ارزیابی سیستم‌های تحت وب به کمک شبکه‌های عصبی مصنوعی

سید رضا شاه امیری[۱] ، سید امیرحسن منجمی[۲] ، ناصر نعمت بخش[۳]
۱- کارشناس ارشدکامپیوتر، دانشگاه آزاد اسلامی واحد نجف آباد، reza@rsh.ir
۲- استادیار دانشکده مهندسی، دانشگاه اصفهان، monadjemi@eng.ui.ac.ir
۳- استادیار دانشکده مهندسی، دانشگاه اصفهان، nemat@eng.ui.ac.ir

چکیده

تست یک سیستم یکی از مراحل مهم چرخه حیاتی توسعه آن سیستم می‌باشد. فاز تست، مرحله‌ای است که قسمت‌های زیادی از آن باید به صورت دستی توسط عامل انسانی انجام پذیرد. بدین علت که احتمال وجود خطا در پروسه‌های انسانی زیاد است، قابلیت اطمینان نرم افزار در معرض تهدید قرار خواهد گرفت و احتمال تست نشده ماندن جنبه‌هایی از سیستم همواره وجود خواهد داشت. لذا خودکارسازی و هوشمند سازی هرچه بیشتر این پروسه، منجر به تولید سیستمی با هزینه کمتر و مطمئن‌تر می‌شود. در این مقاله ما روشی برای خودکارسازی فرآیند تست نوع خاصی از سیستم‌های یعنی نرم افزارهای تحت وب توسط شبکه های عصبی مصنوعی را ارائه خواهیم کرد که در آن شبکه عصبی مصنوعی توسط زوج مرتب‌های ورودی و خروجی معادل در سیستم آموزش دیده و بدین وسیله یک ابزار تست نرم افزار را مدلسازی می‌کند، سپس از این مدل برای ارزیابی و تست سیستم استفاده خواهد شد. ما نتیجه کار را به کمک یک شبکه عصبی مصنوعی بر روی نسخه تغییر یافته‌ای از یک نرم افزار ثبت نام و انتخاب واحد تحت وب که در آن خطاهایی را تزریق کرده ایم، ارزیابی خـواهیم کرد و مشخص خـواهیم نمود کـه می‌توان توسط شبکه های عصبی مصنوعی صحت نرم افزار را بررسی کرد.

واژه های کلیدی

ارزیابی خودکار سیستم، نرم افزارهای تحت وب، تست جعبه سیاه، شبکه های عصبی مصنوعی.

۱– مقدمه

هدف اصلی تست نرم افزار مشخص کردن این است که تا چه اندازه نرم افزار بر خصوصیاتی که از آن انتظار میرود منطبق است. دو رهیافت اصلی در تست نرم افزار وجود دارد : تست جعبه سفید[۱] و تست جعبه سیاه[۲] در تست جعبه سفید به بررسی و تحلیل کدهای برنامه می پردازیم و جزئیات و نحوه تولید خروجی از روی ورودی را مورد بررسی قرار می‌دهیم. برای اینکار به درک عمیقی از صورت مسئله و جزئیات پیاده‌سازی نیازمندیم. در تست جعبه سیاه بدون نیاز به کدها و جزئیات پیاده سازی و نحوه عملکرد درونی سیستم، فقط با در اختیار داشتن و بررسی کردن ورودیها و خروجیها صحت عملکرد سیستم مشخص می‌شود.

ارزیابی (تست) نرم افزار به سه مرحله تقسیم میشود : تولید داده های تست، ارائه داده ها به عنوان ورودی به نرم افزار و ارزیابی

نتایج در گذشته تست نرم افزار توسط یک تست کننده انسانی انجام می‌شد که موارد تست را انتخاب، اجرا، ونتایج را بررسی می‌کرد، ولی در نرم افزارهای بزرگ امروزی اینکار بسیار زمانبر و پرهزینه خواهد بود. خصوصا در نرم افزارهای مبتنی بر وب که سرعت تولید[۳] در آنها بسیار مهم تر از نرم فزارهای دیگر است، عملا امکان تست انسانی به نحو مطلوب غیر ممکن است. زیرا انتخاب تمامی موارد ممکن ترکیبات تست و بررسی اجرای آنها بسیار زمانبر و پر هزینه خواهد بود. بدین منظور متخصصین نرم افزار، سعی در ارائه راهکارهای مختلفی نموده اند تا پروسه تست را خودکار و هوشمند سازند تا بدین وسیله نقش انسانی و در پی آن زمان و هزینه تست را به حداقل و قابلیت اطمینان سیستم را به حداکثر برسانند. در این بین از میان تکنیک‌های هوش مصنوعی، ایده استفاده از شبکه‌های عصبی مصنوعی تا حدی مطرح بوده است. به عنوان نمونه در [۱] مایر- هاورز و همکاران روشی ارائه کرده‌اند تا با استفاده از شبکه‌های عصبی مصنوعی مجموعه کل موارد تست بررسی شده تا مناسب ترین آنها که می تواند بیشترین خطاها را مشخص کند انتخاب شود. در [۲] خوش‌گفتار و دیگران، از شبکه‌های عصبی به عنوان ابزاری برای تحقیق و بررسی خطاهای درون یک نرم افزار استفاده شده است تا بتوان برآوردی از تعداد خطاهای درون نرم افزار ارائه داد. در [۳] ایدری و همکاران، از خاصیت جعبه سیاه مانند شبکه ها عصبی برای ارزیابی هزینه های ساخت نرم افزار کمک گرفته‌اند. در [۴] وان مالی و همکاران از شبکه های عصبی به عنوان مدلی از نرم افزار برای تشخیص خطا در نسخه‌های جدید آن نرم افزار استفاده شده است.

