

بهینه‌سازی همزمان شبکه‌ی عصبی بازگشتی برای بهبود شناسایی و مدل‌سازی دینامیک غیرخطی هواپیما

علیرضا رودباری* (دانشجوی دکتری)

فریبرز نقفی (دانشیار)

دانشکده‌ی مهندسی هوافضا، دانشگاه صنعتی شریف

مهندسی مکانیک شریف، (پیاورد ۱۳۹۴)
دوره‌ی ۳ - ۳۱، شماره‌ی ۱، ص. ۳۳-۳۴

در این نوشتار چگونگی استفاده از الگوریتم ژنتیک در آموزش شبکه‌های عصبی، و به‌طور همزمان بهینه‌سازی ساختاری آن‌ها به‌منظور مدل‌سازی دینامیک غیرخطی هواپیماهایی با قابلیت مانور بالا، بررسی می‌شود. ارتباط‌های وزنی، معماری شبکه و قوانین یادگیری از مشخصاتی هستند که نقش بسیار مهمی در کیفیت آموزش و تعمیم شبکه‌های عصبی برای مدل‌سازی سیستم‌های غیرخطی ایفا می‌کنند. لذا تنظیم درست این پارامترها کمک شایانی به بهبود قابلیت تعمیم‌دهی شبکه‌ی عصبی آموزش دیده می‌کند. در این کار از روش الگوریتم ژنتیک عادی و اصلاح شده در کنار ساختارهایی متفاوت از شبکه‌های عصبی استفاده شده است. اعتباربخشی روش از طریق مقایسه و ارزیابی نتایج تحلیلی با داده‌های تجربی حاصل از آزمایش پرواز یک هواپیمای جنگنده‌ی نسل چهارم صورت پذیرفته است. بررسی‌ها نشان‌گر دقت بالای روش در مدل‌سازی دینامیک هواپیماست.

واژگان کلیدی: دینامیک غیرخطی هواپیما، شناسایی سیستم، شبکه‌های عصبی، الگوریتم بهینه‌سازی.

alirezaroudbari@ae.sharif.edu
saghafi@sharif.edu

۱. مقدمه

سابقه‌ی بیش از دو دهه استفاده‌ی گسترده از شبکه‌های عصبی مصنوعی در کاربردهای گوناگون -- نظیر بهینه‌سازی، مدل‌سازی و شناسایی سیستم -- و نیز قابلیت تقریب‌زنی، یادگیری و تعمیم‌دهی سبب شده است که این شبکه‌ها جایگزین مناسبی برای روش‌های کلاسیک شناسایی سیستم‌های غیرخطی باشند.^[۱] اغلب کارهای پژوهشی که تاکنون در زمینه‌ی کاربرد شبکه‌های عصبی در تخمین پارامتر و شناسایی دینامیک وسایل پرنده منتشر شده، در زمینه‌ی طراحی کنترل‌کننده و عمدتاً از نوع تطبیقی آن بوده است.^[۱۱-۱۳] در سال‌های اخیر تحقیقاتی در زمینه‌ی مدل‌سازی و شناسایی سیستم دینامیکی هواپیما به‌کمک شبکه‌ی عصبی انجام شده که نتایج مطلوب حاصل از این تحقیقات حاکی از پتانسیل این روش در چنین کاربردهایی است.^[۱۴-۱۹] با توجه به این توانایی‌ها، می‌توان شبکه‌های عصبی را ابزار مناسبی برای جایگزینی روش‌های موجود مدل‌سازی وسایل پرنده^[۲۰] محسوب کرد.

