارات دیگر داده‌کاوی مانند ماشین بردار پشتیبان و شبکه بیزین استفاده نمود. درحالیکه اگر بتوان برای تولید قوانین از دیدگاه آموزشی استفاده کرد، آنگاه می‌توان از این روش بر روی ابزارات دیگر داده کاوی نیز استفاده نمود. روش‌های مبتنی بر دیدگاه آموزشی (بر خلاف روش‌های مبتنی بر دیدگاه تجزیه‌ای) محدودیت استفاده از یک ابزار خاص را برای تولید قوانین ندارند.

پیوست‌ها

پیوست ۱: معرفی مختصر مجموعه داده‌های نامبرده در این پایان‌نامه

در این قسمت برخی از مجموعه داده‌های نامبرده شده در این پایان نامه که توسط محققین قبلی به‌منظور آزمایش روش‌های ابداع شده مورد استفاده قرار گرفتند، به‌صورت مختصر توضیح داده می‌شوند. این مجموعه داده‌ها در دو دسته مجموعه داده‌های مورد استفاده برای مسائل نوع کلاس‌بندی و مجموعه داده‌های مورد استفاده در مسائل تخمین تابع قرار می‌گیرند.

مجموعه داده‌های کلاس‌بندی:

- **Golf Playing Problem**: این مسئله برای تعیین انجام و یا عدم انجام بازی گلف در شرایط آب و هوایی مختلف می‌باشد. مجموعه داده مورد استفاده در این مسئله شامل ۱۴ نمونه بوده و برای هر نمونه ۴ ویژگی ورودی وجود دارد. در این مجموعه داده ۹ نمونه عضو کلاس انجام بازی و باقی نمونه‌ها عضو کلاس عدم انجام بازی می‌باشند.
- **Wine Problem**: این مسئله به‌منظور شناسایی انواع مختلف نوشیدنی در ایتالیا بوده که از طریق آزمایشات شیمیایی انجام می‌شود. مجموعه داده مورد استفاده شامل ۱۷۸ نمونه بوده که برای هر نمونه ۱۳ ویژگی وجود دارد. این نمونه‌ها در سه کلاس مختلف دسته‌بندی می‌شوند که ۵۹ نمونه در کلاس اول، ۷۱ نمونه در کلاس دوم و باقی نمونه‌ها در کلاس سوم قرار می‌گیرند.
- **Glass Problem**: این مسئله به‌منظور شناسایی شیشه‌های پنجره‌ای و غیر پنجره‌ای مطرح می‌شود. در این مسئله از مجموعه داده‌ای استفاده می‌گردد که شامل ۲۱۴ نمونه می‌باشد و برای هر نمونه ۹ پارامتر ورودی وجود دارد. از این نمونه‌ها تعداد ۱۶۳ عدد در کلاس شیشه‌های پنجره‌ای و باقی نمونه‌ها در کلاس شیشه‌های غیر پنجره‌ای قرار می‌گیرند.
- **Iris Flower Problem**: این مجموعه داده شناخته شده ترین مجموعه داده در زمینه دسته‌بندی می‌باشد که استفاده‌های بیشماری از آن می‌گردد. در این مسئله هدف شناسایی

نوع گل زنبق بر اساس پارامترهای بدست آمده از گلبرگ و کاسبرگ آن است. مجموعه داده موجود برای این مسئله شامل ۱۵۰ نمونه بوده که برای هر نمونه ۴ ویژگی ورودی وجود دارد. این نمونه‌ها در ۳ کلاس مجزا دسته‌بندی می‌شوند که هر کلاس شامل ۵۰ نمونه می‌باشد.

- **Wisconsin Breast Cancer Problem**: این مسئله جهت انجام عملیات شناسایی سرطان سینه می‌باشد. در این مسئله سعی می‌شود تا نوع تومور[1] (خوش‌خیم[2] و بدخیم[3] بودن) را در بین بیماران بر اساس اطلاعات جمع‌آوری شده میکروسکوپی از سلول‌ها تشخیص داد. مجموعه داده‌ای که برای حل این مسئله مورد استفاده قرار می‌گیرد شامل ۶۹۹ نمونه بوده که برای هر نمونه ۹ ویژگی ورودی وجود دارد. برای این مسئله دو کلاس خوش‌خیم و بدخیم وجود دارد که ۴۵۸ نمونه در کلاس خوش‌خیم و باقی نمونه‌ها در کلاس بدخیم قرار دارند.
- **Mushroom Problem**: در این مسئله نمونه قارچ‌های مختلف به‌منظور دسته‌بندی در یکی از دو گروه سمی و خوراکی، مورد بررسی قرار می‌گیرند. در مجموعه داده مورد استفاده در این مسئله ۸۱۲۴ نمونه قارچ گوناگون وجود دارد که هر کدام از این نمونه‌ها عضو یکی از کلاس‌های سمی یا خوراکی هستند. تعداد نمونه‌های موجود در کلاس سمی برابر ۳۹۱۶ عدد و کلاس خوراکی شامل ۴۲۰۸ نمونه می‌باشد. جهت دسته‌بندی این نمونه‌ها از ۲۲ ویژگی در این مجموعه استفاده گردیده است.
- **BUPA Problem**: این مسئله یکی دیگر از مجموعه داده‌های مورد استفاده در عملیات دسته‌بندی می‌باشد که شامل یک مجموعه داده از افراد بیمار و غیر بیمار است. در این مجموعه ۳۴۵ نمونه وجود دارد و برای هر نمونه ۶ ویژگی موجود است. هر کدام از این نمونه‌ها توسط ویژگی‌های موجود در یکی از کلاس‌های سالم و یا بیمار قرار می‌گیرند.