در این تحقیق، ما کاربرد دیگری از شبکه های عصبی را در پروسه تست ارائه کرده، این مبحث را در نرم افزارهای تحت وب[۴] که از انواع مهم نرم افزارهای فعلی هستند مورد بررسی قرار می‌دهیم و حاصل را با یک نرم افزار ثبت نام و انتخاب واحد تحت وب ارزیابی می‌کنیم. برای اینکار ما از یک شبکه عصبی مصنوعی سه لایه برای مدل سازی ماژولهایی از نرم افزار که قابلیت مدل شدن توسط شبکه عصبی را دارند، استفاده کرده‌ایم. ما توسط قوانین حرفه ای[۵]، که ماژول تحت تست وظیفه پیاده‌سازی آنها را به عهده دارد، شبکه عصبی را با زوج مرتبهای ورودی- خروجی تعلیم داده و از این شبکه آموزش دیده برای تست این ماژول استفاده کرده‌ایم. بدین صورت که یک ورودی خاص، هم در اختیار ماژول نرم افزاری و هم در اختیار شبکه عصبی آموزش دیده قرار داده شده و خروجی تولید شده توسط این دو با هم مقایسه می‌شوند. خواهیم دید که خروجی شبکه عصبی خروجی صحیح می‌باشد و می‌توان از آن به عنوان معیاری برای تشخیص خطای ماژول نرم افزاری استفاده کرد. اینکار توسط یک ابزار مقایسه‌گر انجام شده و هرگونه تفاوتی میان خروجی این دو به عنوان یک خطا در نرم افزار تحت وب گزارش خواهد شد. در حقیقت ما مدل جدیدی از تستهای جعبه سیاه را معرفی کرده ایم که در شکل(۱) نمای کلی این متد نمایش داده شده است.

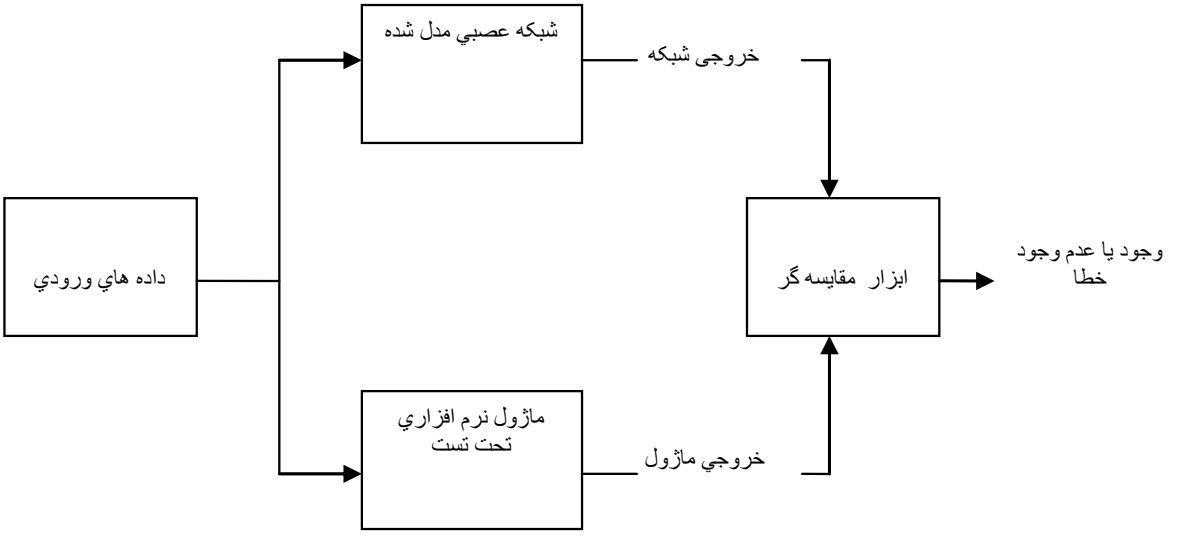
استفاده از مدل شبکه های عصبی برای ارزیابی نرم افزار مزایای زیادی را در پی دارد. این‌کار باعث صرفه جویی زیادی در زمان و هزینه تست خواهد شد و همچنین می‌تواند پروسه تست را ساده‌تر نماید. نشان خواهیم داد که این ایده در ارزیابی صحیح قوانین حرفه‌ای که برای پیاده‌سازی آنها نیاز به ساختارهای پیچیده تصمیم گیری می‌باشد موثر است. زیرا خطایابی در ساختارهای تصمیم گیری بزرگ و تو در تو عملی بس دشوار و پر مخاطره خواهد بود که منجر به به خطر افتادن بالقوه قابلیت اطمینان نرم افزار خواهد شد. در نتیجه استفاده از شبکه های عصبی منجر به تولید نرم افزار مطمئن تری خواهد گردید.

