
بسم الله الرحمن الرحیم

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد مهندسی هوش مصنوعی

بهبود دقت قوانین استخراج شده توسط شبکه‌های عصبی چند لایه با بهره‌گیری از الگوریتم تکاملی در آموزش و هرس کردن شبکه

نگارنده: یاسر ایرانی

استاد راهنما

دکتر حمید حسن‌پور

بهمن ۱۳۹۷

شماره: ۶۱۳/ر/ف/ک

تاریخ: ۹۷/۱۲/۱۱

باسمه‌تعالی

مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد **آقای یاسر ایرانی** با شماره دانشجویی ۹۵۰۲۱۵٤ رشته مهندسی کامپیوتر **گرایش هوش مصنوعی و رباتیکز** تحت عنوان **بهبود دقت قوانین استخراج شده توسط شبکه‌های عصبی چند لایه با بهره‌گیری از الگوریتم تکاملی در آموزش و هرس کردن شبکه** که در تاریخ ۱۳۹۷/۱۱/۷ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با درجه: عالی) ☑ **مردود ☐**

نوع تحقیق: نظری ☐ **عملی ☐**

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضاء
۱ـ استادراهنمای اول	حمید حسن‌پور	استاد	
۲- نماینده تحصیلات تکمیلی	محسن فرهادی	مربی	
۳- استاد ممتحن اول	محسن بیگلری	استادیار	
۴- استاد ممتحن دوم	هادی گرایلو	استادیار	

نام و نام خانوادگی رئیس دانشکده:

تاریخ و امضاء و مهر دانشکده:

تبصره: در صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تقدیم به

سپاسگزار کسانی هستم که سرآغاز تولد من هستند. از یکی زاده می شوم و از دیگری جاودانه. استادی که سپیدی را بر تخته سیاه زندگیم نگاشت و مادری که تار مویی از او بپای من سیاه نماند.

تقدیم به

مقدس ترین واژه ها در لغت نامه دلم، مادر مهربانم که زندگیم را مدیون مهر و عطوفت آن می دانم.

پدر، مهربانی مشفق، بردبار و حامی.

برادر و خواهرانم، همراهان همیشگی و پشتوانه های زندگیم.

سپاسگزاری

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست. از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر حمید حسن‌پور، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید.

یاسر ایرانی

بهمن ۱۳۹۷

تعهد نامه

اینجانب **یاسر ایرانی** دانشجوی دوره کارشناسی ارشد رشته **مهندسی کامپیوتر** دانشکده **مهندسی کامپیوتر** دانشگاه صنعتی شاهرود، نویسنده پایان‌نامه با عنوان **بهبود دقت قوانین استخراج شده توسط شبکه‌های عصبی چند لایه با بهره‌گیری از الگوریتم تکاملی در آموزش و هرس کردن شبکه** ، تحت راهنمایی **حمید حسن‌پور** متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام‘‘ دانشگاه صنعتی شاهرود ’’ یا “ Shahrood University of Technology ” به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

یاسر ایرانی

بهمن ۱۳۹۷

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

چکیده

امروزه شاهد جمع‌آوری داده‌های فراوانی در زمینه‌های مختلفی از جمله پزشکی، تجاری و صنعتی هستیم. این داده‌ها برای مقاصد گوناگونی گردآوری می‌شوند. یکی از اهداف مهم جمع آوری داده‌ها، تحقیق روی آن‌ها برای بدست آوردن نتایج و الگوهای مفید موجود در آن‌ها است. پیچیدگی و حجم زیاد داده‌ها، موجب سردرگمی در تحلیل این داده‌ها شده است. از این رو، روش‌های داده‌کاوی متعددی برای تجزیه و تحلیل داده‌ها ارائه شده است. استخراج قانون یکی از کاربردهای مهم داده‌کاوی برای کشف دانش و روابط بین داده‌ها است. استخراج قانون به کمک ابزارهای مختلفی مانند شبکه‌عصبی، درخت تصمیم و ماشین بردار پشتیبان انجام می‌شود. شبکه‌عصبی، به دلیل عدم نیاز به دانش اولیه در مورد داده‌ها و داشتن دقت بالا، بیشتر مورد توجه قرار گرفته است.

اغلب روش‌های ارائه شده برای استخراج قانون متکی بر هرس‌سازی شبکه هستند، هرس‌سازی باعث کاهش دقت و کارایی شبکه‌عصبی و در نتیجه ضعف در قوانین استخراج شده می‌شود. در این پژوهش روش نوینی ارائه می‌شود که با بهره‌گیری از الگوریتم ژنتیک، وزن‌های نرون‌های شبکه‌عصبی را به گونه‌ای تنظیم می‌کند که خسارت ناشی از مرحله هرس‌سازی را به میزان قابل توجهی تقلیل دهد. این روش در مرحله اول، به هنگام آموزش شبکه‌عصبی و به شکل سراسری و در مرحله بعد، پس از اتمام آموزش به صورت محلی با بررسی وزن‌های هر نرون از لایه میانی و خروجی، به اصلاح وزن‌های اتصالات شبکه می‌پردازد. آزمایشات انجام شده بر روی چند پایگاه‌داده استاندارد نشان می‌دهد که قوانین استخراج شده به کمک روش پیشنهادی از نظر سادگی بهتر از روش‌های موجود و از نظر دقت بطور متوسط دو درصد بهبود می‌یابد.

کلمات کلیدی: استخراج قانون، الگوریتم ژنتیک، شبکه‌عصبی مصنوعی، کشف روابط بین داده‌ای، هرس‌سازی شبکه‌عصبی، کشف دانش

لیست مقالات مستخرج از پایان‌نامه

۱. ایرانی، ی. حسن‌پور، ح. (۱۳۹۷)، ’’استفاده از خوشه‌بندی برای بهبود گسسته‌سازی خروجی لایه مخفی’’ بیست و چهارمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، دانشگاه صنعتی شریف، تهران.

۲. ایرانی، ی. حسن‌پور، ح. (۱۳۹۸)، ’’استفاده از الگوریتم ژنتیک برای بهبود هرس‌سازی شبکه‌عصبی در استخراج قانون’’ بیست و هفتمین کنفرانس مهندسی برق ایران، دانشگاه یزد، یزد.

فهرست مطالب

فهرست تصاویر

فهرست جداول

فهرست اختصارات

فصل ۱

مقدمه

۱.۱ مقدمه

امروزه با توجه به تولید حجم عظیمی از داده‌ها در زمینه‌های مختلف از جمله پزشکی [۳] ، تجاری [۲] و صنعتی [۳۸]، کشف روابط میان این داده‌ها برای اهداف مختلف از اهمیت ویژه‌ای برخوردار است. یادگیری ماشین[1] شاخه‌ای از هوش مصنوعی[2] است که به دنبال استقراء[3] یا پالایش[4] دانش است. استنتاج مفاهیم عمومی از نمونه‌های آموزشی خاص، با شناسایی ویژگی‌هایی که بتوانند نمونه‌های منفی از مثبت را تشخیص دهند، برای یادگیری ماشین اهمیت دارد. در طول دهه گذشته، یادگیری ماشین از حالت تئوری به جایی رسیده

[1] Machine learning
[2] Artificial intelligence
[3] Knowledge induction
[4] Knowledge refinement

است که ارزش تجاری قابل توجهی پیدا کرده است. الگوریتم های یادگیری ماشین با موفقیت توانسته‌اند در مسائل کاربردی گوناگونی به کار گرفته شوند [۵۳، ۶، ۳۲]. استخراج قانون[1] یکی از کاربردهای مهم یادگیری ماشین و داده‌کاوی[2] برای کشف روابط پنهان میان داده‌ها است. استخراج قانون برای یک مدل پیش‌بینی کننده غیرشفاف و داده‌هایی که با آن آموزش دیده است، یک توصیف قابل درک برای فرضیه های مدل پیش بینی کننده ارائه می دهد که رفتاری نزدیک به آن را دارد. استخراج قانون به کمک ابزارهای گوناگونی مانند شبکه‌عصبی مصنوعی (ANN[3]) [۲۰]، درخت تصمیم[4] [۵۲] و ماشین بردار پشتیبان (SVM[5]) [۴۰] انجام می‌گیرد. در این میان شبکه‌های عصبی، به خصوص شبکه‌عصبی چند لایه پرسپترون (MLP[6])، به دلیل عدم نیاز به دانش اولیه در مورد داده‌ها و دقت[7] بالا، یکی از پرکاربردترین ابزارهای موجود در این حوزه است.

اگرچه شبکه‌های عصبی به طور کلی عملکرد بهتری را در مسائل طبقه‌بندی[8] الگوها ارائه می‌دهند، اغلب از آن‌ها به عنوان یک جعبه سیاه[9] یاد می‌شود [۱۱، ۱۳]. به عبارت دیگر، پیش‌بینی‌های آن‌ها به اندازه درخت‌های تصمیم قابل تفسیر نیستند. با استفاده از یک الگوریتم برای استخراج قانون، می‌توان پیش‌بینی‌های انجام شده توسط شبکه‌عصبی را درک کرد، این قوانین در عین حال که می‌توانند دقت شبکه‌عصبی را تا حد خوبی حفظ کنند، دقت بالاتری را در مقایسه با درخت‌های تصمیم ارائه می‌دهند. پس اگر بتوانیم قوانینی را از شبکه‌های عصبی مانند قوانین تولید شده از درخت‌های تصمیم استخراج کنیم، مطمئنا پیش‌بینی انجام شده توسط شبکه را می‌توانیم بهتر تفسیر کنیم. علاوه بر این، قوانین یک نوع دانش هستند که کارشناسان می‌توانند به راحتی آنها را تأیید، رد و یا توسعه دهند [۴۹].

سیستم‌های یادگیري ماشین نمادین[10]، ارائه‌ای صریح از دانش موجود در یک مسئله دارند که می‌تواند

---

[1] Rule Extraction
[2] Data mining
[3] Artificial Neural Network
[4] Decision tree
[5] Support Vector Machine
[6] Multi-layer Perceptron
[7] Accuracy
[8] Classification
[9] Black Box
[10] Symbolic machine learning

چگونگی دستیابی به نتایج را برای کاربر شرح دهد. یک سیستم کاربر پسند[1]، به طور گسترده در برنامه‌های کاربردی[2] مورد نیاز است. کاربرد موفق روش‌ها و سیستم‌های شبکه‌عصبی در حوزه‌هایی مانند تجارت[۱۷]، مهندسی، پزشکی[۳۲]، اجتماعی [۵]و غیره نشان دهنده توانایی بالای این ابزار است [۴]. با این حال، شبکه‌های عصبی نیازمند یک مکانیزم توضیحی مشابه سیستم‌های یادگیري ماشین نمادین دارند. این امر کاربران را قادر می‌سازد فرآیند تصمیم گیری خود را درک کنند و از این طریق شبکه‌های عصبی به کاربران گسترده‌تری دست یابند.

۲.۱ اهمیت استخراج قانون از شبکه‌عصبی

برطبق تحقیقات صورت گرفته در این حوزه، مدل‌های جعبه سیاه به طور میانگین عملکرد بهتری را نسبت به مدل‌های جعبه سفید[3] ارائه می‌دهند. در نتیجه قوانین استخراج شده از این الگوریتم‌ها کیفیت بالاتری از مدل‌های جعبه سفید خواهند داشت. از طرفی استفاده از این مدل‌ها در زمینه‌هایی مانند پزشکی و مالی که نیاز به تایید متخصصان دارد، امکان‌پذیر نمی‌باشد. همچنین استفاده از این ابزار در کاربردهایی از داده‌کاوی که هدف بدست آوردن دانش در مورد داده‌های ورودی است، ممکن نیست. استخراج قانون، استفاده از مدل‌های جعبه سیاه در این زمینه‌ها را امکان‌پذیر می‌سازد [۹].

در سال‌های اخیر شبکه‌های عصبی از یک رویکرد تئوری به یک تکنولوژی پرکاربرد در مسائل مختلف تبدیل شده‌اند. بیشتر این کاربردهای مربوط به شناسایی الگو و شبکه‌های عصبی اغلب به عنوان تخمین‌گر غیرخطی[4] استفاده می‌شوند که در مقابل نویز مقاوم هستند. روش‌های یادگیری شبکه‌عصبی مصنوعی، یک رویکرد قدرتمند و غیر خطی برای تقریب تابع هدف برای بسیاری از مسائل طبقه‌بندی، رگرسیون[5] و خوشه‌بندی[6]

[1] User friendly
[2] Application
[3] White Box
[4] Non-linear approximators
[5] Regression
[6] Clustering

است. شبکه‌عصبی عملکرد خوبی را در طیف وسیعی از مسائل عملی نشان داده است. با این حال، دلایل محکمی برای مناسب نبودن شبکه‌های عصبی برای ارائه دانش، وجود دارد. ناتوانی در توضیح ساختارها و قابلیت درک پایین شبکه آموزش دیده، این که چرا بعضی از داده‌ها به یک دسته خاص تعلق می‌گیرند، از جمله مهم‌ترین این دلایل هستند. بنابراین، شبکه‌های عصبی، برای افزایش میزان مقبولیت خود برای طیف گسترده‌تری از کاربران به عنوان یک سیستم تصمیم‌گیری[1] و ارتقای میزان سودمندی خود به عنوان یک ابزار مفید و تعمیم‌پذیر یادگیری ماشین، باید بر محدودیت‌های خود غلبه کند [۱].

مسئله نحوه تصمیم‌گیری یک شبکه آموزش دیده، مشکلی دیرینه در حوزه مدل‌سازی پیوندگرا[2] است. دانش در یک شبکه‌عصبی به صورت توزیع شده در وزن‌های عددی شبکه تعبیه شده است. یک رویکرد امید بخش برای حل این مشکل، نمایش پارامترهای وزنی به صورت یک توصیف نمادین است. تفسیر نمادین دانش تعبیه شده در شبکه آموزش دیده، رویکردی برای غلبه بر مشکل ''جعبه سیاه'' است. این نوع از اصلاح فرموله‌سازی[3] که به عنوان استخراج قانون شناخته می‌شود، می‌تواند رفتار شبکه را به خوبی توضیح و به انتقال دانش بدست آمده، کمک کند. تکنیک‌های شبکه‌عصبی روابط نهفته میان داده‌ها را تشخیص و رفتار ویژگی‌ها را پیش‌بینی می‌کند، تکنیک‌های استخراج قانون ابزارهایی برای درک فرآیند تصمیم‌گیری شبکه‌های عصبی هستند [۳۹].

نیاز به درک رفتار شبکه‌عصبی در حوزه‌های بحرانی[4]، منجر به ارائه رویکردهای زیادی برای توضیح فرآیند تصمیم‌گیری شبکه با استفاده از انواع مختلفی از قوانین شده است. قوانین استخراج شده اغلب به شکل گزاره‌ای[5] هستند. با این حال، محدودیت‌های ذاتی در این نوع قوانین وجود دارد. مجموعه قوانین استخراج شده از شبکه، ممکن است بیش از حد برای داده‌های طبقه‌بندی شده دقیق باشد، در حالی که مسئله واقعی

[1] Decision System
[2] Connectionist modeling
[3] Reformulation
[4] Critical
[5] Propositional

به درستی توصیف نشده باشد [۵۶].

۳.۱ مراحل استخراج قانون از شبکه‌عصبی

اساسا، روش‌های استخراج قانون از شبکه‌عصبی در سه گروه آموزشی [۱] ، تجزیه‌ای [۲] و ترکیبی [۳] قرار می‌گیرند [۶]. در دیدگاه آموزشی، قوانین برطبق یک تحلیل تجربی از الگوهای ورودی–خروجی تولید می‌شوند که به آن رویکرد جعبه سیاه نیز گفته می‌شود. در تکنیک‌های تجزیه‌ای قوانین با بررسی وزن‌های هر نرون[۴] لایه مخفی و خروجی تعیین می‌شوند. در نهایت، دیدگاه ترکیبی هر دو دیدگاه آموزشی و تجزیه‌ای را ادغام می‌کند، این روش‌ها از دانش موجود در ساختار درونی شبکه برای بهبود دیدگاه آموزشی استفاده می‌کنند. در این پژوهش استخراج قانون از شبکه‌عصبی با استفاده از دیدگاه تجزیه ای مورد توجه قرار گرفته است. روش‌های تجزیه‌ای برای استخراج قانون از شبکه‌عصبی، به دلیل توجه به ساختار داخلی شبکه و ارتباطات موجود، نتایج بهتری در مقایسه با دیدگاه آموزشی ارائه می‌دهند. در دیدگاه تجزیه‌ای، استخراج قانون با تجزیه و تحلیل وزن‌ها، بایاس‌ها[۵] و مقادیر تابع فعال‌ساز[۶]، به صورت مجزا برای هر نرون از لایه مخفی و خروجی انجام میشود. دیدگاه آموزشی، بر خلاف دیدگاه تجزیه‌ای، به ساختار داخلی شبکه توجهی ندارد و صرفا با تحلیل خروجی‌های شبکه به عنوان یک جعبه سیاه برای نمونه‌های ورودی به استخراج قانون می‌پردازد [۶].

به طور کلی استخراج قانون با استفاده از دیدگاه تجزیه‌ای طی چند مرحله انجام می‌شود:

- **پیش پردازش[۷] داده‌ها:** معمولا داده‌ها به صورت خام به شبکه اعمال نمی‌شوند. مقادیر خام گستردگی زیادی دارند، ممکن است برای یک مسئله برخی از ویژگی‌ها غیر موثر و یا اضافی باشند. بدین ترتیب

[1] Pedagogical
[2] Decompositional
[3] Eclectic
[4] Neuron
[5] Bias
[6] Activation Function
[7] Preprocessing

مجموعه عملیاتی در قالب پیش پردازش قبل از انجام عملیات اصلی برروی داده‌ها استفاده می‌شود. گسسته‌سازی و رقمی‌کردن[1]، نرمالسازی داده‌ها[2]، نمونه‌برداری[3]، تجمیع[4]، کاهش بعد[5] وغیره از جمله عملیاتی است که برروی داده‌ها انجام می‌شود.

- **تعیین ساختار مناسب شبکه و آموزش آن:** با توجه به داده‌های موجود و قوانین مورد نیاز ساختار شبکه‌عصبی مشخص می‌شود. تعیین ساختار مناسب و بهینه موجب تولید نتایج قابل قبولی خواهد شد. تعداد نرون‌های لایه ورودی و خروجی با توجه به داده‌ها در مرحله قبل تعیین می‌گردد، اما تعداد لایه‌های مخفی و تعداد نرون‌های موجود در هر لایه با توجه به میزان پیچیدگی و پراکندگی داده‌ها تعیین می‌شود.

- **هرس‌سازی[6] شبکه‌عصبی:** در این فاز، هدف، ساده‌سازی شبکه به منظور کاهش ابهام و پیچیدگی به منظور استخراج قوانین است. در این گام برخی اتصالات و گره‌های کم اثر حذف خواهند شد. این امر ممکن است باعث کاهش دقت و کارایی[7] شبکه شود.

- **گسسته‌سازی خروجی لایه مخفی:** استخراج قوانین با تحلیل خروجی لایه مخفی انجام می‌شود. الگوها با استفاده از دسته‌بندی این مقادیر تولید و در نهایت قوانین استخراج می‌شوند. پیچیدگی قوانین استخراج شده از شبکه، وابسته به مقادیر گسسته شده‌ی خروجی لایه مخفی است.

- **استخراج قانون:** در انتها قوانین با تجزیه و تحلیل ساختار هرس شده شبکه و مقادیر گسسته خروجی لایه مخفی تولید می‌شوند.

---

[1] Discretization
[2] Normalization
[3] Sampling
[4] Aggregation
[5] Dimensionality reduction
[6] Pruning
[7] Performance

۴.۱ چالش‌ها و راهکارها

قوانین استخراج شده به کمک شبکه‌های عصبی معمولا دقت پایین‌تری از خود شبکه دارند. دلیل این امر، متکی بودن این روش‌ها به هرس‌سازی شبکه و استخراج قانون از ساختار هرس شده شبکه می‌باشد. با هرس‌سازی، یال‌های کم اهمیت (وزن‌های نزدیک به صفر) حذف می‌شوند که موجب کاهش دقت و کارایی شبکه می‌شود. این عملیات بخش جداناپذیر و مهمی از روش‌های استخراج قانون به کمک شبکه‌عصبی است و با عث شده است که تاثیر به سزایی بر کارایی قوانین استخراج شده داشته باشد.

ساختار هرس شده شبکه‌عصبی، فقط شامل اتصالات موثر شبکه است. با این حال استخراج قانون، به دلیل وجود مقادیر پیوسته در خروجی لایه مخفی، به راحتی امکان پذیر نمی‌باشد. از این رو گسسته‌سازی این مقادیر یکی از اجزای مهم و جدا ناپذیر در استخراج قانون از شبکه‌عصبی با دیدگاه تجزیه‌ای می‌باشد. تشکیل دسته‌های صحیح برای استخراج قوانین مربوط به هر کلاس، سادگی قوانین استخراج شده و کارایی مناسب آن‌ها به طور مستقیم وابسته به عملیات گسسته‌سازی هستند.

FSRE[1] یک روش برای استخراج قانون از شبکه‌عصبی MLP است که توانایی استخراج قوانین گزاره‌ای if-then-else برای مسائل دسته‌بندی[2] را دارد [۵۳]. استخراج قانون در این روش با استفاده از دیدگاه تجزیه‌ای انجام می‌شود. شبکه‌عصبی با استفاده از داده‌های آموزشی در صورت نیاز ابتدا پیش‌پردازش شده و سپس برای آموزش شبکه‌عصبی استفاده می‌شوند. استخراج قانون در گام آخر در چهار مرحله و با استفاده از ساختار هرس شده شبکه و مقادیر گسسته شده در لایه مخفی انجام می‌شود. عملیات گسسته‌سازی خروجی لایه مخفی در روش FSRE با صرف وقت زیاد به صورت دستی و با استفاده از ترسیم خروجی هر یک از نرون لایه مخفی انجام می‌شود.

