

طراحی و پیاده‌سازی دسته‌بندی‌کننده دقیق اهداف مبتنی بر شبکه عصبی تکاملی

سید محمدرضا موسوی میرکلایی^{۱*}، مسعود کاوه^۲، الهه سادات نمازی لواسانی^۳ و محمد خویشه^۴

۱- استاد دانشکده مهندسی برق دانشگاه علم و صنعت ایران

۲- دانشجوی دکتری دانشکده مهندسی برق دانشگاه علم و صنعت ایران

۳- دانشجوی کارشناسی ارشد دانشکده مهندسی برق دانشگاه علم و صنعت ایران

۴- استادیار دانشکده مهندسی برق دانشگاه علوم دریایی امام خمینی (ره) نوشهر

چکیده

پیچیدگی فیزیکی اهداف و شباهت بسیار زیادشان دسته‌بندی آن‌ها، یکی از مسائل چالش برانگیز برای پژوهشگران است. شبکه‌های عصبی چندلایه، یکی از پرکاربردترین ابزار در دسته‌بندی اهداف می‌باشند. می‌توان از آموزش به‌عنوان مهم‌ترین بخش این شبکه‌ها اشاره نمود. در سال‌های اخیر به دلیل ضعف‌های زیاد روش‌های سنتی مانند روش‌های مبتنی بر مشتق و گرادینان نزولی، استفاده از الگوریتم‌های تکاملی برای آموزش این نوع شبکه‌ها بسیار مرسوم گشته است. هدف در این مقاله، استفاده از الگوریتم بهینه شده مبتنی بر جغرافیای زیستی، برای آموزش شبکه‌های عصبی چندلایه، به منظور دسته‌بندی اهداف می‌باشد. شبیه‌سازی و مقایسه نتایج نشان می‌دهند که الگوریتم بهینه شده مبتنی بر جغرافیای زیستی در این مقاله موجب طراحی یک دسته‌بندی‌کننده بهینه برای دسته‌بندی اهداف شده است. به این منظور دسته‌بندی بلادرنگ دسته‌بندی‌کننده طراحی شده بر روی تراشه FPGA با استفاده از ابزار Xilinx System Generator پیاده‌سازی می‌گردد. در نهایت نیز دسته‌بندی اهداف Iris و Lenses با استفاده از شبکه عصبی به‌صورت سخت‌افزاری و بر بستر تراشه‌های Xilinx Spartan6-XC6SLX45-FGG484-2 و Xilinx Kintex C7K32T-2FFG900C پیاده‌سازی خواهد شد.

کلمات کلیدی: شبکه عصبی، دسته‌بندی‌کننده، الگوریتم‌های فرا ابتکاری، الگوریتم BBO، FPGA.

۱. مقدمه

شبکه‌های عصبی یکی از پر استفاده‌ترین ابزار در دسته‌بندی اهداف می‌باشند. سادگی، هزینه‌ی محاسباتی کم و کارایی بالا، آن‌ها را یک ابزار محاسباتی محبوب ساخته است. به‌طور کلی، شبکه‌های عصبی مصنوعی چندلایه به‌منظور دسته‌بندی الگو، پیش‌بینی داده، تقریب زدن توابع و کنترل هوشمند مورد استفاده قرار می‌گیرند. شبکه‌های عصبی مصنوعی یکی از ابزارهای بسیار پر کاربرد هوش مصنوعی است. با رشد روزافزون اندازه مسائل دنیای واقعی، شبکه‌های عصبی به‌عنوان یک ابزار مدل‌سازی می‌تواند در حل مسائل مختلفی از قبیل مسائل پیچیده مهندسی و مدیریتی مورد

* m_mosavi@iust.ac.ir

استفاده واقع شود [۳-۱]. صرف نظر از کاربردها، توانایی متمایز شبکه‌های عصبی مصنوعی چندلایه، یادگیری می‌باشد. یادگیری به این معنی است که این شبکه‌ها همانند مغز انسان می‌توانند از یک تجربه یا آزمایش یاد بگیرند. این ویژگی (یادگیری) بخش ضروری همه‌ی شبکه‌های عصبی است که ممکن است به دو نوع تقسیم گردد: یادگیری با نظارت و یادگیری بدون نظارت [۴]. برای شبکه‌های عصبی مصنوعی چندلایه (در بیشتر کاربردها)، از الگوریتم‌های پس‌انتشار بهینه‌شده و یا استاندارد به‌عنوان روش یادگیری استفاده می‌گردد که از خانواده‌ی یادگیری با نظارت می‌باشند [۵]. الگوریتم پس‌انتشار، بر مبنای گرادینان است که اشکالاتی همچون همگرایی آهسته و بکارگیری در یک محدوده‌ی کوچک را دارد و بنابراین برای کاربردهای عملی قابل اعتماد نیست.