علی‌رغم موفقیت چشمگیر شبکه‌های عصبی در علوم مختلف، تعیین الگوریتم آموزش و معماری شبکه برای تعمیم‌دهی مناسب در کاربردهای مدل‌سازی، هنوز به‌عنوان یک مشکل اساسی باقی مانده است. مقصود از تعمیم‌دهی، داشتن عملکرد مناسب به‌ازای داده‌هایی است که در آموزش شبکه دخالت نداشته‌اند. تعمیم‌دهی در موضوع شناسایی به‌صورت آفلاین (بدون داشتن پس‌خور از سیستم اصلی) بسیار مهم است. هرچه شبکه‌ی عصبی از قدرت تعمیم‌دهی بالاتری برخوردار باشد، در پیش‌بینی رفتار سیستم معتبرتر خواهد بود. افزودن پس‌خور داخلی باعث

وسایل پرنده از جمله سیستم‌های دینامیکی غیرخطی پیچیده و مزدوج‌اند^۱ که تحت تأثیر شرایط پروازی مختلف قرار دارند. دسترسی به مدل قابل قبولی از رفتار هواپیما -- چه در تونل باد و چه در پرواز -- مستلزم آزمایشات متعددی است. «شناسایی سیستم» روشی است مناسب برای مدل‌سازی که در آن با استفاده از داده‌های به دست آمده از آزمایشات تجربی، مدل ریاضی سیستم تعیین می‌شود. تاکنون روش‌های زیادی در شناسایی سیستم مورد توجه محققین قرار گرفته است.^[۱-۴] روش‌های کلاسیک شناسایی دینامیک غیرخطی، نظیر تخمین کم‌ترین مربعات،^[۵] شبه‌خطی‌سازها^[۶] و مدل‌های تصادفی^[۷] روش‌های مناسبی هستند، اما در این روش‌ها تخمین پارامترهای مورد نیاز سیستم نیازمند داشتن دانش قبلی از ساختار مدل است و معمولاً مدل بهینه‌ی برای دینامیک سیستم تقریب زده نمی‌شود. شناسایی با روش‌های کلاسیک از قبیل مدل‌های چندجمله‌ی، سری‌های ولترا و وینر نیز با پیچیدگی‌های قابل ملاحظه‌ی مواجه است. در روش‌های دیگر شناسایی مانند خودرگرسیون غیرخطی با ورودی از خارج (NARX)^۳، مدل‌های خودرگرسیون غیرخطی با میانگین متحرک^۴ (NARMAX)^[۸] و روش غیرخطی خطای خروجی^۵ (NOE)^[۹] نیز که براساس مدل‌های رگرسیونی به کار گرفته می‌شوند، معمولاً تعیین بردار رگرسیون مشکل است.

* نویسنده مسئول

تاریخ دریافت: ۱۳۹۱/۱۲/۲۳، اصلاحیه ۱۳۹۲/۹/۲۰، پذیرش ۱۳۹۲/۱۱/۹.

«افزایش حافظه‌ی دینامیکی» شبکه می‌شود و این یکی از راهکارهای افزایش قدرت تعمیم‌دهی در مدل‌سازی و شناسایی سیستم‌های دینامیکی غیرخطی است^[۱۹] اگرچه باعث بزرگ‌تر شدن شبکه نیز می‌شود. بزرگ‌تر شدن شبکه به یادگیری سرعت می‌بخشد اما بر تعمیم‌دهی تأثیر منفی خواهد داشت.^[۲۱] کوچک نگه داشتن شبکه از راه‌های مختلفی میسر است، از جمله: هرس کردن^۶، الگوریتم کاهش^۷ و الگوریتم افزایشی^۸.^[۲۲] این روش‌ها معمولاً بسیار وقت‌گیر است و نیز برای بهبود تعمیم‌دهی تضمین قطعی نمی‌دهند.^[۲۳] بنابراین علاوه بر بهینه‌سازی در الگوریتم آموزش شبکه، بهینه‌سازی معماری نیز می‌تواند به تعمیم‌دهی و عملکرد بهتر شبکه در سیستم‌های پیچیده کمک شایانی کند. عملکرد بهتر شبکه را می‌توان به میزان قابل ملاحظه‌ای به روش‌ها و الگوریتم‌های جست‌وجوی استفاده شده در حین آموزش نسبت داد. اغلب روش‌های آموزش برای یافتن پارامترهای وزنی شبکه‌های عصبی مبتنی بر الگوریتم‌های گرادیانی و نوعاً گونه‌ی از پس انتشار خطا^۹ هستند. اما محدودیت‌های استفاده از این الگوریتم‌ها در مسائل بهینه‌سازی موجب عملکرد متناقض^{۱۰} و غیرقابل پیش‌بینی^{۱۱} می‌شود.^[۲۳] همچنین آموزش شبکه‌های عصبی با الگوریتم‌های جست‌وجوی گرادیانی باعث انطباق بیش از حد^{۱۲} شبکه به داده‌های آموزشی می‌شود. انطباق بیش از حد شبکه، به نوبه خود باعث کاهش قابلیت تعمیم‌دهی شبکه خواهد شد. انطباق بیش از حد نیز هنگامی رخ می‌دهد که با وجود این که شبکه در نگاشت داده‌های آموزشی به خروجی مطلوب بسیار دقیق است، اما به خاطر فرآیند نگرختن الگوی حاکم در پاسخ به ورودی‌های جدید، ناموفق عمل می‌کند. علت این امر را می‌توان در وجود پیچیدگی بیش از حد نیاز در وزن‌ها، گره‌ها یا پس‌خورهای اضافی شبکه جست‌وجو کرد.^[۲۴]