[1] Four Step Rule Extraction
[2] Classification

در این پایان نامه قصد داریم به بهبود این روش در دو فاز بپردازیم. در فاز اول با ارائه یک راهکار نوین و با استفاده از الگوریتم ژنتیک[1] خسارات ناشی از هرس‌سازی را به میزان قابل توجهی کاهش می‌دهیم. این عملیات در دو گام به صورت سراسری[2] و محلی[3] به اصلاح وزن‌های شبکه می‌پردازد، در این روش قصد داریم با غیرمتوازن[4] کردن وزن‌ها آن‌ها را در دو دسته بی‌اهمیت و بااهمیت قرار دهیم تا ساختار هرس شده بهینه‌ای برای استخراج قانون تولید شود. در فاز بعد به منظور بهبود کارایی، سرعت و سادگی قوانین استخراج شده، یک الگوریتم خوشه‌بندی ابتکاری برای گسسته‌سازی خروجی نرون‌های لایه مخفی ارائه خواهیم داد.

۵.۱ ساختار پایان نامه

در این فصل به بیان کاربرد و اهمیت استخراج قانون، شبکه‌عصبی و رابطه این دو با یکدیگر پرداختیم. در نهایت روش استخراج قانون FSRE و چالش‌های موجود در این روش بررسی و راهکارهای ارائه شده به صورت مختصر بیان شد. در فصل ۲ به معرفی انواع قوانین، برخی دسته‌بندی‌های موجود برای روش‌های استخراج قانون، معرفی روش استخراج قانون FSRE و برخی روش‌های دیگر موجود در این حوزه می‌پردازیم. در فصل ۳ روش پیشنهادی ارائه خواهد شد. در فصل ۴ به بررسی و تحلیل نتایج و تاثیر روش پیشنهادی در بهبود استخراج قانون می‌پردازیم. در نهایت در فصل ۵ نتیجه‌گیری پایان نامه و راهکارهایی برای بهبود این پژوهش در اینده بیان خواهد شد.

---

[1] Genetic algorithm
[2] Global
[3] Local
[4] Unbalance

فصل ۲

ادبیات پژوهش

۱.۲ مقدمه

در فصل ۱ کلیات استخراج قانون با استفاده از شبکه‌عصبی مورد بررسی قرار گرفت. همچنین چالش‌های موجود در روش FSRE بررسی شد. در این فصل به معرفی انواع قوانین قابل استخراج از شبکه‌عصبی خواهیم پرداخت. همچنین دسته‌بندی روش‌های استخراج قانون از دیدگاه‌های مختلف مورد بررسی قرار خواهد گرفت. در ادامه روش استخراج قانون FSRE به عنوان روش پایه این پژوهش و سایر روش‌های مهم موجود معرفی خواهند شد.

۲.۲ انواع قوانین قابل استخراج

قوانین استخراج شده می توانند در ماهیت خود به طور قابل توجهی متفاوت باشند، از این رو خوانایی آن‌ها بسته به روش استخراج استفاده شده متفاوت است. هر نوع از قوانین، به شیوه‌ای متفاوت مفهوم و همبستگی درونی پارامترهای ورودی را توصیف می‌کند. اغلب یک مصالحه [1] بین خوانایی و نرخ بالای طبقه‌بندی وجود دارد. در ادامه به طور خلاصه هر کدام از قوانین منطقی را شرح خواهیم داد [۹]:

- **قوانین گزاره‌ای If-Then / If-Then-Else:** این نوع قوانین که متداول‌ترین نوع قوانین هستند شامل یک جزء شرطی If و یک جزء طبقه‌بند Then هستند. یک قانون می‌تواند در صورت نیاز شامل یک بخش طبقه‌بند Else نیز باشد. جزء شرطی If در یک قانون گزاره‌ای، یک ترکیبی بولی [2] از شرطهای منطقی بر روی ویژگی‌های ورودی است. بخش شرطی می‌تواند شامل عبارات عطفی ، فصلی و نقیض باشد. نمونه‌ای از یک قانون گزاره‌ای در رابطه ۱.۲ آمده است. برای مقادیر ورودی پیوسته معمولا شرطها به صورت محدودیت در یک بازه تعیین می‌شوند. رابطه ۲.۲ مثالی از این نوع قانون است. نمونه‌هایی از این قوانین در [۱۶، ۲۵] استفاده شده‌اند.

$$\textbf{If } X = x \textbf{ and } Y = y \textbf{ then } Class = A \tag{۱.۲}$$

$$\textbf{If } X \in [c_1, c_2] \textbf{ and } Y > c_3 \textbf{ then } Class = A \textbf{ where } c_1, c_2, c_3 \in \mathbb{R} \tag{۲.۲}$$

[1] Trade off

[2] Boolean

در این رابطه‌ها X و Y متغیرهای ورودی، x، y و c_i مقادیر ممکن برای این متغیرها و A یک نمونه کلاس خروجی است.

- **قوانین M-of-N:** این نوع قوانین بسیار مشابه قوانین گزاره‌ای هستند. در این نوع قانون کلاس‌بندی با محقق شدن حداقل، دقیقا و یا حداکثر M شرط از N شرط موجود انجام می‌شود. معمولا این قوانین را می‌توان به سادگی به قوانین گزاره‌ای تبدیل کرد. برای مثال رابطه ۴.۲ نمونه قانون گزاره‌ای تبدیل شده از قانون M-of-N در رابطه ۳.۲ است. همانطور که مشاهده می‌شود در صورتی که M برابر با یک باشد می‌توان قانون را به شکل یک ترکیب فصلی[1] از تمام شروط موجود بیان کرد و در صورتی که M برابر با N باشد، این ترکیب عطفی[2] خواهد بود. [۴۵] نمونه‌ای از استخراج این قوانین است.

$$\textbf{If exactly } ۲ - of - X = a_۱,\ Y = a_۲,\ Z = a_۳ \textbf{ then } Class = A \qquad (۳.۲)$$

$$\textbf{If } ((X = a_۱ \textbf{ and } Y = a_۲) \textbf{ or } (X = a_۱ \textbf{ and } Z = a_۳) \textbf{ or } (Y = a_۲ \textbf{ and } Z = a_۳)) \textbf{ then } Class = A \qquad (۴.۲)$$

- **قوانین ضمنی [3]:** در حالی که قوانین فوق نشان دهنده‌ی مناطقی از ابرفضاها[4] هستند که محورهای مختصات را با استفاده از بازه‌هایی جدا می‌کنند، گاهی ممکن است برای تعریف یک زیرفضا[5] نیاز به

[1] Disjunction
[2] Conjunction
[3] Oblique
[4] Hyperspace
[5] Subspace

تعداد شرایط زیادی برای دستیابی به یک طبقه‌بند[1] با کیفیت مطلوب داشته باشیم. قوانین ضمنی قادر هستند ابرفضاها را به زیرفضاهای دلخواهی تقسیم کنند. با استفاده از این قوانین می‌توان زیرفضاهای مورد نیاز را با کیفیت مطلوب و با تعداد قوانین کمتری تولید کرد. نمونه‌ای از این قوانین در رابطه ۵.۲ آمده است. نمونه‌ای از استخراج این قوانین در [۵۰] بیان شده است.

$$\textbf{If } (a_1X_1 + a_2X_2 + a_3X_3 > a_4) \textbf{ and } (a_4X_1 + a_5X_3 > a_6) \textbf{ then } Class = A \quad (۵.۲)$$

- **قوانین چند جمله‌ای:** این نوع قوانین شباهت زیادی به قوانین ضمنی دارند. قوانین تولید شده از معادلات پیچیده‌تری برای تولید بازه‌های مورد نظر خود استفاده می‌کند. به طور مثال رابطه ۶.۲ یک قانون برای تولید یک زیرفضای بیضوی است. اشکال اصلی این قوانین دشوار بودن درک آن‌ها است.

$$\textbf{If } (a_1X_1^2 + a_2X_2^2 + a_3X_1X_2 + a_4X_1 + a_5X_2 + a_6) \leq a_7 \textbf{ then } Class = B \quad (۶.۲)$$

- **قوانین فازی** [2]**:** این نوع قوانین شباهت بسیار زیادی به قوانین گزاره‌ای دارند، با این تفاوت که از معیارهای توصیفی به جای معیارهای کمی و بولی استفاده می‌کنند. در قوانین فازی زبان گفتمان متغیرها یک مجموعه فازی است. برای مثال رابطه ۷.۲ یک قانون فازی را بیان می‌کند. در این رابطه $high$ ، $medium$ و low اعضای مجموعه فازی هستند. در مجموعه‌های فازی ارتباط هر عنصر با با مجموعه استفاده از یک درجه تعلق[3] مشخص می‌شود. استفاده از مفاهیم زبانی[4] سبب شده است که قوانین فازی به خوبی قوانین گزاره‌ای قابل درک باشند. [۳۶، ۲۹] نمونه‌هایی از تولید این قوانین

[1] Classifier
[2] Fuzzy
[3] Corresponding grade
[4] Linguistic concepts

هستند.

$$\textbf{If } X \ is \ high \textbf{ and } Y \ is \ medium \textbf{ then } Class = A \qquad (۷.۲)$$

لازم به ذکر است که دانش موجود در شبکه‌عصبی را می‌توان به شکل‌های مختلفی از جمله ماشین‌های حالت[1] و نمایش بصری[2] استخراج کرد. کیفیت[3] اصلی‌ترین معیار برای ارزیابی قوانین استخراج شده است. به طور خاص، کیفیت معادل میزان قابل درک بودن است که این معیار ممکن است بیشتر و یا کمتر از دقت باشد، اغلب افزایش یکی از این معیارها منجر به کاهش دیگری خواهد شد.

۳.۲ دسته‌بندی روش‌های استخراج قانون

روش‌های زیادی برای استخراج قانون از شبکه‌عصبی ارائه شده‌اند. هرکدام از این روش‌ها دارای خصوصیات متفاوتی هستند که می‌توان آن‌ها را با استفاده از این خصوصیات دسته‌بندی نمود. در این بخش پنج دسته‌بندی مختلف معرفی خواهند شد.

۱.۳.۲ بر اساس نوع شبکه

در این دسته‌بندی نوع شبکه استفاده شده برای استخراج قانون به عنوان پارامتر دسته‌بندی استفاده می‌شود. استخراج قانون از شبکه‌های عصبی مختلفی نظیر شبکه‌عصبی چندلایه پرسپترون (MLP) [۵۵]، شبکه‌های تابع پایه شعاعی (RBF[4]) [۵۸، ۳۷] و شبکه‌های نگاشت خود سازمان یافته (SOM[5]) [۳۵] صورت گرفته است.

[1] State machines
[2] Visual representation
[3] Quality
[4] Radial Basis Function
[5] Self-Organizing Map

هر کدام از این شبکه‌ها کاربرد خاص خود را دارند. شبکه‌های MLP که رایج‌ترین و پر کاربردترین نوع شبکه‌عصبی برای استخراج قانون می‌باشند، برای مسائل دسته‌بندی و خوشه‌بندی استفاده می‌شوند و از دقت بسیار خوبی بهره می‌برند. شبکه‌های RBF همانند شبکه‌های چند لایه عمدتا برای مسائل خوشه‌بندی مورد استفاده قرار می‌گیرند اما کارایی و دقت این شبکه‌ها در مقایسه با شبکه‌های MLP در حد پایین‌تری قرار دارد. شبکه‌های نگاشت خود سازمان یافته در نمایش همسایگی‌ها در مجموعه داده و یا شباهت/ تفاوت بین نمونه‌های مختلف کاربرد دارند و کاربرد چندانی در عملیان دسته بندی ندارد. به همین دلیل روش‌های استخراج قانون برای این شبکه کمتر می‌باشند. مزیت شبکه‌های SOM در نمایش داده‌هاست اما قوانین استخراج شده با این روش ساده بوده و تنها برای مسائل ساده کاربرد دارد.

۲.۳.۲ بر اساس کاربرد

همانطور که بیان شد استخراج قانون در مسائل گوناگونی کاربرد دارد. دسته‌بندی[۲۵] و تخمین توابع[1] [۴۸، ۱۲] نمونه‌هایی از این کاربردها هستند. در این دسته‌بندی نوع کاربرد شبکه به عنوان معیار برای دسته‌بندی روش‌های موجود استفاده می‌شود. در مسائل دسته‌بندی هدف تعیین عضویت یک نمونه ورودی به یکی از دسته‌های خروجی موجود است. این دسته‌ها توسط مقادیر گسسته مشخص شده و بدون روی هم افتادگی[2] هستند. این مسائل اغلب با استفاده از شبکه‌عصبی MLP انجام می‌شود، در این شبکه معمولا برای هر دسته موجود یک نرون در لایه خروجی در نظر گرفته می‌شود، قوانین برای هر نرون لایه خروجی به صورت جداگانه تولید می‌شوند و کلاس مربوط به هر دسته از قوانین تعیین می‌شود. در مسائل مربوط به تخمین توابع هدف تعیین مقدار (حقیقی) خروجی برای هر نمونه بر اساس پارامترهای ورودی آن نمونه است. در این مسائل لایه خروجی تنها شامل یک نرون خواهد بود که مقدار خروجی هر نمونه را تخمین میزند.

[1] Regression
[2] Overlap

۳.۳.۲ بر اساس دیدگاه استخراج قانون

همانطور که در ۳.۱ نیز اشاره شد، روش‌های موجود قوانین را بر اساس سه دیدگاه آموزشی [۴۱، ۴۲، ۱۲]، تجزیه‌ای[۲۳] و ترکیبی[۴۷] از شبکه‌عصبی استخراج می‌کنند. دیدگاه تجزیه‌ای پر کاربردترین و پربازده‌ترین روش موجود برای استخراج قانون از شبکه های عصبی می‌باشد. در این روش قوانین با تجزیه و تحلیل ساختار درونی شبکه، اتصالات موجود و خروجی نرون‌های لایه میانی استخراج می‌شوند. در این دیدگاه اتصالات بی اهمیت با استفاده از تکنیک‌های هرس‌سازی حذف و اتصالات و نرون‌های موثر شبکه شناسایی می‌شوند. ویژگی‌های موثر هر نرون لایه میانی با توجه به اتصالات باقی‌مانده مشخص می‌شود، سپس به ازای هر نمونه ورودی خروجی نرون‌های لایه میانی تعیین و نمونه‌های ورودی بر اساس این خروجی‌ها خوشه‌بندی و کلاس خروجی آن‌ها تعیین می‌شود. در نهایت، قوانین مربوط به هر خوشه با توجه به نمونه‌های ورودی و ویژگی‌های موثر هر خوشه استخراج می‌شوند.

دیدگاه آموزشی برخلاف دیدگاه تجزیه‌ای به ساختار درونی شبکه توجهی ندارد. در روش‌های مبتنی بر این دیدگاه به دلیل عدم توجه به اتصالات شبکه، معمولا عملیات هرس‌سازی انجام نمی‌شود. قوانین با استفاده از تحلیل خروجی‌های شبکه، به عنوان یک جعبه سیاه، به ازای هر نمونه ورودی استخراج می‌شوند. استخراج قوانین در دیدگاه ترکیبی با ادغام این دو دیدگاه صورت می‌گیرد. عموما دیدگاه ترکیبی سعی دارد تا با بهره‌گیری از برخی اطلاعات درونی شبکه به عنوان مکمل دیدگاه آموزشی، قوانین با کیفیت‌تری را تولید نماید.

۴.۳.۲ بر اساس نوع قوانین استخراج شده

قوانین می‌توانند به شکل‌های مختلفی از شبکه‌عصبی آموزش دیده استخراج شوند. در ۲.۲ برخی از شکل‌های قوانین قابل استخراج از شبکه بیان شده‌اند. همانطور که مشاهده می‌شود، هر کدام از این قوانین کاربرد خاص

خود را دارند. در این نوع دسته‌بندی، شکل قوانین استخراج شده به عنوان مبنای دسته‌بندی در نظر گرفته می‌شود. قوانین گزاره‌ای، M-of-N، و فازی از متداول‌ترین انواع این قوانین هستند.

۵.۳.۲ براساس نوع ویژگی‌های موجود در پایگاه داده

در این دسته‌بندی قابلیت اعمال روش‌های مختلف استخراج قانون برروی انواع مجموعه‌های داده بررسی می‌شود. مجموعه داده‌های موجود ممکن است شامل ویژگی‌های پیوسته، گسسته و یا کیفی باشند. ویژگی‌های کیفی برای استفاده نیاز به تبدیل به یکی از ویژگی‌های گسسته و یا پیوسته دارند. برخی از روش‌های استخراج قانون موجود از ویژگی‌های پیوسته[۲۵]، برخی دیگر از ویژگی‌های گسسته[۱۶] و گروهی دیگر از هر دو نوع داده پیوسته و گسسته[۵۳] پشتیبانی می‌کنند. عموما ویژگی‌های پیوسته قابل گسسته‌سازی هستند و در صورت نیاز می‌توان آن‌ها را به مقادیر گسسته تبدیل کرد، این عملیات با توجه به روش استخراج قانون و یا برای بهبود دقت قوانین انجام می‌شود. معمولا مسائل مربوط به تخمین توابع فقط از مقادیر پیوسته نظیر دما، قد و وزن که غیر قابل شمارش هستند، پشتیبانی می‌کنند. برخی از روش‌ها فقط از ویژگی‌های گسسته مانند جنسیت، گروه خونی و مذهب که قابل شمارش هستند پشتیبانی می‌کنند. استخراج قانون با استفاده از ویژگی‌های گسسته ساده‌تر و قابل درک‌تر است اما ممکن است که نسبت به دلیل عملیات گسسته‌سازی نسبت به روش‌های مبتنی بر ویژگی‌های پیوسته از دقت پایین‌تری برخوردار باشد.

۴.۲ کارهای مرتبط

در این بخش به بررسی برخی از روش‌های مهم ارائه شده تا کنون و بررسی و تحلیل هر روش از دیدگاه‌های مختلف دسته‌بندی بیان شده در بخش قبل می‌پردازیم. در ادامه به تشریح روش FSRE می‌پردازیم که در این

پایان نامه قصد داریم به بهبود آن بپردازیم.

۱.۴.۲ مروری بر کارهای انجام شده

Setiono در [۴۳] یک روش هرس‌سازی و استخراج قانون برای تشخیص سرطان سینه بر روی پایگاه‌داده Wisconsin Breast Cancer [۸] ارائه داده است. آموزش شبکه با استفاده از یک تابع خطای اصلاح شده برای تولید وزن‌های غیرمتوان انجام شده است. در این روش، برای دست‌یابی به وزن‌های مناسب، هر ویژگی ورودی با ارائه یک رابطه به ۱۰ مقدار دودویی گسسته‌سازی می‌شود. البته، این عمل باعث ده برابر شدن تعداد نرون‌های لایه ورودی می‌شود. هرس‌سازی شبکه با استفاده از یک تابع مجازات[1] انجام شده است. در پایان، خروجی تابع فعال‌ساز برای هر نرون لایه میانی گسسته‌سازی می‌شود. قوانین نهایی با ترکیب روابط بدست آمده از بررسی اتصالات باقی مانده برای هر نرون میانی، استخراج می‌شوند. روش‌های ارائه شده در این مقاله، با توسعه تکنیک‌های هرس سازی در [۴۴]، و تعمیم روش استخراج قانون با عنوان روش RX در [۳۴] ارائه شده‌اند.

در روش RX پس از آموزش و هرس کردن، مقادیر بدست آمده در لایه مخفی توسط عملیات خوشه‌بندی به مقادیر گسسته تبدیل می‌گردند. سپس رابطه بین مقادیر گسسته بدست آمده با خروجی‌های شبکه تعیین می‌شود. آنگاه برای هر نرون لایه مخفی تعداد اتصالات ورودی به هر نرون پس از هرس‌سازی تعیین می‌شود. در صورتیکه این تعداد کمتر از حد آستانه باشد، قوانین برای دستیابی به مقادیر گسسته شده این نرون بر اساس ورودی‌ها بدست می‌آیند. در غیر این صورت نرون لایه مخفی و ورودی‌های متصل به آن به عنوان یک شبکه جداگانه در نظر گرفته می‌شوند:

- هر مقدار گسسته بدست آمده از نرون لایه مخفی به عنوان یک خروجی شبکه عصبی جدید در نظر گرفته

[1] Penalty

می‌شود.

- هر ورودی متصل به نرون لایه مخفی به عنوان ورودی شبکه جدید در نظر گرفته می‌شود.
- لایه مخفی جدیدی با تعداد نرون مناسب ایجاد می‌شود.
- این شبکه توسط داده‌های موجود آموزش دیده و مراحل استخراج قانون برای آن تکرار می‌شود.

با ترکیب قوانین بدست آمده مجموعه قوانین نهایی بدست می‌آید.

Setiono و Leow در [۴۷] یک روش ترکیبی به نام FERNN برای استخراج سریع قانون ارائه دادند. در این روش برای جلوگیری از پردازش‌های اضافه از فاز هرس‌سازی چشم پوشی می‌شود، و قوانین به شکل یک درخت تصمیم با روش C4.5 استخراج می‌شوند. این کار با استفاده از معیار بهره اطلاعات [1] برای تعیین اولویت نرون‌های لایه مخفی برای دسته‌بندی نمونه‌ها صورت می‌پذیرد. نرون دارای بیشترین اولویت به عنوان ریشه قرار می‌گیرد و مقادیر خروجی آن به دو زیر بازه تقسیم می‌شود، این تقسیم‌بندی طوری انجام می‌شود که بتواند بیشترین کارایی را داشته باشد. این عملیات برای سایر نرون‌ها تکرار می‌شود تا در نهایت درخت تصمیم برای دسته‌بندی نمونه‌ها تکمیل گردد. این روش با استفاده از دیدگاه ترکیبی عمل می‌کند. استفاده از شبکه عصبی به عنوان یک جعبه‌سیاه برای استخراج قوانین، سبب شده است تا در فرآیند استخراج قانون نیازی به هرس‌سازی شبکه وجود نداشته باشد.