هدف نهایی فرآیند یادگیری در شبکه‌های عصبی، پیدا کردن بهترین ترکیب از وزن‌ها و بایاس آن‌ها است به‌طوری که در آموزش شبکه و نمونه‌های آزمون، کمترین مقدار خطا را داشته باشیم [۶]. مطالعات جدید نشان می‌دهد که الگوریتم جستجوی ابتکاری یا اکتشافی می‌تواند جایگزین الگوریتم‌های یادگیری مبتنی بر گرادینان باشد، زیرا ماهیت تصادفی این الگوریتم‌ها اجازه می‌دهد تا حداقل خطای بهتری نسبت به روش‌های مبتنی بر گرادینان داشته باشیم. به‌علاوه، همان‌طوری که بررسی خواهد شد، نرخ همگرایی الگوریتم جستجوی تکاملی می‌تواند سریع‌تر از الگوریتم پس‌انتشار باشد. روش‌های تکاملی گوناگونی از قبیل بهینه‌سازی گروهی ذرات^۱، الگوریتم ژنتیک^۲، الگوریتم کلونی مورچه‌ها^۳ و الگوریتم‌های تکامل‌پذیر^۴ برای آموزش شبکه‌های عصبی بکار گرفته شده است [۱۰-۶]. الگوریتم بهینه‌شده بر مبنای جغرافیای زیستی^۵ یک الگوریتم تکاملی است که مکانیزم تکاملی خاصی را برای هر فرد در یک جمعیت پیشنهاد می‌کند. این الگوریتم که بر پایه علم جغرافیای زیستی شکل گرفته است، به بررسی چگونگی توزیع گونه‌ها در یک جمعیت از زیستگاه‌ها می‌پردازد. الگوریتم BBO دارای سه عمل‌گر اصلی نخبه‌گرایی^۶، مهاجرت^۷ و جهش^۸ می‌باشد. عمل‌گر مهاجرت مسئولیت به اشتراک گذاشتن ویژگی‌ها در میان راه حل‌ها و تعدیل پتانسیل بهره‌برداری الگوریتم را بر عهده دارد. همچنین نرخ جهش وظیفه ایجاد تنوع در میان جمعیت و کنترل قابلیت اکتشاف را دارا می‌باشد [۱۱].

در این مقاله از شبکه عصبی ادراکی چندلایه^۹ به عنوان دسته‌بندی کننده و از الگوریتم‌های تکاملی به منظور آموزش آن استفاده می‌گردد. با توجه به ساختار موازی MLP، پیاده‌سازی سخت‌افزاری آن بر بستر FPGA باعث بهبود بسیار زیاد از لحاظ سرعت و زمان پردازش خواهد گردید. اما نقطه چالش برانگیز MLP برای تعیین ساختار و الگوریتم یادگیری آن است، چرا که با توجه به ابعاد بالای برخی داده‌ها نیاز به پردازش بلادرنگ باید از روش‌هایی برای آموزش MLP استفاده کرد که هم قدرت دسته‌بندی مناسبی داشته و هم پیچیدگی زیادی نداشته باشند. با توجه به موارد بیان شده در این مقاله از الگوریتم‌های فراابتکاری با ساختار تصادفی (به‌جای ساختار معین الگوریتم‌های آموزش سنتی) استفاده شد. برای طراحی

^۱ Particle Swarm Optimization (PSO)

^۲ Genetic Algorithm (GA)

^۳ Ant Colony Optimization (ACO)

^۴ Evolutionary Strategies (ES)

^۵ Biogeography-Based Optimization (BBO)

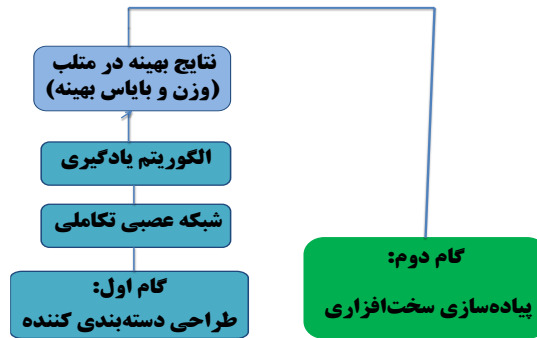
^۶ Elitism

^۷ Migration

^۸ Mutation

^۹ Multi-Layer Perceptron Neural Network (MLP NN)

این دسته‌بندی‌کننده دادگان با ابعاد متوسط Iris و دادگان با ابعاد کم Lense دو گام اصلی طرح‌ریزی گردیده است. روند کلی طراحی دسته‌بندی‌کننده در شکل (۱) نمایش داده شده است.



شکل (۱): شمای کلی دسته‌بندی‌کننده طراحی شده.