[1] Tumor
[2] Benign
[3] Malignant

- **Diabetes Problem**: در این مسئله هدف تشخیص بیماری دیابت در بین افراد با استفاده از اطلاعات شخصی آنها می‌باشد. در مجموعه داده مورد استفاده در این مسئله ۷۶۸ نمونه وجود داشته که برای هر نمونه ۸ مقدار واقعی از ویژگی‌های مختلف ثبت شده است. تمامی مقادیر ثبت شده برای این ویژگی‌ها به‌صورت اعداد اعشاری می‌باشد. این مسئله یک مسئله دسته‌بندی دو کلاسه می‌باشد که که در آن ۶۵٫۱٪ از نمونه‌ها دارای نتیجه منفی در آزمایش دیابت می‌باشند. این مجموعه داده با نام Pima Indians Diabetes در پایگاه داده UCI وجود دارد.
- **Season Problem**: این مجموعه داده شامل مقادیر گسسته بوده که برای تشخیص نوع فصل از سال مورد استفاده قرار می‌گیرد. این مجموعه داده شامل ۱۱ نمونه بوده که برای هر نمونه ۳ ویژگی مجزا وجود دارد. هر نمونه در یکی از ۴ کلاس خروجی، مطابق با فصول مختلف سال دسته‌بندی می‌شوند.
- **Lung Cancer Problem**: این مسئله به تشخیص بیماری سرطان ریه پرداخته و حالت بیماری را در بین افراد تشخیص می‌دهد. در مجموعه داده استفاده شده برای این مسئله تعداد ۳۲ نمونه وجود داشته و برای هر نمونه ۵۶ ویژگی با مقادیر گسسته وجود دارد. هر کدام از نمونه‌های موجود در این مجموعه داده در یکی از سه کلاس خروجی (که حالت بیماری را نشان می‌دهند) قرار می‌گیرند.
- **Monks Problem**: این مسئله به بررسی و مقایسه الگوریتم‌های یادگیری متفاوت می‌پردازد. در مجموعه داده مورد استفاده برای این مسئله ۴۳۲ نمونه وجود دارد و برای هر نمونه مقادیر ۶ ویژگی مختلف در قالب مقادیر گسسته ذخیر شده است. نمونه‌های این مجموعه در دو کلاس متفاوت دسته‌بندی می‌شوند.
- **Credit Problem**: در این مسئله استفاده و عدم استفاده از کارت‌های اعتباری مورد بررسی قرار می‌گیرند. برای این مسئله تعداد ۶۹۰ نمونه جمع‌آوری گردیده است که هر نمونه شامل

مقادیر ۱۵ ویژگی مختلف می‌باشد. مقادیر ویژگی‌های این نمونه‌ها به‌صورت ترکیبی از مقادیر عددی و اسمی می‌باشد. برای این مسئله دو کلاس مثبت و منفی وجود دارد که تعداد نمونه‌های این مجموعه در کلاس مثبت برابر ۳۰۷ و تعداد نمونه‌های کلاس منفی برابر ۳۸۳ عدد می‌باشد.

مجموعه داده‌های تخمین تابع:

- **Pine Problem**: این مسئله برای بررسی نوعی کرم بر روی درختان کاج می‌باشد. اهمیت این مسئله در تعیین تاثیر شرایط مختلف در کلونی این کرم‌ها در جنگل‌های گوناگون است. این مجموعه داده دارای ۳۳ نمونه بوده که هر نمونه از یک منطقه خاص گرفته شده است و برای هر نمونه موجود تعداد ۱۰ ویژگی متفاوت ذخیره گردیده است.
- **Abalone Problem**: این مسئله برای تخمین سن صدف‌های دریایی مطرح می‌گردد. در مجموعه داده مورد استفاده برای این مسئله تعداد ۴۱۷۷ نمونه جمع‌آوری گردیده است. برای هر نمونه تعداد ۸ ویژگی مختلف از نوع صحیح، اسمی و اعشاری وجود دارد. در این مجموعه داده سن صدف‌های دریایی از ۱ سال تا ۲۹ سال تخمین زده می‌شود.
- **Sin-Cos Problem**: این مسئله یک مسئله تخمین تابع ریاضیاتی می‌باشد. نمونه‌های این مسئله شامل ۴ ویژگی ورودی هستند که سه ویژگی آن (θ, x, s) دارای مقادیر اعشاری و یک ویژگی $(\emptyset)$ دارای مقادیر صحیح (گسسته) می‌باشد. مقدار خروجی در این مسئله توسط تابع رابطه (۱) تولید می‌گردد. در این فرمول پارامترهای a ، b و c توسط کاربر برای تولید نمونه‌ها اعمال می‌گردد. در مجموعه داده تولید شده توسط محققین، مقادیر این پارامتر ها برای مجموعه آموزش بدین صورت بوده است: $a = 0.3$، $b = 0.0$ و $c = 0.2$. مقدار پارامتر c به‌عنوان میزان نویز در نظر گرفته شده است.

$$y = \sin(4\pi\theta - \emptyset) + ax + bs + c \qquad (۱)$$

- **Boston Housing Problem**: این مسئله به قیمت خانه‌های حوالی شهر بوستن می‌پردازد. در مجموعه داده مورد استفاده برای این مسئله تعداد ۵۰۶ نمونه وجود داشته که برای هر نمونه ۱۳ ویژگی با مقادیر پیوسته وجود دارد. هر یک از این نمونه‌ها، مشخصات یک خانه در این مجموعه داده می‌باشد که خروجی این مسئله نیز قیمت آن خانه بر حسب ۱۰۰۰ دلار است.