در [۲۳] روشی تحت عنوان REANN ارائه شده است. این روش به صورت تکراری مراحل استخراج قانون، خوشه‌بندی قانون، هرس کردن قانون را انجام می‌دهد تا همه نمونه‌ها پوشش داده شود. این روش برای مسائل ساده به خوبی پاسخگو بوده، اما در مقابل مجموعه داده‌های حجیم و مسائل پیچیده از پیچیدگی زمانی بالایی برخوردار است. این الگوریتم از ابتدای ایجاد شبکه عصبی اعمال شده و در انتها با تعیین قوانین مورد نیاز

[1] Information Gain

برای دسته‌بندی نمونه‌های هر کلاس خاتمه می‌یابد. پس از تعیین تعداد نرون‌های ورودی و خروجی، آموزش با یک نررون در لایه مخفی آغاز می‌گردد. 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 شبکه آموزش دیده هرس می‌شود. پس از هرس سازی، مقادیر فعال شده هر نرون لایه مخفی توسط عملیات خوشه‌بندی به مقادیر گسسته تبدیل می‌شود. استخراج قانون با استفاده از روش REx [۲۱] انجام می‌شود. در این روش پس از گسسته‌سازی خروجی لایه مخفی، رابطه بین مقادیر گسسته لایه مخفی و دسته‌های موجود در لایه خروجی تعیین می‌گردد. سپس رابطه بین نمونه‌های ورودی و مقادیر گسسته محاسبه می‌شود. در نهایت با ترکیب روابط بدست آمده قوانین نهایی حاصل می‌شود.

Hruschka و Ebecken در [۱۹] برای بهبود روش RX ، به تغییر الگوریتم خوشه‌بندی مقادیر تابع فعال‌ساز نرون‌های لایه میانی پرداخته‌اند. گسسته‌سازی در این روش با استفاده از الگوریتم ژنتیک صورت می‌گیرد. در نهایت قوانین توسط روش RX استخراج می‌شوند.

روش NeuroRule در [۵۱] توسط Setiono و Tanaka ارائه شده است. این روش از با استفاده از یک شبکه‌عصبی MLP با ساختار استاندارد سه لایه به استخراج قوانین برای مسائل دسته‌بندی با استفاده از دیدگاه تجزیه‌ای می‌پردازد. در این روش از یک تابع مجازات برای هرس کردن شبکه‌عصبی استفاده می‌شود. پس از گسسته‌سازی مقادیر خروجی نرون‌های لایه میانی، استخراج قانون انجام می‌شود. در این روش ابتدا روابط موجود بین لایه ورودی و مقادیر گسسته بدست می‌آید. در مرحله بعد روابط بین مقادیر گسسته و نرون‌های لایه خروجی محاسبه شده و در نهایت قوانین با ترکیب این روابط بدست می‌آید.

کارایی این روش بر روی مسئله LED ارزیابی شده است. در این مسئله هدف شناسایی رقم نشان داده شده بر اساس حالات مختلف بر روی یک دیود نوری هفت بخشی[1] است که شامل ۱۲۸ حالت مختلف است.

[1] Seven segment

نتایج بدست آمده قادر خواهد بود که این مسئله را با دقت ۱۰۰% حل نماید. پیش از این نیز در [۲۷] یک روش جهت استخراج قوانین فازی برای این مسئله ارائه شده بود که پیچیدگی بالاتری نسبت به این قوانین داشته است.

ElAlami در [۱۴] از روش تخریبی برای هرس کردن شبکه استفاده کرده است. اتصالات شبکه در روش تخریبی با یک مقدار آستانه مقایسه می‌شوند، وزن‌هایی که مقادیر کمتری نسبت به آستانه دارند، از شبکه حذف خواهند شد. هرس‌سازی برای هر کلاس خروجی به صورت جداگانه انجام می‌شود تا اتصالات موثر آن کلاس مشخص شوند. در نهایت قوانین هر کلاس، از شبکه هرس شده مربوط به همان کلاس، در دو مرحله استخراج می‌شوند. ابتدا توانایی کلاس‌بندی داده‌ها به کمک یک ویژگی بررسی می‌شود. اگر یک ویژگی توانایی تشخیص یک کلاس را با دقت مطلوب داشت، قوانین مربوط به آن کلاس استخراج می‌شود. در گام بعد توانایی دسته‌بندی هر کلاس با استفاده از ترکیب‌های مختلف ویژگی‌ها بررسی می‌شود تا قوانین برای سایر کلاس‌ها نیز استخراج گردد.

روش دیگری تحت عنوان LREX در [۳۷] مطرح شده است که قادر است قوانین ساده‌ای برای خوشه‌بندی داده‌ها با استفاده از شبکه‌های عصبی RBF استخراج کند. در این روش باید هر کدام از نرون‌های لایه مخفی تنها متعلق به یک کلاس باشند تا بتوان نتایج مناسبی بدست آورد. پس از آموزش شبکه‌عصبی، مقادیر مختصات مرکز خوشه‌ها (نرون‌های لایه مخفی) و وزن‌های لایه خروجی استخراج می‌شوند. وزن یال‌های ورودی برای هر خوشه، مرکز خوشه را تعیین می‌کند. برای هر خوشه یک بردار حاصل می‌شود که مراکز ویژگی‌های ورودی برای این خوشه را نشان می‌دهد. کلاس خروجی برای هر خوشه (نرون مخفی) با توجه به بزرگترین وزن یال متصل به آن در لایه خروج مشخص می‌گردد. نمونه‌های موثر برای هر نرون لایه مخفی مشخص می‌گردند. این نمونه‌های موثر در هنگام اعمال به شبکه عصبی باعث تولید مقدار خروجی بیشتری

برای نرون مخفی مورد نظر، نسبت به دیگر نرون‌های مخفی می‌شوند. مقدار بدست آمده از تابع شعاع گوسی، به عنوان انحراف معیار خوشه‌ها در نظر گرفته می‌شود. حد بالا و پایین ویژگی‌های هر خوشه با استفاده از مقادیر میانگین و انحراف معیار بدست می‌آید. در نهایت با ترکیب مقادیر حد بالا و پایین یک گزاره شرطی برای یک ویژگی در خوشه مورد نظر بدست می‌آید که با ترکیب این ویژگی‌ها قانون خوشه مورد نظر بدست خواهد آمد.

روشی برای استخراج قانون با کمک شبکه‌عصبی SOM مبتنی بر ماتریس فاصله یکپارچه [1] در [۳۵] ارائه شده است. این ماتریس شامل فاصله اقلیدسی وزن همه نرون‌های خروجی در شبکه عصبی SOM می‌باشد که به منظور تعیین محدوده خوشه‌ها مورد استفاده قرار می‌گیرد و حالتی مانند رابطه ۸.۲ دارد:

$$U = [U(۱),\ U(۱,۲),\ U(۲),\ U(۲,۳),\ U(۳),\ U(۳,۴),\ U(۴)] \tag{۸.۲}$$

در این رابطه $U(i,j)$ فاصله بین نرون‌های i و j و $U(k)$ میانگین وزن بین نرون‌های مجاور است.

در این روش ابتدا نرون‌های لایه خروجی (معمولا در فضای دو بعدی قرار دارند) دسته‌بندی می‌شوند. بنابراین اولین نرون خروجی انتخاب می‌شود. برای نرون انتخاب شده و نرون‌های مجاور با استفاده از رابطه ۹.۲ مقدار اختلاف مرز تولید می‌شود.

$$BDV = \frac{M_L - M_O}{R_O} \tag{۹.۲}$$

در این رابطه M_L میانگین بین واحد کاندید برای مرز و واحدهای انتخاب شده قبلی، M_O میانگین بین دیگر

[1] U-matrix

واحدهای مجاور باقیمانده و R_O محدوده دیگر واحدهای مجاور باقیمانده است.

نرونی که حاوی مقدار تولید شده بزرگتری باشد به عنوان نرون بعدی انتخاب می‌شود. 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. 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، به عنوان مرزهای جداسازی خوشه‌ها تعیین می‌گردند. تعداد مرزهای انتخاب شده به تعداد مورد نیاز برای جداسازی خوشه‌های موجود است. 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، در صورتیکه این مرزهای بدست آمده با مرزهای ماتریس U مطابقت داشت، ویژگی ورودی به عنوان ویژگی مناسب انتخاب می‌شود. در نهایت میانگین مقادیر مرز در ویژگی‌های انتخاب شده به عنوان گزاره‌های شرطی انتخاب می‌شوند.

روش دیگری تحت عنوان REFANN در [۲۴] برای تخمین توابع ارائه شده است. در این روش ابتدا مراحل آموزش و هرس‌سازی با استفاده از روش N2PFA [۴۶] انجام 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 به حداقل برسد. سپس جهت هرس کردن این شبکه عصبی، نرون‌های مخفی غیر ضروری و ورودی‌های نامرتبط به صورت آزمایش و خطا مشخص و حذف می‌گردند. این عملیات تا زمانی ادامه می‌یابد که میزان خطای حاصل از ساختار جدید بیشتر از حد آستانه نباشد. با اعمال داده‌های موجود به ساختار هرس شده، مقادیر خروجی هر نرون مخفی بدست آمده که به صورت یک منحنی می‌باشد. برای کشف معادله انجام شده در هر نرون مخفی، منحنی آنرا به سه یا پنج قسمت

تقسیم کرده و توسط توابع ریاضیاتی سه بخشی و پنج بخشی تابع مورد نظر برای منحنی تخمین زده می‌شود. در نهایت، با توجه به توابع تخمین زده شده، قوانین تولید می‌گردد.

روش ANN-DT [۴۱] یک نمونه از روش‌هایی است که عملیات استخراج قانون را با استفاده از دیدگاه آموزشی انجام می‌دهد. قوانین استخراج شده با استفاده از این الگوریتم به شکل درخت تصمیم هستند. این روش توانایی استخراج قانون بدون در نظر گرفتن ساختار شبکه و یا نوع ویژگی‌های ورودی را دارد. از این روش می‌توان در مواردی مانند رگرسیون که خروجی شبکه مقادیر پیوسته‌ای می‌باشد نیز استفاده کرد. خروجی این الگوریتم یک درخت تصمیم تک متغییری است که در هر گره تنها شرایط یک متغیر به همراه یک مقدار آستانه بررسی می‌شود. هدف این روش تعیین ویژگی‌های موثر و مقادیر آستانه برای ویژگی است.

مراحل استخراج قانون:

۱. شبکه تا رسیدن به سطح قابل قبولی آموزش می‌بیند.

۲. یک مجموعه داده برای استخراج قانون تعیین می‌شود. زیر مجموعه‌ای از نمونه‌ها (با در نظر گرفتن توزیع داده‌های اصلی) نمونه‌برداری می‌شوند که سبب کاهش پیچیدگی زمانی، کاهش تعداد قوانین و افزایش کارایی می‌گردد.

۳. انتخاب ویژگی‌های مناسب و تعیین مقادیر آستانه برای تولید قوانین از روی نمونه‌های انتخاب شده در این فاز صورت می‌گیرد. به صورت 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.

۴. در این فاز زمان توقف عملیات بازگشتی تعیین می‌گردد، که برای این کار از حالتی مانند صفر شدن انحراف استاندارد یا واریانس در دسته‌ها استفاده می‌شود.

این روش بازگشتی است و می‌تواند در شبکه های عصبی نوع چندلایه و تابع پایه شعاعی مورد استفاده قرار می‌گیرد.

KDRuleEx [۴۲] روش دیگری برای استخراج قانون مبتنی بر دیدگاه آموزشی است. در این روش عملیات انجام شده توسط شبکه عصبی در قالب یک جدول تصمیم نمایش داده می‌شود. این الگوریتم به لطف استفاده از تکنیک آموزشی پیچیدگی زمانی کمتری دارد. این الگوریتم به دلیل عدم استفاده از حالت بازگشتی، مدیریت حافظه بهتری در مقایسه با سایر روش‌های مبتنی بر دیدگاه آموزشی دارد. این روش در مسائل ساده و نه چندان پیچیده کاربرد دارد.

مراحل استخراج قانون:

۱. تبدیل ویژگی‌های پیوسته به گسسته.

۲. تعیین بهترین ویژگی جداکننده. تقسیم مقادیر این ویژگی به بخش‌هایی که بتوان جداسازی بین داده ها را ایجاد کرد.

۳. تقسیم‌بندی ویژگی‌ها تا زمانی ادامه می‌یابد که هزینه محاسبات و قوانین بدست آمده از میزان تعیین شده بیشتر نباشد.

۴. تصمیمات حاصل شده در جدول مورد نظر وارد شده و بهترین ویژگی بعدی انتخاب می‌شود.

۵. این عملیات تا زمانی که به میزان درستی تعیین شده دست یابد ادامه می‌یابد.

در روش دیگر با استفاده از دوگام زیر باعث ساده‌سازی استخراج قوانین به صورت M-of-N می‌شود. سپس با تعیین تابع فعالساز مناسب برای لایه مخفی شبکه عصبی آموزش می‌بینند. در نهایت با هرس کردن شبکه قوانین مورد نظر بدست می‌آیند:

- در گام اول مقادیر نمونه‌های موجود قبل از اعمال به شبکه به مقادیر گسسته ـ۱ و ۱ تبدیل می‌شوند که باعث محدود شدن مقادیر ورودی می‌گردد.

- در گام دوم از تابع تانژانت هایپربولیک برروی همه اتصالات بین لایه ورودی و مخفی استفاده می‌شود که موجب محدود شدن مقادیر ورودی به نرون‌های لایه مخفی می‌شود.

روش دیگری در [۲۹] ارائه شده است که قادر است قوانین فازی را برای مجموعه‌هایی شامل دو نوع ویژگی پیوسته و گسسته بدست آورد. در این روش مقادیر ویژگی‌های گسسته به مقادیر باینری به فرم ۱ از n تبدیل می‌شوند. مقادیر ویژگی‌های پیوسته با توجه به کمترین و بیشترین مقدار ویژگی و طبق تابع عضویت پنج متغییری به مقادیر فازی تبدیل می‌شود. سپس شبکه با استفاده از روش DE [۵۴] آموزش می‌بیند. DE یک روش آموزش بهینه شده بر اساس جمعیت است و با استفاده از داده‌های جدید بدست آمده آموزش می‌بیند. در پایان، از روش TACO [۳۳] جهت استخراج قوانین فازی استفاده می‌شود.

۲.۴.۲ معرفی روش FSRE

در [۵۳] روش دیگری تحت عنوان FSRE برای استخراج قانون ارائه شده است. در این روش پس از آموزش شبکه، وزن‌های شبکه در دو مرحله هرس می‌شوند. در گام اول اتصالات مابین لایه مخفی و خروجی هرس می‌شود که باعث شناسایی نرون‌های موثر هر کلاس خروجی می‌گردد. ممکن است یک نرون میانی به دلیل قطع شدن تمام اتصالاتش با خروجی، از شبکه حذف شود. اگر اتصالات یک نرون دارای مقادیر مختلفی باشند، اتصالات دارای وزن‌های بزرگ‌تر حفظ، و بقیه اتصالات حذف می‌شوند. در گام بعد اتصالات مابین لایه ورودی و مخفی هرس می‌شوند تا پارامترهای ضروری هر نرون لایه مخفی شناسایی و بقیه حذف می‌شوند. در این الگوریتم پس از حذف هر اتصال کارایی شبکه سنجیده می‌شود.

استخراج قانون در روش FSRE ، پس از گسسته‌سازی نرون‌های لایه میانی، در چهار مرحله انجام می‌شود. ابتدا با اعمال داده‌های آموزشی به ساختار هرس شده شبکه، الگویی برای هر نمونه ورودی تعیین می‌شود. این الگو با کنار هم قرار گرفتن مقادیر خروجی گسسته نرون‌های لایه میانی شکل می‌گیرد. در مرحله بعد، نمونه‌های دارای الگوی مشابه در یک دسته قرار می‌گیرند. یک کلاس خروجی برای هر دسته با توجه به خروجی اکثریت نمونه‌های آن، تعیین می‌شود. در مرحله سوم، با توجه به حذف شدن برخی ویژگی‌های ورودی در فاز هرس‌سازی، ممکن است نمونه‌های موجود در یک دسته مقادیر یکسانی داشته باشند. همچنین برخی نمونه‌ها ممکن است که دارای مقادیر مشابهی باشند. در این مرحله نمونه‌های تکراری حذف و نمونه‌های مشابه ترکیب می‌شوند. در نهایت دسته‌های دارای کلاس خروجی یکسان با هم ادغام و بار دیگر عملیات حذف و ترکیب نمونه‌ها بر روی دسته‌های متعلق به یک کلاس انجام می‌شود. در انتها برای اینکه قوانین بتوانند نمونه‌های دیده نشده را نیز کلاس‌بندی کنند، یکی از کلاس‌ها به عنوان پاسخ پیش‌فرض تعیین می‌گردد. دسته‌های دارای بالاترین مقدار خطا، بیشترین تعداد نمونه و یا بیشترین قوانین تولید شده، می‌توانند کاندید انتخاب به عنوان پاسخ پیش‌فرض باشند.

در فصل بعد مراحل استخراج قانون به کمک روش FSRE بررسی و رویکردی برای حل چالش‌های موجود ارائه خواهد شد.

۵.۲ نتیجه‌گیری

در این فصل با انواع قوانین قابل استخراج از شبکه‌های عصبی آشنا شدیم. در ادامه به بررسی دسته‌بندی‌های موجود برای روش‌های استخراج قانون ارائه شده پرداختیم. هر روش استخراج قانون دارای خصوصیاتی می‌باشد که سبب می‌شود در دسته‌های مختلفی قرار بگیرد. در ادامه برخی از مهم‌ترین روش‌هایی که تا کنون ارائه

شده‌اند را مرور کردیم. روش FSRE یکی از جدیدترین و کاراترین روش‌های موجود برای استخراج قانون است که توانایی حل مسائل پیچیده را دارد. در فصل بعد به تشریح این روش خواهیم پرداخت.

فصل ۳

روش پیشنهادی

۱.۳ مقدمه

در این فصل ابتدا به تشریح مراحل استخراج قانون توسط روش FSRE می‌پردازیم. سپس چالش‌های موجود و در این روش و راهکارهای پیشنهادی و اهمیت اعمال آن‌ها بیان خواهد شد. در انتها روش پیشنهادی ارائه خواهد شد.

۲.۳ مراحل استخراج قانون به روش FSRE

FSRE یک روش برای استخراج قانون از شبکه‌عصبی چند لایه پرسپترون است که توانایی استخراج قوانین گزاره‌ای if-then-else برای مسائل دسته‌بندی را دارد. استخراج قانون در این روش با استفاده از دیدگاه تجزیه‌ای انجام می‌شود. داده‌های آموزشی در صورت نیاز ابتدا پیش‌پردازش شده و سپس برای آموزش به شبکه‌عصبی ارسال می‌شوند. استخراج قانون در گام آخر در چهار مرحله و با استفاده از ساختار هرس شده شبکه انجام می‌شود. در ادامه به بررسی هر کدام از این مراحل خواهیم پرداخت.

۱.۲.۳ پیش‌پردازش داده‌ها

برای دستیابی به کارایی مناسب، اغلب نیاز است قبل از آموزش شبکه، پیش‌پردازش‌هایی بر روی داده‌های خام با هدف تبدیل مجموعه داده‌های خام به داده‌های کارآمد و بهینه، اعمال شود. نرمال‌سازی، گسسته‌سازی، کاهش بعد و نمونه‌برداری از جمله پیش‌پردازش‌های رایجی هستند که می‌توانند این داده‌های خام را به داده‌های کارآمد در جهت افزایش کارایی شبکه تبدیل کنند. این پیش‌پردازش‌ها متناسب با مجموعه داده مورد نظر انتخاب می‌شوند. روش FSRE با هر دو نوع داده پیوسته و گسسته سازگاری دارد و محدودیتی برای استفاده از هر کدام از این داده‌ها وجود ندارد.

پراکندگی و اندازه‌های متفاوت طول بردار ویژگی در پایگاه‌داده، سبب می‌شود که یک ویژگی با مقادیر بزرگ نسبت به سایر ویژگی‌ها، تاثیر بیشتری در آموزش شبکه داشته باشد. برای رفع این مشکل، مقادیر هر ویژگی

ورودی با استفاده از رابطه ۱.۳ به کمک میانگین و انحراف معیار نرمال‌سازی می‌شوند.

$$x_i^{'} = \frac{x_i - \mu}{\sigma} \tag{۱.۳}$$

در این رابطه x_i مقدار اولیه یک ویژگی برای نمونه i اُم، μ میانگین مقادیر آن ویژگی برای تمام نمونه‌ها، σ انحراف معیار استاندارد ویژگی مورد بررسی و $x_i^{'}$ مقدار نرمال شده برای نمونه i اُم می‌باشد.

از روش بهره اطلاعاتی[1] برای ارزیابی ویژگی‌ها از نظر میزان کارایی و اهمیت استفاده می‌شود. این روش، یک خصوصیت آماری مبتنی بر بی‌نظمی[2]می‌باشد که عبارت است از میزان کاهش بی‌نظمی که به واسطه جداسازی نمونه‌ها از طریق یک ویژگی حاصل می‌شود و برای هر ویژگی به صورت مجزا بدست می‌آید.

درصورت نیاز می‌توان در مرحله پیش‌پردازش مقادیر پیوسته را گسسته‌سازی کرد. روش‌های CAIM [۳۱]، Fast-CAIM [۳۰]، Modified CAIM [۵۷] و ur-CAIM [۱۰] رایج‌ترین روش‌های موجود برای گسسته‌سازی هستند.

پس از انجام عملیات پیش‌پردازش باید با استفاده از یک روش مناسب داده‌ها کدگذاری شوند، تا بهترین نتیجه حاصل شود. روش‌های باینری موجب افزایش ورودی‌های شبکه (بخصوص ۱ از N) و ساده‌تر شدن مسئله می‌شوند. در حالت‌های نرمال‌شده فضای تغییرات هر ویژگی در محدوده کوچک‌تری قرار می‌گیرد که از نوسان در آموزش جلوگیری می‌کند. همچنین حالت فازی برای مسائل فازی و تولید قوانین فازی استفاده می‌شود. در انتهای عملیات انجام شده در این فاز، داده‌ها به سه دسته مجزای آموزش، ارزیابی و تست تقسیم می‌شوند.