در مقایسه‌ی بین روش‌های مختلف آموزش شبکه‌های عصبی -- از قبیل الگوریتم ژنتیک، تابکاری (آنیلینگ) القایی و پس‌انتشار خطا -- برتری الگوریتم ژنتیک در این زمینه به اثبات رسیده است.^[۲۵] در فرایند آموزش شبکه و اصولاً شبکه‌های عصبی بازگشتی با پس‌خور داخلی به وسیله‌ی الگوریتم ژنتیک می‌توان وزن‌های ارتباط‌های پس‌خور را نیز در کروموزوم‌ها اضافه کرد و در فرایند آموزش دخالت داد.^[۲۶] روش پیشنهادی در این تحقیق، استفاده از یک الگوریتم ژنتیک اصلاح‌شده (MGA)^{۱۳} یا همان شیوه‌ی جست‌وجوی سراسری است. الگوریتم ژنتیک اصلاح‌شده از طریق صفر کردن وزن‌ها (یا ارتباط‌های) غیر ضروری و حذف آن‌ها موجب بهبود تعمیم‌پذیری می‌شود. این الگوریتم با بهینه‌سازی همزمان ارتباط‌های وزنی، معماری و آموزش همزمان، قادر به جست‌وجو و یافتن یک شبکه‌ی بهینه و صرفه‌جو^{۱۴} خواهد بود که تعمیم‌دهی بهتری دارد. الگوریتم مذکور برای آموزش شبکه‌های عصبی بازگشتی جزئی و یک شبکه‌ی ترکیبی پیشنهادی با داده‌های تجربی و داده‌های شبیه‌سازی شده مورد استفاده قرار گرفته است. داده‌ها تجربی، از آزمون پرواز یک هواپیمای جنگنده‌ی عملیاتی موجود در کشور به دست آمده است. مقایسه‌ی نتایج حاصل از آموزش شبکه با الگوریتم ژنتیک معمولی (NGA)^{۱۵} و الگوریتم ژنتیک اصلاح‌شده نشان می‌دهد که بهبود قابل توجهی در تعمیم‌دهی شبکه‌های عصبی برای شناسایی و مدل‌سازی دینامیک غیرخطی هواپیما حاصل شده است.