۲. آموزش شبکه عصبی با استفاده از الگوریتم BBO

الگوریتم BBO برای اولین بار در سال ۲۰۰۸ پیشنهاد گردید. ایده اصلی این الگوریتم از رشته‌ای در زیست‌شناسی الهام گرفته شده است که درباره طرز انتشار و پخش حیوانات و نباتات (در زمان و مکان) بحث می‌کند. در این زمینه، اکوسیستم‌های متفاوت (محل سکونت یا قلمروها)، برای یافتن ارتباط بین گونه‌های مختلف (ساکنین) بر حسب مهاجرت به بیرون^۱، مهاجرت به داخل^۲ و جهش بررسی می‌شوند [۱۵-۱۲]. تکامل اکوسیستم‌ها با در نظر گرفتن انواع گونه‌های مختلف و تأثیر مهاجرت و جهش برای رسیدن به یک شرایط پایدار، زیربنای الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی می‌باشد. همانند الگوریتم ژنتیک، الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی، از تعدادی عوامل جست‌وجو به نام محل‌های سکونت^۳ استفاده می‌کند. این محل‌های سکونت، مشابه کروموزوم‌ها الگوریتم ژنتیک هستند. الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی، هر محل سکونت را به‌صورت بردارهایی از ساکنین^۴ (مشابه ژن‌ها در الگوریتم ژنتیک) در نظر می‌گیرد که متغیرهای مسأله را نشان می‌دهند. به‌علاوه برای هر محل سکونت، شاخص مناسب بودن محل سکونت^۵ نیز تعریف می‌گردد. بالا بودن این شاخص به منزله داشتن شرایط بهتر می‌باشد. در هر زمان محل‌های سکونت براساس سه قانون اصلی به شرح زیر تعیین می‌گردند:

الف) ساکنینی که در مکان‌های با HSI بالا اقامت دارند، بیشتر تمایل به مهاجرت به مکان‌هایی با HSI پایین‌تر دارند.

ب) ساکنینی موجود در مکان‌های با HSI پایین، تمایل بیشتری به جذب مهاجران از مکان‌های با HSI بالا دارند.

ج) مکان‌ها بدون توجه به مقدار شاخص خود باید به‌صورت تصادفی ساکنان‌شان را تغییر دهند.

در طبیعت، این پدیده موجب ایجاد تعادل میان اکوسیستم‌های مختلف می‌گردد. به‌عبارت دیگر طبیعت به بهبود پایداری نواحی زیستی مختلف تمایل دارد. الگوریتم BBO، این مفاهیم را برای بهبود شاخص همه‌ی مکان‌های زیستی به‌کار می‌برد. در ادامه سه عمل‌گر مهاجرت، جهش و نخبه‌گزینی به‌عنوان عمل‌گرهای اصلی این الگوریتم، مورد بررسی قرار

^۱ Emigration

^۲ Immigration

^۳ Habitat

^۴ Habitant

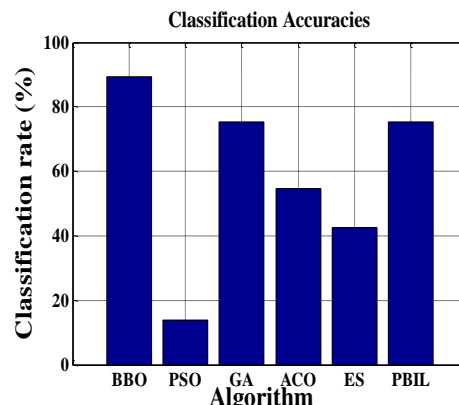
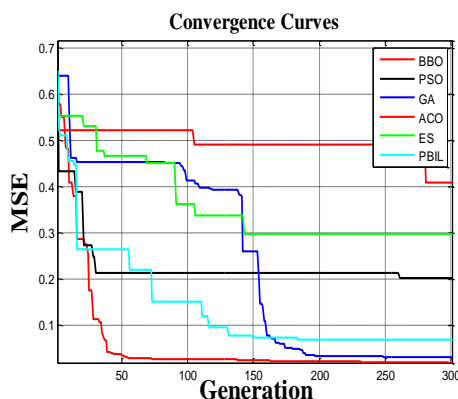
^۵ Habitat Suitability Index (HSI)

می‌گیرند. در این مقاله، الگوریتم‌های فراابتکاری با استفاده از شیوهی به‌کارگیری آن‌ها به‌منظور یافتن ترکیب وزن یال‌ها و بایاس‌گره‌ها برای داشتن کمترین مقدار خطا در یک شبکه‌ی عصبی MLP مورد استفاده قرار می‌گیرند. به‌منظور طراحی یک الگوریتم آموزش‌دهنده برای شبکه‌های عصبی MLP از مراجع [۱۶ و ۱۵] استفاده شده است. در ادامه کارآیی الگوریتم BBO با الگوریتم‌هایی مانند الگوریتم ژنتیک، الگوریتم ازدحام ذرات، الگوریتم کلونی مورچه، استراتژی تکاملی و الگوریتم یادگیری افزایشی مبتنی بر جمعیت مورد مقایسه قرار می‌گیرد. برای معتبر بودن ارزیابی و شبیه‌سازی و جامع بودن مقایسه‌ها، دو نوع مختلف از داده‌ها مورد بررسی قرار می‌گیرند: دادگان با ابعاد کم (Lense) و دادگان با ابعاد متوسط (Iris) [۱۶]. داده‌ها در قسمت‌های بعدی بررسی شده و ویژگی‌های اولیه آن‌ها در جدول (۱) در دسترس می‌باشد. در هر قسمت الگوریتم BBO در موارد دقت دسته‌بندی، سرعت همگرایی و قابلیت اجتناب از گیر کردن در کمینه‌های محلی با دیگر الگوریتم‌های نامبرده، در غالب شکل‌ها و جدول‌ها مقایسه می‌شود. پارامترهای مهم و مقادیر اولیه الگوریتم‌ها [۱۶] قابل مشاهده می‌باشند.