[1] Information gain
[2] Entropy

۲.۲.۳ تعیین ساختار مناسب و آموزش شبکه

آموزش شبکه با استفاده از یک شبکه استاندارد سه لایه MLP انجام می‌شود. تعداد نرون‌های لایه ورودی متناسب با تعداد ویژگی‌های ورودی پایگاه‌داده تعیین می‌شود. تعداد نرون‌های لایه خروجی نیز متناسب با تعداد کلاس‌های موجود انتخاب می‌شوند. تعداد نرون‌های لایه میانی در ابتدای آموزش یک در نظر گرفته می‌شود. در صورت عدم دستیابی به کارایی مناسب یک واحد به تعداد آن‌ها افزوده می‌شود، این کار تا زمان دست‌یابی شبکه به کارایی مطلوب ادامه می‌یابد. آموزش با داده‌های آموزشی و کارایی با مجموعه تست محاسبه می‌گردد. تابع MSE[1] برای محاسبه خطا استفاده می‌شود. جهت اندازه‌گیری میزان کارایی از پارامترهای دقت، حساسیت [2] و اختصاصی بودن [3] استفاده می‌شود. حساسیت و اختصاصی بودن تنها برای حالت دو کلاسه استفاده می‌شود. توابع فعال‌ساز نرون‌های لایه میانی و خروجی و همچنین وجود بایاس در هر لایه متناسب با شبکه انتخاب می‌شود. در پایان آموزش شبکه با استفاده از روش گرادیان نزولی[4] و الگوریتم یادگیری پس انتشار خطا[5] صورت می‌گیرد.

۳.۲.۳ هرس‌سازی شبکه

وجود اتصالات زیاد در شبکه باعث می‌شود که استخراج قانون بسیار پیچیده یا غیرممکن شود. به همین منظور برخی اتصالات و گره‌های کم اهمیت در این مرحله حذف می‌شوند. هرس‌سازی باعث ساده‌تر شدن ساختار شبکه، شناسایی گره‌ها و اتصالات موثر در شبکه و در نهایت تعیین ویژگی‌های پراهمیت برای هر کلاس می‌شود. این عملیات علیرغم کاهش ابهام در شبکه، باعث کاهش کارایی شبکه نیز می‌شود که سبب می‌شود قوانین

[1] Mean Squared Error
[2] Sensitivity
[3] Specificity
[4] Gradient descent
[5] Backpropagation

استخراج شده از این ساختار دقت کمتری نسبت به شبکه اصلی داشته باشد. در نتیجه هرس‌سازی باید به گونه‌ای صورت بگیرد که علاوه بر کاهش پیچیدگی، دقت شبکه تا حد مطلوبی حفظ شود.

هرس‌سازی در روش FSRE در دو مرحله صورت می‌پذیرد:

- در گام اول اتصالات مابین لایه مخفی و خروجی هرس شده که باعث شناسایی نرون‌های موثر هر کلاس خروجی می‌شود. در این الگوریتم ممکن است تمام اتصالات یک نرون با خروجی حذف گردند. در این حالت این نرون و همه اتصالات آن از شبکه حذف می‌شود. اگر اتصالات یک نرون دارای مقادیر مختلفی باشند، اتصالاتی که دارای بزرگترین وزن هستند حفظ و باقی حذف می‌شوند.
- در گام بعد اتصالات مابین لایه ورودی و مخفی هرس می‌شوند تا پارامترهای ورودی موثرتر شناسایی شوند. پارامترهای ضروری هر نرون لایه مخفی شناسایی و باقی حذف می‌شوند. در صورتیکه تمام اتصالات یک ورودی حذف گردند، این ویژگی ضروری نبوده و حذف می‌شود. در این الگوریتم یک اتصال حذف می‌شود و سپس کارایی شبکه سنجیده می‌شود.

۴.۲.۳ استخراج قانون

عملیات نهایی استخراج قانون با استفاده از ساختار هرس شده شبکه انجام می‌شود. قوانین گزاره‌ای در چهار مرحله برای مسائل دسته‌بندی استخراج می‌شوند. در این روش از الگوی نمونه‌های ورودی در لایه مخفی جهت تولید قوانین استفاده می‌گردد، تشکیل این الگوها با توجه به خروجی تابع فعال‌ساز نرون‌های لایه میانی انجام می‌پذیرد.

وجود مقادیر پیوسته در خروجی لایه مخفی موجب تشکیل الگوهای بسیار زیادی می‌شود. از این رو استخراج قانون با استفاده از این الگوها بسیار پیچیده و حتی غیرممکن خواهد بود. به همین منظور نیاز است

تا خروجی تابع فعال‌ساز برای هر نرون لایه میانی گسسته‌سازی شود. روش‌های گسسته‌سازی مختلفی برای اینکار ارائه شده است. در این روش، با توجه به توزیع مقادیر خروجی هر نرون به صورت مجزا، گسسته‌سازی با تعیین مقادیر آستانه و به صورت دستی انجام می‌شود.

پس از اتمام فاز گسسته‌سازی، عملیات استخراج قانون با استفاده از مقادیر گسسته در لایه میانی انجام می‌شود. این عملیات در چهار مرحله صورت صورت می‌گیرد:

۱. ابتدا تمام نمونه‌ها به شبکه هرس شده اعمال می‌شوند تا الگوی هر نمونه تعیین شود. الگوی یک نمونه با کنار هم قرار دادن مقادیر گسسته شده خروجی تابع فعال‌ساز نرون‌های لایه میانی، تشکیل می‌شود.

۲. برخی نمونه‌های ورودی مقادیر گسسته مشابهی در لایه مخفی تولید می‌کنند، که باعث تولید شدن الگوهای یکسانی برای برخی نمونه‌ها می‌شود. هر الگوی بدست آمده متعلق به یک کلاس خروجی مشخص است که با توجه به خروجی شبکه تعیین می‌گردد. در این گام برای هر الگوی بدست آمده یک دسته جدا در نظر گرفته می‌شود. در هر دسته نمونه‌های ورودی مشابه قرار می‌گیرند.

۳. با توجه به حذف شدن برخی ویژگی‌های ورودی در فاز هرس‌سازی، ممکن است نمونه‌های موجود در یک دسته مقادیر یکسانی داشته باشند. همچنین ممکن است برخی نمونه‌ها مقادیر مشابهی داشته باشند. در این مرحله نمونه‌های تکراری حذف و نمونه‌های مشابه ترکیب می‌شوند. در نهایت دسته‌های دارای کلاس خروجی یکسان با هم ادغام می‌شوند. بار دیگر عملیات حذف نمونه‌های تکراری و ترکیب نمونه‌های مشابه بر روی دسته‌های متعلق به یک کلاس انجام می‌شود.

۴. در انتها برای اینکه قوانین بتوانند نمونه‌های دیده نشده را نیز کلاس‌بندی کنند، یکی کلاس‌ها به عنوان پاسخ پیش‌فرض تعیین می‌گردد. دسته‌های دارای بالاترین مقدار خطا، بیشترین تعداد نمونه و

یا بیشترین قوانین تولید شده می‌توانند کاندید انتخاب به عنوان پاسخ پیش‌فرض باشند.

۳.۳ چالش‌ها و راهکارها

وزن‌های بدست آمده با استفاده از روش پس انتشار خطا و گرادیان نزولی، تنها با هدف کاهش خطا و افزایش کارایی شبکه تولید می‌شوند. برخی از این وزن‌ها مقادیر بزرگی دارند که باعث می‌شود در تعیین خروجی تاثیر قابل توجهی داشته باشند، گروهی دیگر از اتصالات دارای مقادیر کوچک و نزدیک به صفر هستند، در نتیجه اغلب تاثیر چندانی در خروجی شبکه ندارند و در مرحله هرس‌سازی حذف می‌شوند. گروه سوم اتصالاتی هستند که تاثیر آن‌ها در کارایی شبکه کم است اما بی‌تاثیر نیستند. الگوریتم‌های هرس‌سازی در جهت حفظ اتصالات موثر و حذف اتصالات بی‌اثر عمل می‌کنند، اما چالش اصلی موجود مربوط به اتصالاتی است که تاثیر کمی در خروجی شبکه دارند. رویکرد الگوریتم‌های هرس‌سازی، در رابطه با این اتصالات، حذف آن‌ها تا حد امکان، به منظور استخراج قوانین ساده‌تر می‌باشد.

قوانین استخراج شده به کمک شبکه‌های عصبی معمولا دقت پایین‌تری از خود شبکه دارند. دلیل این امر، متکی بودن این روش‌ها به هرس‌سازی شبکه و استخراج قانون از ساختار هرس شده شبکه می‌باشد. با هرس‌سازی، یال‌های کم اهمیت (وزن‌های نزدیک به صفر) حذف می‌شوند که موجب کاهش دقت و کارایی شبکه می‌شود. از آنجایی که این عملیات بخش جداناپذیر و مهمی از روش‌های استخراج قانون به کمک شبکه‌عصبی است، با ارائه یک راهکار نوین در این تحقیق خسارات ناشی از آن را به میزان قابل توجهی کاهش می‌دهیم.

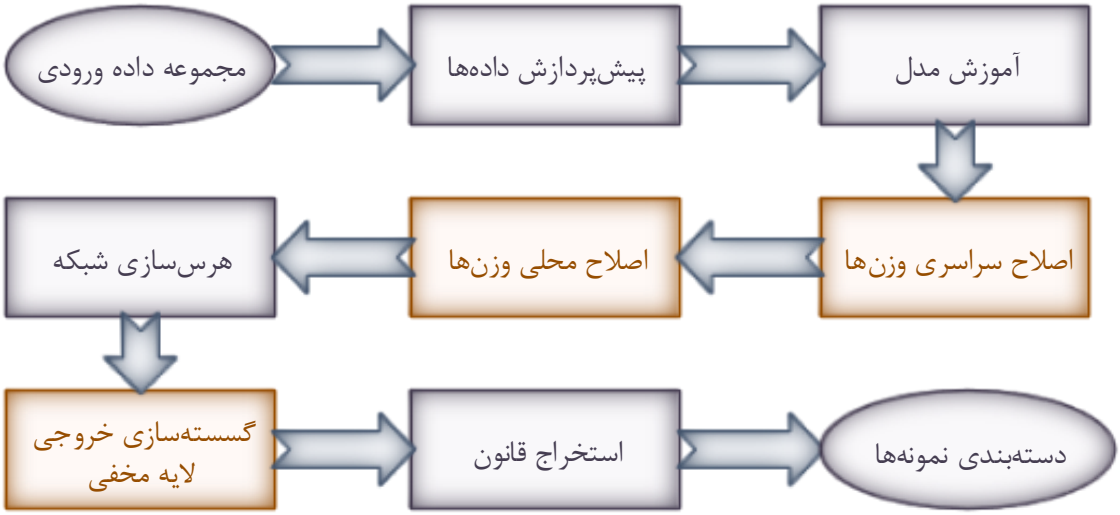
وزن‌های یک شبکه عصبی آموزش دیده را می‌توان به سه دستهٔ کم اهمیت، اهمیت متوسط، و پراهمیت تقسیم بندی کرد. وزنهای کم اهمیت، به صفر نزدیک هستد و با حذف آنها (هرس یالهای مربوطه) تاثیر

چندانی بر عملکرد شبه عصبی نخواهد گذاشت. در مرحله هرس، یالهای با اهمیت حفظ می‌شوند. چنانچه با هرس یالهای کم اهمیت نتوان مجموعه قوانین مناسبی را استخراج نمود، یالهای بااهمیت متوسط نیز حذف خواهند شد. البته حذف یالهای بااهمیت متوسط به میزان قابل توجهی بر دقت شبکه و در نتیجه دقت مجموعه قوانین استخراج شده تاثیر خواهد داشت. در روش پیشنهادی، با استفاده از الگوریتم ژنتیک، آموزش به گونه‌ای انجام می‌شود که وزن‌های تولید شده غیرمتوازن باشند. به عبارت دیگر، شبکه فاقد یال‌های با اهمیت متوسط باشد. براین اساس، با حفظ وزن‌های پر اهمیت و حذف وزن‌های کم اهمیت خسارت ناشی از مرحله هرس‌سازی تقلیل می‌یابد. این امر باعث می‌شود که شبکه عصبی پس از هرس سازی، و در نتیجه مجموعه قوانین استخراج شده از آن با کاهش دقت همراه باشد. روش پیشنهادی در این تحقیق، با هدف تولید هدفمند وزن‌های غیر متوازن با استفاده از الگوریتم ژنتیک، کاهش دقت شبکه عصبی هرس شده را کنترل نماید. این کار به گونه‌ای در مرحله آموزش شبکه عصبی انجام می‌شود که ضمن حفظ دقت شبکه، مجموعه وزنهای آن به دو گروه موثر و کم تاثیر (نزدیک به صفر) قابل دسته بندی باشند.

ساختار هرس شده شبکه‌عصبی، فقط شامل اتصالات موثر شبکه است. با این حال استخراج قانون، به دلیل وجود مقادیر پیوسته در خروجی لایه میانی، به راحتی امکان پذیر نمی‌باشد. از این رو گسسته‌سازی این مقادیر یکی از اجزای مهم و جدا ناپذیر در استخراج قانون از شبکه‌عصبی با دیدگاه تجزیه‌ای می‌باشد. تشکیل دسته‌های صحیح برای استخراج قوانین مربوط به هر کلاس، سادگی قوانین استخراج شده و کارایی مناسب آن‌ها به طور مستقیم وابسته به عملیات گسسته‌سازی هستند.

عملیات گسسته‌سازی خروجی لایه میانی در روش FSRE با صرف وقت زیاد به صورت دستی و با استفاده از ترسیم خروجی هر یک از نرون لایه میانی انجام می‌شود. از این رو گسسته‌سازی نیز علاوه بر هرس‌سازی یکی از چالش‌هایی است که قصد داریم در این پژوهش به آن بپردازیم. برای رفع این مشکل قصد داریم با استفاده از

الگوریتم خوشه‌بندی ارائه شده در [۴۳]، دقت و کارایی قوانین استخراج شده با استفاده از این روش را بهبود دهیم. فرآیند استخراج قانون با استفاده از روش پیشنهادی در شکل ۱.۳ نشان داده شده است.



شکل ۱.۳: فرآیند استخراج قانون با استفاده از روش پیشنهادی

۴.۳ معرفی الگوریتم ژنتیک

غیرخطی بودن توابع هدف یا محدودیت ها، گسسته بودن فضای حل مسئله و اندازه مسئله از جمله عوامل تاثیرگذار در پیچیدگی محاسباتی مدل های تصمیم گیری هستند. روش های دقیق در این شرایط قادر به پیدا کردن جواب بهینه در زمان قابل قبولی نیستند. هدف روش های فراابتکاری، پیدا کردن بهترین جواب ممکن (نزدیک به بهینه) در زمان قابل قبول است.

در سی سال گذشته، نوع جدیدی از الگوریتم های تقریب ظهور یافته‌اند که اساساً هدف از آنها ترکیب روش‌های ابتکاری در چارچوبهای کلان تر به منظور کاوش کارا و اثربخش فضای جستجو می‌باشد. امروزه از این روش‌ها با عنوان روشهای فرا ابتکاری[1] نام برده می‌شود. یک روش فرا ابتکاری، در حقیقت، یک روش

[1] Meta-heuristic

ابتکاری برای حل یک طبقه بسیار عمومی از مسائل است و ترکیبی کارآمد از توابع هدف یا روش‌های ابتکاری مبتنی بر توابع هدف می باشد.

الگوریتم ژنتیک، الهامی از علم ژنتیک و نظریه تکامل داروین است و بر اساس بقای برترین‌ها یا انتخاب طبیعی استوار است. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن بعنوان تابع بهینه کننده است. الگوریتم ژنتیک ابزار سودمندی در بازشناسی الگو، انتخاب ویژگی، درك تصویر و یادگیری ماشینی است. در الگوریتم‌های ژنتیکی، نحوه تکامل ژنتیکی موجودات زند ەشبیه‌سازی می‌شود. این الگوریتم‌ها با الهام از روند تکاملی طبیعت مسائل را حل می‌نمایند. یعنی مانند طبیعت یک جمعیت از موجودات را تشکیل می‌دهند و با اعمالی بر روی این مجموعه به یک مجموعه بهینه و یا موجود بهینه دست می‌یابند. با توجه به خصوصیات خاص خودشان به خوبی از عهده حل مسائلی که نیاز به بهینه‌سازی دارند بر می‌آیند.

الگوریتم ژنتیک نوع خاصی از الگوریتم‌های تکاملی است که از تکنیک‌های زیست‌شناسی مانند وراثت و جهش استفاده می‌کند. در این الگوریتم ابتدا یک جمعیت اولیه تصادفی از کروموزوم‌ها[1] تولید شده و با استفاده از تابع برازندگی[2] مناسب، ارزیابی می‌شوند. در هر مرحله تعدادی از کروموزوم‌های برتر به عنوان کروموزوم نخبه به نسل بعد منتقل می‌شوند. سایر کروموزوم‌های نسل بعد با استفاده از عملگرهای ژنتیک تولید می‌شوند. این عملیات تا زمانی ادامه می‌یابد که شرط خاتمه الگوریتم برقرار شود. در نهایت کروموزوم برتر به عنوان جواب مسئله به خروجی منتقل می‌شود.

۱.۴.۳ کروموزوم

برای اینکه بتوانیم یک مسئله را بوسیله الگوریتم‌های ژنتیک حل کنیم، بایستی آن را به فرم مخصوص مورد نیاز این الگوریتم‌ها تبدیل کنیم. در این روند بایستی راه حل مورد نیاز مسئله را به گونه‌ای تعریف کنیم که

[1] Chromosome
[2] Fitness Function

قابل نمایش بوسیله یک کروموزوم باشد. اینکه چه نوع بازنمائی[1] برای مسئله استفاده شود، به شخص طراح و فرم مسئله بستگی دارد. چند نمونه از بازنمائی‌هایی را که معمولاً استفاده می شوند:

- اعداد صحیح
- رشته‌های بیتی
- اعداد حقیقی در فرم نقطه شناور
- اعداد حقیقی به فرم رشته‌های بیتی
- یک مجموعه از اعداد حقیقی یا صحیح
- ماشین‌های حالت محدود
- هر فرم دیگری که بتوانیم عملگرهای ژنتیک را بر روی آنها تعریف کنیم.

۲.۴.۳ تابع برازندگی

برای اینکه بتوانیم موجودیت‌های[2] بهتر را درون جمعیت تشخیص بدهیم بایستی معیاری را تعریف کنیم که بر اساس آن موجودیت‌های بهتر را تشخیص دهیم. به این کار، یعنی تعیین میزان خوبی یک موجود، ارزیابی آن موجود می گویند. ارزیابی، اینگونه است که بر حسب اینکه موجود چقدر خوب است یک عدد به آن نسبت می دهیم، این عدد که برای موجودات بهتر بزرگتر (یا کوچکتر) است را شایستگی آن موجود می نامیم. به عنوان مثال در صورتی که به دنبال مینیمم یک تابع هستیم، مقدار شایستگی را می توانیم ورودیهایی که مقادیر تابع

[1] Representation
[2] Entity

برای آنها کمتر است در نظر بگیریم که ورودیهای بهتری هستند. بسته به نوع مساله ما می خواهیم شایستگی را بیشینه و یا کمینه کنیم.

۳.۴.۳ عملگرهای ژنتیك

برای پیدا کردن یك نقطه در فضای جستجو باید از عملگرهای ژنتیك استفاده کرد. در ادامه عملگرهای انتخاب[1]، ترکیب[2] و جهش[3] توضیح داده خواهند شد.

عملگر انتخاب

در مرحله انتخاب، یك جفت از کروموزوم‌ها برگزیده می‌شوند تا با هم ترکیب شوند. عملگر انتخاب رابط بین دو نسل است و بعضی از اعضای نسل کنونی را به نسل آینده منتقل می‌کند. بعد از انتخاب، عملگرهای ژنتیك روی دو عضو برگزیده اعمال می‌شوند. معیار در انتخاب اعضاء ارزش تطابق آنها می‌باشد اما روند انتخاب حالتی تصادفی دارد.

شاید انتخاب مستقیم و ترتیبی به این شکل که بهترین اعضا دو به دو انتخاب شوند در نگاه اول روش مناسبی به نظر برسد اما باید به نکته‌ای توجه داشت. در الگوریتم ژنتیك ما با ژنها روبرو هستیم. یك عضو با تطابق پایین اگرچه در نسل خودش عضو مناسبی نمی‌باشد اما ممکن است شامل ژن‌هایی خوب باشد و اگر شانس انتخاب شدنش ۰ باشد، این ژن‌های خوب نمی‌توانند به نسل‌های بعد منتقل شوند. پس روش انخاب باید به گونه‌ای باشد که به این عضو نیز شانس انتخاب شدن بدهد. راه حل مناسب، طراحی روش انتخاب به گونه‌ای است که احتمال انتخاب شدن اعضای با تطابق بالاتر بیشتر باشد. انتخاب باید به گونه‌ای صورت بگیرد که تا جایی که ممکن است هر نسل جدید نسبت به نسل قبلی‌اش تطابق میانگین بهتری داشته باشد.

[1] Selection
[2] CrossOver
[3] Mutation

روش‌های متداول انتخاب عبارتند از:

- انتخاب چرخ رولت[1]
- انتخاب ترتیبی[2]
- انتخاب بولتزمن[3]
- انتخاب حالت پایدار[4]
- نخبه سالاری[5]
- انتخاب رقابتی[6]

عملگر ترکیب

مهمترین عملگر در الگوریتم ژنتیك، عملگر ترکیب است. ترکیب، فرآیندی است که در آن نسل قدیمی کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها بوجود بیاید. جفت‌هایی که در قسمت انتخاب، به عنوان والد در نظر گرفته شدند، در این قسمت ژنهای‌شان را با هم مبادله می‌کنند و اعضایی جدید بوجود می‌آورند.

بر اساس مباحث تئوری ترکیب اعضا با تطابق بالا باعث بوجود آمدن اعضایی می‌شود که از تطابق میانگین، تطابق بیشتری دارند. ترکیب در الگوریتم ژنتیك باعث از بین رفتن پراکندگی یا تنوع ژنتیکی جمعیت می‌شود. زیرا اجازه می‌دهد ژن‌های خوب یکدیگر را بیابند. شکل‌های مختلفی از عملگر ترکیب وجود دارد که مهمترین‌های آنها عبارتند از ترکیب تك نقطه‌ای، ترکیب دو نقطه‌ای، ترکیب n نقطه‌ای، ترکیب یکنواخت و ترکیب حسابی.