پیوست ۲: توابع فعال‌ساز (Activation Function)

اصولاً وقتی مقادیر ورودی به نرون با اعمال ماتریس وزن W وزن‌دار شدند، مقادیر وزن‌دار توسط تابع تحریک (فعال‌ساز) مقدار واقعی خروجی نرون را به خود می‌گیرند.

$$n = W \times P + b \quad (۲)$$

$$a_{out} = f(n) \quad (۳)$$

P: بردار ورودی

b: بایاس

a_{out}: بردار خروجی

$f()$: تابع فعال‌ساز

تابع فعال‌ساز بر اساس نیاز خاص حل مسئله انتخاب می‌شود که در عمل تعداد محدودی از این توابع مورد استفاده قرار می‌گیرند. توابعی که به خصوص در کاربردهای مهندسی مورد استفاده بیشتری قرار می‌گیرند، عبارتنداز: تابع حدی دومقداره (باینری)، تابع فعال‌ساز سیگموئید و یا تابع فعال‌ساز تانژانت هایپربولیک که در ادامه نشان داده می‌شوند.

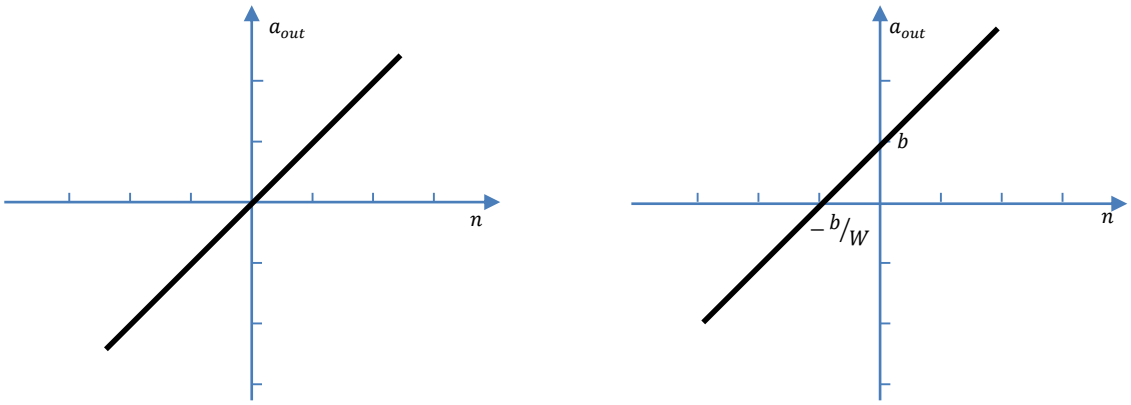
تابع تحریک خطی (Linear Function)

این تابع به‌صورت زیر تعریف می‌گردد.

$$f(n) = n \qquad (۴)$$

این بدان معنی است که خروجی این تابع برابر ورودی آن است. این تابع بیشتر در شبکه‌های آدلاین مورد استفاده قرار می‌گیرند.

باید توجه شود که جمله بایاس b که در رابطه (۲) نشان داده شده است، موجب جابه‌جایی منحنی در فضای ورودی می‌گردد که اهمیت آن را در شکل زیر می‌بینید.

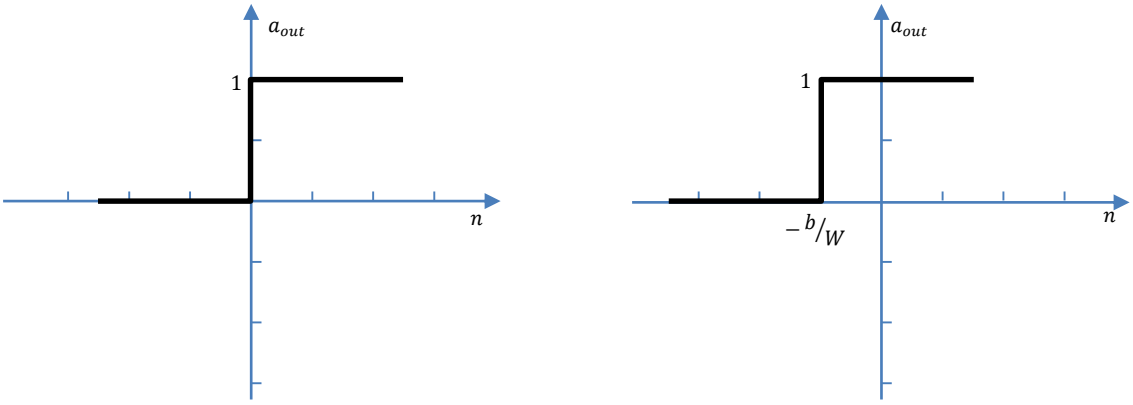


شکل ۱ یک تابع خطی و تاثیر اضافه شدن مقدار بایاس به آن

تابع حدی دو مقداری

خروجی این تابع معمولاً (0 یا 1) و یا (+1 یا -1) است. اگر مقدار ورودی وزن‌دار بزرگتر از $-b/W$ باشد مقدار تابع +1 و در غیر انصورت مقدار آن -1 و یا صفر (بسته به خواسته مسئله) می‌باشد. این تابع چون مقادیر را بین دو مقدار محدود می‌کند در مسائلی به کاربرده می‌شوند که بخواهیم داده‌ها را به طور خطی از هم جدا کنیم (مانند تابع فعال‌ساز خطی)