جدول (۱): دادگان مورد استفاده به‌منظور دسته‌بندی اهداف.

سال	ابعاد	ویژگی‌ها	ویژگی مشخصه	نوع داده	کاربرد	نام
۱۹۸۸	۱۵۰	۴	واقعی	چند متغیره	طبقه بندی	Iris
۱۹۹۰	۲۴	۴	قطعی	چند متغیره	طبقه بندی	Lense

دادگان گلبرگ دارای ۱۵۰ نمونه و ۴ ویژگی است که عبارتند از طول گلبرگ، پهنای گلبرگ، طول کاسبرگ و پهنای کاسبرگ می‌باشند. همچنین دادگان گلبرگ می‌تواند به سه دسته، دسته‌بندی گردد. برای دسته‌بندی این دادگان از یک شبکه عصبی ادراکی چند لایه با ساختار (۴،۹،۳) استفاده شده است. شکل (۲) و جدول (۲) نتایج دسته‌بندی این دادگان با استفاده از روش پیشنهادی را نشان می‌دهند که در آن‌ها الگوریتم BBO به‌طور قابل ملاحظه‌ای دارای دقت و سرعت بیشتری می‌باشد.



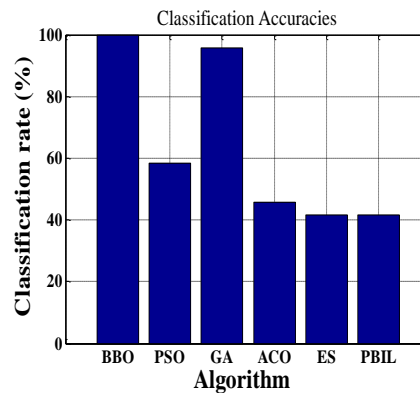
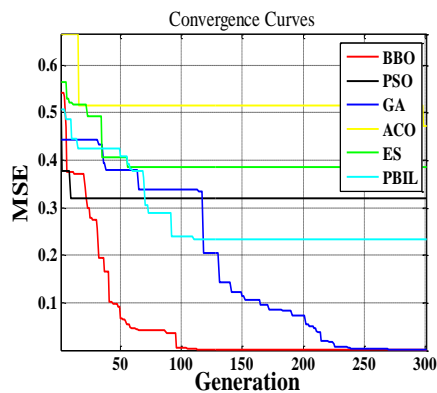
شکل (۲): نرخ دسته‌بندی و سرعت همگرایی برای الگوریتم BBO و دیگر الگوریتم‌ها برای دادگان گلبرگ.

دادگان لنز از ۲۴ نمونه تشکیل شده است و دارای ۳ دسته زیر تقسیم می‌شود: الف) بیمار باید از لنزهای تماسی سخت

استفاده کند. (ب) بیمار باید از لنزهای تماسی نرم استفاده نماید. (ج) بیمار نیازی به استفاده از لنز تماسی ندارد. برای دسته‌بندی این دادگان از یک شبکه عصبی ادراکی چند لایه با ساختار (۳ و ۹ و ۴) استفاده شده است. شکل (۳) و جدول (۳) نتایج دسته‌بندی این دادگان با استفاده از روش پیشنهادی را نشان می‌دهند که در آن‌ها الگوریتم BBO به‌طور قابل ملاحظه‌ای دارای دقت و سرعت بیشتری می‌باشد.

جدول (۳-۳) مقایسه الگوریتم BBO با دیگر الگوریتم‌ها برای دادگان گلبرگ.

الگوریتم	MSE (Mean)	MSE (Variance)	P-value	نرخ دسته‌بندی
BBO	0.017782	0.0122	1	89.33%
PSO	0.20109	0.0034	$1.4616e - 76$	14.00%
GA	0.029297	0.0420	$4.4274e - 66$	75.33%
ACO	0.40933	$7.9052e - 04$	$2.5940e - 93$	54.66%
ES	0.29785	0.0076	$9.5441e - 86$	42.66%
PBIL	0.06742	0.0118	$3.4865e - 64$	75.33%



شکل (۳) : نرخ دسته‌بندی و سرعت همگرایی برای الگوریتم BBO و دیگر الگوریتم‌ها برای دادگان لنز.

جدول (۳) : مقایسه الگوریتم BBO با دیگر الگوریتم‌ها برای دادگان لنز.