---

[1] Roulette wheel Selection
[2] Rank Selection
[3] Boltzmann Selection
[4] Steady-State Selection
[5] Elitism
[6] Tournament Selection

ترکیب لازم نیست در هر نسل اتفاق بیفتد. در واقع ممکن است نسل‌هایی بدون عملگر ترکیب به نسل‌های جدید، تبدیل شوند. برای تعیین رخ دادن یا ندادن ترکیب از پارامتری به نام احتمال ترکیب P_c، استفاده می‌شود که این پارامتر مقدار بین ۰ و ۱ است. از آنجا که ترکیب نقشی اساسی در رشد مقدار میانگین تطابق جمعیت دارد، مقدار P_c بین ۰,۵ تا ۰,۸ و بیشتر بین ۰,۷ تا ۰,۸درنظر گرفته می شود.

عملگر جهش

بعد از اینکه یك عضو در جمعیت جدید بوجود آمد، هر ژن آن با احتمال جهش، جهش می‌یابد. در جهش ممکن است ژنی از مجموعه ژن‌های جمعیت حذف شود یا ژنی که تا به حال در جمعیت وجود نداشته است به آن اضافه شود. جهش یك ژن به معنای تغییر آن ژن است و وابسته به نوع کدگذاری، روش‌های متفاوت جهش استفاده می‌شود.

همانطور که گفته شد، هر عضو وابسته به احتمال جهش، جهش می‌یابد. احتمال جهش، ، P_m مقداری است که توسط کاربر تعیین می‌شود. در الگوریتم استاندارد ژنتیك، مقدار این پارامتر، بسیار کوچك، مثل $P_m = ۰,۰۱$ یا حتی $P_m = ۰,۰۰۱$ در نظر گرفته می‌شود.

P_m مقداری بین ۰ و ۱ است و معمولا عددی کوچك انتخاب می‌شود. در فرزندی که بوجود آمده است (توسط ترکیب)، به ترتیب مقداری تصادفی بین ۰ و ۱ به هر ژن اختصاص می‌یابد. اگر این مقدار اختصاص داده شده از P_m کمتر باشد، ژن جهش می‌یابد و اگر بیشتر باشد، ژن تغییر نمی‌کند. نرخ بالای جهش، باعث تنوع و پراکندگی ژنتیکی در جمعیت می‌شود که این پراکندگی ممکن است همگرایی را به تاخیر بیندازد. به همین دلیل، پیشنهاد می‌شود برای جمعیت‌های بزرگ یا در نسل‌های آخر، از P_m های کوچکتر و برای جمعیت‌های کوچك یا در نسل‌های ابتدایی از P_m های بزرگتر استفاده شود. وارونه‌سازی بیت، تغییر ترتیب قرارگیری و تغییر مقدار نمونه‌هایی از انواع جهش هستند.

۵.۳ روش پیشنهادی

همانطور که بیان شد چالش‌های اساسی مورد بررسی در روش FSRE در مرحله هرس‌سازی و گسسته‌سازی می‌باشد. در این بخش قصد داریم به معرفی راهکارهای ارائه شده برای بهبود این الگوریتم در این دو بخش بپردازیم. همانطور که در بخش ۳.۳ اشاره شد چالش اصلی، حذف برخی وزن‌های میانی است که باعث می‌شود برخی از دانش موجود در شبکه از دست برود. برای رفع این مشکل رویکردی مبتنی بر الگوریتم ژنتیک ارائه شده است که قادر است وزن‌های شبکه‌عصبی آموزش دیده را در دو مرحله به صورت سراسری و محلی به شکلی اصلاح کند که وزن‌های حاصل به صورت غیرمتوازن باشند. به صورت ساده‌تر قصد داریم وزن‌ها فقط در دو دسته موثر و بی‌اثر قرار بگیرند و دسته میانی تا جای ممکن حذف شود. این امر باعث می‌شود هرس‌سازی به شکل بهینه و کاراتر انجام شود. الگوریتم سراسری، وزن‌های شبکه را با استفاده از الگوریتم ژنتیک به هنگام آموزش شبکه اصلاح می‌کند. الگوریتم محلی نیز به اصلاح وزن‌های شبکه آموزش دیده برای هر نرون لایه میانی به صورت جداگانه می‌پردازد. این الگوریتم‌ها در ادامه تشریح خواهند شد.

اهمیت فاز گسسته‌سازی در استخراج قانون برای استخراج قوانین دقیق تر و کاراتر بیان شد. در گام بعد سعی داریم که با ارائه یک الگوریتم خوشه‌بندی برای گسسته‌سازی مقادیر خروجی تابع فعال‌ساز نرون های لایه میانی به اهداف زیر دست یابیم.

- این الگوریتم عملیات گسسته‌سازی را به صورت پویا و خودکار انجام دهد.

- گسسته‌سازی به شکل کاملا بهینه و با حداقل خطای ممکن انجام شود.

۱.۵.۳ غیرمتوازن کردن وزن‌ها به صورت سراسری

همانطور که بیان شد، هدف روش پیشنهادی تولید وزن های شبکه در دو دسته موثر و کم تاثیر می‌باشد. بدین منظور آموزش شبکه با استفاده از الگوریتم ژنتیک انجام می‌شود. حل مسئله با استفاده از الگوریتم ژنتیک، نیازمند تعریف کروموزوم و تابع برازندگی مناسب است. با توجه به اینکه جواب مسئله ما در الگوریتم ژنتیک وزن‌های شبکه هستند، در نتیجه ساختار کروموزوم را این وزن‌ها تشکیل خواهد داد. بدین صورت که وزن‌های لایه ورودی به مخفی و لایه مخفی به خروجی را با ترتیب مشخصی به صورت متوالی کنار هم قرار می‌دهیم. هر وزن شبکه‌عصبی به عنوان یک ژن در کروموزوم قرار خواهد گرفت.

چالش اصلی این بخش تعریف یک تابع برازندگی مناسب جهت امتیازدهی به این کروموزوم‌ها می‌باشد. تابع برازندگی باید در عین حال که کارایی شبکه را حفظ می‌کند، به انتخاب کروموزوم‌هایی بپردازد که شامل وزن‌های غیرمتوازن بهتری باشد. در نتیجه تابع برازندگی مورد نیاز در این مسئله از دو بخش تشکیل می‌شود.

- قسمت اول مربوط به کارایی شبکه می‌شود، و می‌توان از معیارهای مجموع مربعات خطا (SSE[1])، میانگین مربعات خطا (MSE)، یا دقت (Accuracy) شبکه استفاده کرد.
- در قسمت دوم باید یک معیار جهت امتیازدهی به میزان دو دسته شدن وزن‌ها در هر کروموزوم تعریف کنیم. رابطه ۲.۳ به صورت تجربی برای امتیازدهی به غیر متوازن بودن وزن‌ها تعریف شده است.

$$score = \frac{۱}{\sqrt{var(|w|)} * R} \qquad (۲.۳)$$

در این رابطه، $score$ امتیاز مربوط به یک کروموزوم است که با توجه به میزان دو دسته شدن وزن‌های

[1] Sum of Squared Error

آن تعیین می‌شود. w بردار وزن‌های موجود در یک کروموزوم، R نسبت تعداد وزن‌های کم‌تاثیر یک کروموزوم به کل وزن‌ها، var واریانس بردار ورودی و $|.|$ بیانگر قدرمطلق بردار وزن‌ها است.

استفاده از واریانس در این رابطه به منظور ایجاد پراکندگی در وزن‌های شبکه می‌باشد. استفاده از این معیار به تنهایی ممکن است تعداد وزن‌های موثر شبکه را به میزان قابل توجهی افزایش دهد، در این صورت این وزن‌ها جهت هرس‌سازی شبکه مناسب نخواهند بود. پارامتر R برای جلوگیری از انتخاب این دسته از کروموزوم‌ها تعریف شده است. با انتخاب یک مقدار آستانه مناسب می‌توان تخمینی از تعداد وزن‌های غیر موثر در هر کروموزوم را بدست آورد. پارامتر R حاصل تقسیم این عدد بر تعداد کل اتصالات است. بنابراین مقدار این پارامتر همواره عددی بین ۰ و ۱ خواهد بود. در صورتیکه تعداد وزن‌های بی‌اثر کم باشد، R عددی نزدیک به صفر خواهد بود که باعث می‌شود مقدار $score$ عددی بزرگ باشد. با توجه به اینکه در نهایت الگوریتم ژنتیک سعی در کمینه‌سازی $score$ را دارد، با این کار احتمال انتخاب این کروموزوم کاهش خواهد یافت.

با ادغام این دو معیار می‌توانیم وزن‌های مورد نظر خود را تولید کنیم. با ترکیب معیار $score$ با یک معیار ارزیابی شبکه، که در اینجا از SSE استفاده کرده‌ایم، تابع برازندگی نهایی در رابطه ۳.۳ بیان شده است که نتایج قابل قبولی را ارائه می‌دهد.

$$fitness\_value = \frac{SSE + score}{۲} \tag{۳.۳}$$

در این رابطه، $score$ همانطور که بیان شد امتیاز غیرمتوازن بودن وزن‌ها، SSE معیار ارزیابی شبکه و $fitness\_value$ مقدار تابع برازندگی می‌باشد.

عملیات آموزش شبکه با استفاده از الگوریتم ژنتیک در جهت اصلاح وزن‌ها انجام خواهد شد. الگوریتم ژنتیک سعی در کمینه‌سازی تابع برازندگی معرفی شده دارد. با کمینه کردن این تابع علاوه بر دست‌یابی به وزن‌های مناسب برای هرس‌سازی، کارایی شبکه را تا حد مطلوبی حفظ خواهد شد. برای اینکه کارایی شبکه پس از آموزش به روش گرادیان نزولی و الگوریتم پس انتشار خطا حفظ شود، وزن‌های تولید شده را به عنوان یک کروموزوم در جمعیت اولیه قرار می‌دهیم. این کروموزوم به عنوان یک کروموزوم نخبه باعث سرعت بخشیدن به الگوریتم و حفظ حداقل کارایی شبکه خواهد شد.

۲.۵.۳ غیر متوازن کردن وزن‌ها به صورت محلی

در این الگوریتم بهبود وزن‌ها با استفاده از الگوریتم ژنتیک با بررسی وزن‌های ورودی به هر نرون به صورت جداگانه صورت می‌گیرد. در این روش ورودی هر نرون لایه خروجی و مخفی را به صورت جداگانه با استفاده از الگوریتم ژنتیک اصلاح می‌کنیم. این روش می‌تواند به دو صورت انجام پذیرد:

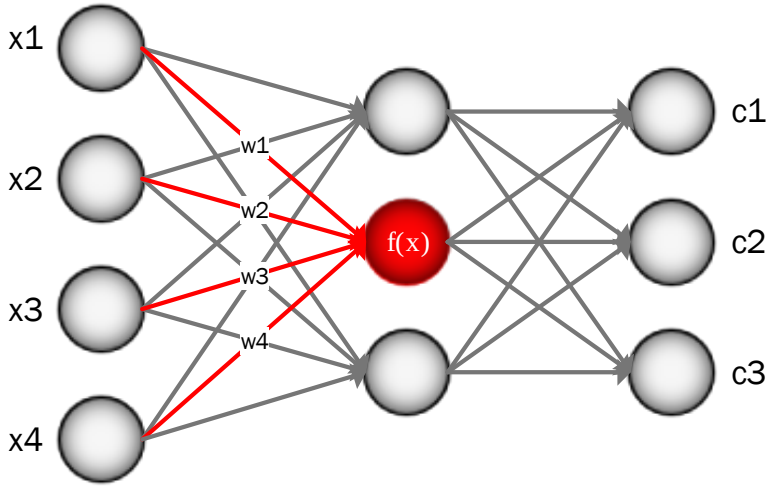
- در روش اول سعی می‌شود با اصلاح وزن‌های ورودی به شکل دلخواه، خروجی هر نرون به ازای نمونه‌های ورودی آموزش بدون تغییر باقی بماند.
- روش دوم به عنوان یک روش ساده‌تر، استفاده از یک تابع برازندگی برای بررسی عملکرد کلی شبکه به ازای تغییر ورودی‌های یک نرون است.

تابع برازندگی در روش اول با تشکیل یک چند جمله‌ای به ازای هر نرون لایه میانی تعریف می‌شود. برای نمونه رابطه ۴.۳ مطابق شکل ۲.۳ تابع برازندگی مربوط به یک نرون لایه میانی می‌باشد.

$$diffrence\_value = |f(x) - [(w_۱ * x_۱) + (w_۲ * x_۲) + (w_۳ * x_۳) + (w_۴ * x_۴)]| \qquad (۴.۳)$$

دراین رابطه w_i ها وزن‌های ورودی برای یک نرون و x_i ها مقدار ویژگی نمونه‌های ورودی به شبکه هستند. $f(x)$ خروجی نرون به ازای این ورودی‌ها است. در این رابطه مقدار $f(x)$ باید همواره به ازای وزن‌های اولیه شبکه ثابت بماند و با خروجی حاصل از وزن‌های جدید مقایسه شود.

$diffrence\_value$ مقدار تابع برازندگی خواهد بود که همواره مثبت است. با کمینه‌سازی این تابع خروجی نرون حفظ خواهد شد. همانند روش قبل برای غیرمتوازن کردن وزن‌ها، $diffrence\_value$ می‌بایست با مقدار $score$ از رابطه ۲.۳ ترکیب شود تا با حفظ کارایی شبکه، وزن‌های شبکه غیرمتوازن شوند. ساختار کروموزوم در این روش تنها شامل وزن های ورودی به یک نرون خواهد بود و الگوریتم ژنتیک به ازای هر نرون به صورت جداگانه اجرا خواهد شد.



شکل ۲.۳: بررسی ورودی یک نرون به صورت جداگانه

۳.۵.۳ معرفی الگوریتم گسسته‌سازی

ایده اصلی این الگوریتم، انتخاب مقادیر گسسته با حداقل فاصله $\epsilon \in (۰,۱)$ از یکدیگر است. ϵ های خیلی کوچک تضمین می‌کنند که همواره دقت شبکه با استفاده از این مقادیر گسسته حفظ شود. هر چه ϵ کوچک

باشد، تعداد مقادیر گسسته تولید شده بیشتر خواهد بود. اما وجود مقادیر گسسته زیاد سبب تولید الگوهای بیشتر و قوانین پیچیده‌تر خواهد شد. بنابراین برای تولید دسته‌های کمتر و قوانین ساده‌تر، مطلوب این است که گسسته‌سازی با تعداد کمی مقدار گسسته انجام شود. الگوریتم با یک ϵ بزرگ آغاز می‌شود، در صورتی که مقادیر گسسته قادر نباشند کارایی شبکه را حفظ کنند، الگوریتم دوباره برای ϵ کوچک‌تر اجرا می‌شود. این عمل تا جایی ادامه می‌یابد تا شبکه با حداقل تعداد مقادیر گسسته، کارایی مناسبی داشته باشد. خوشه‌بندی با انتخاب یک ϵ، برای هر نرون لایه میانی به شرح زیر است:

۱. اگر $\alpha_۱$ خروجی اولین نمونه برای نرون لایه میانی باشد، آنگاه $H_۱ = \alpha_۱$ مرکز اولین خوشه، $Count_۱ = ۱$ و $Sum_۱ = \alpha_۱$ ، به ترتیب تعداد و مجموع نمونه‌های عضو این خوشه و $D = ۱$ تعداد خوشه‌های تشکیل شده برای این نرون خواهد بود.

۲. برای تمام نمونه‌های $i = ۲, ۳, ..., k$ در مجموعه آموزش، اگر یک j وجود داشته باشد بطوریکه $\min_{j=۱,...,D} |_i - H_j| \leq \epsilon$ ، در این صورت $Count_j+ = ۱$ و $Sum_j+ = \alpha_i$ و در غیر این صورت یک خوشه جدید تشکیل می‌شود.

۳. در هر خوشه، مقدار H با میانگین تمام مقادیر عضو آن خوشه، جایگزین می‌شود.

زمانی که مقادیر گسسته برای تمام نرون‌های لایه میانی بدست آمدند، دقت شبکه با استفاده از مقادیر گسسته دوباره مورد بررسی قرار می‌گیرد. اگر دقت شبکه به پایین‌تر از حد تعیین شده کاهش یافت، الگوریتم دوباره با مقدار کوچک‌تر ϵ اجرا می‌شود. برای ϵ به اندازه کافی کوچک، همواره می‌توان به دقت شبکه با مقادیر پیوسته دست یافت، هر چند که تعداد مقادیر گسسته افزایش می‌یابند.

۶.۳ نتیجه‌گیری

در این فصل مراحل استخراج قانون با استفاده از روش FSRE تشریح شد. چالش‌های موجود در این روش برای بهبود قوانین استخراج شده مطرح و راهکارهایی ارائه شد. هرس‌سازی با اتلاف و گسسته‌سازی نامناسب در این روش موجب می‌شود تا قوانین کیفیت مناسبی را نداشته باشند. در پایان راهکارهای ارائه شده در سه مرحله توضیح داده شد. در مرحله اول وزن‌ها در حین آموزش در جهت تولید ساختار مناسب برای بهبود هرس‌سازی، اصلاح می‌شوند. در مرحله دوم این اصلاح به صورت محلی و برای هر نرون به صورت جداگانه انجام می‌شود. در پایان برای بهبود قوانین تولید شده، یک روش خوشه‌بندی برای گسسته‌سازی خروجی نرون‌های لایه میانی ارائه شد.

فصل ۴

بررسی و تحلیل نتایج

۱.۴ مقدمه

در این فصل به بررسی و آزمایش روش ارائه شده برای استخراج قانون از شبکه‌عصبی برای برخی از مجموعه داده‌های استاندارد موجود می‌پردازیم. مسائل مورد بررسی در این فصل عمدتا ساده می‌باشند و جهت مقایسه نتایج روش پیشنهادی نسبت به سایر روش‌های موجود، به خصوص روش FSRE می‌باشد. نتایج بدست آمده برای هر مجموعه داده به صورت مجزا و با ذکر جزئیات در بخش‌های بعد ارائه شده است. در پایان فصل نیز این نتایج با نتایج حاصل از دیگر روش‌های استخراج قانون مورد مقایسه قرار می‌گیرد.

ماتریس درهم‌ریختگی[۲] ابزار مناسبی برای مسائل دسته‌بندی است که قادر است به سادگی میزان کارایی

[2] Confusion matrix

شبکه و خطای موجود در دسته‌بندی نمونه‌های مختلف را بررسی کند. این ماتریس، یک ماتریس مربعی است که تعداد سطر و ستون آن به تعداد کلاس‌های موجود در مسئله مورد نظر است. مثالی از این ماتریس در جدول ۱.۴ نمایش داده شده است. در این مثال اگر فرض کنیم الگوریتمی برای کلاس بندی بین گربه‌ها، سگ‌ها و خرگوش‌ها طراحی کرده‌ایم. فرض کنیم در این مثال ۸ گربه، ۶ سگ و ۱۳ خرگوش داریم. در سطر مربوط به گربه‌ها، ۵ مورد به عنوان گربه و ۳ مورد به عنوان سگ دسته بندی شده‌اند. در صورتی که در سطر مربوط به خرگوش‌ها، تنها چند مورد اشتباه وجود دارد. به سادگی مشاهده می‌شود که عملکرد الگوریتم در تمییز دسته‌های خرگوش‌ها نسبت به گربه‌ها بسیار بهتر است. مشخص است که اعداد روی قطر اصلی ماتریس نمایش تعداد کلاس بندی‌های درست هستند. لذا در صورتی که تمام اعداد غیر روی قطر اصلی صفر باشند، الگوریتم دارای دقت حداکثر است.

جدول ۱.۴: نمونه‌ای از جدول در هم‌ریختگی

		کلاس واقعی		
		گربه	سگ	خرگوش
کلاس پیش‌بینی شده	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.

در این بخش مجموعه داده‌های Iris flower، Glasses و Wisconsin Breast Cancer برای ارزیابی میزان کارایی روش پیشنهادی از مجموعه پایگاه‌های داده UCI [۸] تهیه و مورد استفاده قرار گرفتند.

در تمام مسائل مورد بررسی تعداد نرون‌های لایه ورودی به تعداد ویژگی‌های موجود در مجموعه داده و تعداد نرون‌های لایه خروجی به تعداد کلاس‌های موجود در مسئله هستند. همچنین همواره مقادیر کلاس‌های

خروجی به صورت ۱ از N گسسته‌سازی می‌شود که سبب می‌شود برای هر کلاس تنها یکی از مقادیر برابر با ۱ و سایر مقادیر صفر باشد. همچنین توابع فعال‌ساز برای نرون‌های لایه میانی تانژانت هیپربولیک ($tanh$) و برای نرون‌های لایه خروجی سیگموید ($sigmoid$) در نظر گرفته شده است. مقدار ضریب یادگیری برای آموزش با استفاده از گرادیان نزولی ۰/۰۰۱ قرار داده شده است، همچنین شرط خاتمه یادگیری با استفاده از داده‌های اعتبارسنجی و معیار Cross-Entropy تنظیم می‌شود.

الگوریتم‌های ژنتیک استفاده شده در این بخش از تابع‌های برازندگی تعریف شده در روابط ۲.۳ و ۳.۳ برای تولید پاسخ بهینه استفاده می‌کنند. جمعیت اولیه شامل ۲۰۰ کروموزوم در بازه بین ـ۰/۲۵ تا ۰/۲۵ به صورت تصادفی تولید می‌شود. پارامترهای مربوط به الگوریتم‌های ژنتیک مطابق جدول ۲.۴ تعریف شده‌اند.

