Archive of SID

طراحی و آموزش شبکه‌های عصبی مصنوعی به‌وسیلهٔ استراتژی تکاملی با جمعیت‌های موازی

فردین احمدی‌زر[1]، خه‌بات سلطانیان[2] و فردین اخلاقیان‌طاب[3]

[1] گروه مهندسی صنایع، دانشکدهٔ مهندسی، دانشگاه کردستان، کردستان، ایران

[2و3] گروه مهندسی نرم‌افزار و فناوری اطلاعات، دانشکدهٔ مهندسی، دانشگاه کردستان، کردستان، ایران

چکیده

کاربرد شبکه‌های عصبی مصنوعی در حوزه‌هایی از قبیل دسته‌بندی تصاویر و سیگنال‌های صوتی، مؤید توانایی این ابزار قدرتمند هوش مصنوعی در حل مسائل دنیای امروز است. طراحی و آموزش شبکه‌های عصبی همواره یک فرآیند زمان‌بر و مشکل بوده است. یک مدل عصبی مناسب باید بتواند الگوی داده‌های آموزشی را فراگرفته و نیز قابلیت تعمیم داشته باشد. در این مقاله، از جمعیت‌های موازی برای طراحی معماری شبکهٔ عصبی و همچنین از استراتژی تکاملی برای آموزش آن استفاده شده است، به‌طوری‌که در هر جمعیت شبکه‌ای با معماری خاصی تکامل می‌یابد. به کمک یک روش انتخاب دومعیاره مبتنی بر میزان خطا و پیچیدگی شبکه‌ها، الگوریتم ارائه شده قادر است شبکه‌های ساده با قابلیت تعمیم بالا تولید کند. برای ارزیابی کارایی الگوریتم پیشنهادی از هفت مسأله استاندارد دسته‌بندی استفاده شده است. روش ارائه‌شده با روش‌های تکامل اوزان، تکامل معماری و نیز الگوریتم‌های تکامل هم‌زمان معماری و اوزان مورد مقایسه قرار گرفته است. نتایج آزمایش‌ها کارایی و پایداری این روش را نسبت به روش‌های مورد مقایسه نشان می‌دهد. در این مقاله، همچنین تأثیر وجود جمعیت‌های موازی، روش انتخاب دومعیاره و نیز عملگر ادغام در الگوریتم ارائه‌شده مورد بررسی قرار گرفته است. از مزایای اصلی این روش بهره‌گیری از پردازش موازی به‌وسیلهٔ جمعیت‌های مستقل است.

واژگان کلیدی: شبکه‌های عصبی مصنوعی، استراتژی تکاملی، جمعیت‌های موازی.

۱- مقدّمه

شبکه‌های عصبی مصنوعی یکی از ابزارهای بسیار پرکاربرد هوش مصنوعی است. با رشد روزافزون اندازهٔ مسائل دنیای واقعی، شبکه‌های عصبی به‌عنوان یک ابزار مدل‌سازی می‌تواند در حل مسائل مختلفی از قبیل مسائل پیچیده مهندسی و مدیریتی مورد استفاده واقع شود. از جمله کاربردهای مدل‌های عصبی می‌توان به پیش‌بینی، دسته‌بندی و کنترل هوشمند اشاره کرد. شبکه‌های عصبی مصنوعی درواقع براساس مدلی بسیار ساده از ساختار و کارکرد مغز انسان شکل گرفته است (هیکین، ۱۹۹۴)؛ به‌گونه‌ای که از شبکه‌های عصبی مصنوعی برای پژوهش در زمینهٔ نحوهٔ کارکرد واقعی سامانه عصبی مغز نیز بهره گرفته می‌شود (سعیدی و توحیدخواه، ۱۳۸۸).

طراحی معماری یک شبکهٔ عصبی و نحوهٔ آموزش آن در کارایی شبکه تأثیر بسیار زیادی دارد و از این‌رو تکنیک‌های مختلفی در این زمینه ارائه و استفاده شده‌اند. طرح معماری شبکه با تعداد لایه‌های آن، تعداد نرون‌های هر لایه، نحوه ارتباط این نرون‌ها و توابع گذار مشخص می‌شود. همچنین وزن‌های اعصاب مصنوعی شبکه با آموزش تخصیص می‌یابند به‌طوری‌که شبکه، قدرت تعمیم قابل‌قبولی داشته باشد. توانایی الگوریتم‌های تکاملی در حل مسائل بهینه‌سازی سبب ترغیب پژوهش‌گران به استفاده از این دسته از الگوریتم‌ها برای طراحی و آموزش شبکه‌های عصبی شده است (یائو، ۱۹۹۹). الگوریتم‌های تکاملی از روش‌های جستجوی تصادفی مبتنی بر جمعیت محسوب می‌شوند (میچل، ۱۹۹۸). اساس این الگوریتم‌ها که از تکامل طبیعی موجودات زنده الهام گرفته است، انتخاب متناسب با

www.SID.ir

Archive of SID

برازندگی افراد جمعیت می‌باشد. در یک الگوریتم تکاملی، جمعیتی از راه‌حل‌های نامزد با استفاده از عملگرهای ژنتیکی و در طی نسل‌های متوالی تکامل می‌یابند؛ در شرایطی که احتمال گیرافتادن در دام نقاط بهینهٔ محلی پایین است.

یکی از متداول‌ترین روش‌ها برای آموزش شبکه‌های عصبی، الگوریتم‌های مبتنی بر کاهش گرادیان است. الگوریتم پس‌انتشار خطا یکی از نمونه‌های پرکاربرد این نوع الگوریتم‌ها است که در نسخه‌های مختلفی استفاده شده است (لانگ و همکاران، ۱۹۹۰). یکی از معایب روش‌های مبتنی بر گرادیان، افتادن در دام نقاط بهینه محلی تابع خطا است (وایتلی و همکاران، ۱۹۹۰). همچنین اگر تابع خطا چندبعدی بوده یا مشتق‌پذیر نباشد، روش پس‌انتشار خطا، نقطه بهینه عمومی را نمی‌تواند پیدا کند؛ در مرجع (هرتز، ۱۹۹۱) جزییات روش پس‌انتشار خطا و سایر الگوریتم‌های یادگیری آمده است. استفاده از الگوریتم‌های تکاملی برای یافتن مجموعهٔ اوزان شبکه‌های عصبی درواقع راهی برای غلبه بر مشکلات روش‌های آموزش مبتنی برگرادیان است. الگوریتم‌های تکاملی به‌صورت عمومی جستجو می‌کنند که این امر باعث می‌شود بتوانند از دام بهینه‌های محلی فرار کنند. همچنین، این الگوریتم‌ها فضای جواب را بدون نیاز به اطلاعات گرادیان جستجو می‌کنند (وایتلی و همکاران، ۱۹۹۰)؛ و ازاین‌رو برای بسیاری از مسائل از قبیل کنترل و یادگیری تقویتی قابل استفاده و بسیار سودمند هستند (استنلی و میکالینن، ۲۰۰۲).

بخش دوم این مقاله به مرور انواع الگوریتم‌های تکاملی موجود برای توسعه شبکه‌های عصبی اختصاص دارد. الگوریتم ارائه شده در بخش ۳ توضیح داده می‌شود. نتایج محاسباتی حاصل از پیاده‌سازی الگوریتم بر روی داده‌های استاندارد و نیز مقایسه این نتایج با الگوریتم‌های دیگر در بخش ۴ ارائه می‌شود. اهمیت و تأثیر اجزاء الگوریتم در بخش ۵ تحلیل شده و در بخش پایانی نیز نتیجه‌گیری آورده می‌شود.

۲- طراحی و آموزش شبکه‌های عصبی به‌وسیلهٔ الگوریتم‌های تکاملی

الگوریتم‌های ارائه‌شده برای تکامل شبکه‌های عصبی را می‌توان در سه دسته تقسیم‌بندی کرد. دستهٔ نخست به روش‌هایی اختصاص دارد که معماری شبکه از قبل مشخص بوده و فقط اوزان اتصالات شبکه تکامل داده می‌شود. در این راستا می‌توان به استفاده از الگوریتم ژنتیک برای یافتن مجموعه اوزان بهینه اشاره کرد؛ نمایش وزن‌ها به صورت باینری و به‌کارگیری الگوریتم ژنتیک استاندارد از قدیمی‌ترین کارهای این دسته محسوب می‌شود (وایتلی، ۱۹۸۹). بعضی از پژوهش‌گران نیز از بردار اعداد حقیقی برای بازنمایی اوزان استفاده کرده‌اند (مونتانا و دیویس، ۱۹۸۹). یکی از مشکلات این الگوریتم‌ها، موضوع جایگشت نرون‌های یک شبکهٔ واحد است که می‌تواند باعث ناکارامدی عملگر ادغام[1] شود (یائو، ۱۹۹۹؛ حفیداسان، ۲۰۱۰). استراتژی تکاملی[2] و برنامه‌نویسی تکاملی[3] از معمول‌ترین روش‌ها برای تکامل بردارهای حقیقی محسوب می‌شوند (یائو، ۱۹۹۹). در سال‌های اخیر همچنین الگوریتم‌های مختلفی برای تکامل بردارهایی از اعداد حقیقی ارائه شده است (دب و همکاران، ۲۰۰۲) که در زمینه تکامل وزن‌های شبکه‌های عصبی نیز مورد استفاده قرار گرفته‌اند (کانتو-پاز و کمث، ۲۰۰۵).