الگوریتم	MSE (Mean)	MSE (Variance)	P-value	نرخ دسته‌بندی
BBO	$6.3088e - 06$	0.0134	1	100%
PSO	0.31912	$1.8302e - 04$	$2.4265e - 83$	58.33%
GA	0.00042269	0.0275	$3.2687e - 43$	95.83%
ACO	0.47252	0.0011	$5.9707e - 106$	45.83%

ES	0.38481	0.0018	$1.1833e - 95$	41.66%
PBIL	0.23424	0.0065	$5.6037e - 73$	41.66%

دلیل کارایی بهتر الگوریتم BBO در مقایسه با الگوریتم ژنتیک در اکثر مسائل، نرخ‌های متفاوت مهاجرت به بیرون و مهاجرت به داخل هر زیستگاه‌ها است. الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی در مقایسه با الگوریتم ژنتیک که برای همه‌ی زیستگاه‌های جامعه آماریش، یک نرخ تولید مجدد کلی دارد (ادغام)، دو نرخ (مهاجرت به بیرون و مهاجرت به داخل) برای هر زیستگاه دارد. این مسأله باعث ایجاد رفتار تکاملی و قدرت شناسایی متفاوت می‌گردد. نقطه قوت الگوریتم BBO در قدرت اکتشاف ایجاد تولید جمعیت‌های جدید و توانایی بهره‌برداری است که تا حد زیادی این مهم را مدیون عملگر مهاجرت خود با دو پارامتر کنترلی مهاجرت به داخل و مهاجرت به خارج می‌باشد. به‌طور کلی آموزش شبکه‌های عصبی یک چالش برای مسائل بهینه‌سازی غیرخطی می‌باشد. لذا در این مقاله برای آموزش شبکه عصبی از الگوریتم‌های فراابتکاری استفاده شده است.

۳. پیاده‌سازی سخت‌افزاری

برای رسیدن به هدف پردازش بلادرنگ، طراحی دسته‌بندی کننده (شبکه عصبی) به‌صورت سخت‌افزاری و بر بستر تراشه‌های Xilinx Kintex C7K32T-2FFG900C و Xilinx Spartan6-XC6SLX45-FGG484-2 خواهد شد. البته با توجه به نحوه‌ی طراحی و با توجه به تحلیل صورت گرفته در مورد سطح تراشه مصرفی و فرکانس کاری مدار طراحی شده می‌توان این مدار را بر روی تراشه‌های ارزان قیمت‌تر با منابع سخت‌افزاری کمتر DSP48E1s پیاده‌سازی نمود. هم‌چنین در این فصل پیچیدگی محاسباتی دسته‌بندی کننده‌های مبتنی بر BBO محاسبه خواهد شد. طراحی سخت‌افزاری Spartan6 در محیط ISE و Kintex7 در محیط Vivado انجام می‌پذیرد.

شبکه عصبی عمدتاً در نرم‌افزار پیاده‌سازی شده‌اند، هر چند نسخه سخت‌افزاری دارای اهمیت هستند. پیاده‌سازی نرم‌افزاری انعطاف‌پذیری برای طیف گسترده‌ای از مدل شبکه‌های عصبی را ارائه می‌دهد، در حالی که اجرای سخت‌افزار شبکه‌های عصبی در FPGA به دو دلیل مورد توجه بیشتری قرار گرفته است. اول اینکه اگرچه عملکرد پردازشگر معمولی به‌طور مستمر بهبود می‌یابد، سریع‌ترین پردازشگر متوالی هنوز قادر به ارائه پاسخ بلادرنگ برای شبکه عصبی با تعداد زیادی از نورون‌ها نیست. دوم، اغلب پیاده‌سازی‌ها به شکل شبیه‌سازی‌های در حال اجرا بر روی کامپیوتر شخصی به خاطر فقدان نرم‌افزار قابل حمل برای اجرای شبکه عصبی انجام شده‌اند. به همین ترتیب، شبکه عصبی را می‌توان در FPGA اجرا کرد، زیرا پلت فرم قابل حمل بیشتری برای کاربردهای دنیای واقعی متعدد فراهم می‌کند. FPGA به‌طور کامل قابل برنامه‌ریزی است و ساختار موازی شبکه عصبی آن را به‌طور بالقوه برای محاسبه یک کار خاص سریع می‌سازد و منجر به کاهش زمان پردازش می‌شود. با این حال، اجرای شبکه عصبی در FPGA به سادگی اجرا در نرم‌افزار نیست به خصوص زمانی که شامل تعداد زیادی نورون و محاسبه معادله پیچیده مانند تابع فعال‌ساز می‌شود. یکی از عناصر محاسباتی مهم در یک شبکه عصبی چند لایه پرسپترون تابع فعال‌سازی غیرخطی سیگموئید است. در این مقاله، تابع سیگموئید به‌عنوان تابع فعال‌سازی مورد استفاده قرار می‌گیرد. این جز مهم از شبکه عصبی است که عمدتاً برای محدود کردن خروجی نورون است. تابع سیگموئید به راحتی می‌تواند در مدارهای آنالوگ به دلیل ویژگی‌های غیرخطی ذاتی در دستگاه‌های CMOS