جدول ۲.۴: پارامترهای الگوریتم ژنتیک

پارامتر	مقدار
جمعیت اولیه	۲۰۰
بازه تولید جمعیت اولیه	[۰/۲۵، ۰/۲۵ـ]
عملگر انتخاب	رقابتی
احتمال ترکیب	۰/۸
احتمال جهش	۰/۵
انحراف معیار توزیع گوسی عملگر جهش	۰/۴
تعداد کروموزوم‌های نخبه	۱۰
حداکثر تعداد نسل	۵۰۰
حداکثر زمان اجرا	۷۰۰ ثانیه

۲.۴ مجموعه داده Iris

مجموعه داده Iris یکی از پرکاربردترین مجموعه داده‌هایی است که در بررسی نحوه کارایی سیستم‌های شناسایی‌گر استفاده می‌شود. این مجموعه شامل نمونه‌های مختلف گل زنبق می‌باشد که در سه دسته متفاوت تقسیم‌بندی می‌شوند. مشخصات این مجموعه در جدول ۳.۴ ارائه شده است.

با توجه به پراکندگی مقادیر ویژگی‌های مختلف، این داده‌ها برای اعمال به شبکه‌عصبی نیاز به آماده‌سازی

جدول ۳.۴: اطلاعات کلی در مورد مجموعه داده Iris Flower

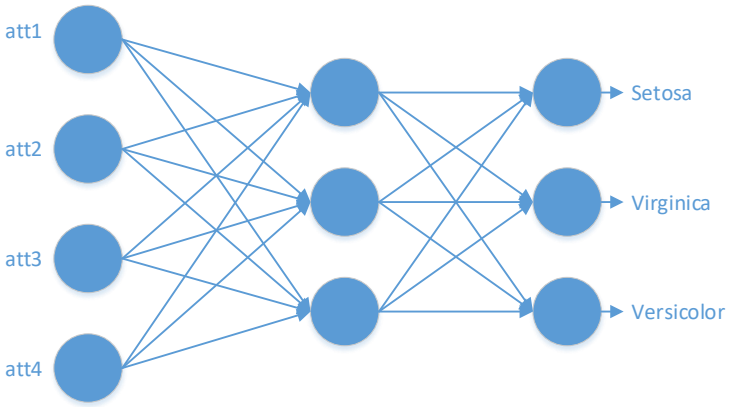
نام مجموعه داده	Flower Iris		
تعداد کلاس‌های خروجی	۳		
تعداد ویژگی‌های ورودی	۴		
تعداد نمونه‌ها	۱۵۰		
تعداد نمونه در هر کلاس	Setosa=۵۰	Virginica=۵۰	Versicolor=۵۰

دارند. عملیات نرمال‌سازی با استفاده از رابطه ۱.۳ بیان شده در بخش ۱.۲.۳ برای یکنواخت کردن پراکندگی تمام ویژگی‌ها انجام می‌شود. همچنین خروجی شبکه به صورت ۱ از N گسسته‌سازی می‌شود تا هر نرون خروجی بیانگر یک کلاس باشد.

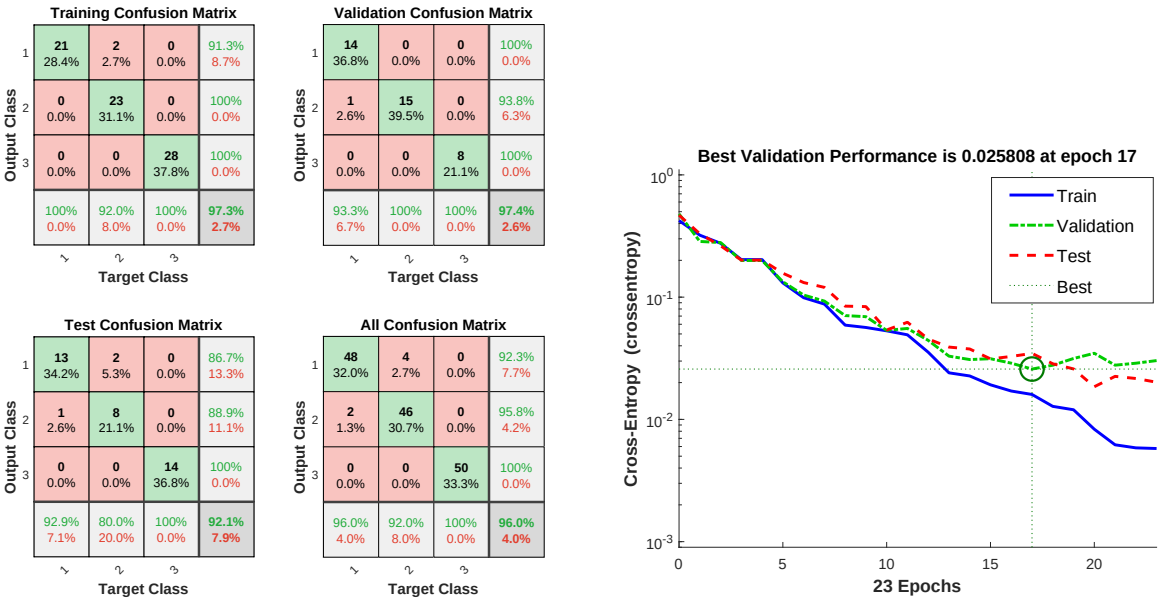
داده‌ها برای اعمال به شبکه به سه دسته مجموعه آموزش، اعتبار سنجی و ارزیابی تقسیم می‌شوند. برای این مجموعه داده، ۵۰% داده‌ها برای آموزش شبکه در مجموعه آموزش، ۲۵% داده‌ها برای اعتبارسنجی در مجموعه داده اعتبار سنجی و ۲۵% باقی مانده داده‌ها در مجموعه ارزیابی برای بررسی کارایی شبکه برای نمونه‌های دیده نشده قرار می‌گیرند.

با توجه به تعداد ویژگی‌های ورودی و کلاس‌های خروجی که در جدول ۳.۴ بیان شده است، ۴ نرون در لایه ورودی و ۳ نرون در لایه خروجی قرار می‌گیرد. تعداد نرون‌های لایه میانی با توجه به کارایی شبکه تعیین می‌شود، بدین صورت که در ابتدا کارایی شبکه با ۱ نرون در لایه میانی محاسبه می‌شود، در صورتی که کارایی شبکه مطلوب نباشد یک نرون به لایه میانی اضافه می‌شود. این عملیات تا رسیدن به کارایی مطلوب، ادامه پیدا می‌کند. در این مسئله با ۳ نرون لایه میانی به کارایی مطلوبی دست یافتیم. ساختار اولیه شبکه در شکل ۱.۴ نمایش داده شده است. این شبکه قادر است با دقت ۹۶% عملیات دسته‌بندی را انجام دهد.

ابتدا آموزش شبکه با استفاده از این ساختار به روش گرادیان نزولی و الگوریتم پس انتشار خطا انجام می‌شود. روند آموزش شبکه و کارایی آن در شکل ۲.۴ نمایش داده شده است. همانطور که مشاهده می‌شود این شبکه قادر است نمونه‌های هر کلاس را به خوبی شناسایی کند.



شکل ۱.۴: ساختار اولیه شبکه برای آموزش برای مجموعه داده Iris

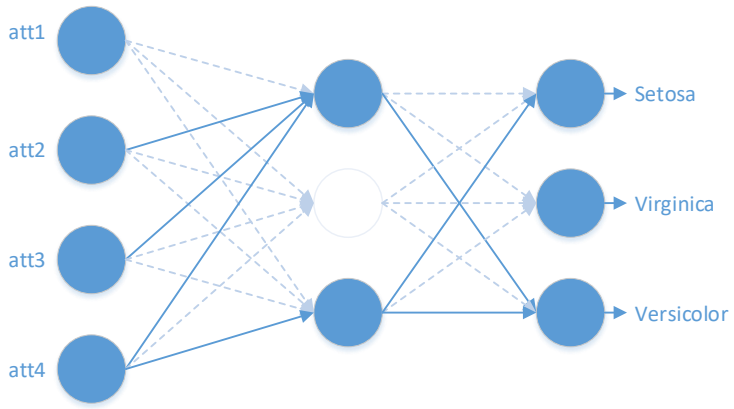


(آ) روند آموزش شبکه برای مجموعه داده Iris (ب) ماتریس درهم‌ریختگی بدست آمده برای مجموعه داده Iris

شکل ۲.۴: روند آموزش و کارایی شبکه برای مجموعه داده Iris

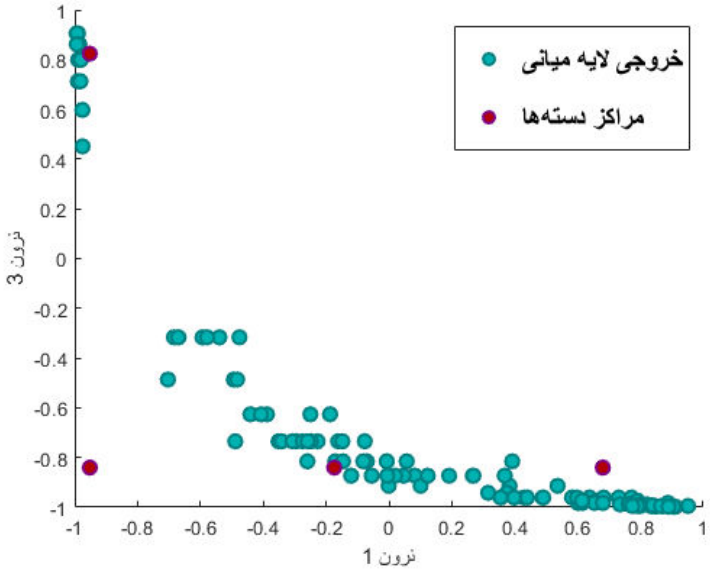
در مرحله بعد پس از آموزش شبکه، با استفاده از الگوریتم‌های ارائه شده در بخش‌های ۱.۵.۳ و ۲.۵.۳ وزن‌های شبکه را برای هرس‌سازی آماده می‌کنیم. پس از اصلاح وزن‌ها شبکه حاصل را با استفاده از الگوریتم معرفی شده در ۳.۲.۳ هرس‌سازی می‌کنیم تا به یک ساختار بهینه هرس شده برای استخراج قانون دست یابیم. نتایج هرس‌سازی در شکل ۳.۴ نمایش داده شده است. همانطور که مشاهده می‌شود تعداد کمی از اتصالات در شبکه هرس شده باقی مانده است اما این شبکه همچنان قادر است تا نمونه‌های ورودی را با دقت ۹۶/۶۶٪

دسته‌بندی نماید.



شکل ۳.۴: ساختار نهایی هرس شده شبکه برای مجموعه داده Iris

در نهایت استخراج قانون با استفاده از الگوریتم FSRE بیان شده در بخش ۴.۲.۳ انجام خواهد شد. برای اینکار ابتدا مقادیر خروجی نرون‌های لایه میانی با استفاده از الگوریتم خوشه‌بندی ارائه شده در ۳.۵.۳ گسسته‌سازی می‌شوند. همانطور که مشاهده می‌شود، تنها ۲ نرون در شبکه هرس شده باقی مانده است. نتایج حاصل از گسسته‌سازی بر روی این مجموعه داده در شکل ۴.۴ نمایش داده شده است. شبکه با استفاده از این مقادیر گسسته قادر است تا نمونه‌های موجود را با دقت ۹۷/۳۳% دسته‌بندی نماید.



شکل ۴.۴: نتایج حاصل از گسسته‌سازی نرون‌های لایه میانی برای مجموعه داده Iris

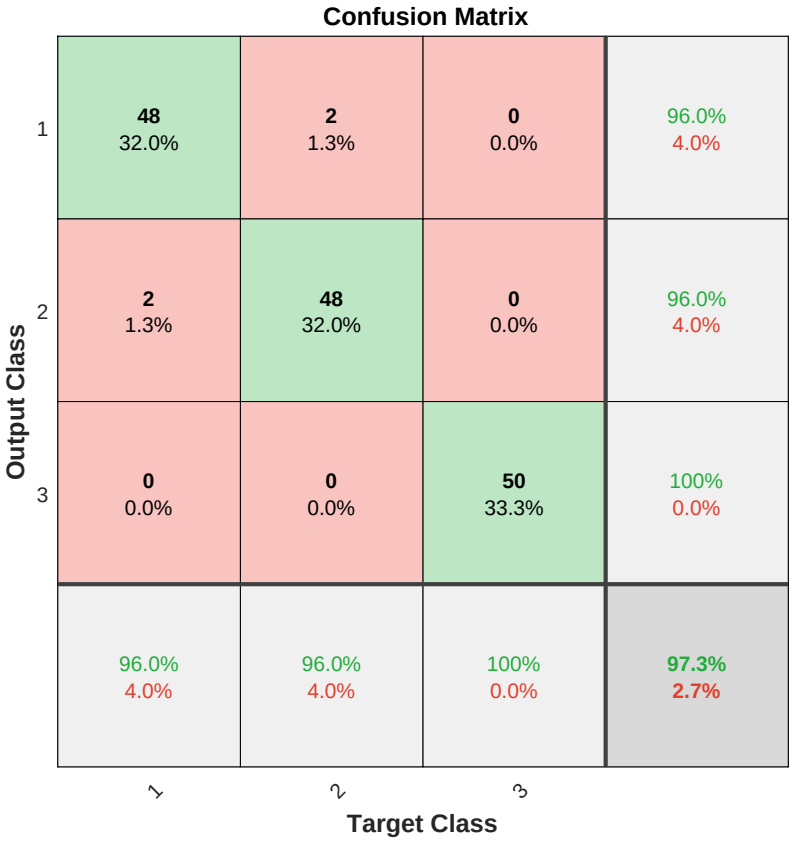
با توجه به اتصالات باقی‌مانده در شبکه هرس شده، ویژگی اول به طور کامل غیر موثر تشخیص داده شده است و از شبکه حذف شده است. با توجه به اینکه اولین کلاس خروجی فقط از سومین نرون لایه میانی تاثیر می‌پذیرد و این نرون تنها از ویژگی چهارم ورودی می‌گیرد، در نتیجه قوانین تولید شده برای این کلاس فقط به این ویژگی وابسته خواهد بود. ورودی‌های اولین کلاس خروجی در ساختار هرس شده به طور کامل حذف شده‌اند، در نتیجه در این ساختار نمی‌توان برای این کلاس قانونی استخراج کرد و این کلاس در قانون پیش فرض قرار خواهد گرفت. قوانین مربوط به آخرین کلاس نیز به همین ترتیب استخراج خواهد شد. قوانین استخراج شده برای این مجموعه داده در قالب رابطه ۱.۴ بیان شده‌اند که قادرند نمونه‌های این مجموعه داده را با دقت مطلوب ۹۷/۳۳% دسته‌بندی نمایند. ماتریس درهم‌ریختگی برای بررسی کارایی قوانین استخراج شده در شکل ۵.۴ ارائه شده است.

$$\begin{aligned}
&\textbf{if} \quad (att۴ \leq ۰/۶) \quad \textbf{then} \quad Setosa \\
&\textbf{else if} \quad (۲ \leq att۲ \leq ۳/۴) \quad \textbf{and} \quad (۳ \leq att۳ \leq ۵/۱) \\
&\qquad \textbf{and} \quad (۱ \leq att۴ \leq ۱/۶) \quad \textbf{then} \quad Versicolour \\
&\textbf{else} \quad Virginica
\end{aligned} \tag{۱.۴}$$

۳.۴ مجموعه داده Wisconsin Breast Cancer

مجموعه داده سرطان سینه یا Wisconsin Breast Cancer یکی از مجموعه داده های دسته‌بندی موجود در زمینه پزشکی و تشخیص بیماری می‌باشد. در این مجموعه داده دو نوع سرطان سینه خوش‌خیم [۱] و بدخیم [۲]

[1] Benign [2] Malignant



شکل ۵.۴: ماتریس درهم ریختگی برای قوانین استخراج شده مجموعه داده Iris

وجود دارد که با استفاده از ۹ ویژگی موجود در این مجموعه داده، عملیات دسته‌بندی داده‌ها انجام می‌شود. جدول ۴.۴ اطلاعات کلی را در رابطه با این مجموعه داده نشان می‌دهد. همانطور که مشاهده می‌شود، تعداد نمونه‌های دو دسته موجود اختلاف زیادی دارند، بطوریکه تعداد نمونه‌های خوش‌خیم حدود ۶۵/۵% از این مجموعه را تشکیل داده و مابقی نمونه‌ها نیز متعلق به حالت بدخیم هستند.

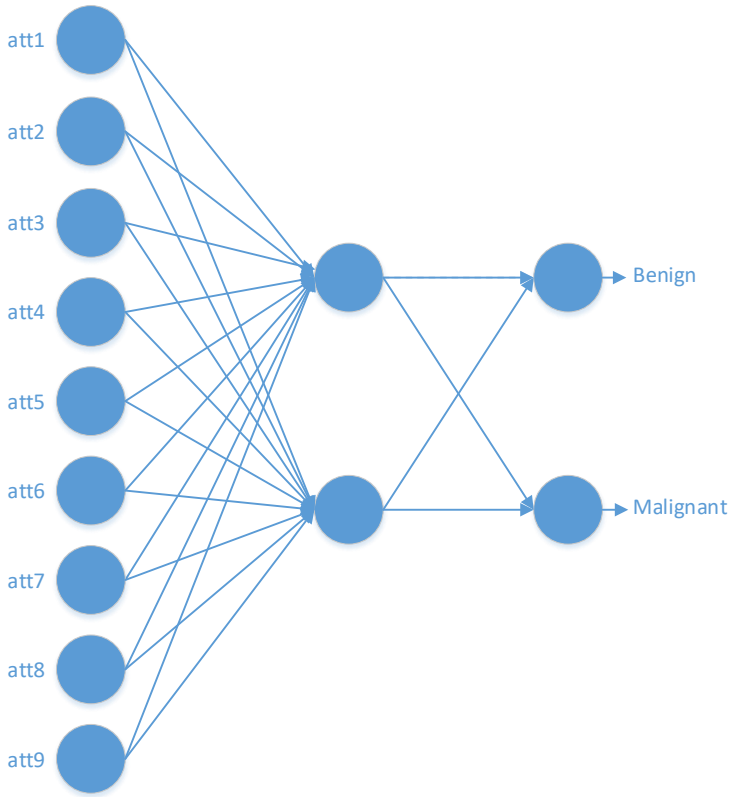
جدول ۴.۴: اطلاعات کلی در مورد مجموعه داده Wisconsin Breast Cancer

نام مجموعه داده	Cancer Breast Wisconsin
تعداد کلاس‌های خروجی	۲
تعداد ویژگی‌های ورودی	۹
تعداد نمونه‌ها	۶۹۹
تعداد نمونه در هر کلاس	Benign=۴۵۸ Malignant=۲۴۱

این مجموعه داده شامل ویژگی‌ها با مقادیر گسسته بوده که نیاز است تا قبل از اعمال به شبکه عصبی، نرمال‌سازی شوند. برای این کار از رابطه‌ی ۱.۳ استفاده می‌گردد. در نهایت این مجموعه داده به سه مجموعه

آموزش، اعتبارسنجی و ارزیابی با نسبت ۵۰ـ۲۵ـ۲۵ تقسیم می‌شود.

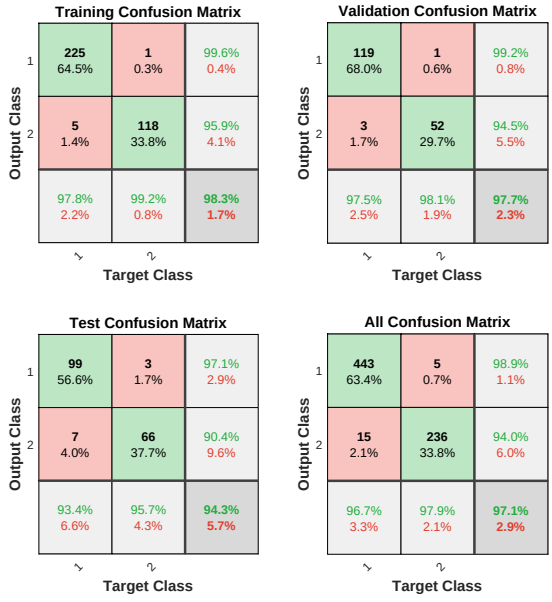
ساختار اولیه بدست آمده برای آموزش شبکه با ۹ نرون در لایه ورودی، ۲ نرون در لایه میانی و ۲ نرون در لایه خروجی است. شکل ۶.۴ ساختار اولیه شبکه را نشان می‌دهد.



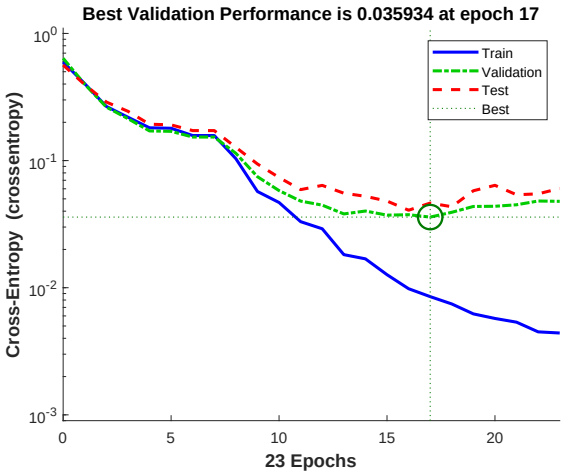
شکل ۶.۴: ساختار اولیه شبکه برای آموزش برای مجموعه داده Wisconsin Breast Cancer

ساختار اولیه حاصل را با استفاده از روش گرادیان نزولی و الگوریتم پس انتشار خطا بر روی نمونه‌های آموزشی، آموزش می‌دهیم. نتایج حاصل از آموزش و روند یادگیری شبکه در شکل ۷.۴ ارائه شده است.

این شبکه قادر است نمونه‌های موجود را با دقت مناسبی دسته‌بندی نماید. برای استخراج قانون از این شبکه ابتدا به اصلاح وزن‌های شبکه در دو مرحله سراسری و محلی می‌پردازیم. سپس با استفاده از الگوریتم هرس‌سازی ارائه شده، اتصالات بی‌اهمیت را از شبکه حذف می‌کنیم. ساختار هرس شده شبکه در شکل ۸.۴ نمایش داده شده است.