در ادامه این مقاله، در بخش ۲، پیش زمینه لازم در مورد شبکه های عصبی به اختصار بیان خواهد شد. در بخش ۳، نرم افزارهای تحت وب و خصوصیاتی که آنها را از بقیه نرم افزارها متفاوت می‌کند بیان می‌شود . در بخش ۴، نرم افزار مورد تست و قوانین حرفه‌ای مورد نظر معرفی می‌شوند. بخش ۵، متدولوژی تست و نحوه انجام آن را تشریح کرده، نهایتا در بخش ۶ نیز نتایج کار مورد بررسی و جمع‌بندی قرار خواهند گرفت.

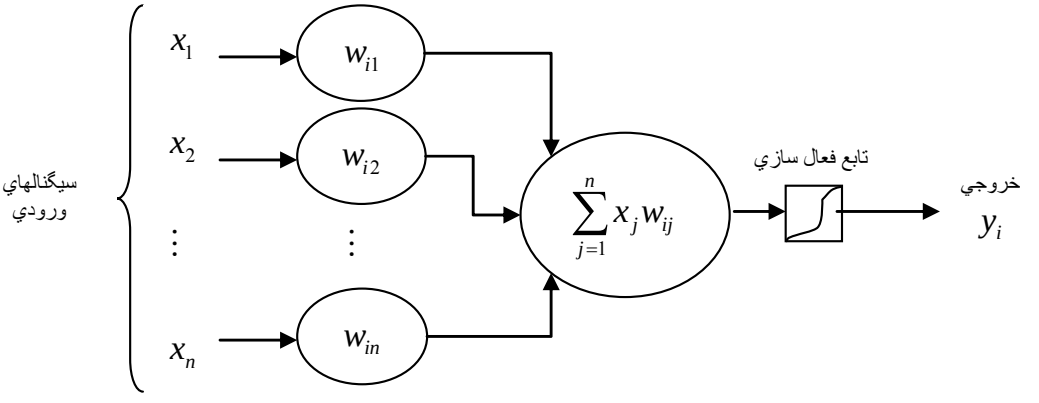
۲- شبکه های عصبی مصنوعی

شبکه‌های عصبی مصنوعی برای شبیه سازی توانائی پردازش اطلاعات ساختار مغز انسان طراحی شده اند. در معماری یک شبکه عصبی - واحد های محاسبه گر ساده که مشابه نرونهای مغزی انسان می‌باشند – به کار می‌روند. هر شبکه عصبی از یک یا چند لایه از این نرونها (واحدهای محاسبه‌گر) و اتصالات داخلی مابین آنها که هرکدام یک وزن سیناپسی[۶] دارند، تشکیل شده است. هر نرون در شبکه برای انجام برخی از عملیات محاسبه بر اساس آنچه در یک فرآیند آموزش یا یادگیری[۷] مشخص می‌شود، به کار می‌رود. در وزنهای سیناپسی، اطلاعاتی که برای انجام عملیات محاسبه هر نرون لازم هستند ذخیره می‌شوند. مقادیر این اوزان نیز در فرآیند یادگیری تعیین می‌شوند. بنابراین شبکه عصبی یک سیستم موازی پردازش اطلاعات با کنترل توزیع شده می‌باشد که میتواند محیط پیرامون خویش را درک کرده و آنرا ذخیره کند.

دو نکته اساسی در شبکه های عصبی، توانایی پردازش موازی و توزیع شده و امکان برون‌یابی اطلاعات ذخیره شده برای تولید خروجیهایی که در طی فرآیند یادگیری به شبکه معرفی نشده‌اند می‌باشد. این دو خصیصه تا حدودی امکان حل مسائل پیچیده را به شبکه‌های عصبی می‌دهند.