دسته دوم به روش‌هایی اختصاص دارد که معماری شبکه عصبی را تکامل می‌دهند. طرح معماری در کارایی و قدرت پردازش شبکه تأثیر مستقیم دارد. اگر تعداد نرون‌های شبکه و ارتباطات بسیار کم باشد، ممکن است شبکه هیچ وقت همگرا نشود. از سوی دیگر، اگر پیچیدگی شبکهٔ عصبی بسیار زیاد باشد، امکان یادگیری بیش از حد[4] آن وجود دارد (کانتو-پاز و کمث، ۲۰۰۵). چنانچه معماری شبکه توسط فرد خبره طراحی شود، گریزی از سعی و خطا نخواهد بود. در تکامل معماری شبکه‌های عصبی، دو رهیافت مختلف برای بازنمایی ژنوتایپ وجود دارد. در رهیافت نخست که بازنمایی مستقیم گفته می‌شود، تمام جزئیات شبکه و ارتباطات در کروموزوم کد می‌شود. درنظرگرفتن شبکه عصبی به‌صورت یک گراف جهت‌دار و استفاده از ماتریس مجاورت به‌عنوان نمایش ژنوتایپی شبکه، یکی از روش‌های بازنمایی مستقیم است. درایه‌های ماتریس دودویی بوده و وجود یا عدم وجود ارتباط بین دو گره را نشان می‌دهند (وایتلی، ۱۹۸۹؛ کیتانو، ۱۹۹۰؛ کانتو-پاز و کمث، ۲۰۰۵). در رهیافت دوم تکامل معماری از بازنمایی غیرمستقیم استفاده می‌شود؛ میزان اطلاعاتی که در کروموزوم کد می‌شود، براساس دانشی که از شبکه مورد نظر داریم تعیین می‌شود. نمایش پارامتری یکی از انواع بازنمایی غیرمستقیم است. کدکردن قواعد توسعه شبکه عصبی در کروموزوم نیز

[1] Crossover Operator
[2] Evolution Strategy
[3] Evolutionary Programming
[4] Over-fitting

www.SID.ir

Archive of SID

یکی دیگر از روش‌های شناخته شده بازنمایی غیرمستقیم است. کیتانو (کیتانو، ۱۹۹۰) نخستین رهیافت مبتنی بر گرامر را ارائه داد. صدیقی و لوکاس (صدیقی و لوکاس، ۱۹۹۸) نشان دادند که کارایی بازنمایی مستقیم حداقل به اندازه این روش است.

یکی از بزرگ‌ترین مشکلات تکامل معماری، وجود نوفه و نامعتبربودن تابع برازندگی است که منبع آن یک‌به‌چند بودن نگاشت از سطح ژنوتایپ به سطح فنوتایپ است. برای حل این مشکل می‌توان از تکامل معماری و وزن‌ها به‌صورت هم‌زمان استفاده کرد که در دسته سوم سامانه‌های تکامل شبکه‌های عصبی جای می‌گیرد. در این راستا نیز انواع نمایش‌های مستقیم و غیرمستقیم مورد استفاده واقع شده است. یائو (یائو، ۱۹۹۹) یک سامانهٔ خودکار برای تکامل هم‌زمان معماری و وزن‌ها ارائه کرد که مبتنی بر برنامه‌نویسی تکاملی است. از جمله کارهای موفق در این زمینه باید به NEAT[1] (استنلی و میکالینن، ۲۰۰۲) اشاره کرد. همچنین از الگوریتم تکامل گرامری برای طراحی معماری و آموزش شبکه‌های عصبی استفاده شده است (تسولوس و همکاران، ۲۰۰۸). ریورو و همکاران (ریورو و همکاران، ۲۰۱۰) در سال ۲۰۱۰ یکی از جدیدترین روش‌های تکامل هم‌زمان معماری و اوزان را ارائه دادند. درهمین‌اواخر در مقالهٔ (کاستلانی، ۲۰۱۳) یک مقایسهٔ آماری بین رهیافت‌های مختلف تولید شبکه‌های عصبی انجام شده است. همچنین در مقاله مذکور یک الگوریتم جدید برای تکامل توأم معماری و اوزان با نمایش مستقیم ارائه شده و مورد تحلیل قرار گرفته است. علاوه‌بر موارد ذکرشده، ایدهٔ ترکیب شبکه‌های عصبی برای دست‌یابی به قدرت تعمیم بیشتر از طرف پژوهش‌گران مختلف در حال پی‌گیری است (یائو و اسلام، ۲۰۰۸؛ محمدزاده و همکاران، ۱۳۹۰).

با وجود توانایی الگوریتم‌های تکاملی برای آموزش شبکه‌های عصبی، لزوم طراحی معماری که به‌طورمعمول توسط فرد خبره و با سعی و خطا انجام می‌شود، بزرگ‌ترین مشکل این رهیافت محسوب می‌شود. ازاین‌رو در این مقاله، الگوریتم جدیدی برای تعیین همزمان معماری و اوزان بر پایهٔ استراتژی تکاملی ارائه شده است. معمول‌ترین راه برای تکامل هم‌زمان معماری و اوزان، قراردادن شبکه‌های با معماری مختلف (به‌عنوان‌مثال شبکه‌های با تعداد نرون میانی متفاوت) در یک جمعیت است؛ اما از آنجایی که اطلاعات یک شبکهٔ عصبی پخش شده است، انجام عمل‌گر ادغام بر روی شبکه‌های با معماری متفاوت منجر به تولید فرزندان با کیفیت پایین می‌شود (آنجلاین و همکاران، ۱۹۹۴؛ استنلی، ۲۰۰۴). به همین دلیل، در الگوریتم ارائه‌شده از چند جمعیت (گونه) برای تکامل معماری‌های مختلف بهره گرفته می‌شود تا به معماری‌های مناسب فرصت تکامل داده شود. جمعیت‌های موازی، این امکان را فراهم می‌سازند تا الگوریتم به تعداد نرون‌های میانی مورد نیاز دست پیدا کند؛ اما همان‌طور که پیش‌تر گفته شد، اگر پیچیدگی یک شبکه بیشتر از حد لازم باشد، ممکن است به یادگیری بیش از حد آن منجر شود که این امر سبب کاهش قدرت تعمیم شبکه می‌شود. قابل ذکر است که پیاده‌سازی شبکه‌های ساده‌تر با سهولت بیشتری انجام می‌شود. همچنین استخراج دانش از شبکه‌های عصبی آموزش‌دیده با افزایش تعداد نرون‌ها مشکل‌تر می‌شود. بنابراین، در الگوریتم پیشنهادی، راهی اندیشیده شده است تا از بین شبکه‌های تکامل‌یافته در جمعیت‌های موازی، شبکه‌ای انتخاب شود که علاوه‌بر خطای آموزش، ساده نیز باشد. جزئیات این الگوریتم در بخش بعد شرح داده شده است.

۳- الگوریتم ارائه‌شده

یکی از اساسی‌ترین مسائل در طراحی معماری یک شبکهٔ عصبی تعیین تعداد نرون‌های میانی است که خود بر ارتباطات شبکه تأثیر خواهد گذاشت. طرح معماری همراه با مجموعه اوزان تعیین‌کنندهٔ کارایی و قدرت پردازش شبکه هستند. الگوریتم ارائه‌شده از چند جمعیت موازی استفاده می‌کند؛ به‌گونه‌ای که شبکه‌های عصبی موجود در هر جمعیت از لحاظ معماری یکسان هستند؛ درحالی‌که تعداد نرون‌های میانی از یک جمعیت به جمعیت دیگر متفاوت است. تکامل هر جمعیت به‌صورت مستقل و با استراتژی تکاملی انجام می‌گیرد.

۳-۱- طراحی معماری و انتخاب جواب نهایی

در تعیین تعداد نرون‌های هر جمعیت از این گزارهٔ تجربی که "*شبکهٔ ساده‌تر می‌تواند منجر به قدرت تعمیم بیشتر شود*"، استفاده شده است. به این منظور، برای مسائلی که دانش خاصی در مورد آن‌ها در دست نیست، جستجو از شبکه‌های دارای کمترین تعداد نرون‌های میانی شروع می‌شود (به عبارت دیگر، شبکه‌های جمعیت نخست دارای دو نرون، شبکه‌های جمعیت دوم دارای سه نرون و ... خواهند بود).

[1] Neuro Evolution of Augmenting Topologies

www.SID.ir

Archive of SID

یکی دیگر از روش‌های شناخته شده بازنمایی غیرمستقیم است. کیتانو (کیتانو، ۱۹۹۰) نخستین رهیافت مبتنی بر گرامر را ارائه داد. صدیقی و لوکاس (صدیقی و لوکاس، ۱۹۹۸) نشان دادند که کارایی بازنمایی مستقیم حداقل به اندازه این روش است.

یکی از بزرگ‌ترین مشکلات تکامل معماری، وجود نوفه و نامعتبربودن تابع برازندگی است که منبع آن یک‌به‌چند بودن نگاشت از سطح ژنوتایپ به سطح فنوتایپ است. برای حل این مشکل می‌توان از تکامل معماری و وزن‌ها به‌صورت هم‌زمان استفاده کرد که در دسته سوم سامانه‌های تکامل شبکه‌های عصبی جای می‌گیرد. در این راستا نیز انواع نمایش‌های مستقیم و غیرمستقیم مورد استفاده واقع شده است. یائو (یائو، ۱۹۹۹) یک سامانهٔ خودکار برای تکامل هم‌زمان معماری و وزن‌ها ارائه کرد که مبتنی بر برنامه‌نویسی تکاملی است. از جمله کارهای موفق در این زمینه باید به NEAT[1] (استنلی و میکالینن، ۲۰۰۲) اشاره کرد. همچنین از الگوریتم تکامل گرامری برای طراحی معماری و آموزش شبکه‌های عصبی استفاده شده است (تسولوس و همکاران، ۲۰۰۸). ریورو و همکاران (ریورو و همکاران، ۲۰۱۰) در سال ۲۰۱۰ یکی از جدیدترین روش‌های تکامل هم‌زمان معماری و اوزان را ارائه دادند. درهمین‌اواخر در مقالهٔ (کاستلانی، ۲۰۱۳) یک مقایسهٔ آماری بین رهیافت‌های مختلف تولید شبکه‌های عصبی انجام شده است. همچنین در مقاله مذکور یک الگوریتم جدید برای تکامل توأم معماری و اوزان با نمایش مستقیم ارائه شده و مورد تحلیل قرار گرفته است. علاوه‌بر موارد ذکرشده، ایدهٔ ترکیب شبکه‌های عصبی برای دست‌یابی به قدرت تعمیم بیشتر از طرف پژوهش‌گران مختلف در حال پی‌گیری است (یائو و اسلام، ۲۰۰۸؛ محمدزاده و همکاران، ۱۳۹۰).