پرسپترون^{۱۸} (MLP) است. پرسپترون چندلایه یکی از ساده‌ترین و در عین حال کارآمدترین چیدمان‌های پیشنهادی ساختار شبکه‌های عصبی‌اند که در علوم مختلف مورد استفاده قرار می‌گیرند. در این شبکه‌ها نورون‌ها به صورت رو به جلو به هم متصل‌اند. شبکه‌های پرسپترون از قابلیت درون‌یابی و برون‌یابی در خروجی مدل بادقت بالا، برخوردار هستند، اما چون از نوع استاتیک‌اند قادر به تشخیص اثر زمان در فرایند نبوده و در نتیجه برای مدل‌سازی و شناسایی سیستم‌های دینامیکی مناسب نیستند. با افزودن سابقه‌ی ورودی‌ها و خروجی‌ها به عنوان ورودی‌های اضافی این مشکل حل می‌شود. در این صورت شبکه‌ی پرسپترون تبدیل به شبکه‌ی بازگشتی خواهد شد. شبکه‌های بازگشتی، شبکه‌هایی چندلایه‌اند که دارای اتصالات بازگشتی با تأخیر زمانی هستند. در شبکه‌های بازگشتی اطلاعات هم در مسیر رو به جلو، و هم در مسیر رو به عقب بین گره‌ها حرکت می‌کنند و موجب افزایش ظرفیت دینامیکی شبکه می‌شوند. اتصالات بازگشتی یا تأخیرهای زمانی به دو صورت تأخیر زمانی درون‌بازگشتی و تأخیر زمانی برون‌بازگشتی در شبکه‌های عصبی وجود دارند.

برای مدل‌سازی یک سیستم دینامیکی نظیر هواپیما، تخمین اثر ورودی‌ها و حالت درونی سیستم روی خروجی‌ها ضروری است. با توجه به توصیف سیستم به کمک متغیرهای حالت، یک سیستم دینامیکی را می‌توان در فضای گسسته با رابطه‌ی ۱ نشان داد:

$$\begin{aligned} X(k+1) &= F(X(k), U(k)) \\ Y(k) &= G(X(k), U(k)) \end{aligned} \quad (1)$$

که در آن X بردار متغیر حالت، U بردار ورودی و Y بردار خروجی است. معادله‌ی اول ارتباط حالت سیستم در یک زمان مشخص را با ورودی و حالت سیستم در لحظه‌ی قبلی برقرار می‌کند. معادله‌ی دوم ارتباط و وابستگی خروجی سیستم را در لحظه‌ی k با ورودی و حالت سیستم در همان لحظه نشان می‌دهد.

برای مدل‌سازی سیستم‌های دینامیکی با معادلات فوق، شبکه‌های ترکیبی که ویژگی‌های بازگشتی و غیر بازگشتی دارند، پیشنهاد شده است. برای معادله‌ی اول یا همان حالت‌های سیستم می‌توان از شبکه‌های بازگشتی، و برای معادله‌ی دوم یا خروجی سیستم از شبکه‌ی غیر بازگشتی مانند شبکه‌های رو به جلوی چندلایه سود برد.

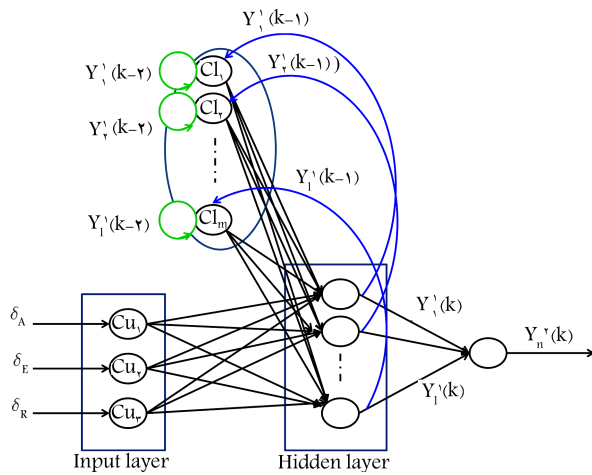
«شبکه‌های درون بازگشتی جزئی» نوعی شبکه ترکیبی محسوب می‌شوند که مناسب شناسایی سیستم‌اند. دو ساختار معروف این گروه از شبکه‌های عصبی عبارت‌اند از شبکه‌های المان^[۲۷] ۱۹ و جردن^[۲۸] ۲۰ که در این پروژه مورد استفاده قرار گرفته‌اند. علاوه بر این دو شبکه، شبکه‌ی ترکیبی با نام شبکه‌ی بازگشتی ترکیبی پیشنهاد شده مورد استفاده قرار گرفته که عملکرد خوبی در شناسایی و مدل‌سازی دینامیک غیرخطی دارد. نتایج حاصل از این شبکه‌ی ترکیبی پیشنهادی در مقاله‌ی دیگری در حال انتشار است.