شکل۱- شمای کلی سیستم تست به کمک شبکه عصبی مصنوعی



شکل۲ – ساختار یک نرون در شبکه عصبی

در متد ما، از شبکه های عصبی برای تولید خروجی مشخص به ازاء سیگنال ورودی استفاده شده است. در حین مرحله یادگیری، اوزان سیناپسی شبکه بر اساس نمونه های مجموعه یادگیری بروز رسانی و تنظیم می‌شوند. این مجموعه یادگیری شامل بردارهای ورودی و خروجی متناظر می‌باشد. یک دوره تعلیم، زمانی به پایان می‌رسد که کلیه مجموعه یادگیری یکبار در اختیار شبکه قرار گیرند. بنابراین یک شبکه عصبی به صورت یک ماشین نگاشت ورودی به خروجی تعبیر می‌شود که توانایی تعمیم به نمونه های جدید را نیز دارد.

هر نرون در شبکه عصبی از سه قسمت تشکیل شده است : مجموعه‌ای از n وزن سیناپسی ،به ازاء هر اتصال یکی ; یک جمع کننده برای ادغام سیگنال ورودی با اوزان سیناپسی و یک تابع فعالسازی یا تابع فشرده سازی[۸] که وظیفه محدود کردن خروجی ما بین مقادیری خاص را بعهده دارد. در شکل(۲) ساختار هر نرون نشان داده شده است.

اگر سیگنال ورودی به نرون i برابر x_i باشد، وزن سیناپسی مربوط به اتصال ما بین سیگنال ورودی و نرون برابر w_{ij} می‌باشد. مجموعه سیگنال ورودی در مجموعه اوزان ضرب می شود. حاصل برای محاسبه خروجی به تابع فعال‌سازی وارد می‌شود.

$$y_i = F(net_i) \quad net = \sum_{j=1}^{n} x_j w_{ij} \qquad (۱)$$

یک شبکه عصبی چند لایه جلوروندە[۹] شامل یک لایه ورودی غیر محاسبه گر (به ازاء هر ورودی یکی) و یک یا چند لایه پنهان

از نرونهای محاسبه‌گر و یک لایه از نرونهای خروجی محاسبه‌گر می‌باشد. انتشار عقب[۱۰] یکی از الگوریتم‌های استاندارد یادگیری شبکه های عصبی چند لایه جلورونده می‌باشد. این الگوریتم شامل انتشار به جلو سیگنال ورودی و انتشار به عقب بردار خطا می‌باشد که در طی آن اوزان سیناپسی به تدریج تنظیم می‌شود تا به سطح معینی از خطا برسیم. پس از آن شبکه عصبی آموزش دیده و آماده تولید خروجی می‌باشد. در حقیقت الگوریتم انتشار به عقب یک روال جستجوی گرادیان کاهنده است که به دنبال تعیین اوزان به صورتی است که حداقل اختلاف را ما بین خروجی‌های فعلی شبکه و خروجی‌های مورد انتظار ایجاد کند. مفاهیم کامل شبکه های عصبی مصوعی در [۸،۷] تشریح شده است .

۳- نرم افزارهای تحت وب

امروزه نرم افزارها تفاوت‌های زیادی با گذشته کرده اند. خصوصا با گسترش وب، نرم افزارهایی که تحت شبکه اینترنت اجرا شوند بسیار شایع شده‌اند. به علت خواص و ویژگی‌های شبکه مانند و تحت وب بودن، این گونه از نرم افزارها با نرم افزارهای سنتی تفاوت‌های بسیاری دارند. در [۵]، کاپل و دیگران نرم افزارهای تحت وب را اینگونه تعریف کرده اند:

یک نرم افزار تحت وب، سیستمی نرم افزاری است که مبتنی بر تکنولوژی‌ها و استانداردهای کنسرسیوم World Wide Web(W3C) عمل می‌کند و منابع خاصی از وب را همانند محتوا و سرویس‌ها، توسط یک واسط کاربری بنام مرورگر وب[۱۱]، در اختیارکاربران قرار می‌دهد.

علاوه بر اینکه نرم افزارهای تحت وب توسط یک واسط کاربری بنام مرورگر قابل دستیابی هستند، خصوصیات مختلف دیگری نیز مطرح است که مهندسی نرم افزار را در آنها خاص می‌کند. به عنوان مثال در [۶] ، پرسمن بیان می کند که خاصیت شبکه‌ای بودن، دسترسی همزمان غیر قابل پیش بینی، قابلیت دسترسی کنترل شده و توزیع شده، حساسیت به محتوا، امنیت و زیبایی، فاکتورهایی هستند که هرچند در نرم افزارهای سنتی هم مطرح هستند، اما در نرم افزارهای تحت وب از اهمیت بیشتری برخوردارند.

تست نرم افزارهای تحت وب نیز تفاوت های زیادی با دیگر نرم افزارها دارد. به عنوان نمونه خیلی از خطاها در اثر تنظیمات مختلف مرورگرهای سطح سرویس گیرنده[۱۲] رخ می‌دهند که عموما توسط تست کننده قابل بررسی نیستند. همچنین تنوع طرح‌های زیر بنای[۱۳] سخت افزاری مختلفی که ممکن است نرم افزار در آن اجرا شود نیز مسئله مهمی است. مثلا ممکن است کاربر به جای PC از یک دستگاه PDA برای کار با نرم افزار استفاده کند. لذا در تست اینگونه نرم افزارها باید فاکتورهای مختلفی را رعایت کرد.