(ب) ماتریس درهم‌ریختگی بدست آمده برای مجموعه داده Wisconsin Breast Cancer

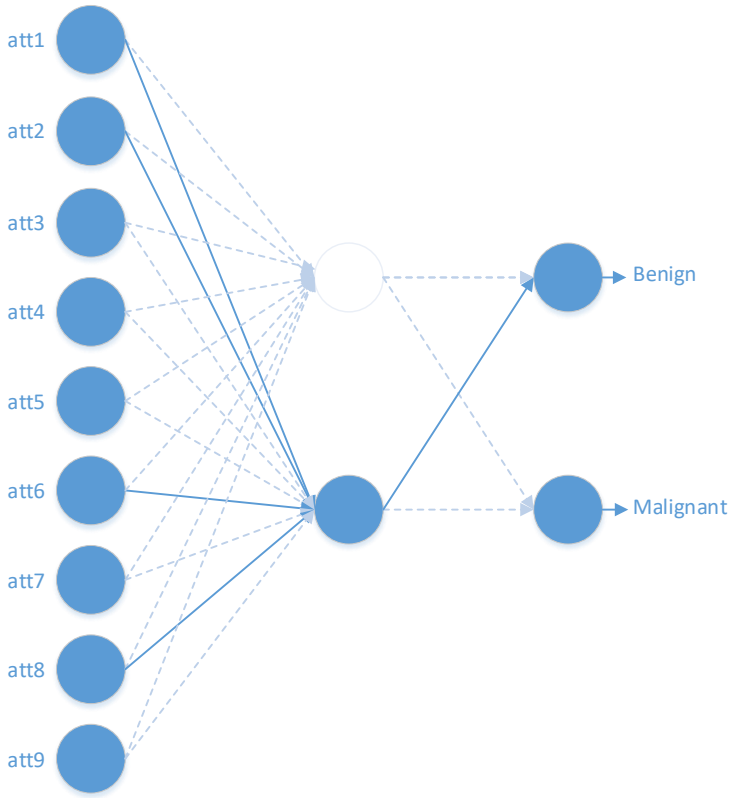


(آ) روند آموزش شبکه برای مجموعه داده Wisconsin Breast Cancer

شکل ۷.۴: روند آموزش و کارایی شبکه برای مجموعه داده Wisconsin Breast Cancer

دقت دسته‌بندی با استفاده از شبکه هرس شده برای تمام نمونه‌ها ۹۶/۸۵٪ است. برای استخراج قانون ابتدا خروجی نرون‌های لایه میانی با استفاده از الگوریتم خوشه‌بندی معرفی شده گسسته‌سازی می‌شوند، سپس قوانین نهایی با استفاده از این مقادیر گسسته شده و با بهره‌گیری از الگوریتم ارائه شده در ۴.۲.۳ استخراج می‌شوند. شبکه با استفاده از مقادیر گسسته در نرون‌های لایه میانی نمونه‌های موجود را با دقت ۹۶/۶۵٪ دسته‌بندی می‌کند.

همانطور که مشخص است در ساختار هرس شده شبکه فقط کلاس خوش‌خیم دارای ورودی از لایه مخفی می‌باشد و از ۴ ویژگی ورودی تاثیر می‌پذیرد. در نتیجه قوانین مربوط به این کلاس استخراج و کلاس بدخیم به صورت قانون پیش فرض در نظر گرفته می‌شود. قوانین استخراج شده برای این مجموعه داده در رابطه ۲.۴

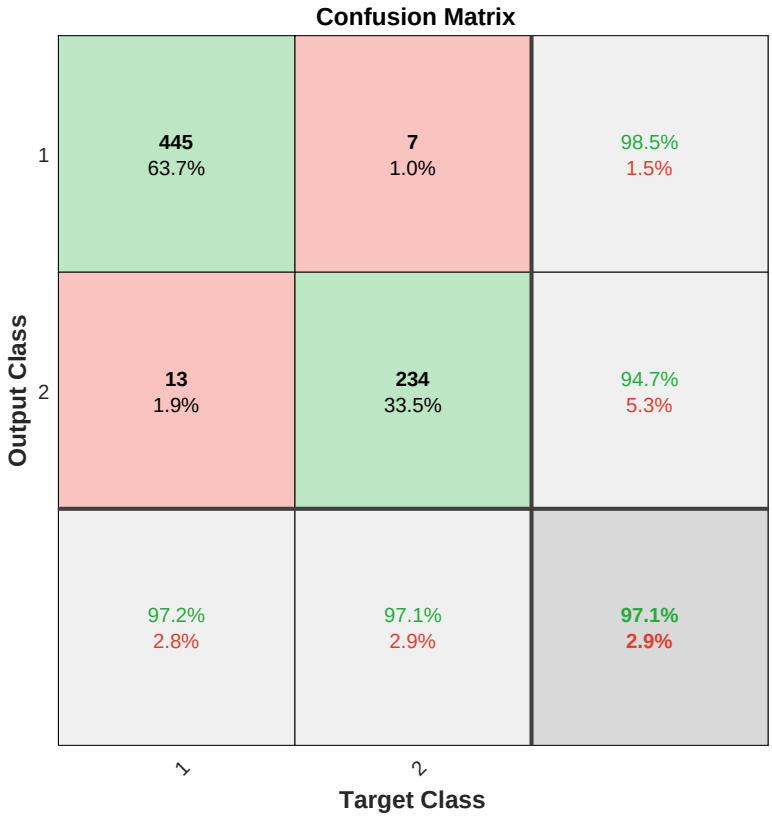


شکل ۸.۴: ساختار نهایی هرس شده شبکه برای مجموعه داده Wisconsin Breast Cancer

بیان شده‌اند.

$$\begin{aligned} &\textbf{if} \quad (att۱ \leq ۰/۶) \quad \textbf{and} \quad (att۲ \leq ۰/۴) \quad \textbf{and} \quad (att۶ \leq ۰/۶) \\ &\qquad \textbf{and} \quad (att۸ \leq ۰/۸) \quad \textbf{then} \quad Benign \\ &\textbf{else} \quad Malignant \end{aligned} \tag{۲.۴}$$

این قوانین قادرند نمونه‌های موجود را با دقت ۹۷/۱۳% دسته‌بندی نمایند. نتایج حاصل از دسته‌بندی این مجموعه داده با استفاده از قوانین استخراج شده در قالب ماتریس درهم‌ریختگی در شکل ۹.۴ قرار گرفته است.



شکل ۹.۴: ماتریس درهم ریختگی برای قوانین استخراج شده مجموعه داده Wisconsin Breast Cancer

۴.۴ مجموعه داده Glasses

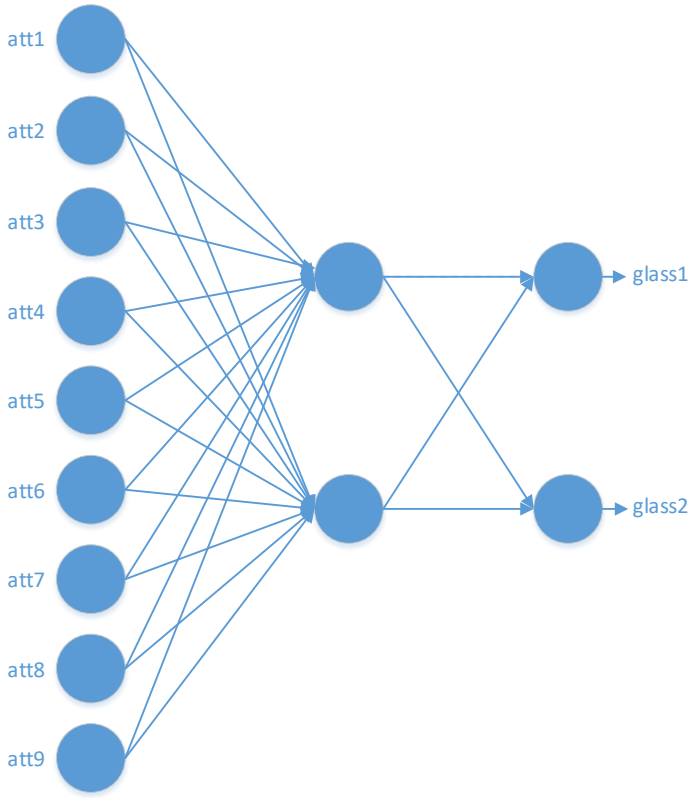
این مجموعه داده یکی دیگر از مجموعه داده‌های دسته‌بندی در UCI بوده که شامل ۲۱۴ نمونه در ۲ کلاس متفاوت است. در این مجموعه داده به منظور دسته‌بندی نمونه‌ها از ۹ ویژگی ورودی استفاده می‌شود. مشخصات کلی این مجموعه داده در جدول ۵.۴ نشان داده شده است. ویژگی‌های موجود در این مجموعه داده دارای مقادیر پیوسته می‌باشد. در فاز پیش پردازش ابتدا این مقادیر نرمال‌سازی شده و سپس نمونه‌های موجود را به سه مجموعه آموزش، اعتبار سنجی و ارزیابی با نسبت ۵۰ـ۲۵ـ۲۵ تقسیم شده است. سعی شده است نسبت توزیع نمونه‌ها در این مجموعه‌ها رعایت شود.

ساختار اولیه شبکه عصبی همانند قبل با توجه به تعداد ویژگی‌های ورودی و کلاس‌های خروجی برای

جدول ۵.۴: اطلاعات کلی در مورد مجموعه داده Glasses

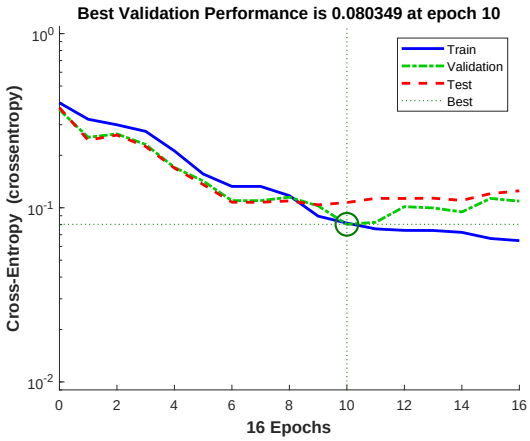
نام مجموعه داده	Glasses	
تعداد کلاس‌های خروجی	۲	
تعداد ویژگی‌های ورودی	۹	
تعداد نمونه‌ها	۲۱۴	
تعداد نمونه در هر کلاس	Glass۱=۵۱	Glass۲=۱۶۳

لایه‌های ورودی و خروجی تعیین می‌شود. همچنین تعداد نرون‌های لایه میانی با توجه به کارایی شبکه با کمترین تعداد نرون در لایه میانی تعیین می‌شود. ساختار اولیه شبکه در شکل ۱۰.۴ نمایش داده شده است. همچنین کارایی شبکه در قالب ماتریس درهم‌ریختگی و روند آموزش شبکه در شکل ۱۱.۴ بیان شده است. همانطور که مشاهده می‌شود شبکه قادر است با دقت مطلوبی داده‌های موجود را دسته‌بندی نماید.

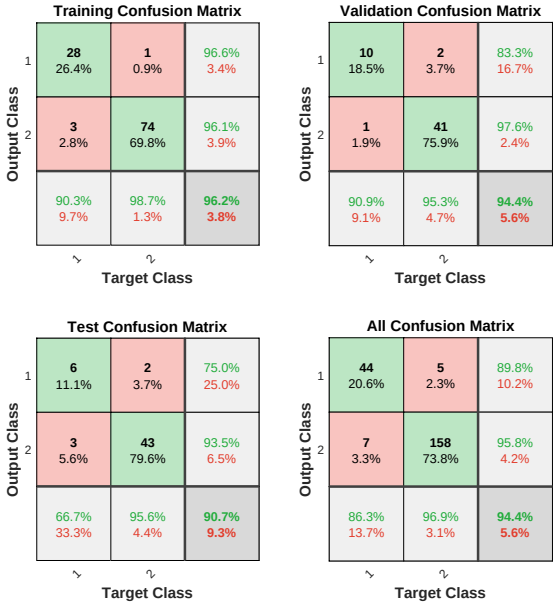


شکل ۱۰.۴: ساختار اولیه شبکه برای آموزش برای مجموعه داده Glasses

در گام بعد وزن‌های آموزش دیده به روش معمول توسط الگوریتم ژنتیک و در دو مرحله اصلاح و برای هرس‌سازی آماده می‌شوند. پس از اعمال الگوریتم هرس‌سازی شبکه حاصل به صورت شکل ۱۲.۴ بدست آمده



(آ) روند آموزش شبکه برای مجموعه داده Glasses



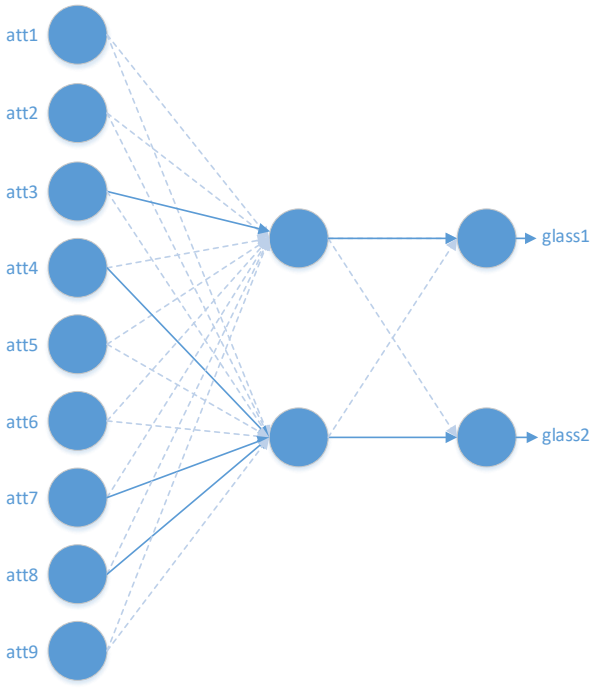
(ب) ماتریس درهم‌ریختگی بدست آمده برای مجموعه داده Glasses

شکل ۱۱.۴: روند آموزش و کارایی شبکه برای مجموعه داده Glasses

است. دقت شبکه حاصل برابر ۹۴/۳۹٪ برای تمام نمونه‌ها است. همانطور که مشاهده می‌شود ساختار هرس شده همچنان قادر است با دقت بسیار بالایی عملیات دسته‌بندی را انجام دهد. همچنین دقت این شبکه با استفاده از مقادیر گسسته شده ۹۳/۴۵٪ بدست آمده است.

با توجه به ساختار هرس شده بدست آمده تعداد و قوانین به دست آمده برای کلاس اول ساده‌تر است. در نتیجه کلاس دوم را به عنوان قانون پیش فرض در نظر گرفته و قوانین برای کلاس اول استخراج می‌شوند. همانطور که مشاهده می‌شود کلاس اول تنها از ویژگی سوم تاثیر می‌پذیرد. قوانین نهایی استخراج شده در رابطه ۳.۴ آمده است.

$$\textbf{if} \quad (att۳ \leq ۲/۷) \quad \textbf{then} \quad glass۱ \qquad \textbf{else} \quad glass۲ \tag{۳.۴}$$

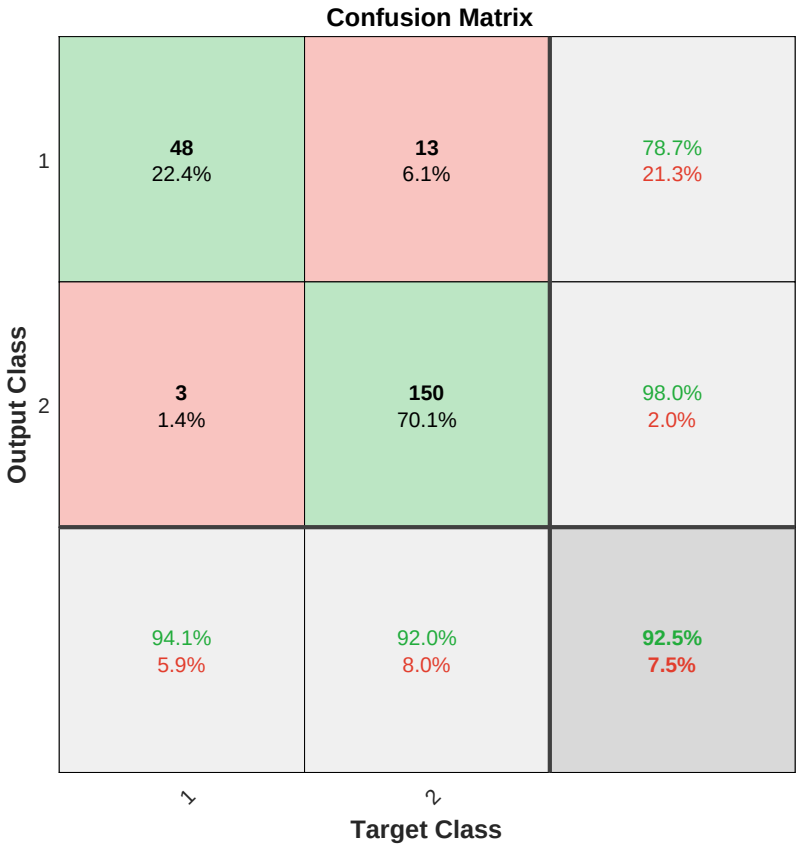


شکل ۱۲.۴: ساختار نهایی هرس شده شبکه برای مجموعه داده Glasses

کارایی این قوانین برای دسته‌بندی نمونه‌ها در قالب ماتریس درهم‌ریختگی در شکل ۱۳.۴ بیان شده است. این قوانین قادرند تا نمونه‌های ورودی را با دقت مطلوب ۹۲/۵۲% دسته‌بندی نمایند.

۵.۴ تحلیل و بررسی نتایج

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 Iris flower، Glasses و Wisconsin Breast Cancer است که از پایگاه داده UCI جمع‌آوری شده و مورد استفاده قرار گرفته‌اند. همچنین روش‌هایی را که به منظور مقایسه در این بخش مورد بررسی قرار گرفته‌اند شامل روش‌های REANN [۲۵]، RGANN [۲۲]، ESRNN [۲۶]، DIFACONN [۲۹]، Rex [۲۱]، Rex-P [۳۶]، Rex-M [۳۶]، SV-DT [۱۵]، RBFGD [۵۸]، ERBF [۷]، TB-RBF [۲۸]، MMDT [۵۲]، MDTF [۱۸]، GRBF [۷]، FSRE [۵۳] است.



شکل ۱۳.۴: ماتریس درهم ریختگی برای قوانین استخراج شده مجموعه داده Glasses

نتایج آزمایشات برای مجموعه داده Iris به صورت جدول ۶.۴ بدست آمده است.

جدول ۶.۴: نتایج مقایسه روش پیشنهادی نسبت به سایر روش‌ها برای مجموعه داده Iris

نام مجموعه داده	ویژگی‌ها	روش پیشنهادی	FSRE	DIFACONN	ESRNN	SV-DT	Rex-P	Rex-M	MDTF
	تعداد قوانین	۳	۳	۳/۲۷	۳	۷	۸/۸	۸/۸	۵
Iris	میانگین تعداد شرایط	۱/۳۳	۱	–	۱	۳/۵۷	–	–	–
	دقت قوانین	۹۷/۳۳%	۹۶/۶۶%	۹۸/۰۷%	۹۸/۶۷%	۸۰%	۹۸/۶۷%	۹۷/۶%	۹۷/۳۳%

همانطور که مشاهده می‌شود روش پیشنهادی توانسته با تولید تعداد کم قوانین عملکرد مناسبی برای دسته‌بندی نمونه‌های این مجموعه داده داشته باشد. همچنین این روش عملکرد روش FSRE را بهبود بخشیده است.

نتایج آزمایشات برای مجموعه داده Glasses در جدول ۷.۴ قرار داده شده است. با توجه به نتایج بدست آمده، روش پیشنهادی با حداقل تعداد قوانین بهترین عملکرد را برای دسته‌بندی نمونه‌های این مجموعه داشته

جدول ۷.۴: نتایج مقایسه روش پیشنهادی نسبت به سایر روش‌ها برای مجموعه داده Glasses

نام مجموعه داده	ویژگی‌ها	روش پیشنهادی	FSRE	DIFACONN	RBF-GD	GRBF	ERBF	TB-RBF	MMDT
	تعداد قوانین	۲	۲	۱۷/۶۳	۶	۱۶	۱۶	۱۶/۴	۱۳/۲۷
Glasses	میانگین تعداد شرایط	۰/۵	۳	–	۳/۳۳	–	–	–	۶۵
	دقت قوانین	۹۲/۵۲%	۸۷/۸۵%	۸۴/۱۳%	۸۶/۲۱%	۶۴/۷۶%	۶۸/۵۷%	۶۲/۸%	۸۶/۵۸%

است و به میزان قابل توجهی روش پایه را بهبود بخشیده است.

نتایج آزمایشات برای مجموعه داده Wisconsin Breast Cancer نیز در جدول ۸.۴ قرار دارد.

جدول ۸.۴: نتایج مقایسه روش پیشنهادی نسبت به سایر روش‌ها برای مجموعه داده Wisconsin Breast Cancer

نام مجموعه داده	ویژگی‌ها	روش پیشنهادی	FSRE	REANN	RGANN	Rex	SV-DT	Rex-p	Rex-M
	تعداد قوانین	۲	۲	۲	۲	۲	۴	۳	۱۵/۶
Cancer Breast Wisconsin	میانگین تعداد شرایط	۲	۳	۳	۳	۳	۲/۲۵	–	–
	دقت قوانین	۹۷/۱۳%	۹۶/۵۶%	۹۶/۲۸%	۹۶/۲۸%	۹۶/۲۸%	۹۲/۹۸%	۹۵/۹۷%	۹۳/۴۳%

همانطور که مشاهده می‌شود روش پیشنهادی به خوبی توانسته در این مسئله نیز عملکرد مناسبی داشته باشد و با تعداد کم‌تری از قوانین قادر است نمونه‌های این مجموعه داده را به خوبی دسته‌بندی نماید.

۶.۴ نتیجه‌گیری

در این فصل روش پیشنهادی بر روی برخی از پایگاه‌های داده UCI اعمال شد. نتایج بدست آمده نشان می‌دهد که این روش قادر است با حداقل تعداد قوانین، نمونه‌ها را با دقت مطلوبی دسته‌بندی نماید. همچنین در ادامه به مقایسه روش پیشنهادی با سایر روش‌های موجود، به خصوص روش FSRE پرداختیم. نتایج بدست آمده نشان می‌دهد که روش پیشنهادی به طور میانگین به میزان دو درصد روش پایه را بهبود داده است.