$$f(n) = \begin{cases} 0 & n < 0 \\ 1 & n > 0 \end{cases} \qquad (۵)$$



شکل ۲ تابع فعال‌ساز دو مقداری با عرض از مبداء صفر و اعمال مقدار بایاس

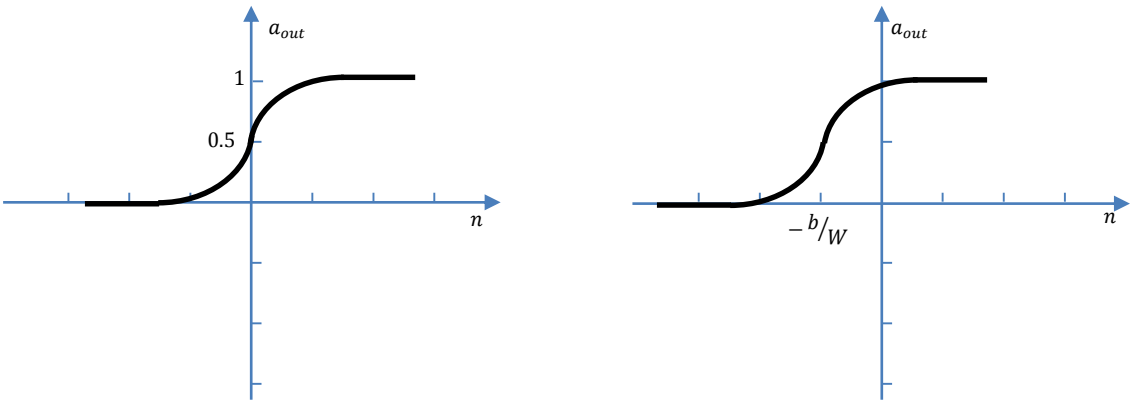
$$f(n) = \begin{cases} 0 & n < \dfrac{-b}{W} \\ 1 & n > \dfrac{-b}{W} \end{cases} \qquad (۶)$$

تابع فعال‌ساز سیگموئید (Sigmoid Function)

این تابع نیز مقادیر ورودی را بین صفر و یک محدود می‌کند؛ با این تفاوت که میزان تغییرات آن غیر خطی است. مزیت این تابع در عملکرد آن با ورودی‌های بزرگ و یا کوچک است.

هنگامی که مقادیر ورودی به سمت اعداد منفی بزرگ میل کنند مقدار صفر، هنگامی که مقادیر ورودی به سمت صفر میل کند مقدار تابع به سمت ۰٫۵ و هنگامی که مقادیر ورودی به سمت مقادیر مثبت بزرگ میل کند مقدار تابع به سمت ۱ میل خواهد کرد. فرم این تابع مطابق شکل ۳ بوده که مطابق رابطه (۷) تعریف می‌شود.

$$f(n) = \begin{cases} 0 & n < \dfrac{-b}{W} \\ 1 & n > \dfrac{-b}{W} \end{cases} \qquad (۷)$$

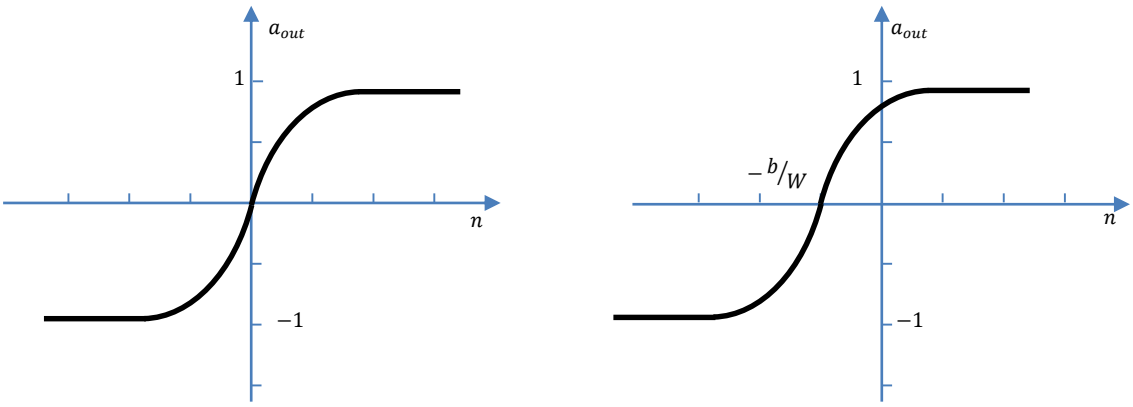


شکل ۳ تابع فعال‌ساز سیگموئید

تابع فعال‌ساز تانژانت هایپربولیک (Tangent Hyperbolic Function)

تابع فعال‌ساز دیگری که اغلب توسط بیولوژیست‌ها به‌عنوان مدل ریاضی از تحریک سلول عصبی مورد استفاده قرار می‌گیرد، تابع تانژانت هایپربولیک است که از نظر شکل شبیه تابع منطقی است.

$$f(n) = \tanh(n) = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}} \qquad (۸)$$



شکل ۴ تابع فعال‌ساز تانژانت هایپربولیک

پیوست ۳: الگوریتم یادگیری پس انتشار خطا (Back-propagation)

الگوریتم پس انتشار خطا، الگوریتم یادگیری است که برای شبکه‌های چند لایه پرسپترون به کار می‌رود. در این روش از آموزش با ناظر[1] استفاده می‌شود، بدین صورت که بردار خروجی مطلوب (Target) به‌عنوان ناظر در نظر گرفته می‌شود. مقدار خروجی شبکه (a_{out}) در طول فرایند یادگیری شبکه به بردار مطلوب نزدیک می‌گردد.