امروزه به علت وجود تکنولوژی‌های جدید و قدرتمند نرم افزاری و سخت افزاری، امکان ساخت نرم افزارهای تحت وب بسیار پیچیده و قدرتمند فراهم شده است که می‌توانند همانند هر نرم افزاری از پایگاه داده خاص خود استفاده کنند و قوانین منطقی و تجاری پیچیده را پیاده‌سازی کنند. در این تحقیق تمرکز ما برروی خودکارسازی ارزیابی و تست قوانین منطقی نرم افزارهای تحت وب می‌باشد.

۴- توصیف نرم افزار مورد تست

نرم افزار مورد نظر ما، یک سیستم انتخاب واحد دانشگاهی تحت وب می‌باشد که هدف بررسی صحت اعتبار عملیات انتخاب واحد، بر اساس قوانین انتخاب واحد یک دانشگاه می‌باشد. شبکه عصبی و نرم افزار مورد تست هر دو قوانین انتخاب واحد را پیاده سازی می‌کنند و بر اساس آن مشخص می‌کنند که آیا بر اساس اطلاعات دانشجو، انتخاب واحد دانشجو صحیح می‌باشد یا خیر. این کار بر اساس اطلاعات مختلف دانشجو (همانند تعداد واحدهای درخواستی، تعداد واحدهای گذرانده و ...) که به صورت خودکار در سیستم ذخیره می‌شوند انجام خواهد شد. برای این‌کار یک پروسه تصمیم‌گیری پیچیده مورد نیاز است که احتمال اشتباه در آن نیز بالا می‌باشد.

۱ـ در صورتیکه دانشجو سنوات تحصیلی نداشته باشد در هیچ صورتی قادر به انتخاب واحد نیست.
۲ـ در صورتیکه دانشجو ترم آخر نباشد، در ترم تابستان می تواند حداکثر تا ۷ واحد با هر معدلی اخذ کند.
۳ـ در ترم معمولی(غیر تابستان) در صورتیکه دانشجو ترم آخر نباشد :
A: اگر معدل دانشجو کمتر از ۱۲ است می تواند حداکثر تا ۱۴ واحد اخذ کند.
B: اگر معدل دانشجو مابین ۱۲ تا ۱۷ است، دانشجو می تواند حداکثرتا ۲۰ واحد اخذ کند.
C: اگر معدل دانشجو بالای ۱۷ بود، دانشجو می تواند تا حداکثر ۲۷ واحد اخذ کند.
۴ـ در صورتی که دانشجو ترم آخر باشد:
A: در ترم معمولی حداکثر می تواند ۲۷ واحد با هر معدل اخذ کند.
B: در ترم تابستانی می تواند حداکثر تا ۱۴ واحد با هر معدل اخذ کند.
۵ـ در صورتیکه دانشجو از حد مجاز مشروطی تجاوز کرده باشد در هیچ صورتی مجاز به انتخاب واحد نیست.

شکل ۳ - قوانین انتخاب واحد

در شکل(۳) الگوریتمی که برای پیاده سازی قوانین موجود در سیستم آموزش (قوانین حرفه ای) که در حقیقت شبکه عصبی مدل کننده آنها است، نشان داده شده است .

برای بررسی قوانین مطرح شده در شکل(۳) به فیلدهای زیر نیاز

جدول ۱- دسته بندی اطلاعات

نام پارامتر	حوزه ها و مقادیر	توضیح
mustGo	۱	تعداد دفعات مجاز مشروطی دانشجو به پایان رسیده است
	۰	تعداد دفعات مجاز مشروطی دانشجو به پایان نرسیده است
sanavat	۰	سنوات تحصیلی مجاز دانشجو به پایان رسیده است
	۱	سنوات تحصیلی مجاز دانشجو به پایان نرسیده است
lastTerm	۰	دانشجو، دانشجوی ترم آخر نیست
	۱	دانشجو، دانشجوی ترم آخر است
term	۰	عملیات انتخاب واحد برای ترم تابستان است
	۱	عملیات انتخاب واحد برای ترم غیر تابستان است
lastTermAvg	۰	معدل ترم گذشته دانشجو کمتر از ۱۲ می‌باشد
	۱	معدل ترم گذشته دانشجو مابین ۱۲ تا ۱۷ می‌باشد
	۲	معدل ترم گذشته دانشجو بیش از ۱۷ می‌باشد
totalGetCourse	۰	تعداد واحدهای درخواست شده توسط دانشجو کمتر از ۱۴ واحد می‌باشد
	۱	تعداد واحدهای درخواست شده توسط دانشجو مابین ۱۴ تا ۲۰ واحد می‌باشد
	۲	تعداد واحدهای درخواست شده توسط دانشجو مابین ۲۰ تا ۲۷ واحد می‌باشد
	۳	تعداد واحدهای درخواست شده توسط دانشجو کمتر از ۷ واحد می‌باشد
	۴	تعداد واحدهای درخواست شده توسط دانشجو بیش از ۲۷ واحد می‌باشد