۱.۲. شبکه المان و المان اصلاح شده

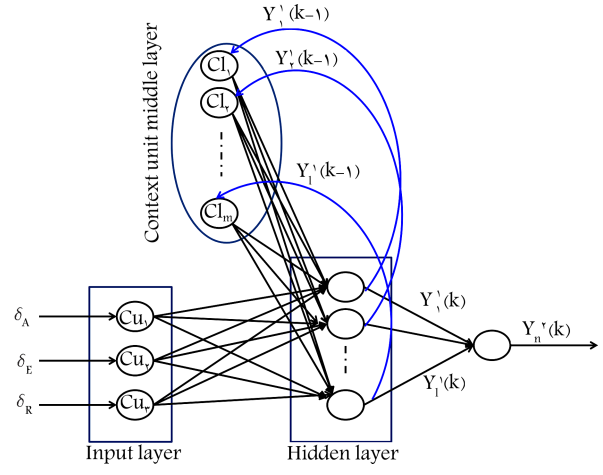
شبکه‌ی المان (شکل ۱) یک شبکه‌ی چندلایه است که در سطح ورودی شبکه یک لایه‌ی مخفی به نام واحدهای زمینه وجود دارد. این لایه از آن جهت مخفی است که واحدهای آن تنها با نورون‌های درونی شبکه تعامل دارند و با دنیای خارج از شبکه مرتبط نیستند. لایه‌ی مخفی در این شبکه آن را قادر می‌سازد برای هر ورودی اصلی یک خروجی مطلوب تولید کند. نورون‌های لایه‌ی زمینه نیز دارای تاریخچه‌ی از حالت داخلی سیستم‌اند؛ لذا نورون‌های لایه‌ی پنهان علاوه بر ورودی اصلی شبکه،

۲. شبکه‌ی عصبی در شناسایی سیستم‌های دینامیکی

دو معماری کلی برای شبکه‌های عصبی که در شناسایی سیستم مورد استفاده قرار می‌گیرند وجود دارد: شبکه‌های پیش‌خور^{۱۶} و شبکه‌های بازگشتی^{۱۷}. شبکه‌های پیش‌خور ساختار ساده‌ی دارند و مهم‌ترین آنها شبکه‌های چندلایه‌ی



شکل ۲. معماری شبکه‌ی المان اصلاح شده.



شکل ۱. معماری شبکه‌ی المان.

خروجی لایه‌ی میانی را نگه نمی‌دارند و فقط ارزش آخرین حالت ذخیره می‌شود. اما در شبکه‌ی المان اصلاح شده نورون‌های زمینه حالت‌های گذشته‌ی خروجی لایه‌ی میانی را حفظ می‌کنند؛ لذا حافظه‌ی دینامیکی شبکه افزایش می‌یابد. این نوع شبکه‌ها در شناسایی و شبیه‌سازی سیستم‌های دینامیکی هواپیما مناسب‌اند.^[۱۰] روابط استفاده شده برای این شبکه عبارت‌اند از:

$$Y_l^i(k) = \tan \operatorname{sig} \left(\frac{IW \times U + b^i + CW^i \times Y_l^i(k-1) + CW^i \times Y_l^i(k-2) + CW^i \times Y_l^i(k-3)}{CW^i \times Y_l^i(k-2) + CW^i \times Y_l^i(k-3)} \right) \quad (8)$$

$$Y_n^i(k) = \operatorname{purlin}(LW \times Y_l^i(k) + b^i) \quad (9)$$

۲.۲. شبکه‌ی جردن

در این شبکه برخلاف شبکه‌ی المان، واحد زمینه از لایه‌ی میانی بازخورد نمی‌شود. بلکه خروجی شبکه به‌عنوان واحد زمینه به ورودی اصلی شبکه بازخورد می‌شود. بردار ورودی در این شبکه همانند بردار ورودی شبکه‌ی المان است و تنها تفاوت آن در ارزش نورون‌های زمینه است. روابط به‌کار رفته در این شبکه نیز عبارت خواهند بود از:

$$Y_l^i(k) = \tan \operatorname{sig} (IW \times U + b^i + JW^i \times Y_n^i(k-1)) \quad (10)$$

$$Y_n^i(k) = \operatorname{purlin}(LW \times Y_l^i(k) + b^i) \quad (11)$$

۳.۲. شبکه‌ی بازگشتی ترکیبی پیشنهادی

در ساختار شبکه‌ی درون بازگشتی المان یا جردن، واحد زمینه تنها در یکی از لایه‌های شبکه وجود دارد. برای افزایش ظرفیت دینامیکی و داشتن تاریخچه‌ی زمانی حالت‌های سیستم، می‌توان واحد زمینه را در تمامی لایه‌های شبکه ایجاد کرد. بنابراین در این شبکه دو نوع واحد زمینه با نام‌های «واحد زمینه‌ی لایه‌ی میانی» و «واحد زمینه‌ی لایه‌ی خارجی» وجود خواهد داشت. لذا شبکه‌ی ترکیبی پیشنهادی شامل دو بخش است: ۱. شبکه‌ی چندلایه‌ی رو به جلو؛ ۲. واحدهای زمینه و خودبازگشتی مربوط به هر نورون. واحد زمینه نیز شامل دو بخش واحد زمینه‌ی لایه‌ی خروجی و واحد زمینه‌ی لایه‌ی میانی است. ورودی‌های شبکه شامل سه سری ورودی خواهد بود:

مجبور به نگاهی تاریخی‌تری از حالت‌ها به خروجی هستند، در نتیجه شبکه اثر زمان را در فرایند درک می‌کند.^[۱۰] ساختار این شبکه به‌عنوان یک شبکه‌ی چندلایه عمل می‌کند که لایه‌ی ورودی آن شامل ورودی‌های خارجی (ورودی‌های سیستم) و خروجی نورون‌های لایه‌ی زمینه است. با این ویژگی که ورودی خارجی شبکه و نورون‌های زمینه یک بردار جدید ورودی ایجاد می‌کنند که به‌صورت زیر است:

$$U = [U_1 \quad U_2 \quad \dots \quad U_I \quad Cl_1 \quad Cl_2 \quad \dots \quad Cl_L]^T \quad (2)$$

نورون‌های لایه‌ی زمینه و بازگشتی، نمایانگر حالت‌های داخلی‌اند، زیرا خروجی لایه‌ی میانی شبکه را به‌عنوان ورودی به شبکه بازخورد می‌دهد؛ به عبارت دیگر این نورون‌ها از حالت‌های گذشته به‌عنوان ورودی استفاده می‌کنند. این بدان معناست که واحد زمینه، تاریخچه‌ی از حالت‌های گذشته‌ی سیستم را که توسط ماتریس وزنی CW تنظیم می‌شوند در خود ذخیره می‌کند. روابط مورد استفاده در این شبکه نیز عبارت‌اند از:

$$Cn(k+1) = F(Cn(k), U(k)) = F_i \left(\sum_{j=1}^n W_{ij} X_j - \theta_i \right) \quad (3)$$