با وجود توانایی الگوریتم‌های تکاملی برای آموزش شبکه‌های عصبی، لزوم طراحی معماری که به‌طورمعمول توسط فرد خبره و با سعی و خطا انجام می‌شود، بزرگ‌ترین مشکل این رهیافت محسوب می‌شود. ازاین‌رو در این مقاله، الگوریتم جدیدی برای تعیین همزمان معماری و اوزان بر پایهٔ استراتژی تکاملی ارائه شده است. معمول‌ترین راه برای تکامل هم‌زمان معماری و اوزان، قراردادن شبکه‌های با معماری مختلف (به‌عنوان‌مثال شبکه‌های با تعداد نرون میانی متفاوت) در یک جمعیت است؛ اما از آنجایی که اطلاعات یک شبکهٔ عصبی پخش شده است، انجام عمل‌گر ادغام بر روی شبکه‌های با معماری متفاوت منجر به تولید فرزندان با کیفیت پایین می‌شود (آنجلاین و همکاران، ۱۹۹۴؛ استنلی، ۲۰۰۴). به همین دلیل، در الگوریتم ارائه‌شده از چند جمعیت (گونه) برای تکامل معماری‌های مختلف بهره گرفته می‌شود تا به معماری‌های مناسب فرصت تکامل داده شود. جمعیت‌های موازی، این امکان را فراهم می‌سازند تا الگوریتم به تعداد نرون‌های میانی مورد نیاز دست پیدا کند؛ اما همان‌طور که پیش‌تر گفته شد، اگر پیچیدگی یک شبکه بیشتر از حد لازم باشد، ممکن است به یادگیری بیش از حد آن منجر شود که این امر سبب کاهش قدرت تعمیم شبکه می‌شود. قابل ذکر است که پیاده‌سازی شبکه‌های ساده‌تر با سهولت بیشتری انجام می‌شود. همچنین استخراج دانش از شبکه‌های عصبی آموزش‌دیده با افزایش تعداد نرون‌ها مشکل‌تر می‌شود. بنابراین، در الگوریتم پیشنهادی، راهی اندیشیده شده است تا از بین شبکه‌های تکامل‌یافته در جمعیت‌های موازی، شبکه‌ای انتخاب شود که علاوه‌بر خطای آموزش، ساده نیز باشد. جزئیات این الگوریتم در بخش بعد شرح داده شده است.

۳- الگوریتم ارائه‌شده

یکی از اساسی‌ترین مسائل در طراحی معماری یک شبکهٔ عصبی تعیین تعداد نرون‌های میانی است که خود بر ارتباطات شبکه تأثیر خواهد گذاشت. طرح معماری همراه با مجموعه اوزان تعیین‌کنندهٔ کارایی و قدرت پردازش شبکه هستند. الگوریتم ارائه‌شده از چند جمعیت موازی استفاده می‌کند؛ به‌گونه‌ای که شبکه‌های عصبی موجود در هر جمعیت از لحاظ معماری یکسان هستند؛ درحالی‌که تعداد نرون‌های میانی از یک جمعیت به جمعیت دیگر متفاوت است. تکامل هر جمعیت به‌صورت مستقل و با استراتژی تکاملی انجام می‌گیرد.

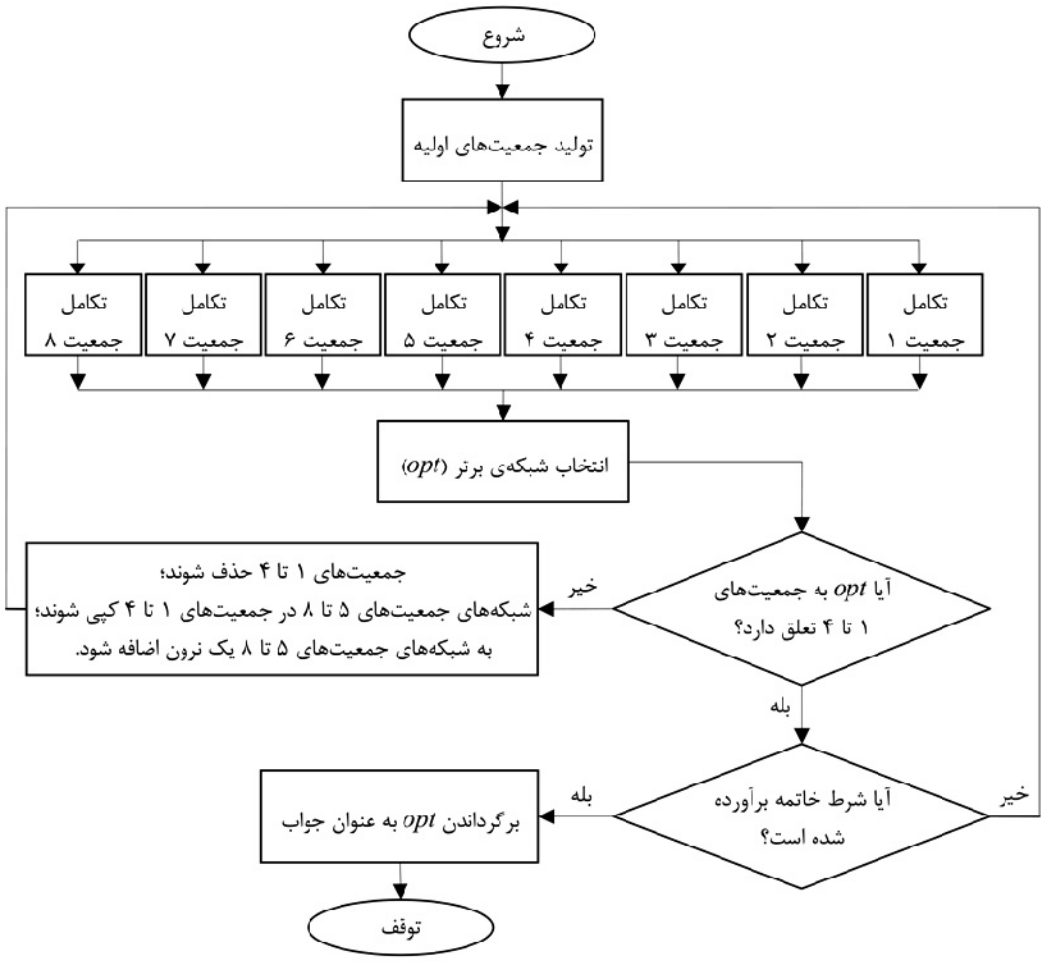
۳-۱- طراحی معماری و انتخاب جواب نهایی

در تعیین تعداد نرون‌های هر جمعیت از این گزارهٔ تجربی که "*شبکهٔ ساده‌تر می‌تواند منجر به قدرت تعمیم بیشتر شود*"، استفاده شده است. به این منظور، برای مسائلی که دانش خاصی در مورد آن‌ها در دست نیست، جستجو از شبکه‌های دارای کمترین تعداد نرون‌های میانی شروع می‌شود (به عبارت دیگر، شبکه‌های جمعیت نخست دارای دو نرون، شبکه‌های جمعیت دوم دارای سه نرون و ... خواهند بود).

[1] Neuro Evolution of Augmenting Topologies

www.SID.ir

Archive of SID



(شکل- ۱): شمای الگوریتم ارائه شده

با نرون خروجی)، تعداد کل عناصر بردار مذکور $h(m+2)$ خواهد بود که آن را با l نشان می‌دهیم. به ازای هر وزن یک طول گام جهش نیز در کروموزوم قرار داده می‌شود و بنابراین طول آن برابر با $2l$ خواهد بود؛ این کروموزوم را می‌توان به‌صورت $\langle w_1, w_2, \ldots, w_l, \sigma_1, \sigma_2, \ldots, \sigma_l \rangle$ در نظر گرفت که در آن w_i وزن یال iام شبکه و σ_i پارامتر استراتژی (طول گام جهش) متناظر آن است. توجه شود که وجود یک طول گام جهش به‌ازای هر وزن این امکان را به الگوریتم می‌دهد تا در جهات مختلف با سرعت‌های متفاوت جستجو کند.

شکل (۲) یک شبکهٔ عصبی با دو نرون ورودی، دو نرون میانی و یک نرون خروجی را همراه با نمایش ژنوتایپ آن نشان می‌دهد. در این نمایش وزن‌های ارتباطات ورودی، یال جبران‌ساز و نیز ارتباط خروجی هر نرون میانی در کنار همدیگر قرار داده می‌شوند.

۳-۲-۲- انتخاب والدین

همانند استراتژی تکاملی استاندارد، والدین به صورت تصادفی و بدون توجه به میزان برازندگی از توزیع یکنواخت انتخاب می‌شوند. از آن‌جایی‌که در یک استراتژی تکاملی مناسب تعداد فرزندان تولیدشده باید از تعداد افراد جمعیت بیشتر باشد، در این پژوهش، تعداد فرزندان تولیدشده هشت برابر تعداد اعضای جمعیت است.

۳-۲-۳- عمل‌گر ادغام والدین

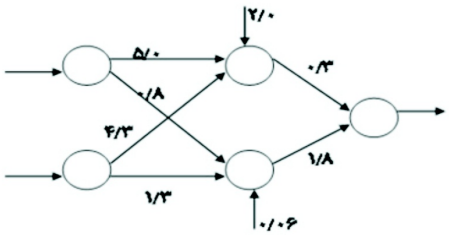
ادغام والدین در دو مرحله انجام می‌گیرد. در یک مرحله، ترکیب گسسته یکنواخت بر روی وزن‌ها انجام شده و در مرحله دیگر، ترکیب پیوسته میانی بر روی پارامترهای استراتژی صورت می‌گیرد.

۳-۲-۴- عمل‌گر جهش

ازآنجایی‌که در الگوریتم ارائه‌شده به‌ازای هر وزن یک طول گام جهش (پارامتر استراتژی) در هر کروموزوم در نظر گرفته

www.SID.ir

Archive of SID



(الف)

(ب)

۵/۰	۴/۳	۲/۰	۰/۳	۰/۸	۱/۳	۰/۰۶	۱/۸	۰/۲	۱/۹	۰/۰۰۳	۰/۱۴	۱۱	۸/۴	۱/۰۱	۵/۸

(شکل- ۲- (الف) یک شبکه عصبی با ۲ نرون ورودی، ۲ نرون میانی و ۱ نرون خروجی. (ب) نمایش این شبکه عصبی در سطح ژنوتایپ که شامل وزن‌های شبکه و طول گام‌های جهش وزن‌هاست

شده است، عملگر جهش در دو مرحله انجام می‌شود. در مرحله نخست، عمل جهش گوسی بر روی پارامترهای استراتژی انجام می‌شود و پارامترهای استراتژی جدید (σ_i') جایگزین پارامترهای استراتژی قبلی (σ_i) می‌شوند. در مرحله دوم، با استفاده از این پارامترها به‌عنوان طول گام‌های جهش، وزن‌ها (w_i ها) مورد جهش گوسی قرار می‌گیرند.