است:

الف ـ جمع کل تعداد واحدهای گذرانده شده برای مشخص کردن دانشجوی ترم آخر (lastTerm)

ب ـ تعداد ترمهای مشروطی دانشجو برای مشخص کردن اینکه آیا دانشجو مجاز به انتخاب واحد است یا خیر (mustGo)

ج ـ معدل ترم قبل (lastTermAvg)

دـ نیمسال تحصیلی معمولی یا تابستانی (term)

هـ ـ تعداد ترمهای ثبت نام شده دانشجو برای بررسی سنوات تحصیلی (sanavat)

ی ـ تعداد واحدهای درخواست شده دانشجو برای انتخاب واحد ترم جاری (totalGetCourse)

بدین منظور که بتوان از اطلاعات دانشجو به منظور تصمیم گیری و همچنین آموزش شبکه استفاده کرد، در مرحله اول ما آنها را کوانتیزه[14] کرده و آنها را به بازه‌هایی عددی دسته بندی کردیم. جدول(۱) این دسته بندی را نشان می‌دهد. تمامی این مقادیر به صورت خودکار در نرم افزار مورد تست تولید شده، دسته بندی شده و به ماژول تست کننده ارسال می‌شود. در شکل(۴) پیاده‌سازی قوانین آموزشی نشان داده شده است.

۵- توصیف متودولوژی تست

همانگونه که گفته شد روش تست ارائه شده نوعی تست جعبه سیاه می‌باشد. در فاز یادگیری شبکه عصبی، کلیه ورودی‌ها و خروجی‌های دسته‌بندی شده متناظر با موارد تست، در اختیار شبکه عصبی قرار می‌گیرند. پس از فاز یادگیری، شبکه عصبی توانایی تولید خروجی از ورودی متناظر را خواهد داشت و می‌توان از این خروجی برای بررسی صحت خروجی نرم افزار استفاده کرد. به عبارتی دیگر، ما ماژول بررسی کننده قوانین حرفه‌ای را توسط شبکه عصبی مدلسازی کرده ایم.

در جدول(۲) نمونه‌ای از زوج مرتب‌های ورودی– خروجی که برای آموزش شبکه استفاده می‌شود، نشان داده شده است.

شبکه عصبی استفاده شده در این تحقیق، یک شبکه عصبی پرسپترون[15] با شش نرون در لایه ورودی، سه نرون در لایه پنهان و یک نرون در لایه خروجی با تابع فشرده‌سازی سیگموید می‌باشد. برای آموزش شبکه عصبی از ۲۳۲ زوج مرتب ورودی/خروجی به عنوان داده‌های تست نرم افزار استفاده شده است. خروجی **صفر** شبکه به معنی عدم تایید صحت عملیات انتخاب واحد، و **یک** به معنی تایید عملیات انتخاب واحد می‌باشد.

در زمان فراخوانی ماژول تست، اطلاعات دسته بندی شده دانشجو به ماژول تست کننده و شبکه عصبی آموزش دیده فرستاده می‌شود. ماژول تست کننده اطلاعات ورودی را دریافت و خروجی صفر یا یک احتمالا خطاداری را تولید می‌کند. همچنین شبکه عصبی نیز خروجی صحیح را تولید می‌کند. هر دو خروجی به ابزار مقایسه‌گر فرستاده شده و ابزار مقایسه‌گر هر تفاوتی مابین این دو خروجی را به عنوان بروز یک خطا در ماژول تست کننده اعلام

جدول ۲ – نمونه ای از الگوهایی که برای آموزش شبکه عصبی استفاده شده است

must Go	Sanavat	Last Term	Term	Last Term Avg	Total Get course	Output
0	0	0	0	2	4	0
0	1	0	0	0	3	1
0	1	0	1	0	3	1
0	1	1	1	1	0	1
1	1	1	1	2	3	0
1	1	1	1	2	4	0
0	1	1	1	2	4	0

می‌دارد.

به منظور بررسی صحت عملکرد روش ارائه شده، ما خطاهایی را در ماژول تست کننده تزریق کردیم که این خطاهای تزریق شده از انواع متداول در برنامه نویسی می‌باشند. به عنوان نمونه استفاده اشتباه از یک and بجای or در یک ساختار تصمیم‌گیری بسیار متداول است. یا در قسمت‌هایی که باید مقایسه ای صورت بگیرد. جابجا شدن یک صفر یا یک نیز بسیار محتمل است. در جدول(۳) نمونه هایی از این خطاها نشان داده شده‌اند که در این آزمون به کار رفته‌اند.