$$Y(k) = G(Cn(k), U(k)) \quad (4)$$

$$Y_i^i(k) = Cn_i(k-1) \quad \forall i = 1, 2, \dots, k \quad (5)$$

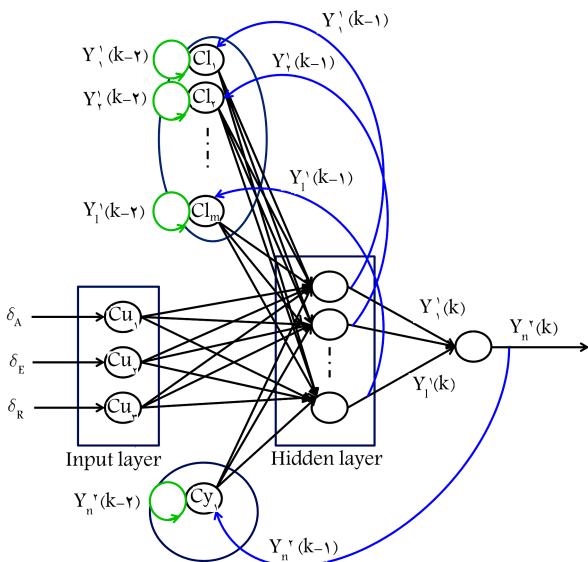
شبکه‌ی المان براساس روابط ۵ تا ۷ در این پروژه مورد استفاده قرار گرفته است. در لایه‌ی میانی از تابع تبدیل تانژانت سیگموئید و در لایه‌ی خروجی از تابع خطی استفاده شده است. نتایج استفاده از این نوع روابط در بهبود عملکرد شبکه، در بخش‌های بعد نشان داده خواهد شد.

$$Y_l^i(k) = \tan \operatorname{sig} (IW \times U + b^i + CW^i \times Y_l^i(k-1)) \quad (6)$$

$$Y_n^i(k) = \operatorname{purlin}(LW \times Y_l^i(k) + b^i) \quad (7)$$

$$U = [\delta_E \quad \delta_A \quad \delta_R]^T \quad (8)$$

شبکه‌ی المان اصلاح شده نیز همانند شبکه‌ی المان است (شکل ۲)، با این تفاوت که برای بالابردن ظرفیت دینامیکی شبکه‌ی عناصر خودبازخورد در لایه‌ی زمینه به هریک از نورون‌ها افزوده شده است. یعنی خروجی قبلی نورون‌های زمینه‌ی به خودش بازگشت داده می‌شود. در شبکه‌ی المان نورون‌های زمینه حالت‌های گذشته‌ی



شکل ۳. معماری شبکه‌ی ترکیبی پیشنهادی.

انتخاب توابع F و G بستگی به انتخاب تابع تبدیل لایه‌های مختلف و ساختار بلوک‌های مختلف شبکه دارد. با انتخاب تابع تبدیل تانژانت سیگموئید در لایه‌ی میانی و تابع تبدیل خطی در لایه‌ی خروجی بهترین جواب برای مدل‌سازی دینامیک هواپیما حاصل شده است. با این انتخاب خروجی شبکه‌ی عصبی و خروجی لایه‌ی میانی عبارت خواهد بود از:

$$Y_l^*(k) = \tan sig \left(\begin{matrix} IW \times U + b^1 + CW^1 \times Y_l^*(k-1) \\ + CW^2 \times Y_l^*(k-2) + JW^1 \\ \times Y_l^*(k-1) + JW^2 \times Y_l^*(k-2) \end{matrix} \right) \quad (17)$$

$$Y_n^*(k) = \text{purlin}(LW \times Y_l^*(k) + b^t) \quad (18)$$

در این شبکه ارتباط‌های وزنی واحد زمینه در لایه‌ی میانی و لایه‌ی خروجی وجود دارد. یعنی ترم‌های اضافه شده به شبکه عبارت‌اند از:

$$CW^1 \times Y_l^*(k-1) \dots CW^L \times Y_l^*(k-i) + JW^1 \times Y_l^*(k-1) + \dots + JW^j \times Y_l^*(k-j) \quad (19)$$

در شبکه‌ی اصلاح شده‌ی ترکیبی پیشنهادی حتی می‌توان برای بالا بردن حافظه‌ی دینامیکی، عناصر خودبازخورد از هر مرتبه‌ی دلخواه n را به شبکه اضافه کرد. چنانچه عناصر خودبازخورد به شبکه اضافه شود، یعنی خروجی‌های قبلی نورون‌های واحد زمینه در لایه‌ی میانی و لایه‌ی خروجی به خودشان بازگشت داده شوند، بردار Y^1 به روابط شبکه اضافه خواهد شد:

$$\left[Y_l^*(k-1) \quad Y_l^*(k-2) \quad \dots \quad Y_l^*(k-i) \quad Y_n^*(k-1) \quad \dots \quad Y_n^*(k-j) \right] \quad (20)$$