جهش پارامترهای استراتژی و وزن‌ها به‌صورت زیر انجام می‌گیرد:

$$\sigma_i' = \sigma_i . e^{N(0,\tau') + N_i(0,\tau)} \quad (۲)$$

$$w_i' = w_i + N_i(0, \sigma_i') \quad (۳)$$

که در آن $\tau' = 1/\sqrt{2l}$ نرخ یادگیری عمومی و $\tau = 1/\sqrt{2\sqrt{l}}$ نرخ یادگیری اختصاصی است. برای آنکه پارامترهای استراتژی به صفر نزدیک نشوند، یک قاعده نیز در نظر گرفته شده است به این صورت که اگر پارامتری از مقدار مشخص ۰/۲ کوچکتر شد، با ۰/۲ جایگزین می‌شود.

۳-۲-۵- انتخاب نسل جدید

فرزندان تولیدشده براساس برازندگی مرتب‌شده و سپس بهترین‌های آنها جایگزین نسل فعلی می‌شوند. در واقع، تمامی جواب‌های فعلی با جواب‌های جدید جایگزین می‌شوند که این امر الگوریتم را قادر خواهد ساخت تا بتواند از دام نقاط بهینه محلی فرار کند. همچنین به‌دلیل انتخاب بهترین فرزندان، فشار انتخاب زیاد است؛ این موضوع خود باعث کاهش اثر مشکل جای‌گشت می‌شود (هنکاک، ۱۹۹۲).

۴-نتایج محاسباتی

در این بخش، نتایج آزمایش‌ها و نیز تحلیل‌های صورت گرفته ارائه می‌شود. برای ارزیابی میزان کارایی الگوریتم ارائه‌شده از هفت مجموعه داده‌ای استاندارد دسته‌بندی (بلیک و مرز، ۱۹۹۸) استفاده شده است. در جدول (۱) مشخصات این مجموعهٔ داده‌ها آمده است.

(جدول- ۱): مشخصات مجموعه داده‌ها

مجموعه داده	تعداد نمونه	تعداد کلاس	تعداد ویژگی	نام مجموعه داده
۱	۶۹۹	۲	۹	Breast Cancer
۲	۱۰۰۰	۲	۶۲	Credit-German
۳	۳۰۳	۲	۲۶	Heart-Cleveland
۴	۳۵۱	۲	۳۴	Ionosphere
۵	۱۵۰	۳	۴	Iris
۶	۲۰۸	۲	۶۰	Sonar
۷	۱۷۸	۳	۱۳	Wine

در آزمایش‌های صورت‌گرفته از روش اعتبارسنجی متقابل دوبخشی استفاده شده است (کانتو-پاز و کمث، ۲۰۰۵). در این روش، داده‌ها با توزیع یک‌نواخت به دو بخش مساوی تقسیم می‌شوند. در هر آزمایش دو بار الگوریتم اجرا شده است. در اجرای نخست، از بخش نخست داده‌ها به‌عنوان مجموعهٔ آموزش و از بخش دیگر به‌عنوان مجموعهٔ آزمون استفاده شده است، درحالی‌که در اجرای دوم، بخش دوم برای آموزش و بخش نخست نیز برای آزمون مورد استفاده قرار گرفته است. این آزمایش با هر یک از هفت مجموعه

www.SID.ir

Archive of SID

داده‌ها پنج بار تکرار شده است. میانگین و انحراف از معیار صحت دسته‌بندی (بر روی مجموعه‌های آزمون) در ۱۰ بار اجرا به عنوان شاخصی برای مقایسه الگوریتم‌ها استفاده شده است؛ نتایج روش‌های مورد مقایسه نیز از این نوع آزمایش به‌دست آمده است (کانتو-پاز و کمث، ۲۰۰۵؛ ریورو و همکاران، ۲۰۱۰). در این مقاله، علاوه‌بر صحت دسته‌بندی، سادگی شبکه‌های به‌دست آمده به‌عنوان معیار دوم برای مقایسهٔ دو الگوریتم مورد استفاده قرار گرفته است. با یک آزمون آماری، به نحوی منطقی و با اولویت بر معیار نخست، از این دو معیار برای ارزیابی معنی‌داربودن اختلاف عملکرد الگوریتم‌ها استفاده شده است. این روش که در مقالهٔ (کاستلانی، ۲۰۱۳) مورد استفاده واقع شده است، به‌راحتی به ما امکان می‌دهد که نتایج به‌دست آمده از آزمایش‌ها را تفسیر کنیم. این روش در بخش‌های ۴-۱ و ۴-۲ به‌تفصیل شرح داده شده است. همچنین، زمان اجرا یا هزینه محاسباتی الگوریتم‌های مختلف نیز در یک مقایسهٔ منصفانه باید مورد ارزیابی قرار گیرد. در این مقاله، از تعداد فراخوانی‌های تابع ارزیابی به‌عنوان هزینهٔ محاسباتی استفاده شده است. این شاخص در مرجع (کانتو-پاز و کمث، ۲۰۰۵) گزارش نشده است؛ امّا می‌توان آن‌را با ضرب تعداد افراد جمعیت در تعداد نسل‌ها که در آن مقاله گزارش شده است، محاسبه کرد. این روش محاسبه در مرجع (ریورو و همکاران، ۲۰۱۰) نیز استفاده شده است. الگوریتم‌های مورد مقایسه با توجه به هزینهٔ محاسباتی به دو دسته تقسیم شده‌اند. دستهٔ نخست شامل الگوریتم پس‌انتشار خطا، الگوریتم تکامل بردار اعداد حقیقی G3PCX (کانتو-پاز و کمث، ۲۰۰۵) و الگوریتم تکامل بردار دودویی است. دسته دوم روش‌های مورد مقایسه نیز شامل الگوریتم‌های طراحی معماری و الگوریتم‌های تکامل همزمان معماری و اوزان است. از آن‌جایی که هزینه محاسباتی الگوریتم‌های دسته نخست بسیار کمتر از دسته دوم است، الگوریتم ارائه‌شده با دو مجموعه پارامتر متفاوت اجرا شده است که جزئیات آنها در جدول (۲) آمده است. از تنظیم نخست برای مقایسه الگوریتم پیشنهادی با الگوریتم‌های تکامل اوزان و از تنظیم دوم برای مقایسه آن با بقیه‌ی الگوریتم‌ها استفاده شده است. در هر یک از این مقایسه‌ها، شرط خاتمه به‌گونه‌ای انتخاب شده است که تعداد فراخوانی‌های تابع برازندگی توسط الگوریتم ارائه‌شده در حدود تعداد فراخوانی‌های الگوریتم‌های مورد مقایسه باشد. البته، حد بالای تعداد فراخوانی‌های الگوریتم ارائه‌شده در تمام آزمایش‌ها دویست‌هزار درنظر گرفته شده است.

(جدول- ۲): انواع تنظیمات الگوریتم ارائه شده

پارامتر	تنظیم اول	تنظیم دوم
تعداد جمعیت‌های موازی	۴	۸
تعداد افراد هر جمعیت	۲۰	۴۰
نسبت تعداد فرزندان به والدین	۸	۸

۴-۱- نحوهٔ مقایسهٔ آماری دو الگوریتم

دو الگوریتم، ابتدا براساس معیار نخست، یعنی صحت دسته‌بندی شبکه‌های تولیدشده، با انجام آزمون آماری t در سطح اطمینان ۹۵٪ با هم مقایسه می‌شوند (همان‌گونه که پیش‌تر ذکر شد، هر الگوریتم به‌ازای هر مجموعه داده ۱۰ بار اجرا می‌شود و لذا ۱۰ شبکهٔ عصبی مرتبط با آن خواهیم داشت). چنانچه اختلاف دو الگوریتم معنی‌دار باشد، الگوریتم تولیدکنندهٔ شبکه‌های با خطای کمتر را برنده و الگوریتم دیگر را بازنده می‌نامیم؛ اما اگر این اختلاف معنی‌دار نباشد، الگوریتم‌ها این‌بار براساس معیار دوم، یعنی تعداد نرون‌های میانی شبکه‌های تولیدشده، به‌صورت آماری با هم مقایسه می‌شوند. اگر این اختلاف نیز معنی‌دار نباشد، دو الگوریتم مساوی در نظر گرفته می‌شوند؛ درغیراین‌صورت، الگوریتمی که شبکه‌های ساده‌تر تولید کرده باشد، برنده و دومی بازنده خواهد بود.

۴-۲- نحوهٔ مقایسهٔ چند الگوریتم

برای مقایسهٔ چند الگوریتم بر روی یک مجموعه داده مشخص، هر جفت الگوریتم با روش بیان‌شده در بخش قبل با هم مقایسه می‌شوند. به‌ازای هر بار برنده‌شدن یک الگوریتم در مقایسه با الگوریتمی دیگر، دو امتیاز می‌گیرد و در صورت مساوی‌شدن، یک امتیاز می‌گیرد. در حالت بازنده‌شدن نیز امتیازی دریافت نمی‌کند. جمع امتیازهای اخذشدهٔ یک الگوریتم، پس از انجام این مقایسه‌های زوجی، امتیاز آن الگوریتم بر روی مجموعه‌دادهٔ مورد بحث را مشخص می‌کند.

۴-۳- مقایسهٔ الگوریتم پیشنهادی با روش‌های آموزش شبکه

الگوریتم پیشنهادی با الگوریتم‌های تکاملی موجود برای آموزش شبکه‌های عصبی شامل الگوریتم G3PCX (کانتو-پاز و کمث، ۲۰۰۵) که از بردار اعداد حقیقی برای بازنمایی اوزان استفاده می‌کند و نیز الگوریتم تکامل اوزان با نمایش دودویی مقایسه می‌شود. همچنین از الگوریتم پس‌انتشار خطا که یکی از روش‌های سنتی آموزش شبکه‌های عصبی

www.SID.ir

Archive of SID

است، نیز برای مقایسه استفاده شده است. جزئیات پیاده‌سازی الگوریتم‌های مورد مقایسه در مرجع (کانتو-پاز و کمث، ۲۰۰۵) آمده است و از این‌رو نتایج مربوطه از این مرجع اقتباس شده است.