قانون پس انتشار خطا (BP) از دو مسیر اصلی تشکیل می‌شود. مسیر اول مسیر به مسیر رفت موسوم است. در این مسیر بردار ورودی به شبکه عصبی چند لایه پرسپترون (MLP) اعمال می‌شود و تأثیراتش از طریق لایه میانی یا همان لایه‌های پنهان به لایه‌های خروجی انتشار می‌یابد و توابع فعال‌ساز بر روی تک تک نرون‌های هر لایه عمل می‌کنند. در طول این مسیر پارامترهای شبکه ثابت بوده و بدون تغییر باقی می‌ماند. این مسیر با معادلات زیر بیان می‌شود.

$$a^{i+1}(K) = f^{i+1}\left(W^{i+1}(K) \times a^{i} + b^{i+1}(K)\right) \qquad (۹)$$
$$i = 0,1,\dots,L-1$$

i: شماره لایه در حال محاسبه می‌باشد که a^i به‌عنوان خروجی لایه قبلی برای محاسبات لایه $i+1$ ام به‌عنوان ورودی عمل می‌کند.

در مسیر دوم یا همان مسیر برگشت، برعکس مسیر رفت پارامترهای شبکه تغییر و تنظیم می‌گردند. این تغییرات بر اساس پروسه اصلاح خطایی که سیگنال خطا در لیه خروجی تشکیل داده است انجام می‌شوند. همانطور که می‌دانید، بردار خطا، اختلاف بین بردار پاسخ مطلوب و پاسخ واقعی شبکه می‌باشد که بعد از محاسبه در مسیر برگشت از لایه خروجی به سمت لایه‌های پیشین و از طریق لایه‌های شبکه در کل شبکه توزیع می‌گردد.

[1] Supervised learning

هنگامی که بردار مطلوب مقایسه می‌شود مقدار خطا در خروجی نرون n ام از لایه آخر برای K امین الگو (نمونه) به‌صورت زیر تعریف می‌شود.

$$e_n(K) = t_n(K) - a_n(K) \quad (۱۰)$$

بنابراین می‌توانیم مقدار لحظه‌ای خطا را برای نرون n ام از لایه خروجی، به‌صورت $e_n^2(K)$ تعریف کنیم. همین طور میزان خطای شبکه را با شاخص زیر که برابر مجموع مربعات خطا است، مشاهده کنیم.

$$\hat{F}(K) = \sum_{j=1}^{n} e_j^2(K) \quad (۱۱)$$

و برای اینکه بدانیم تا چه حد شبکه MLP آموزش دیده است و تابع $\hat{F}_{av}$ را که تقریبی از $\hat{F}$ می‌باشد، تعریف می‌کنیم.

$$\hat{F}_{avr}(K) = \frac{1}{n}\sum_{j=1}^{n} \hat{F}(j) \quad (۱۲)$$

در شبکه‌های چند لایه پرسپترون، برخلاف شبکه‌های تک لایه آن، مدل هر نرون دارای یک تابع فعال‌ساز غیر خطی مشتق‌پذیر باید باشد، زیرا برای محاسبه خطا از مشتق تابع فعال‌ساز استفاده می‌شود. از آنجا که مشتق توابع سیگموئید و تانژانت هایپربولیک به راحتی به دست می‌آیند، می‌توان در MLP از این توابع استفاده کرد.

$$f_{sig}(n) = \frac{1}{1 + e^{-n}} \quad (۱۳)$$

$$f'_{sig}(n) = f(n) \times \big(1 - f(N)\big)$$

$$f_{tanh}(n) = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}} \quad (۱۴)$$

$$f'_{tanh}(n) = 1 - f^2(n)$$

مشتق تابع محرک در $\frac{\partial \hat{F}(K)}{\partial W_n(K)}$ و $\frac{\partial \hat{F}(K)}{\partial b_n(K)}$ نمایان می‌شود به قسمی که با داشتن پارامترهای شبکه در هر لایه i می‌توانیم بنویسیم

$$\frac{\partial \hat{F}(K)}{\partial W^i(K)} = \frac{\partial \hat{F}(K)}{\partial e(K)} \times \frac{\partial e(K)}{\partial a^i(K)} \times \frac{\partial a^i(K)}{\partial n^i(K)} \times \frac{\partial n^i(K)}{\partial W^i(K)} \quad (۱۵)$$

$$\frac{\partial \hat{F}(K)}{\partial b^i(K)} = \frac{\partial \hat{F}(K)}{\partial e(K)} \times \frac{\partial e(K)}{\partial a^i(K)} \times \frac{\partial a^i(K)}{\partial n^i(K)} \times \frac{\partial n^i(K)}{\partial b^i(K)} \quad (۱۶)$$