اجرا شوند. در کاربردهای دیجیتال تقریب سیگموئید را می‌توان از یک جدول جست‌وجو^۱ و یا یک محاسبه مستقیم به دست آورد. از آنجا که سخت‌افزار قادر به محاسبه عدد شناور در مقایسه با نرم‌افزار نیست، به خصوص زمانی که شامل تابع فعال‌ساز می‌شود، روش جدید برای محاسبه مورد نیاز است و در عین حال خروجی مشابه محاسبات نرم‌افزاری است. دسته‌بندی کننده طراحی شده در محیط Xilinx System Generator (XSG) پیاده‌سازی گردید. پیاده‌سازی شبکه‌های عصبی تکاملی در دو مرحله انجام شد. بدین صورت که در مرحله اول در محیط نرم‌افزار Matlab و به صورت ممیز شناور^۲ با دقت ۳۲ بیت انجام گردید و پس از ثابت شدن وزن‌ها و بایاس‌ها مقادیر بهینه آن به فرمت ممیز ثابت^۳ با دقت ۱۸ بیت تبدیل شد و برای ساخت شبکه عصبی تکاملی در محیط XSG استفاده گردید. در رویکرد جست‌وجوی جدولی، مقدار y در رابطه با هر مقدار x در حافظه ذخیره می‌شود و x به عنوان یک آدرس برای جدول جست‌وجو استفاده می‌شود. اگرچه این تکنیک برای پیاده‌سازی ساده است، اما نیاز به یک ناحیه سیلیکونی بزرگ دارد، به خصوص زمانی که دقت بالاتر مورد نیاز است و بنابراین نمی‌توان جدول جست‌وجو به ازای هر عنصر پردازشی در یک شبکه عصبی بسیار موازی را در بر گرفت. منابع سخت‌افزاری یک مدار FPGA برای جداول LUT استفاده شده است تنها در صورتی که حافظه فقط خواندنی FPGA با RAM از پیش تعریف شده پیاده‌سازی نشده است، بسیار بزرگ هستند.

به منظور داشتن نتایج مشابه Matlab کار پردازش با قالب ممیز ثابت با ۳۶ بیت شروع می‌شود، با این حال پهنای بیت‌ها با گذر از بلوک‌های مختلف XSG کاهش پیدا می‌کند. از فرمت FIX-22-17 استفاده شده است که به معنای پهنای ۲۲ بیت با قسمت اعشاری ۱۷ بیتی است. تراشه‌های در نظر گرفته شده برای بردهای پردازشی در نظر گرفته شده با درجه سرعت ۲- هستند. جدول (۴) میزان منابع مصرفی، توان کلی مصرفی و بیشینه فرکانس کاری که توسط Xilinx Vivado اجرا و تایید شده است را نمایش می‌دهد.

جدول (۴): میزان منابع سخت‌افزاری موجود در تراشه‌ها.

Spartan6-XC6SLX45-FGG484-2		Kintex XC7 K32T-2FFG900C		تراشه استفاده شده
استفاده شده	موجود در تراشه	استفاده شده	موجود در تراشه	
۵۶۹	۲۷۲۸۸	۲۱۱۸	۲۰۳۸۰۰	LUT
۸۷	۵۸	۹۹	۸۴۰	DSP
۱۳۷	۳۱۶	۱۳۸	۵۰۰	IO
۱	۱۶	۲	۳۲	BUFG
۱۳۰/۸۹ مگا هرتز		۷۰۶/۲۱ مگا هرتز		حداکثر فرکانس کاری
-		۰/۴۹ وات		مجموع توان مصرفی

معماری مورد نظر از ۹۹ بلوک DSP48E1s در تراشه Kintex XC7 استفاده می‌کند. این موارد با معماری موازی

^۱ Look-Up table

^۲ Floating-point

^۳ Fixed-point

شبکه عصبی MLP که نیازمند ۴ ضرب کننده برای هر نرون لایه مخفی و ۹ ضرب کننده برای هر نرون لایه خروجی است، توجیه می‌شود. بنابراین شبکه عصبی MLP به $9 \times 4 + 3 \times 9$ ضرب کننده ۱۸ بیتی نیاز دارد. با در نظر گرفتن این که هر رشته DSP48E1s شامل یک ضرب کننده 18×25 است، ضرب دو عدد ۳۶، ۲۴ و ۱۸ بیتی (ممیز ثابت) به ترتیب نیازمند ۴، ۲ و ۱ عدد DSP48E1s است. بنابراین کل طرح به $4 \times 1 + 4 \times 2 + 2 \times 4 + 1 \times 4 + 1 \times 63 = 87$ عدد DSP48E1s نیازمند است. بجز تعداد زیاد DSP48E1s این معماری فقط بخش کوچکی از منابع در دسترس این مدل‌های FPGA را مصرف می‌کند که در جدول (۴) نمایش داده شده است. چون تعداد DSP48 مورد نیاز بر روی تراشه Spartan6 به تعداد کافی نیست، پس این شبکه در عمل بر روی بورد Spartan6 قابل پیاده‌سازی نبوده و لذا مجموع توان مصرفی نمی‌توان برایش در نظر گرفت.