از آنجایی که در الگوریتم پیشنهادی دو معیار (شامل خطای آموزش و سادگی شبکهٔ عصبی) در نظر گرفته می‌شود، در مقایسهٔ این روش با سایر روش‌ها نیز از این دو معیار استفاده شده است. جدول (۳) خلاصه نتایج محاسباتی را از نظر صحت دسته‌بندی، پیچیدگی شبکه‌های به‌دست‌آمده و تعداد ارزیابی‌ها نشان می‌دهد. در این جدول، نتایج هر الگوریتم بر روی هر مجموعه داده در سه سطر آورده شده است. در سطرهای نخست و دوم به‌ترتیب میانگین و انحراف از معیار صحت دسته‌بندی و تعداد نرون‌های میانی شبکه‌های به‌دست‌آمده، آورده شده است. در سطر سوم نیز تعداد فراخوانی‌های تابع برازندگی قید شده است. برای هر مجموعه داده، بیشترین صحت دسته‌بندی به‌صورت پررنگ مشخص شده است که نشان می‌دهد روش پیشنهادی برای سه مجموعه از داده‌ها، به‌دقتی بالاتر از روش‌های موجود دست یافته است. همچنین میانگین صحت دسته‌بندی روش ارائه‌شده از سایر روش‌ها بیشتر است؛ درحالی‌که پیچیدگی شبکه‌هایی که تولید کرده، بسیار کمتر بوده است.

در ادامه، برای درک بهتر نتایج حاصله و بررسی معنی‌داربودن برتری الگوریتم پیشنهادی، از روش دو معیاره‌ی بخش ۴-۲ استفاده می‌شود. جدول (۴) نتیجه انجام این ارزیابی را نشان می‌دهد. همان‌گونه که پیداست، الگوریتم ارائه شده در پنج مورد از هفت مجموعهٔ دادهٔ مورد استفاده بر سایر الگوریتم‌ها برتری داشته است. جمع امتیازهای هر الگوریتم در مورد همهٔ مجموعهٔ داده‌ها، امتیاز کلی آن را نشان می‌دهد. این امتیاز برتری الگوریتم پیشنهادی را به‌خوبی نمایان می‌سازد.

(جدول- ۳): نتایج الگوریتم ارائه شده در مقایسه با روش‌های آموزش شبکه‌های عصبی

مجموعه داده	پس‌انتشار خطا		G3PCX		باینری		الگوریتم پیشنهادی	
۱	⊕۹۶/۳۹	⊖۰/۵۸	۹۸/۹۴	۲/۳۵	۹۸/۸۸	۰/۳۲	۹۶/۳۲	۱/۰۸
	۵⊗	۰ ⊙	۵	۰	۵	۰	۲	۰
	†۱۱۵۰۰		۱۱۵۰۰		۸۴۰۰		۱۱۵۰۰	
۲	۷۰/۱۲	۱/۳۹	۳۰/۰۰	۱/۴۶	۷۰/۹۴	۱/۴۹	۶۹/۹۸	۱/۶۰
	۱۰	۰	۱۰	۰	۱۰	۰	۲/۷	۱/۰
	۳۹۱۰۰		۳۹۱۰۰		۳۰۱۰۰		۳۰۱۰۰	
۳	۷۸/۱۷	۳/۱۶	۹۰/۴۲	۲/۱۲	۸۷/۷۲	۳/۴۲	۸۰/۰۲	۲/۰۵
	۵	۰	۵	۰	۵	۰	۲/۳	۰/۶
	۱۷۵۵۰		۱۷۵۵۰		۱۵۲۰۰		۱۷۵۵۰	
۴	۸۴/۷۷	۳/۸۰	۶۴/۱۰	۲/۰۴	۷۴/۱۰	۱/۹۴	۸۹/۱۴	۲/۶۴
	۱۰	۰	۱۰	۰	۱۰	۰	۲/۴	۰/۵
	۲۹۳۵۰		۲۹۳۵۰		۲۲۴۰۰		۲۲۴۰۰	
۵	۹۴/۵۳	۳/۱۲	۸۹/۷۳	۱۱/۷۰	۸۸/۶۷	۶/۰۹	۹۵/۷۳	۱/۶۰
	۵	۰	۵	۰	۵	۰	۲/۱	۰/۲
	۱۰۲۵۰		۱۰۲۵۰		۱۳۹۰۰		۱۳۹۰۰	
۶	۶۹/۶۱	۳/۱۲	۶۷/۴۰	۵/۴۴	۷۳/۶۵	۲/۵۵	۷۴/۱۳	۳/۱۰
	۱۰	۰	۱۰	۰	۱۰	۰	۲/۷	۰/۸
	۳۸۸۰۰		۳۸۸۰۰		۲۹۶۰۰		۲۹۶۰۰	
۷	۹۵/۱۶	۱/۷۶	۸۴/۹۴	۱۱/۴۶	۹۲/۴۷	۴/۵۵	۹۳/۸۲	۰/۹۰
	۵	۰	۵	۰	۵	۰	۲/۳	۰/۵
	۱۶۳۵۰		۱۶۳۵۰		۱۳۰۰۰		۱۰۷۰۰	
میانگین	*۸۴/۱۱*	*۲/۴۲*	*۷۵/۰۸*	*۵/۲۲*	*۸۳/۷۸*	*۲/۹۱*	*۸۵/۵۹*	*۱/۸۵*
	۷/۱	۰	*۷/۱*	۰	*۷/۱*	۰	*۲/۳*	*۰/۵*
	۲۳۲۷۱		*۲۳۲۷۱*		*۱۸۹۴۳*		*۱۹۳۹۳*	

⊕ میانگین صحت دسته‌بندی ⊖ انحراف از معیار صحت دسته‌بندی ⊗ میانگین تعداد نرون‌ها ⊙ انحراف از معیار تعداد نرون‌ها

† تعداد فراخوانی‌های تابع برازندگی

www.SID.ir

Archive of SID

(جدول- ۴): مقایسه‌ی آماری الگوریتم ارائه شده با روش‌های آموزش

مجموعه داده	پس‌انتشار خطا	G3PCX	باینری	الگوریتم پیشنهادی
۱	۱	۶	۶	۳
۲	۴	۱	۴	**۷**
۳	۱	۶	۶	۳
۴	۵	۱	۳	**۷**
۵	۴	۳	۲	**۷**
۶	۲	۲	۵	**۷**
۷	۴	۲	۳	**۷**
امتیاز کلی	۲۱	۲۱	۲۹	***۴۱***

۴-۴- مقایسه الگوریتم پیشنهادی با روش‌های طراحی و طراحی و آموزش توأم

الگوریتم ارائه‌شده با الگوریتم‌های تکامل معماری شبکه‌های عصبی شامل روش نمایش ماتریسی، روش هرس، روش نمایش پارامتری و روش گرامری کیتانو که جزئیات آنها در مرجع (کانتو-پاز و کمث، ۲۰۰۵) آمده است، مقایسه می‌شود. همچنین از الگوریتم‌های تکامل هم‌زمان معماری و اوزان نیز برای مقایسه استفاده شده است؛ نتایج روش برنامه‌نویسی ژنتیکی (ریورو و همکاران، ۲۰۱۰) از مقاله مرجع آورده شده است؛ درحالی‌که نتایج مربوط به روش تکامل گرامری (تسولوس و همکاران، ۲۰۰۸) با پیاده‌سازی و سپس انجام آزمایش به‌دست آمده است. جدول (۵) خلاصه نتایج محاسباتی را نشان می‌دهد (چیدمان داده‌ها در این جدول همانند جدول (۳) است).

با مراجعه به جدول (۵)، مشاهده می‌شود که الگوریتم ارائه‌شده برای چهار مجموعه از داده‌ها به دقتی بالاتر از سایر روش‌ها دست یافته است. این درحالی‌است‌که هزینهٔ محاسباتی آن کمتر از سایر روش‌ها به‌جز روش هرس بوده است. همچنین میانگین صحت دسته‌بندی روش ارائه‌شده از بقیهٔ روش‌ها بیشتر است؛ درحالی‌که پیچیدگی شبکه‌هایی که تولید کرده، کمتر بوده است؛ توجه شود که میانگین صحت دسته‌بندی الگوریتم پیشنهادی برای آن چهار مجموعه از داده‌ها که در مرجع (ریورو و همکاران، ۲۰۱۰) در مورد روش برنامه‌نویسی ژنتیکی گزارش شده، برابر با ۹۰/۹۳ است.

برای بررسی معنی‌داربودن برتری الگوریتم ارائه‌شده، از روش دو معیاره‌ی بخش ۴-۲ استفاده کرده و حاصل آن را در جدول (۶) آورده‌ایم (نام الگوریتم‌ها در این جدول به‌صورت مخفف ذکر شده است). همان‌گونه که پیداست، الگوریتم ارائه‌شده در پنج مورد از هفت مجموعهٔ داده مورد استفاده بر سایر الگوریتم‌ها برتری داشته است. همچنین، از نظر امتیاز کلی نیز الگوریتم پیشنهادی برتری چشم‌گیری نسبت به سایر روش‌ها دارد.

(جدول- ۶): مقایسهٔ آماری الگوریتم ارائه شده با روش‌های طراحی معماری و روش‌های توأم طراحی و آموزش

	روش‌های تکامل معماری				روش‌های تکامل توأم		
مجموعه داده	م	ه	پ	گ	ب.ژ	ت.گ	ا.پ
۱	۷	۷	۷	۷	۶	۶	**۹**
۲	۷	۵	۶	**۸**	--	۷	۳
۳	۶	**۱۳**	۳	۶	۶	۶	۹
۴	۷	۲	۵	۷	۶	۹	**۱۳**
۵	۵	۴	۷	۶	۷	۸	**۱۲**
۶	۵	۷	۶	۶	--	۴	**۸**
۷	۲	۷	**۸**	**۸**	--	۳	**۸**
امتیاز کلی	۳۹	۴۵	۴۲	۴۸	۲۵	۴۳	***۶۲***

نتیجهٔ دیگری که از نتایج آزمایش‌های انجام‌شده می‌توان گرفت این است که الگوریتم ارائه‌شده از نظر پایداری نیز عملکرد بسیار خوبی دارد. با مراجعه به جداول (۳ و ۵)، روشن است که انحراف از معیار صحت دسته‌بندی این روش در مقایسه با سایر روش‌ها تاحدودی کم است. علاوه‌براین، انحراف از معیار تعداد نرون‌های میانی شبکه‌های تولیدشده نیز به‌وضوح تأیید می‌کند که شبکه‌های عصبی که الگوریتم در اجراهای مختلف تولید می‌کند، از نظر معماری شباهت بسیار زیادی به هم دارند.