که در آن:

$$\frac{\partial a^i(K)}{\partial n^i(K)} = f'\left(n^i(K)\right) \quad (۱۷)$$

$$\frac{\partial \hat{F}(K)}{\partial e(K)} = 2e(K) \quad (۱۸)$$

$$\frac{\partial e(K)}{\partial a^i(K)} = -1 \quad (۱۹)$$

$$\frac{\partial n^i(K)}{\partial b^i(K)} = 1 \quad (۲۰)$$

$$\frac{\partial n^i(K)}{\partial W^i(K)} = a^{i-1}(K) \quad (۲۱)$$

حال پارامترهای شبکه عصبی MLP را می‌توان در هر لایه و برای هر نرون آن لایه اصلاح کرد.

$$W_n(K+1) = W_n(k) - \alpha \times \frac{\partial \hat{F}(K)}{\partial W_n(K)} = W_n(K) - \alpha \times \Delta W_n(K) \quad (۲۲)$$

$$b_n(K+1) = b_n(k) - \alpha \times \frac{\partial \hat{F}(K)}{\partial b_n(K)} = b_n(K) - \alpha \times \Delta b_n(K) \quad (۲۳)$$

نحوه ارائه داده‌های یادگیری برای آموزش شبکه مهم است. ترتیب ارائه نمونه‌ها به شبکه عصبی باید طوری باشد که شبکه از امکان انتخاب و اموزش برابری نسبت به نمونه‌ها برخوردار باشد و بدین جهت باید نمونه‌ها را به‌صورت تصادفی به شبکه اعمال کنیم.

مراجع

1. Ahn BS, Cho SS, Kim CY (2000) The integrated methodology of rough set theory and artificial neural network for business failure prediction. Expert systems with applications 18 (2):65-74

2. Molina JM, Isasi P, Berlanga A, Sanchis A (2000) Hydroelectric power plant management relying on neural networks and expert system integration. Engineering Applications of Artificial Intelligence 13 (3):357-369

3. Amato F, Lopez A, Peña-Méndez EM, Vaňhara P, Hampl A, Havel J (2013) Artificial neural networks in medical diagnosis. Journal of applied biomedicine 11 (2):47-58

4. Chandra R, Chaudhary K, Kumar A (2007) The Combination and comparison of neural networks with decision trees for wine classification. School of sciences and technology, University of Fiji, in

5. Farissi IE, Azizi M, Lanet J-L, Moussaoui M (2013) Neural network vs. Bayesian network to detect Java card mutants. American Applied Science Research Institute (AASRI) Procedia Conference on Intelligent Systems and Control 4:132-137

6. Forgác R, Krakovsky R Neural network model for multidimensional data classification via clustering with data filtering support. In: 10th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2012. IEEE, pp 79-84

7. Krakovsky R, Forgac R Neural network approach to multidimensional data classification via clustering. In: 9th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2011. IEEE, pp 169-174

8. Gardner M, Dorling S (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment 32 (14):2627-2636

9. Jivani K, Ambasana J, Kanani S (2014) A Survey on Rule Extraction Approaches Based Techniques for Data Classification Using Neural Network. International Journal of Futuristic Trends in Engineering and Technology 1 (1):4-7

10. Sheela KG, Deepa SN (2013) Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering 2013:2-11

11. Zhang QJ, Gupta KC, Devabhaktuni VK (2003) Artificial neural networks for RF and microwave design-from theory to practice. IEEE Transactions on Microwave Theory and Techniques 51 (4):1339-1350

12. Dumitru C, Maria V (2013) Advantages and Disadvantages of Using Neural Networks for Predictions. Ovidius University Annals, Economic Sciences Series 13 (1):444-449

13. Augasta MG, Kathirvalavakumar T Rule Extraction from Neural networks. In: Proceedings of the international conference on pattern recognition, informatics and medical engineering, Salem, Tamilnadu, 2012. IEEE, pp 21-23

14. Gupta A, Park S, Lam SM (1999) Generalized analytic rule extraction for feedforward neural networks. IEEE Transactions on Knowledge and Data Engineering 11 (6):985-991

15. Setiono R (2000) Extracting M-of-N rules from trained neural networks. IEEE Transactions on Neural Networks 11 (2):512-519

16. Kulluk S, Özbakır L, Baykasoğlu A (2013) Fuzzy DIFACONN-miner: A novel approach for fuzzy rule extraction from neural networks. Expert Systems with Applications 40 (3):938-946

17. Setiono R, Liu H (1997) NeuroLinear: From neural networks to oblique decision rules. Neurocomputing 17 (1):1-24

18. Tsai CJ, Lee CI, Yang WP (2008) A discretization algorithm based on class-attribute contingency coefficient. Information Sciences 178 (3):714-731

19. ElAlami ME (2012) Destructive Algorithm for Rule Extraction based on a Trained Neural Network. International Journal of Computer Applications 42 (21):8-14

20. Kamruzzaman SM, Hasan AR Rule Extraction using Artificial Neural Networks. In: Proceeding International Conference on Information and Communication Technology in Management (ICTM 2005), Malaysia, 2005. Multimedia University, pp 1-14

21. Kamruzzaman SM, Islam M (2006) An algorithm to extract rules from artificial neural networks for medical diagnosis problems. International Journal of Information Technology 12 (8):41-59

22. Kamruzzaman SM, Islam M (2007) Extraction of symbolic rules from artificial neural networks. International Journal of Computer, Information Science and Engineering 1 (10):3022-3028

23. Setiono R, Leow WK (2000) FERNN: An algorithm for fast extraction of rules from neural networks. Applied Intelligence 12 (2):15-25