برای محاسبه حجم پردازش یا همان پیچیدگی محاسباتی روش‌ها رابطه دقیقی وجود ندارد و فقط می‌توان تخمینی از حجم محاسباتی روش‌ها ارائه نمود. این مسأله موجب می‌شود که نتوان همه روش‌ها را از منظر پیچیدگی محاسباتی با هم مقایسه کرد. در واقع فقط می‌توان پیچیدگی محاسباتی روش‌های را مقایسه نمود. با این وجود در این قسمت سعی می‌شود که تا جای ممکن است پیچیدگی روش‌ها با یک معیار یکسان بررسی شود. الگوریتم‌ها توسط زبان برنامه‌نویسی پیاده‌سازی می‌شوند و هر الگوریتم توسط یک برنامه ارائه می‌شود. هر برنامه نیز مثل الگوریتم زمان اجرای خاص خود را دارد. عوامل دخیل در زمان اجرای برنامه عبارتند از: سرعت سخت‌افزار، نوع کامپایلر، اندازه داده ورودی، ترکیب داده‌های ورودی و پیچیدگی زمانی الگوریتم. از این عوامل سرعت سخت‌افزار و نوع کامپایلر به صورت ثابت در زمان اجرای برنامه‌ها دخیل هستند. پارامتر مهم پیچیدگی زمانی الگوریتم است که خود تابعی از اندازه مسأله می‌باشد. ترکیب داده‌های ورودی نیز با بررسی الگوریتم در شرایط مختلف قابل اندازه‌گیری می‌باشد.

برای بررسی الگوریتم تابعی به نام $O(n)$ که تابع زمانی الگوریتم نامیده می‌شود، در نظر گرفته خواهد شد که در آن n اندازه ورودی مسأله است. مسأله ممکن است شامل چند ورودی باشد که در این صورت زمان اجرای الگوریتم $O(n, m)$ نمایش داده می‌شود. در صورتی که تعداد پارامترها بیشتر باشند، آن‌هایی که اهمیت بیشتری در زمان اجرا دارند، در محاسبات وارد می‌شوند و از بقیه صرف نظر می‌گردد. برای محاسبه تابع $O(n)$ برای یک الگوریتم موارد زیر باید در محاسبات در نظر گرفته شوند:

➤ زمان مربوط به اعمال جایگزینی که مقدار ثابت دارند.

➤ زمان مربوط به انجام اعمال محاسباتی که مقدار ثابت دارند.

➤ زمان مربوط به تکرار تعدادی دستور یا دستورالعمل.

➤ زمان مربوط به توابع بازگشتی.

از موارد بیان شده در محاسبه $O(n)$ یک الگوریتم، محاسبه تعداد تکرار عملیات و توابع بازگشتی اهمیت ویژه‌ای دارند و بیشتر پیچیدگی زمانی مربوط به این دو می‌باشد. در اینجا سعی گردید که پیچیدگی محاسباتی را برای دسته‌بندی‌کننده‌های متعارف با استفاده از مراجع [۱۷ و ۱۸] به دست آید. برای محاسبه پیچیدگی دسته‌بندی‌کننده‌ی استفاده شده در این مقاله باید پیچیدگی شبکه عصبی MLP را به همراه الگوریتم‌های بهینه‌ساز که کار آموزش شبکه عصبی را به عهده دارند، مد نظر قرار داد.

باید توجه کرد که پیچیدگی روش BBO وابسته به تعداد نمونه‌های آموزش در مجموعه داده‌ها، ساختار شبکه عصبی MLP، تعداد زیستگاه‌ها، تعداد نسل‌ها، فرآیند جهش و نخبه‌گزینی است. بنابراین پیچیدگی محاسباتی کلی دسته‌بندی‌کننده‌های مبتنی بر BBO طبق رابطه (۷) محاسبه می‌شود:

$$O(MLP, BBO) = O(g(O(MLP) + O(migration) + O(mutation) + O(elitism))) \quad (1)$$

که در این رابطه، g نشان‌دهنده حداکثر تعداد نسل‌ها می‌باشد. پیچیدگی محاسباتی یک MLP با h نرون لایه مخفی، o نرون خروجی و t نمونه آموزش توسط $O(t(h+o))$ محاسبه می‌شود. پیچیدگی محاسباتی مهاجرت $O(mn^2)$ است که در آن m تعداد ساکنین و n مبین تعداد زیستگاه است. البته پیچیدگی بهترین حالت ممکن مهاجرت $O(mn)$ خواهد بود. پیچیدگی محاسباتی عمل‌گر جهش در بدترین حالت $O(mn)$ است. از آن‌جا که در فاز نخبه‌گزینی از روش مرتب‌کردن سریع^۱ برای مرتب کردن بهترین زیستگاه‌ها استفاده می‌شود، پیچیدگی محاسباتی عمل‌گر نخبه‌گزینی از درجه $O(n \log n)$ در بهترین حالت و $O(n^2)$ در بدترین حالت می‌باشد. بنابراین پیچیدگی محاسباتی کلی دسته‌بندی‌کننده‌های مبتنی بر BBO طبق رابطه (۸) محاسبه می‌شود:

$$O(MLP, BBO) = O(g(t(h+o) + mn^2 + nm + n^2)) \quad (2)$$

که در این رابطه، g مبین بیشینه تعداد نسل‌ها، t نشان‌دهنده تعداد نمونه‌های آموزش، h و o به ترتیب مبین تعداد نرون لایه مخفی و خروجی، m مبین تعداد ساکنین و n نشان‌دهنده تعداد زیستگاه‌ها می‌باشد.