فصل ۵

جمع‌بندی و پیشنهادات

۱.۵ مقدمه

در این پایان نامه برخی از مهم‌ترین روش‌های موجود مبتنی بر شبکه‌های عصبی برای استخراج قانون بررسی شده است. روش پایه مورد بررسی در این پایان نامه یک روش جدید چهار مرحله‌ای تحت عنوان FSRE است. این روش توانایی کشف الگوهای بین داده‌ای برای مسائل پیچیده را دارد. همچنین برخلاف برخی روش‌های موجود، انواع داده‌ای گسسته و پیوسته را پشتیبانی می‌کند.چالش‌های اساسی مورد بررسی در این مقاله، کاهش دقت قوانین استخراج شده از شبکه‌عصبی، به دلیل متکی بودن این روش‌ها به عملیات هرس‌سازی شبکه‌عصبی و اهمیت گسسته‌سازی خروجی تابع فعال‌ساز برای نرون‌های لایه میانی است.

۲.۵ نوآوری تحقیق

در این پژوهش، با استفاده از الگوریتم ژنتیک، یک راهکار نوین برای کاهش خسارات ناشی از هرس سازی بر دقت شبکه عصبی ارائه شده است. الگوریتم ژنتیک با استفاده از یک تابع برازندگی مناسب، وزن‌های غیرمتوازنی در دو مرحله سراسری و محلی فراهم می‌کند. در مرحله اول به هنگام آموزش شبکه اعمال می‌شود و سعی دارد تا به تولید وزن‌های شبکه جهت‌دهی کند. در مرحله دوم و بعد از آموزش شبکه، این الگوریتم به صورت محلی، به اصلاح وزن‌های موجود می‌پردازد تا در نهایت ساختار بهینه‌ای در اختیار الگوریتم هرس‌سازی قرار بگیرد. به کمک این وزن‌های غیرمتوازن، شبکه هرس شده بهینه‌تر و با وزن‌های موثرتری تولید خواهد شد که سبب می‌شود، قوانین استخراج شده کارایی بهتری در مقایسه با روش‌های موجود ارائه دهد.

اهمیت گسسته‌سازی خروجی تابع فعال‌ساز برای نرون‌های لایه میانی در روش‌های مبتنی بر دیدگاه تجزیه‌ای بیان و چند روش گسسته‌سازی استفاده شده در این حوزه بیان شد. در این پژوهش با استفاده از یک روش گسسته‌سازی با استفاده از خوشه‌بندی، کارایی روش FSRE را بهبود بخشیدیم. این عملیات به صورت دستی و غیر بهینه انجام می‌شد. استفاده از روش خوشه‌بندی معرفی شده سبب شد تا این عملیات به صورت خودکار و بهینه انجام شود.

۳.۵ پیشنهادات

کارهایی که می‌توان برای بهبود عملکرد این روش در آینده انجام داد عبارت‌اند از:

- اصلاح وزن‌ها در این روش با استفاده از یک تابع برازندگی تجربی انجام می‌شود. در صورتی که بتوان با استفاده از روش‌های ریاضی یک تابع برازندگی مناسب برای این روش تعریف کرد، می‌توان وزن‌ها را به

شکل مناسب‌تری برای مسائل پیچیده اصلاح و نتایج قابل قبول‌تری تولید کرد.

- استخراج قانون در این روش با استفاده از یک شبکه‌عصبی چندلایه پرسپترون با سه لایه استاندارد انجام می‌شود. امروزه استفاده از شبکه‌های عمیق برای حل مسائل پیچیده رونق زیادی پیدا کرده است. با تعمیم این روش برای شبکه‌های عمیق می‌توان از این روش برای کشف دانش در مسائل کاربردی و پیچیده‌تری استفاده کرد.

- الگوریتم هرس‌سازی استفاده شده در این روش ساختار ساده و کارایی دارد. اگر بتوان با اعمال تغییراتی این الگوریتم را هوشمندسازی کرد، ساختار هرس شده بهینه‌تری تولید خواهد شد که منجر به تولید قوانین بهتری می‌شود.

- استخراج قانون با استفاده از یک ساختار هرس شده برای تمام کلاس‌های موجود انجام می‌شود. در صورتی که برای هر کلاس خروجی یک شبکه هرس شده جداگانه تولید کنیم، وزن‌های موثر هر کلاس به صورت جداگانه شناسایی خواهند شد. در این صورت قوانین تولید شده برای هر کلاس با معناتر خواهند بود.

مراجع

[1] Aharkava, L. (2010). Artificial neural networks and self-organization for knowledge extraction.

[2] Ahn, B. S., S. Cho, and C. Kim (2000). The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert systems with applications* 18(2), 65–74.

[3] Amato, F., A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel (2013). Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine 11*, 47–58.

[4] Andrews, R., J. Diederich, and A. B. Tickle (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems* 8(6), 373–389.

[5] Azcarraga, A. and R. Setiono (2018). Neural network rule extraction for gaining insight into the characteristics of poverty. *Neural Computing and Applications* 30(9), 2795–2806.

[6] Baesens, B., R. Setiono, C. Mues, and J. Vanthienen (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science* 49(3), 312–329.

[7] Behloul, F., B. P. Lelieveldt, A. Boudraa, and J. H. Reiber (2002). Optimal design of radial basis function neural networks for fuzzy-rule extraction in high dimensional data. *Pattern recognition* 35(3), 659–675.

[8] Blake, C. L. (1998). Uci repository of machine learning databases, irvine, university of california. *http://www. ics. uci. edu/~ mlearn/MLRepository. html*.

[9] Bondarenko, A., T. Zmanovska, and A. Borisov (2010). Decompositional rules extraction methods from neural networks. *publication. editionName*, 256–262.

[10] Cano, A., D. T. Nguyen, S. Ventura, and K. J. Cios (2016). ur-caim: improved caim discretization for unbalanced and balanced data. *Soft Computing* 20(1), 173–188.

[11] Carpenter, G. A. and A.-H. Tan (1995). Rule extraction: From neural architecture to symbolic representation. *Connection Science* 7(1), 3–27.

[12] de Fortuny, E. J. and D. Martens (2012). Active learning based rule extraction for regression. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pp. 926–933. IEEE.

[13] Diederich, J. (1992). Explanation and artificial neural networks. *International Journal of Man-Machine Studies* 37(3), 335–355.

[14] ElAlami, M. (2012). Destructive algorithm for rule extraction based on a trained neural network. *International Journal of Computer Applications* 42(21), 8–14.

[15] Farquad, M., J. Sultana, G. Nagalaxmi, and G. Savankumar (2014). Knowledge discovery from data: comparative study. *Training* 1(3), 4.

[16] Gupta, A., S. Park, and S. M. Lam (1999). Generalized analytic rule extraction for feed-forward neural networks. *IEEE transactions on knowledge and data engineering* 11(6), 985–991.

[17] Hayashi, Y. (2016). Application of a rule extraction algorithm family based on the re-rx algorithm to financial credit risk assessment from a pareto optimal perspective. *Operations Research Perspectives 3*, 32–42.

[18] Hong, T.-P. and J.-B. Chen (2000). Processing individual fuzzy attributes for fuzzy rule induction. *Fuzzy sets and systems* 112(1), 127–140.

[19] Hruschka, E. R. and N. F. Ebecken (2006). Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. *Neurocomputing* 70(1-3), 384–397.

[20] Jivani, K., J. Ambasana, and S. Kanani (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.

[21] Kamruzzaman, S. (2010a). Rex: An efficient rule generator. *arXiv preprint arXiv:1009.4988*.

[22] Kamruzzaman, S. (2010b). Rgann: An efficient algorithm to extract rules from anns. *arXiv preprint arXiv:1009.4962*.

[23] Kamruzzaman, S. and A. R. Hasan (2010). Rule extraction using artificial neural networks. *arXiv preprint arXiv:1009.4984*.

[24] Kamruzzaman, S., M. Islam, et al. (2010a). An algorithm to extract rules from artificial neural networks for medical diagnosis problems. *arXiv preprint arXiv:1009.4566*.

[25] Kamruzzaman, S., M. Islam, et al. (2010b). Extraction of symbolic rules from artificial neural networks. *arXiv preprint arXiv:1009.4570*.

[26] Kamruzzaman, S. and A. Sarkar (2011). A new data mining scheme using artificial neural networks. *Sensors* 11(5), 4622–4647.

[27] Kolman, E. and M. Margaliot (2005). Knowledge extraction from neural networks using the all-permutations fuzzy rule base.

[28] Kubat, M. (1998). Decision trees can initialize radial-basis function networks. *IEEE Transactions on Neural Networks* 9(5), 813–821.

[29] Kulluk, S., L. Özbakır, and A. Baykasoğlu (2013). Fuzzy difaconn-miner: A novel approach for fuzzy rule extraction from neural networks. *Expert Systems with Applications* 40(3), 938–946.

[30] Kurgan, L. A. and K. J. Cios (2003). Fast class-attribute interdependence maximization (caim) discretization algorithm. In *ICMLA*, pp. 30–36.

[31] Kurgan, L. A. and K. J. Cios (2004). Caim discretization algorithm. *IEEE transactions on Knowledge and Data Engineering* 16(2), 145–153.

[32] Lakshmi, K. and G. S. Kumar (2014). Association rule extraction from medical transcripts of diabetic patients. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pp. 201–206. IEEE.

[33] Liu, H. and R. Setiono (1995). Chi2: Feature selection and discretization of numeric attributes. In *Tools with artificial intelligence, 1995. proceedings., seventh international conference on*, pp. 388–391. IEEE.

[34] Lu, H., R. Setiono, and H. Liu (1996). Effective data mining using neural networks. *IEEE transactions on knowledge and data engineering* 8(6), 957–961.

[35] Malone, J., K. McGarry, S. Wermter, and C. Bowerman (2006). Data mining using rule extraction from kohonen self-organising maps. *Neural Computing & Applications* 15(1), 9–17.

[36] Markowska-Kaczmar, U. and W. Trelak (2005). Fuzzy logic and evolutionary algorithm—two techniques in rule extraction from neural networks. *Neurocomputing 63*, 359–379.

[37] McGarry, K. and J. Malone (2004). Analysis of rules discovered by the data mining process. In *Applications and Science in Soft Computing*, pp. 219–224. Springer.

[38] Molina, J. M., P. Isasi, A. Berlanga, and A. Sanchis (2000). Hydroelectric power plant management relying on neural networks and expert system integration. *Engineering Applications of Artificial Intelligence* 13(3), 357–369.

[39] Nayak, R. (1999). *GYAN: A methodology for rule extraction from artificial neural networks*. Ph.D. disseration , Queensland University of Technology.

[40] Núñez, H., C. Angulo, and A. Català (2002). Rule extraction from support vector machines. In *Esann*, pp. 107–112.

[41] Schmitz, G. P., C. Aldrich, and F. S. Gouws (1999). Ann-dt: an algorithm for extraction of decision trees from artificial neural networks. *IEEE Transactions on Neural Networks* 10(6), 1392–1401.

[42] Sethi, K. K., D. K. Mishra, and B. Mishra (2012). Kdruleex: A novel approach for enhancing user comprehensibility using rule extraction. In *Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on*, pp. 55–60. IEEE.

[43] Setiono, R. (1996). Extracting rules from pruned neural networks for breast cancer diagnosis. *Artificial Intelligence in Medicine* 8(1), 37–52.

[44] Setiono, R. (1997). A penalty-function approach for pruning feedforward neural networks. *Neural computation* 9(1), 185–204.

[45] Setiono, R. (2000). Extracting m-of-n rules from trained neural networks. *IEEE Transactions on Neural Networks* 11(2), 512–519.

[46] Setiono, R. and A. Gaweda (2000). Neural network pruning for function approximation. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, Volume 6, pp. 443–448. IEEE.

[47] Setiono, R. and W. K. Leow (2000). Fernn: An algorithm for fast extraction of rules from neural networks. *Applied Intelligence* 12(1-2), 15–25.

[48] Setiono, R., W. K. Leow, and J. M. Zurada (2002). Extraction of rules from artificial neural networks for nonlinear regression. *IEEE transactions on neural networks* 13(3), 564–577.

[49] Setiono, R. and H. Liu (1995). Understanding neural networks via rule extraction. In *IJCAI*, Volume 1, pp. 480–485.

[50] Setiono, R. and H. Liu (1997). Neurolinear: From neural networks to oblique decision rules. *Neurocomputing* 17(1), 1–24.

[51] Setiono, R. and M. Tanaka (2010). Neural network rule extraction and the led display recognition problem. In *2010 22nd International Conference on Tools with Artificial Intelligence*, pp. 53–56. IEEE.

[52] Shaabani, B. and H. Sajedi (2014). Mmdt: Multi-objective memetic rule learning from decision tree. *Journal of Computer & Robotics* 7(2), 37–46.

[53] Shahabi, M., H. Hassanpour, and H. Mashayekhi (2017). Rule extraction for fatty liver detection using neural networks. *Neural Computing and Applications*, 1–11.

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

[55] Tsai, C.-J., C.-I. Lee, and W.-P. Yang (2008). A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences* 178(3), 714–731.

[56] Visser, U., R. Nayak, and M. T. Wongy (1998). Rule extraction from trained neural networks & connectionist knowledge representation for the determination of pesticide mixtures. In *Proceedings of the Ninth Australian conference on neural networks*, pp. 138–142.

[57] Vora, S. V. and M. S. R. M. Tech (2012). Mcaim : Modified caim discretization algorithm for classification.

[58] Wang, L. and X. Fu (2005). A simple rule extraction method using a compact rbf neural network. In *International Symposium on Neural Networks*, pp. 682–687. Springer.

واژه‌نامه فارسی به انگلیسی

ابرفضا ..... Hyperspace
اختصاصی بودن ..... Specificity
استخراج قانون ..... Rule Extraction
استقراء دانش ..... Knowledge induction
اصلاح فرموله‌سازی ..... Reformulation
الگوریتم ژنتیک ..... Genetic algorithm
آموزشی ..... Pedagogical
انتخاب ..... Selection
انتخاب بولتزمن ..... Boltzmann Selection
انتخاب ترتیبي ..... Rank Selection
انتخاب چرخ رولت ..... Roulette wheel Selection
انتخاب حالت پایدار ..... Steady-State Selection
انتخاب رقابتی ..... Tournament Selection
بازنمائی ..... Representation
بایاس ..... Bias
بحرانی ..... Critical
برنامه کاربردی ..... Application
بولی ..... Boolean
بهره اطلاعاتی ..... Information gain
بی‌نظمی ..... Entropy
پالایش دانش ..... Knowledge refinement
پس انتشار ..... Backpropagation
پیش‌پردازش ..... Preprocessing

تابع برازندگی Fitness Function

تابع پایه شعاعی Radial Basis Function

تابع فعال‌ساز Activation Function

تجزیه‌ای Decompositional

تجمیع Aggregation

تخمین توابع Regression

تخمین‌گر غیرخطی Non-linear approximators

ترکیب CrossOver

ترکیب عطفی Conjunction

ترکیب فصلی Disjunction

ترکیبی Eclectic

جعبه سفید White Box

جعبه سیاه Black Box

چند جمله‌ای Oblique

چند لایه پرسپترون Multi-layer perceptron

جهش Mutation

حساسیت Sensitivity

خوشه‌بندی Clustering

داده‌کاوی Data mining

درخت تصمیم Decision tree

درجه تعلق Corresponding grade

دسته‌بندی Classification

دقت Accuracy

رگرسیون Regression

روی هم افتادگی Overlap

زیرفضا Subspace

سراسری Global

سیستم تصمیم‌گیری Decision system

شبکه‌عصبی مصنوعی Artificial neural network

طبقه‌بند Classifier

طبقه‌بندی Classification

غیرمتوازن Unbalance

فازی ..... Fuzzy
کارایی ..... Performance
کاربر پسند ..... User friendly
کاهش بعد ..... Dimensionality reduction
کروموزوم ..... Chromosome
کیفیت ..... Quality
گرادیان نزولی ..... Gradient descent
گزاره‌ای ..... Propositional
گسسته‌سازی ..... Discretization
ماشین بردار پشتیبان ..... Support vector machine
ماشین‌های حالت ..... State machines
مجازات ..... Penalty
محلی ..... Local
مدل‌سازی پیوندگرا ..... Connectionist modeling
مصالحه ..... Trade off
مفاهیم زبانی ..... Linguistic concepts
موجودیت ..... Entity
نخبه سالاري ..... Elitism
نرمال‌سازی ..... Normalization
نرون ..... Neuron
نگاشت خود سازمان یافته ..... Self-Organizing Map
نمایش بصری ..... Visual representation
نمونه‌برداری ..... Sampling
هرس‌سازی ..... Pruning
هفت بخشی ..... Seven segment
هوش مصنوعی ..... Artificial intelligence
یادگیری ماشین ..... Machine learning
یادگیري ماشین نمادین ..... Symbolic machine learning

واژه‌نامه انگلیسی به فارسی

دقت ..... Accuracy
تابع فعال‌ساز ..... Activation Function
تجمیع ..... Aggregation
برنامه کاربردی ..... Application
هوش مصنوعی ..... Artificial intelligence
شبکه‌عصبی مصنوعی ..... Artificial neural network
پس انتشار ..... Backpropagation
بایاس ..... Bias
جعبه سیاه ..... Black Box
انتخاب بولتزمن ..... Boltzmann Selection
بولی ..... Boolean
کروموزوم ..... Chromosome
طبقه‌بندی، دسته‌بندی ..... Classification
طبقه‌بند ..... Classifier
خوشه‌بندی ..... Clustering
ترکیب عطفی ..... Conjunction
بحرانی ..... Critical
ترکیب ..... CrossOver
مدل‌سازی پیوندگرا ..... Connectionist modeling
درجه تعلق ..... Corresponding grade
داده‌کاوی ..... Data mining
سیستم تصمیم‌گیری ..... Decision system
درخت تصمیم ..... Decision tree

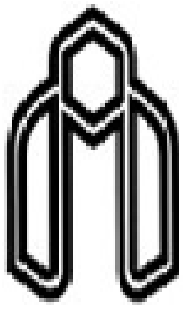
تجزیه‌ای ..... Decompositional
کاهش بعد ..... Dimensionality reduction
گسسته‌سازی ..... Discretization
ترکیب فصلی ..... Disjunction
ترکیبی ..... Eclectic
نخبه سالاري ..... Elitism
موجودیت ..... Entity
بی‌نظمی ..... Entropy
تابع برازندگی ..... Fitness Function
فازی ..... Fuzzy
الگوریتم ژنتیک ..... Genetic algorithm
سراسری ..... Global
گرادیان نزولی ..... Gradient descent
ابرفضا ..... Hyperspace
بهره اطلاعاتی ..... Information gain
استقراء دانش ..... Knowledge induction
پالایش دانش ..... Knowledge refinement
مفاهیم زبانی ..... Linguistic concepts
محلی ..... Local
یادگیری ماشین ..... Machine learning
چند لایه پرسپترون ..... Multi-layer perceptron
جهش ..... Mutation
نرون ..... Neuron
تخمین‌گر غیرخطی ..... Non-linear approximators
نرمال‌سازی ..... Normalization
چند جمله‌ای ..... Oblique
روی هم افتادگی ..... Overlap
آموزشی ..... Pedagogical
مجازات ..... Penalty
کارایی ..... Performance
پیش‌پردازش ..... Preprocessing
گزاره‌ای ..... Propositional

هرس‌سازی ..... Pruning
تابع پایه شعاعی ..... Radial Basis Function
اصلاح فرموله‌سازی ..... Reformulation
رگرسیون، تخمین توابع ..... Regression
استخراج قانون ..... Rule Extraction
انتخاب ترتیبي ..... Rank Selection
بازنمائی ..... Representation
انتخاب چرخ رولت ..... Roulette wheel Selection
نمونه‌برداری ..... Sampling
انتخاب ..... Selection
نگاشت خود سازمان یافته ..... Self-Organizing Map
حساسیت ..... Sensitivity
هفت بخشی ..... Seven segment
اختصاصی بودن ..... Specificity
ماشین‌های حالت ..... State machines
انتخاب حالت پایدار ..... Steady-State Selection
زیرفضا ..... Subspace
ماشین بردار پشتیبان ..... Support vector machine
سیستم‌های یادگیري ماشین نمادین ..... Symbolic machine learning
انتخاب رقابتی ..... Tournament Selection
مصالحه ..... Trade off
غیرمتوازن ..... Unbalance
کاربر پسند ..... User friendly
نمایش بصری ..... Visual representation
کیفیت ..... Quality
جعبه سفید ..... White Box
موجودیت ..... Entity

Aabstract

Nowadays with the growing volume of data, analyzing and discovering the relationships between them, has become very important in data mining science. Rule Extraction is one of the important applications for discovering knowledge and relationships between data. It is done by investigating the internal structure of the network, connections and the output of the hidden layer neurons. Neural networks, especially, have received a lot of attention because of their high accuracy and not requiring prior knowledge about data. Most algorithms in this field of study use decompositional approach and are dependent on pruning phase. However, pruning reduces the efficiency and accuracy of a network which lead to weakening the extracted rules. We proposed a new method to reduce pruning damage using genetic algorithm at two general and local levels. In this method, neural network weights are adjusted in a way, so that the damage caused by the pruning phase will be reduced significantly. The effectiveness of the proposed approach is clearly demonstrated by the experimental results on a set of standard data mining test problems. The accuracy of the rules extracted by our proposed method is improved about 2 percent on average.

Keywords: Rule Extraction, Genetic Algorithm, Artificial Neural Network, Discovering inter-data relationships, Neural Network Pruning, Knowledge discovery



Shahrood University of Technology

Faculty of Computer Engineering

M.Sc. Thesis in Artificial Intelligence Engineering

Improving accuracy of the extracted rules in multilayer neural networks using evolutionary algorithm in training and pruning the network

By: Yasser Irani

Supervisor

Dr. Hamid Hassanpour

February, 2019