۵- تحلیل الگوریتم

در این بخش، ابتدا تأثیر جمعیت‌های موازی (چندجمعیتی بودن) در کارایی الگوریتم ارائه‌شده مورد ارزیابی قرار می‌گیرد؛ سپس، اثر انتخاب شبکهٔ برتر با دو معیار خطا و پیچیدگی اندازه‌گیری می‌شود. در خاتمه نیز، از آن‌جایی که همیشه عمل‌گر ادغام در تکامل شبکه‌های عصبی مورد بحث بوده است، اثر آن در الگوریتم پیشنهادی مورد بررسی قرار می‌گیرد.

برای بررسی اثر جمعیت‌های موازی، نسخه‌ای از الگوریتم پیشنهادی فقط با یک جمعیت که در آغاز شامل شبکه‌هایی با دو تا نه نرون میانی است، مورد استفاده قرار می‌گیرد. سایر شرایط این نسخه جدید همانند الگوریتم اصلی (شامل هشت جمعیت موازی) است. البته در مورد عملگر ادغام به این نکته توجه شده است که برای تولید یک فرزند با h نرون، فقط از والدهایی با همان تعداد نرون استفاده شود (والدی که ژن اول کروموزوم فرزند از آن گرفته

www.SID.ir

Archive of SID

می‌شود، h را تعیین خواهد کرد). نتایج این نسخه در جدول (۷) بعد از نتایج الگوریتم اصلی ارائه شده است. نتایج حاکی از آن است که نسخهٔ جدید در شش مجموعهٔ داده منجر به صحت دسته‌بندی کمتری نسبت به الگوریتم اصلی شده است. همچنین، شبکه‌های به‌دست‌آمده از آن نرون‌های میانی بیشتری نسبت به الگوریتم اصلی داشته است. البته از آنجایی که در این نسخه تعداد نرون‌ها در میان جمعیت ناهمگون است، این فرضیه به ذهن خطور می‌کند که شاید یکی از دلایل این اختلاف عملکرد، ناکارآمدی عملگر ادغام در نسخهٔ تک‌جمعیتی باشد؛ به همین دلیل، نسخه جدید بدون عمل‌گر ادغام نیز مورد آزمایش قرار گرفته است. با مشاهدهٔ نتایج ارائه‌شده در جدول (۷)، روشن می‌شود که باز این نسخه نسبت به الگوریتم اصلی، به‌ویژه از نظر تعداد نرون‌های میانی، عملکرد تاحدودی ضعیف‌تری دارد؛ بنابراین عمل‌گر ادغام تأثیر مخرب زیادی نداشته است و باید دلیل دیگری برای پایین‌بودن کارایی نسخهٔ تک‌جمعیتی وجود داشته باشد. درواقع، دلیل این موضوع را می‌توان در تصادفی‌بودن جمعیت نخستین و فشار انتخاب بسیار زیاد استراتژی تکاملی در تعیین نسل بعد دنبال کرد. این دو عامل در کنار هم می‌تواند منجر به شانس بقای بسیار زیادی برای گونهٔ خاصی از شبکه‌ها شود که این امر می‌تواند منجر به انقراض سایر شبکه‌ها در جمعیت شود. برای روشن‌شدن موضوع، روند تکامل نسخهٔ تک‌جمعیتی (با ادغام) بر روی دو مجموعه داده‌ی نخست و سوم به‌ترتیب در شکل‌های (۳ و ۴) نمایش داده شده است (دو بار اجرا به‌ازای هر مجموعهٔ داده انجام شده است). هر نمودار درصد حضور شبکه‌های با تعداد نرون مختلف در جمعیت را در طی نسل‌های متوالی به‌صورت فاصلهٔ عمودی بین خطوط نشان می‌دهد.

(جدول- ۵): نتایج الگوریتم ارائه شده در مقایسه با روش‌های طراحی معماری و روش‌های طراحی و آموزش همزمان شبکه‌های عصبی

مجموعه داده	روش‌های تکامل معماری				روش‌های تکامل توأم معماری و اوزان		
	ماتریسی	هرس	پارامتری	گرامری (کیتانو)	برنامه‌نویسی ژنتیکی	تکامل گرامری	الگوریتم پیشنهادی
۱	**۹۶/۷۷** ۱/۱۰	۹۶/۳۱ ۱/۲۱	۹۶/۶۹ ۱/۱۳	۹۶/۷۱ ۱/۱۶	⊖۰/۲۱ ⊕۹۶/۲۷	۹۶/۰۲ ۰/۴۴	۹۶/۱۹ ۰/۵۰
	--	--	--	--	⊙ ۰/۵ ⊗۳/۳	۳/۰ ۰/۵	۲/۰ ۰
	۹۲۰۰۰	۱۱۷۰	۱۰۰۰۰۰	۳۰۰۰۰۰	†۹۲۰۰۰	۹۲۰۰۰	۹۲۰۰۰
۲	۷۱/۷۶ ۲/۰۶	۶۹/۷۰ ۲/۴۲	۷۱/۷۰ ۵/۷۵	**۷۲/۱۰** ۱/۶۹	--	۷۱/۵۲ ۲/۵۲	۶۸/۶۶ ۲/۵۰
	--	--	--	--	--	۳/۴ ۱/۰	۳/۲ ۱/۰
	۴۶۸۰۰۰	۳۹۳۰	۱۵۰۰۰۰	۳۸۴۰۰۰	--	۲۵۰۰۰۰	۱۵۰۰۰۰
۳	۷۶/۷۸ ۷/۸۷	۸۹/۵۰ ۳/۳۶	۶۵/۸۹ ۱۳/۵۵	۷۲/۸۰ ۱۲/۵۶	**۸۰/۷۱** ۲/۴۵	۷۹/۶۰ ۲/۶۴	۸۰/۶۸ ۱/۹۵
	--	--	--	--	۲/۹ ۰/۳	۳/۳ ۱/۴	۲/۱ ۰/۳
	۳۰۴۰۰۰	۱۹۴۰	۲۰۰۰۰۰	۶۰۰۰۰۰	۲۰۰۰۰۰	۲۰۰۰۰۰	۲۰۰۰۰۰
۴	۸۷/۰۶ ۲/۱۴	۸۳/۶۶ ۱/۹۰	۸۵/۵۸ ۳/۰۸	۸۸/۰۳ ۱/۵۵	۸۷/۸۳ ۳/۱۴	۸۹/۹۳ ۴/۰۰	**۹۰/۱۶** ۲/۴۴
	--	--	--	--	۹/۳ ۰/۴	۵/۰ ۰/۸	۲/۷ ۰/۶
	۴۶۴۰۰۰	۲۹۴۰	۲۰۰۰۰۰	۶۰۰۰۰۰	۲۰۰۰۰۰	۲۰۰۰۰۰	۲۰۰۰۰۰
۵	۹۲/۴۰ ۲/۶۷	۹۲/۴۰ ۱/۴۰	۹۱/۷۳ ۸/۲۶	۹۲/۹۳ ۳/۰۸	۹۵/۲۲ ۲/۰۵	۹۴/۹۳ ۲/۳۳	**۹۶/۶۷** ۲/۰۸
	--	--	--	--	۸/۹ ۰/۵	۳/۵ ۰/۶	۲/۰ ۰
	۳۲۰۰۰۰	۱۰۸۰	۴۰۰۰۰۰	۱۲۰۰۰۰۰	۳۲۰۰۰۰	۲۵۰۰۰۰	۲۰۰۰۰۰
۶	۷۱/۳۴ ۴/۰۹	۷۳/۹۴ ۳/۹۸	۷۲/۵۹ ۴/۱۶	۷۰/۵۷ ۶/۴۴	--	۶۹/۵۲ ۴/۰۳	**۷۵/۸۶** ۴/۴۵
	--	--	--	--	--	۵/۰ ۱/۲	۳/۱ ۰/۸
	۹۲۴۰۰۰	۳۹۱۰	۳۰۰۰۰۰	۷۶۸۰۰۰	--	۲۵۰۰۰۰	۲۰۰۰۰۰
۷	۹۰/۵۶ ۳/۱۴	۹۳/۳۷ ۲/۳۳	۹۴/۰۴ ۲/۳۷	۹۴/۴۹ ۲/۷۲	--	۹۰/۸۹ ۲/۶۷	**۹۵/۲۸** ۱/۷۲
	--	--	--	--	--	۲/۸ ۱/۰	۲/۴ ۰/۹
	۹۶۰۰۰	۱۶۱۵	۷۵۰۰۰	۱۹۲۰۰۰	--	۷۵۰۰۰	۷۵۰۰۰
میانگین	*۸۳/۸۱ ۳/۳۰*	*۸۵/۵۵ ۲/۳۷*	*۸۲/۶۰ ۵/۴۷*	*۸۳/۹۵ ۴/۱۷*	*۹۰/۰۱ ۲/۱۴*	*۸۴/۶۳ ۲/۶۶*	*۸۶/۲۱ ۲/۲۳*
	--	--	--	--	*۶/۱ ۰/۴*	*۳/۷ ۰/۹*	*۲/۵ ۰/۵*
	۴۷۶۴۲۹	*۲۵۳۲*	*۲۵۴۴۶۴*	*۵۷۷۷۱۴*	*۲۰۳۰۰۰*	*۱۸۸۱۴۳*	*۱۵۹۵۷۱*

⊕ میانگین صحت دسته‌بندی ⊖ انحراف از معیار صحت دسته‌بندی ⊗ میانگین تعداد نرون‌ها ⊙ انحراف از معیار تعداد نرون‌ها