۴. نتیجه‌گیری

در این مقاله یک روش برای دسته‌بندی اهداف مبتنی بر شبکه عصبی MLP پیشنهاد شده است. به منظور آموزش این شبکه، عملکرد الگوریتم‌های مختلف مورد بررسی قرار گرفتند که با توجه به نتایج شبیه‌سازی، الگوریتم BBO بهترین نتایج را در دقت دسته‌بندی و سرعت همگرایی از خود بر جای گذاشت. لذا از این الگوریتم برای آموزش شبکه عصبی و طراحی حالت بهینه دسته‌بندی‌کننده اهداف استفاده گردید. همچنین به منظور پردازش بلادرنگ، دسته‌بندی‌کننده طراحی شده بر روی تراشه FPGA پیاده‌سازی شده و نتایج حاصل از آن نشان‌دهنده انتخاب مناسب شبکه عصبی به دلیل قابلیت اجرای پردازش بلادرنگ به عنوان دسته‌بندی‌کننده می‌باشد. همچنین تحلیل‌های مربوط به پیچیدگی روش پیشنهادی، نشان‌دهنده عملکرد مناسب آن از لحاظ پیچیدگی محاسباتی می‌باشد.

۵. مراجع

- [1] E. Oja, "Unsupervised Learning in Neural Computation," Theoretical Computer Science, Vol.287, pp.187-207, 2002.
- [2] A. Luk, "Fast Convergence for Backpropagation Network with Magnified Gradient Function," IEEE Joint Conference on Neural Networks, Vol.3, pp.1903-1908, 2003.
- [3] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Let a Biogeography-based Optimizer Train Your Multi-Layer Perceptron," Journal of Information Sciences, Vol.269, pp.188-209, June 2014.
- [4] P. N. Le, E. Ambikairajah, J. Epps, V. Sethu and E. H. C. Choi, "Investigation of Spectral Centroid Features for Cognitive Load Classification," Speech Communication, Vol.53, No.4, pp.540-551, 2011.
- [5] P. Auer, H. Burgsteiner and W. Maass, "A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons," Neural Networks, Vol.21, No.5, pp.786-795, June 2008.
- [6] G. Li, J. Na, D. Stoten and X. Ren, "Adaptive Neural Network Feedforward Control for Dynamically Substructured Systems," IEEE Transactions on Control Systems Technology, Vol.22, No.3, pp.944-954, 2014.

^۱ Quicksort



- [7] A. Saxena and A. Saad, "Evolving an Artificial Neural Network Classifier for Condition Monitoring of Rotating Mechanical Systems," *Applied Soft Computing*, Vol.7, pp.441-454, 2007.
- [8] C. Ioana, A. Quinquis and Y. Stepphan, "Feature Extraction from Underwater Signals using Time Frequency Warping Operators," *IEEE Journal of Oceanic Engineering*, Vol.31, No.3, pp.628-646, 2006.
- [9] A. Forero, A. Baxely and L. Straatemeier, "A Multitask Learning Location Mapping Using Passive Sonar," *IEEE Transactions on Signal Processing*, Vol.63, pp.3639-3654, No.14, 2015.
- [10] Y. Chen and Member, "Deep Feature Extraction and Classification of Hyper Spectral Image Based on Convolutional Neural Networks," *IEEE Transactions on GeoScience and Remote Sensing*, Vol.45, No.10, 2016.
- [11] N. Friedman, M. Linial and I. Nachman, "Using Bayesian Networks to Analyze Expression Data," *Journal of Computational*, Vol.7, pp.601-620, 2009.
- [12] E. Wan, "Neural Network Classification: A Bayesian Interpretation," *IEEE Transaction on Neural Networks*, Vol.1, No.4, pp.303-305, 1990.
- [13] Mosavi, M. R. Kaveh, M. and Khishe, M. "Sonar Data Set Classification using MLP Neural Network Trained by Non-linear Migration Rates BBO," *The Fourth Iranian Conference on Engineering Electromagnetics*, pp.1-5. 2016.
- [14] H. Ma and D. Simon, "Analysis of Migration Models of Biogeography-based Optimization using Markov Theory", *Engineering Applications of Artificial Intelligence*, Vol.24, pp.1052-1060, 2011.
- [15] Khishe, M. Mosavi, M. R. and Kaveh, M. "Improved Migration Models of Biogeography-Based Optimization for Sonar Dataset Classification by using Neural Network," *Journal of Applied Acoustics*, Vol.118, pp.15-29. 2017.
- [16] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Let A Biogeography-based Optimizer Train Your Multi Layer Perceptron," *Information Sciences*, Vol.269, pp.188-209, 2014.
- [17] Amin, H. et al., "Piecewise Linear Approximation Applied to Nonlinear Function of a Neural Network," *IEEE Proc. Circuits, Devices Sys*, Vol.144, pp.313-317, 1997.
- [18][60] M. Tommiska, "Efficient Implementation of the Sigmoid Function for Reprogrammable Logic," *IEEE Proceedings-Computers and Digital Techniques*, 2003.