24. Setiono R, Tanaka M Neural network rule extraction and the LED display recognition problem. In: 22nd International Conference on Tools with Artificial Intelligence (ICTAI), 2010. IEEE, pp 53-56

25. Kolman E, Margaliot M (2007) Knowledge extraction from neural networks using the all-permutations fuzzy rule base: the LED display recognition problem. IEEE Transactions on Neural Networks 18 (3):925-931

26. Wang L, Fu X (2005) A simple rule extraction method using a compact RBF neural network. In: Advances in Neural Networks. Springer, Verlag Berlin Heidelberg, pp 682-687

27. McGarry K, Malone J (2004) Analysis of rules discovered by the data mining process. In: Applications and Science in Soft Computing. Springer, pp 219-224

28. Malone J, McGarry K, Wermter S, Bowerman C (2006) Data mining using rule extraction from Kohonen self-organising maps. Neural Computing & Applications 15 (1):1-16

29. Ramadass S, Abhraham A Rule Extraction from SOM for Academic Evaluation. In: International Conference on Affective Computing and Intelligent Interaction, 2012. vol 2. pp 184-189

30. Fortuny EJd, Martens D (2012) Active learning-based pedagogical rule extraction. IEEE Transactions on Neural Networks and Learning Systems 26 (11):1-14

31. Saito K, Nakano R (2002) Extracting regression rules from neural networks. Neural networks 15 (10):1279-1288

32. Setiono R, Thong JYL (2004) An approach to generate rules from neural networks for regression problems. European Journal of Operational Research 155 (1):239-250

33. Setiono R, Leow WK, Zurada JM (2002) Extraction of rules from artificial neural networks for nonlinear regression. IEEE Transactions on Neural Networks 13 (3):564-577

34. Setiono R, Gaweda A Neural network pruning for function approximation. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN) 2000, 2000. IEEE, pp 443-448

35. Kamruzzaman SM REx: An Efficient Rule Generator. In: Proceedings of the 4th International Conference on Electrical Engineering & 2nd Annual Paper Meet, 2006. pp 79-82

36. Schmitz GPJ, Aldrich C, Gouws FS (1999) ANN-DT: an algorithm for extraction of decision trees from artificial neural networks. IEEE Transactions on Neural Networks 10 (6):1392-1401

37. Sethi KK, Mishra DK, Mishra B KDRuleEx: A Novel Approach for Enhancing User Comprehensibility Using Rule Extraction. In: Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 2012. IEEE, pp 55-60

38. Quinlan JR (2014) C4.5: programs for machine learning. Morgan Kaufmann Publisher, San Matio, California

39. R.Hruschka E, Ebecken NFF (2006) Extracting rules from multilayer perceptrons in classification problems: a clustering-based approach. Neurocomputing 70 (1):384-397

40. Kudova P Clustering genetic algorithm. In: 18th International Workshop on Database and Expert Systems Applications, Regensburg, 2007. IEEE, pp 138-142

41. Lu H, Setiono R, Liu H (1996) Effective data mining using neural networks. IEEE Transactions on Knowledge and Data Engineering 8 (6):957-961

42. Kaczmar UM, Trelak W (2005) Fuzzy logic and evolutionary algorithm—two techniques in rule extraction from neural networks. Neurocomputing 63:359-379

43. Yu S, Guo X, Zhu K, Du J (2010) A neuro-fuzzy GA-BP method of seismic reservoir fuzzy rules extraction. Expert Systems with Applications 37 (3):2037-2042

44. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11 (4):341-359

45. Özbakir L, Baykasoğlu A, Kulluk S, Yapıcı H (2009) TACO-miner: an ant colony based algorithm for rule extraction from trained neural networks. Expert Systems with Applications 36 (10):12295-12305

46. Liu H, Setiono R Chi2: Feature selection and discretization of numeric attributes. In: Proceeding of the 7th International Conference on Tools with Artificial Intelligence Herndon, VA, 1995. IEEE, p 388

47. Setiono R (1997) Extracting rules from neural networks by pruning and hidden-unit splitting. Neural Computation 9 (1):205-225

48. Fu X, Wang L Rule extraction using a novel gradient-based method and data dimensionality reduction. In: Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IEEE, pp 1275-1280

49. Roobaert D, Karakoulas G, Chawla NV (2006) Information gain, correlation and support vector machines. In: Feature Extraction. Springer, Berlin Heidelberg, pp 463-470

50. Kurgan L, Cios KJ (2004) CAIM discretization algorithm. IEEE Transactions on Knowledge and Data Engineering 16 (2):145-153

51. Kurgan L, Cios KJ Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm. In: Proceedings of the 2003 International Conference on Machine Learning and Applications, Los Angeles, California, USA, 2003. CSREA Press, pp 30-36

52. Vora SV, Mehta RG (2012) MCAIM: Modified CAIM Discretization Algorithm for Classification. International Journal of Applied Information Systems 3 (5):42-50

53. Cano A, Nguyen DT, Ventura S, Cios KJ (2014) ur-CAIM: improved CAIM discretization for unbalanced and balanced data. Soft Computing:1-16

54. Siraj F, Omer EAOA, Hasan R (2012) Data Mining and Neural Networks: The Impact of Data Representation. In: Karahoca A (ed) Advances in Data Mining Knowledge Discovery and Applications. INTECH Open Access Publisher, pp 97-116

55. Thimm G, Fiesler E (1997) Pruning of neural networks. IDIAP, Martigny, Valais, Switzerland

56. UCI Machine Learning. Bren School. http://archive.ics.uci.edu/ml/machine-learning-databases/. Accessed 27/1/2016 2016