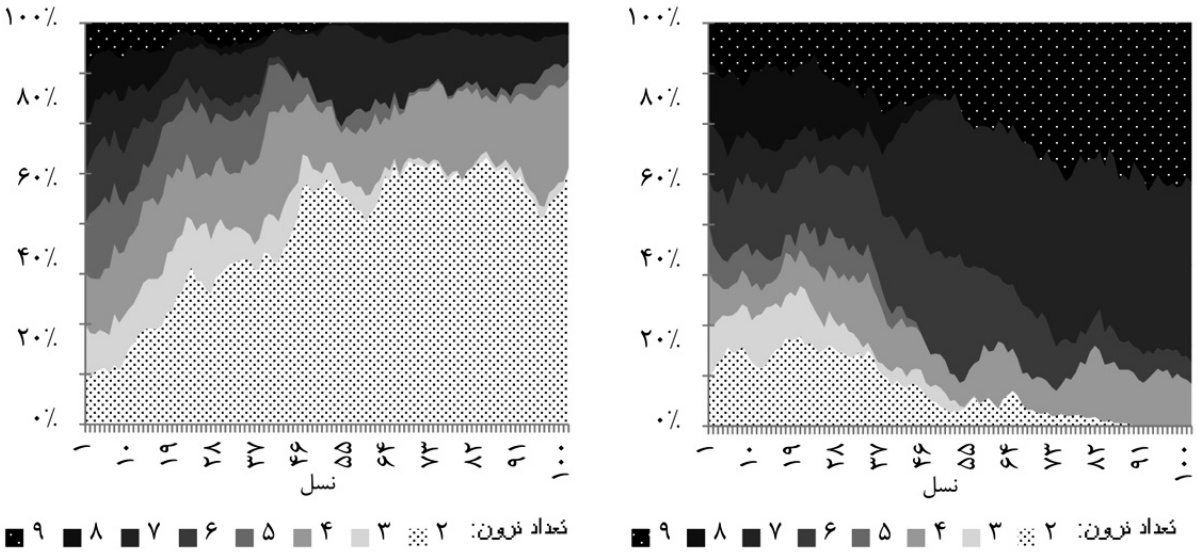
† تعداد فراخوانی‌های تابع برازندگی

www.SID.ir

Archive of SID

(جدول- ۷): اثر مؤلفه‌های مختلف بر عملکرد الگوریتم ارائه شده

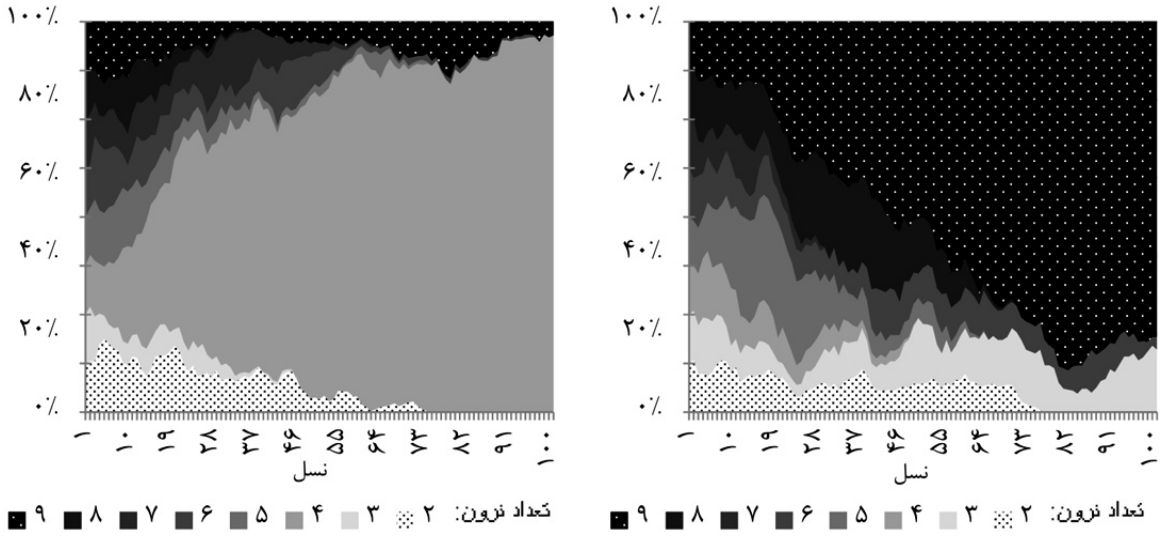
مجموعه داده		الگوریتم اصلی		نسخه‌ی تک‌جمعیتی		نسخه‌ی تک‌جمعیتی بدون ادغام		انتخاب شبکه‌ی برتر مبتنی بر MSE		الگوریتم اصلی بدون ادغام	
		میانگین	انحراف از معیار	میانگین	انحراف از معیار	میانگین	انحراف از معیار	میانگین	انحراف از معیار	میانگین	انحراف از معیار
۱	خطا	۹۶/۱۹	۰/۵۰	۹۶/۱۷	۰/۵۶	**۹۶/۳۹**	۰/۵۷	۹۶/۱۹	۰/۵۰	۹۶/۲۵	۰/۵۰
	تعداد نرون	۲/۰	۰	۴/۳	۲/۹	۴/۷	۱/۵	۴/۰	۱/۳	۲/۲	۰/۶
۲	خطا	۶۸/۶۶	۲/۵۰	۶۹/۰۴	۲/۲۵	۶۹/۰۴	۲/۲۵	**۶۹/۳۲**	۲/۷۳	۶۸/۳۲	۲/۳۲
	تعداد نرون	۳/۲	۱/۰	۵/۲	۲/۴	۵/۲	۲/۴	۴/۸	۱/۹	۲/۹	۱/۰
۳	خطا	۸۰/۶۸	۱/۹۵	۸۰/۰۳	۱/۵۴	۷۸/۳۰	۳/۳۱	۸۰/۱۷	۲/۸۹	**۸۰/۷۳**	۱/۹۳
	تعداد نرون	۲/۱	۰/۳۰	۴/۸	۲/۹	۴/۵	۱/۹	۳/۵	۱/۲	۲/۲	۰/۴
۴	خطا	**۹۰/۱۶**	۲/۴۴	۸۹/۶۴	۲/۵۱	۸۹/۷۴	۳/۵۳	۸۹/۵۱	۳/۴۹	۸۸/۴۹	۱/۷۵
	تعداد نرون	۲/۷	۰/۶	۵/۲	۱/۱	۳/۸	۰/۶	۳/۸	۰/۶	۲/۷	۱/۰
۵	خطا	**۹۶/۶۷**	۲/۰۸	۹۳/۴۶	۲/۱۹	۹۴/۵۳	۱/۷۳	۹۵/۸۶	۱/۴۳	۹۵/۸۶	۱/۸۳
	تعداد نرون	۲/۰	۰/۰	۲/۴	۰/۵	۳/۰	۱/۸	۲/۴	۰/۵	۲/۲	۰/۴
۶	خطا	۷۵/۸۶	۴/۴۵	۷۴/۸۰	۳/۴۹	۷۵/۹۶	۴/۵۱	**۷۶/۵۳**	۳/۱۶	۷۵/۷۷	۳/۱۲
	تعداد نرون	۳/۱	۰/۸	۴/۶	۲/۳	۵/۱	۲/۰	۵/۲	۱/۳	۲/۹	۱/۱
۷	خطا	**۹۵/۲۸**	۱/۷۲	۹۳/۳۷	۲/۹۱	۹۳/۱۴	۲/۸۰	۹۵/۱۶	۱/۵۰	۹۳/۹۳	۳/۸۰
	تعداد نرون	۲/۴	۰/۹	۳/۴	۱/۴	۴/۱	۲/۳	۳/۴	۱/۶	۲/۴	۰/۹
میانگین	*خطا*	*۸۶/۲۱*	*۲/۲۳*	*۸۵/۲۲*	*۲/۲۱*	*۸۵/۳۰*	*۲/۶۷*	*۸۶/۱۱*	*۲/۲۴*	*۸۵/۶۲*	*۲/۱۸*
	تعداد نرون	*۲/۵*	*۰/۵*	*۴/۳*	*۱/۹*	*۴/۳*	*۱/۸*	*۳/۹*	*۱/۲*	*۲/۵*	*۰/۸*



(شکل- ۳): درصد حضور شبکه‌های با تعداد نرون مختلف در جمعیت در طی ۱۰۰ نسل متوالی
(حاصل از آزمایش نسخه‌ی تک‌جمعیتی بر روی مجموعه داده اول)

www.SID.ir

Archive of SID



(شکل- ۴): درصد حضور شبکه‌های با تعداد نرون مختلف در جمعیت در طی ۱۰۰ نسل متوالی
(حاصل از آزمایش نسخه‌ی تک‌جمعیتی بر روی مجموعه داده سوم)

این نسبت در جمعیت نخستین یکسان انتخاب شده است؛ اما به‌تدریج و در اثر تکامل، دیده می‌شود که بعضی از گونه‌ها به‌سرعت اشاعه پیدا کرده و جمعیت را پرمی‌کنند؛ درحالی‌که گونه یا گونه‌های خاصی ممکن است به‌طورکامل از جمعیت حذف شوند. این امر می‌تواند عملکرد پایین این نسخه را توجیه کند. با دقت به این شکل‌ها نمایان است که رفتار الگوریتم در اجراهای مختلف می‌تواند متفاوت باشد. به‌عنوان مثال، نمودار سمت راست شکل ۳ نشان می‌دهد که شبکه‌های با دو نرون در نسل نود از جمعیت حذف شده‌اند، درحالی‌که در اجرای دیگری که در نمودار سمت چپ نمایش یافته است، شبکه‌های با دو نرون در نسل‌های آخر بیش‌ترین سهم را به خود اختصاص داده‌اند. در شکل ۴ نیز این رفتار الگوریتم به‌سادگی قابل درک است. با درنظرگرفتن نتایج جدول ۷ و نمودارهای مورد اشاره، لزوم چندجمعیتی‌بودن برای تکامل معماری شبکه‌های عصبی به‌خوبی مشخص می‌شود. همان‌طور که از نتایج برداشت می‌شود، الگوریتم ارائه‌شده به‌دلیل شروع از شبکه‌های با کمترین تعداد نرون و به‌کارگیری روش انتخاب جواب برتر براساس دو معیار خطا و پیچیدگی، توانسته است، شبکه‌هایی ساده را ایجاد کند. لازم به ذکر است که استفاده از قواعد سرانگشتی برای تعیین تعداد نرون‌های میانی می‌تواند به شبکه‌های بسیار پیچیده‌تری منجر شود. به‌عنوان نمونه، طبق یک قاعدهٔ سرانگشتی مرسوم، می‌توان تعداد نرون‌های میانی را دو برابر تعداد نرون‌های ورودی درنظر گرفت؛ بر طبق این قاعده، باید هیجده نرون میانی برای مجموعهٔ داده نخست و ۱۲۴ نرون میانی برای مجموعهٔ دادهٔ دوم به‌کار برد. درحالی‌که تعداد نرون‌های شبکه‌های حاصل از این الگوریتم به‌طور بسیار چشم‌گیری کمتر است.