همانطور که در جدول(۳) مشاهده می‌شود، خطاهای تزریق شده تمامی قوانین حرفه‌ای را در بر می‌گیرند. به منظور خوانایی بیشتر جدول، کد صحیح، خطای تزریق شده در کد، شماره خط کد در شکل(۴) و خروجی‌های ماژول تست کننده، شبکه عصبی و ابزار مقایسه‌گر به ازاء نمونه‌های تست ارائه شده در این جدول‌ها نشان داده شده‌اند. به این ترتیب صحت عملکرد شبکه عصبی تشریح شده است.

در ساختارهای پیچیده تصمیم‌گیری تمامی این خطاها بسیار اتفاق افتاده و علاوه بر اینکه تشخیص آنها دشوار است، رفع این خطاها امری بس دشوارتر تلقی می‌شود که نیازمند فهم دقیق مسئله و تبحر کافی در برنامه‌سازی و تکنولوژی مورد استفاده می‌باشد.

۶– جمع‌بندی و نتیجه‌گیری

بطور کلی قوانین حرفه‌ای در ماژول مورد نظری که الگوهای یادگیری شبکه عصبی بر اساس آن ایجاد می‌شوند، باید شرایط خاصی را دارا باشند تا مدلسازی آن توسط شبکه عصبی با حداکثر صحت انجام شده و در نتیجه عملیات خطایابی به خوبی انجام شود. در این قوانین حرفه ای باید نحوه تولید خروجی‌ها از ورودی‌ها کاملا واضح و مشخص باشد و تناقضی در آنها وجود نداشته باشد . همچنین باید بتوان وروردیها و خروجیهای متناظر آنها را بر اساس قوانین مذکور ایجاد کرده و از حالات مختلف خروجی و ورودی که نشان دهنده قوانین مختلف می‌باشند در روند یادگیری شبکه عصبی استفاده کرد. این تحقیق نشان می‌دهد که استفاده از این روش در ماژولهای نرم افزاری که خروجی آنها توسط ساختارهای تصمیم گیری هرچند پیچیده ولی خوش تعریف، تولید می‌شوند بسیار مناسب و قابل اعتماد می‌باشد.

در تجربیات ما روش ارائه شده به دو دلیل توانست به دقت صد در صد برسد :

۱ – ماژول تحت تست و قوانین حرفه‌ای انتخاب شده تمامی شرایط بالا را به خوبی دارا بوده و قوانین تولید خروجی‌ها بر اساس ورودی‌ها به خوبی مشخص شده و هیچ‌گونه عدم قطعیت و تضادی در آنها وجود نداشت.

۲ – در روند یادگیری و مدلسازی شبکه عصبی تمامی حالات ورودی و خروجی متناظر از آنها بطور کامل و دقیق به عنوان الگوهای یادگیری در اختیار شبکه عصبی گذاشته شده و تمامی قوانین حرفه ای از هر لحاظ کاملا پوشش دهی شده‌اند.

بنابراین می‌توان توسط شبکه‌های عصبی مصنوعی ماژول‌هایی از نرم افزار را که خصوصیات گفته شده را داشته باشند، بدون درگیر شدن با جزئیات و قوانین پیاده‌سازی، به صورت یک مدل جعبه سیاه به صورت کاملا خودکار تست و ارزیابی کرد. به این ترتیب زمان و هزینه تست کاهش می یابد. با این حال محصول حاصل، به مراتب از قابلیت اطمینان بالاتری برخوردار خواهد بود. زیرا امکان بروز اشتباهات انسانی و نقش انسانی بسیار کاهش یافته و همچنین موارد تست بسیار زیادتری به طور خودکار قابل بررسی خواهند بود.

به علت خاصیت عدم قطعیتی که در شبکه‌های عصبی موجود است، استفاده از این روش در همه حالات و به ازاء هرنوع قانون تجاری ممکن است نتیجه مطلوبی را ارائه نداده و شبکه عصبی با چند درصد خطا در مدلسازی روبرو شود. لذا برای مدلسازی و تست ماژولهای نرم افزاری توسط شبکه های عصبی باید در انتخاب آن ماژول دقت به خرج داد. معایب و محدودیت‌هایی نیز در این روش وجود دارد : اول اینکه این روش در تمامی ماژول‌های یک نرم افزار قابل اعمال نیست و باید شرایط خاص بالا در آنها وجود داشته باشد. دوم، نیاز به دانش شبکه‌های عصبی علاوه بر دانش‌های مورد نیاز ساخت نرم افزار می‌باشد. در نهایت نیز افزایش هزینه سرباری که برای مدلسازی شبکه عصبی نیاز است. البته این هزینه سرباری با در