57. Kamruzzaman SM (2007) RGANN: An Efficient Algorithm to Extract Rules from ANNs. Journal of Electronics and Computer Science 8:19-30

58. Kamruzzaman SM, Sarkar AM (2011) A new data mining scheme using artificial neural networks. Sensors 11 (5):4622-4647

59. Setiono R, Liu H (1996) Symbolic representation of neural networks. Computer 29 (3):71-77

60. Pham DT, Aksoy MS (1993) An algorithm for automatic rule induction. Artificial Intelligence in Engineering 8 (4):277-282

61. Farquad MAH, Sultana J, Nagalaxmi G, Savankumar G (2014) Knowledge Discovery from Data: Comparative Study. Transactions on Engineering and Sciences 2 (6):72-75

62. Markowska-Kaczmar U, Zagórski R Hierarchical evolutionary algorithms in the rule extraction from neural networks. In: ACM Computing SurveysInternational Symposium on Engineering of Intelligent Systems, 2004.

63. Behloul F, Lelieveldt BPF, Boudraa A, Reiber JHC (2002) Optimal design of radial basis function neural networks for fuzzy-rule extraction in high dimensional data. Pattern recognition 35 (3):659-675

64. Kubat M (1998) Decision trees can initialize radial-basis function networks. IEEE Transactions on Neural Networks 9 (5):813-821

65. Shaabani B, Sajedi H (2014) MMDT: Multi-Objective Memetic Rule Learning from Decision Tree. Journal of Computer & Robotics 5 (1):37-46

66. Hong TP, Chen JB (2000) Processing individual fuzzy attributes for fuzzy rule induction. Fuzzy sets and systems 112 (1):127-140

67. Fujiwara S, Hongou Y, Miyaji K, Asai A, Tanabe T, Fukui H, Tsuda Y, Fukuda A, Masuda D, Arisaka Y (2007) Relationship between liver fibrosis noninvasively measured by FibroScan and blood test. Bulletin of the Osaka Medical College 53 (2):93-105

68. Paredes AH, Torres DM, Harrison SA (2012) Nonalcoholic fatty liver disease. Clinics in liver disease 16 (2):397-419

69. Afdhal NH (2012) Fibroscan (transient elastography) for the measurement of liver fibrosis. Gastroenterology & hepatology 8 (9):605-607

70. Lok ASF, Ghany MG, Goodman ZD, Wright EC, Everson GT, Sterling RK, Everhart JE, Lindsay KL, Bonkovsky HL, Bisceglie AMD (2005) Predicting cirrhosis in patients with hepatitis C based on standard laboratory tests: Results of the HALT-C cohort Potential conflict of interest: Nothing to report. Hepatology 42 (2):282-292

71. Forns X, Ampurdanes S, Llovet JM, Aponte J, Quintó L, Bauer EM, Bruguera M, Tapias JMS, Rodés J (2002) Identification of chronic hepatitis C patients without hepatic fibrosis by a simple predictive model. Hepatology 36 (4):986-992

72. Angulo P, Hui JM, Marchesini G, Bugianesi E, George J, Farrell GC, Enders F, Saksena S, Burt AD, Bida JP (2007) The NAFLD fibrosis score: a noninvasive system that identifies liver fibrosis in patients with NAFLD. Hepatology 45 (4):846-854

73. مرگان‌پور ن, حسن‌پور ح (۱۳۹۲) پیش‌بینی شدت آسیب‌های کبدی در بیماران کبد چرب بااستفاده از شبکه های عصبی مصنوعی. دانشگاه شاهرود, شاهرود

74. Jamali R, Jamali A (2010) Non-alcoholic fatty liver disease. Journal of Kashan University of Medical Sciences 14 (2):169-181

75. Janda JM, Abbott SL (2006) The enterobacteria. 2nd edn. American Society of Microbiology, Washington

Abstract

In the last decade the use of artificial intelligence techniques to solve scientific problems has increased dramatically. Artificial neural networks in classification and pattern recognition issues are widely used. The lack of capability in interpreting the results is one of the biggest problems of artificial neural networks. Although, artificial neural networks are able to carry out the classification operations accurately the lack of transparency of the knowledge provided by the neural network makes the results can't be easily used in other systems. Therefore, researchers are trying to fix this problem by extracting the rules of neural networks. Today, there are several methods of rule extraction which are often able to present the operations performed by an artificial neural network on a simple data collection in the form of a set of simple rules. However, these methods are usually less accurate in the case of relatively complex issues and the precision resulted from rules is not comparable to the results of the neural network.

In this project, a new method for extracting the rule from artificial neural networks titled as Four Step Rule Extraction (FSRE) has been introduced. This method is able to convert the classification carried out on a data set into a set of simple, useful and efficient propositional rules. In using this method, there is no limitation on the complexity of the data. To evaluate the performance and results of this method, the common classification data sets (from the database UCI) is used in this thesis. The results showed that the proposed method is more efficient and accurate compared to the available methods of the rule extraction.

Keyword: *Artificial Neural Network (ANN), Rule Extraction, Classification, Data Mining*

Shahrood University of Technology
Department of Computer and Information Technology
Computer & Artificial intelligent

Using neural network for knowledge acuisition in multi-dimensional data

Mojtaba Shahabi

Supervisor:

Dr. Hamid Hassanpour

Adviser:

Dr. Hoda Mashayekhi

February, 2016