در ادامه، اثر انتخاب شبکهٔ برتر با دو معیار خطا و پیچیدگی مورد بررسی قرار می‌گیرد. برای این‌کار، الگوریتم ارائه‌شده به‌گونه‌ای که فقط از معیار خطا برای انتخاب شبکهٔ برتر استفاده کند، نیز مورد ارزیابی قرار گرفته است. نتایج حاصل از آزمایش این نسخه از الگوریتم در ستون ماقبل آخر جدول ۷ آورده شده است. همان‌گونه که پیداست، انتخاب مبتنی بر MSE فقط در دو مورد از هفت مجموعه از داده‌ها منجر به صحت دسته‌بندی بیشتری نسبت به الگوریتم اصلی شده است؛ و همان‌طور که انتظار می‌رفت، شبکه‌های پیچیده‌تری تولید کرده است.

همچنین، نتایج حاصل از الگوریتم پیشنهادی بدون عمل‌گر ادغام در آخرین ستون جدول ۷ ارائه شده است. بر خلاف نظر بعضی از پژوهش‌گران که عمل‌گر ادغام را مخرب معرفی کرده‌اند، مقایسهٔ نتایج حاکی از تأثیر مثبت عمل‌گر ادغام در کارایی این الگوریتم است.

آزمایش‌های انجام‌شده، نشان می‌دهند که طراحی معماری شبکه با استفاده از جمعیت‌های جداگانه (موازی) از آن جهت که به معماری‌های مختلف فرصت یکسانی می‌دهد، برتری به‌طورکامل مشخصی بر الگوریتم‌های تک‌جمعیتی دارد. این عامل همراه با انتخاب جواب نهایی مبتنی بر دو معیار خطای آموزش و سادگی شبکه می‌تواند به شبکه‌های آموزش‌دیده و ساده با قدرت تعمیم بالا منجر شود.

www.SID.ir

Archive of SID

همچنین، مؤثربودن عمل‌گر ادغام در الگوریتم پیشنهادی در کنار عوامل مذکور، سبب برتری نسبی آن بر الگوریتم‌های رقیب است (همان‌گونه که در بخش ۴ مشاهده شد).

۶- نتیجه‌گیری

دقت و قابلیت تعمیم یک شبکهٔ عصبی به طرح معماری و نوع آموزش آن بستگی دارد. در این مقاله، الگوریتمی برای تعیین پیچیدگی و آموزش شبکه‌های عصبی ارائه شده است که از استراتژی تکاملی با جمعیت‌های موازی بهره می‌برد. نتایج روش پیشنهادی برای حل مسائل دسته‌بندی استاندارد با الگوریتم‌های مختلف مقایسه شده است که دقت و پایداری این روش را نسبت به دیگر روش‌ها نشان می‌دهد. الگوریتم ارائه شده قادر است شبکه‌های ساده با قابلیت تعمیم بالا تولید کند. همچنین، آزمایش‌هایی برای نشان‌دادن اثربخشی جمعیت‌های موازی، انتخاب شبکهٔ برتر با دو معیار خطا و پیچیدگی و نیز عمل‌گر ادغام در کارایی الگوریتم پیشنهادی صورت گرفته است.

پیشنهاد برای پژوهش آتی: از آن‌جایی که در روش پیشنهادی، چندین جمعیت تکامل، داده شده و همچنین نوع جواب‌های این جمعیت‌ها از نظر معماری با هم متفاوت هستند، ترکیب جواب‌های (شبکه‌های عصبی) این جمعیت‌ها می‌تواند منجر به شبکه‌ای با قدرت تعمیم بیشتر شود. روش‌های مختلفی برای انجام چنین ترکیبی وجود دارد که در ادبیات شبکه‌های عصبی تجمعی می‌تواند دنبال شود.

۷- مراجع

سعیدی، ساره، توحیدخواه، فرزاد، مدل شبکهٔ عصبی از نگاشت سلول‌های شبکه به سلول‌های مکانی، پردازش علائم و داده‌ها، ۱۳۸۸، (۲) ۱۲، ۵۳-۶۲.

محمدزاده، جواد، مسعودنیا، سعید، آرانی، علی، افزایش نرخ طبقه‌بندی با استفاده از تجمیع ویژگی‌های موثر روش‌های مختلف ترکیب شبکه‌های عصبی، پردازش علائم و داده‌ها، ۱۳۹۰، (۲) ۱۶، ۱۰۱-۱۱۴.

Angeline, P.J., Saunders, G.M., Pollack, J.B., An evolutionary algorithm that constructs recurrent neural networks, IEEE Transactions on Neural Networks, 1994, 5, 54-65.

Blake, C.L., Merz., C.J., UCI Repository of Machine Learning Databases, 1998.

Cantu-Paz, E., Kamath, C., An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 2005, 35, 915-927.

Castellani, M., Evolutionary generation of neural network classifiers – An empirical comparison, Neurocomputing, 2013, 99, 214-229.

Deb, K., Anand, A., Joshi, D., A computationally efficient evolutionary algorithm for real-parameter optimization, Evolutionary Computation, 2002, 10, 371-395.

Haflidason, S., On the significance of the permutation problem in neuroevolution, in the School of Computer Science, The University of Manchester, 2010.

Hancock, P.J.B., Genetic Algorithms and permutation problems: a comparison of recombination operators for neural net structure specification, In: Proceedings of the International Workshop on Combination of Genetic Algorithms and Neural Networks, IEEE Computer Society Press, 1992, pp. 108-122.

Haykin, S., Neural Networks: A Comprehensive Foundation, Macmillan, 1994.

Hertz, J., Introduction to the theory of neural computation (Santa Fe Institute Studies in the Sciences of Complexity), Westview Press, 1991.

Kitano, H., Designing neural networks using genetic algorithms with graph generation system, Complex Systems Journal, 1990, 4, 461-476.

Lang, K.J., Waibel, A.H., Hinton, G.E., A time-delay neural network architecture for isolated word recognition, Neural Networks, 1990, 3, 23-43.

Mitchell, M., An introduction to genetic algorithms (Complex Adaptive Systems), The MIT Press, 1998.

Montana, D. Davis, L., Training feedforward neural networks using genetic algorithms, In: Proceedings of the 11th International Joint Conference on Artificial Intelligence, 1989, pp. 762-767.

Rivero, D., Dorado, J., Rabuñal, J., Pazos, A., Generation and simplification of artificial neural networks by means of genetic programming, Neurocomputing, 2010, 73, 3200-3223.

Siddiqi, A.A., Lucas, S.M., A comparison of matrix rewriting versus direct encoding for evolving neural networks, In: Proceedings of the IEEE International Conference on Evolutionary Computation, 1998, pp. 392-397.

Stanley, K.O., Efficient evolution of neural networks through complexification, The University of Texas at Austin, 2004.

Stanley, K.O., D' Ambrosio, D.B., Gauci, J., A hypercube-based indirect encoding for evolving large-scale neural networks, Artificial Life, 2009, 15, 185-212.

Stanley, K.O., Miikkulainen, R., Evolving neural

www.SID.ir

Archive of SID

networks through augmenting topologies, Evolutionary Computation, 2002, 10, 99-127.

Tsoulos, I., Gavrilis, D., Glavas, E., Neural network construction and training using grammatical evolution, Neurocomputing, 2008, 72, 269-277.

Whitley, D., The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best, In: Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, CA, 1989, pp. 116-123.

Whitley, D., Starkweather, T., Bogart, C., Genetic algorithms and neural networks: optimizing connections and connectivity, Parallel Computing, 1990, 14, 347-361.

Yao, X., Evolving artificial neural networks, In: Proceedings of the IEEE, 87, 1999, pp. 1423-1447.

Yao, X., Islam, Md.M., Evolving artificial neural network ensembles, IEEE Computational Intelligence Magazine, 2008, 3, 31-42.

فردین احمدی‌زر مدرک کارشناسی و کارشناسی ارشد خود را به‌ترتیب در سال‌های ۱۳۷۵ و ۱۳۷۷ در رشتهٔ مهندسی صنایع از دانشگاه‌های صنعتی امیرکبیر و صنعتی اصفهان اخذ کرد. همچنین مدرک دکترای خود را در سال ۱۳۸۸ در رشتهٔ مهندسی صنایع از دانشگاه علم و صنعت ایران دریافت کرد. زمینه‌های پژوهشی مورد علاقهٔ ایشان زمان‌بندی، بهینه‌سازی ترکیباتی و الگوریتم‌های ترکیبی بوده و در حال حاضر عضو هیأت علمی با مرتبه دانشیار در دانشگاه کردستان است.

نشانی رایانامهٔ ایشان عبارت است از:

f.ahmadizar@uok.ac.ir

خه‌بات سلطانیان مدرک کارشناسی نرم‌افزار را در سال ۱۳۸۸ از دانشکدهٔ فنی و مهندسی شهید باهنر شی راز و مدرک کارشناسی ارشد مهندسی کامپیوتر- گرایش هوش مصنوعی را در سال ۱۳۹۰ از دانشگاه کردستان دریافت کرد. زمینه‌های پژوهشی مورد علاقهٔ ایشان شبکه‌های عصبی مصنوعی، الگوریتم‌های تکاملی، سامانه‌های یادگیر گروهی و یادگیری ماشین نیمه‌نظارتی است.

نشانی رایانامهٔ ایشان عبارت است از:

khabat.uok@gmail.com

فردین اخلاقیان‌طاب مدرک کارشناسی مهندسی الکترونیک را در سال ۱۳۶۹ از دانشگاه صنعتی اصفهان و مدرک کارشناسی ارشد مهندسی سیستم‌های مخابراتی را در سال ۱۳۷۳ از دانشگاه تربیت مدرس تهران اخذ کرد. همچنین مدرک دکترای خود را در سال ۱۳۸۵ در رشتهٔ مهندسی برق – گرایش پردازش تصویر از دانشگاه ولنگنگ استرالیا دریافت کرد. زمینه‌های پژوهشی مورد علاقهٔ ایشان بینایی ماشین، پردازش تصویر، شناسایی آماری الگو، یادگیری ماشین و داده‌کاوی بوده و در حال حاضر عضو هیأت علمی با مرتبهٔ استادیار در دانشگاه کردستان است.

نشانی رایانامهٔ ایشان عبارت است از:

f.akhlaghian@uok.ac.ir

www.SID.ir