جدول ۳- نمونه ای از خطاهای تزریق شده به ماژول تست کننده و مقایسه خروجی آن با شبکه عصبی

شماره خطا	شماره خط	کد صحیح	خطای تزریق شده	تاثیر در قانون تجاری	نمونه ای از موارد تست بررسی شده	خروجی ماژول تست (خروجی احتمالا نادرست)	خروجی شبکه عصبی (خروجی صحیح)	خروجی ابزار مقایسه گر
1	5	if (term==0)	if (term==1)	2، 3	0-1-0-0-0-0	1	0	خطا
2					0-1-0-0-1-0	1	0	خطا
3					0-1-1-0-1-0	0	1	خطا
4					0-1-1-0-0-2	0	0	عدم خطا
5	13	(totalGetCourse==0) || (totalgetCourse==3)	(totalGetCourse==0) && (totalgetCourse==3)	3	0-1-0-1-0-0	0	1	خطا
6					0-1-0-1-1-4	0	0	عدم خطا
7	19	(totalGetCourse==0) || (totalgetCourse==3)	(totalGetCourse==2) || (totalGetcourse==3)	3.B	0-1-0-1-1-2	1	0	خطا
8					0-1-0-1-1-0	0	1	خطا
9	25	if (totalgetCourse!=4)	if (totalgetCourse==4)	3.C	0-1-0-1-2-4	1	0	خطا
10	32	if (totalgetCourse!=4)	if (totalgetCourse!=2)	4.A	0-1-1--1-2-2	0	1	خطا
11					0-1-1-1-2-4	1	0	خطا
12	2	if (sanavat==1)	if (sanavat==0)	1	0-1-1-1-2-0	0	1	خطا
13					0-1-0-1-1-3	0	1	خطا
14					0-0-0-1-2-1	1	0	خطا
15	1	if (mustGo==0)	if(mustGo !=0)	5	0-1-0-1-1-0	0	1	خطا
16					0-1-1-1-1-2	0	1	خطا

نظر گرفتن حاصل کار (قابلیت اطمینان بالاتر، هزینه کمتر اجرای تست‌ها و کاهش نقش انسانی) به میزان قابل توجه‌ای قابل صرفنظر است. ایده دیگری که باید در مقوله‌ای جداگانه بررسی شود این است که اصولاً به جای استفاده از یک ماژول نرم افزاری که قوانین حرفه‌ای را پیاده‌سازی کند، از یک شبکه عصبی که مدل کننده قوانین مذکور است در نرم افزار نهایی استفاده شود. این کار می‌تواند باعث حذف پروسه تست ماژول مربوطه شده و نرم افزاری بهتر و احتمالاً کم هزینه تر و مطئن تر ارائه دهد. اثبات این فرضیه نیاز به تحقیق و بررسی بیشتری خواهد داشت. همچنین استفاده از شبکه های عصبی در تست نرم افزارهایی که ماهیتا رفتار آنها غیر مشخص و تصادف[۱۶] است و حالت تعریف شده قطعی ندارند قابل توصیه خواهد بود.

۷- مراجع

[1] C. Anderson, A. Mayrhauser, and r. Mraz, **" On the Use of Neural Networks to Guide Software Testing Activities"**, Proceedings of ITC'95,the International Test Conference, 21 Oct, 1995.

[2] M. T. Khoshgoftar, Pandya, S. Abhijit and Hemant More, **"A Neural Network approach for predicting software faults",** Proceedings of third international symposium on Software Reliability engineering, 1992.

[3] A. Idri, M. T. Khoshgoftar and A. Abra, **"Can Neural Networks be easyli interpreted in software cost estimating?"**, 2002 World Congress on Computational Intelligence Honolulu, Hawaii, 12 May, 2002.

[4] M. Vanmali, M. Last and A. Kandel, **"Using a Neural Network in the Software Testing**

Process", International Journal of Intelligent Systems, 17, 1, 45-62, 2002.
[5] G. Kappel, B. Pröll, S. Reich and W. Retschitzegger, **"Web Engineering: The Discipline of Systematic Development of Web Applications",** John Wiley & Sons Inc, 2006.
[6] R. J. Pressman, **Software Engineering :A Practitioners Approach. Sixth Edition.** McGrawHill, 2005.
[7] L. Fausett, **Fundamentals of Neural Networks : Architecture, Algorithms and Applications**. Prentice Hall, 1994.
[8] Robert Schalkoff, **Artificial Neural Networks,** McGraw-Hill, 1997.

۸– پی‌نوشتها

1- White – Box testing
2- Black – Box testing
3- Immediacy
4- Web Browser
5- Business Rules یا Application logic
6- Synaptic weight
7- Training
8- Squashing
9- Feedforward
10- Backpropagation
11- Web Browser
12- Client
13- Platform
14- Quantization
15- Perseptron
16- Stochastic