برای اغلب پارامترهای دینامیکی هواپیما با دو بار بازگشت برای هر نورون واحدهای زمینه ($i = 1, j = 2$) به شبکه، نتایج بهتری در آموزش و تعمیم‌دهی مشاهده می‌شود. در این صورت تنها ارتباط‌های وزنی 21 به روابط شبکه‌ی عصبی اضافه می‌شود:

$$CW^2 \times Y_l^*(k-2) + JW^2 \times Y_l^*(k-2) \quad (21)$$

۱. ورودی‌های خارجی شبکه که در نورون‌های $Cu = [Cu_1 \quad Cu_2 \quad \dots \quad Cu_i \quad \dots \quad Cu_n]$ ذخیره می‌شود (I تعداد ورودی‌های شبکه). در مدل‌سازی هواپیما این ورودی‌ها عبارت‌اند از: ورودی کنترلی ایرون، رادر و الویتور.

۲. بردار خروجی بخش لایه‌ی میانی که به‌عنوان واحد زمینه‌ی لایه‌ی میانی نام‌گذاری شده است. خروجی‌های این بخش در نورون‌های $Cl = [Cl_1 \quad Cl_2 \quad \dots \quad Cl_i \quad \dots \quad Cl_m]$ ذخیره می‌شوند (l تعداد نورون‌های لایه‌ی میانی). برای افزایش ظرفیت دینامیکی شبکه برای هر نورون نیز می‌توان حالت‌های قبلی خروجی نورون را با وزن‌های اضافی تنظیم، و به شبکه اضافه کرد. یعنی در این واحد زمینه برای هر نورون بردار Y^1 به شبکه اضافه می‌شود.

$$Y^1 = [Y^1(k-1), Y^1(k-2), \dots, Y^1(k-i)] \quad (12)$$

تعداد عناصر (i) این بردار نیز بستگی به انتخاب تعداد نورون‌های لایه‌ی میانی دارد. بهترین حالت برای شناسایی و مدل‌سازی که در این پروژه نتایج خوبی داشت، استفاده از هفت نورون در لایه‌ی میانی و برای هر نورون دو بار تأخیر زمانی (یعنی $Y^1 = [Y^1(k-1), Y^1(k-2)]$) است.

۳. بردار نورون‌های لایه‌ی خروجی شبکه که در نورون‌های $Cy = [Cy_1 \quad Cy_2 \quad \dots \quad Cy_o]$ ذخیره می‌شود (O تعداد خروجی‌های شبکه). این بخش نیز به‌عنوان واحد زمینه‌ی لایه‌ی خروجی شبکه معرفی می‌شود و شامل بردار Y^2 است.

$$Y^2 = [Y^2(k-1), Y^2(k-2), \dots, Y^2(k-j)] \quad (13)$$

این ورودی نیز بستگی به تعداد خروجی شبکه دارد. چنانچه شبکه دارای شش خروجی شامل حالت‌های هواپیما ($y = [p, q, r, u, v, w]$) باشد، بردار ورودی این بخش، حاوی شش نورون برای واحد زمینه لایه‌ی خروجی خواهد بود.

بنابراین بردار ورودی یک شبکه‌ی عصبی با سه ورودی، هفت نورون در لایه‌ی میانی و شش خروجی خواهد شد:

$$U = \left[U_1 \quad Cl \quad Cy \right] = [\delta_E \quad \delta_A \quad \delta_R \quad Cl_1 \quad Cl_2 \quad \dots \quad Cl_m \quad Cy_1 \quad Cy_2 \quad \dots \quad Cy_o]^T \quad (14)$$

در این حالت شبکه دارای ۱۶ ورودی است که شامل سه ورودی کنترلی هواپیما، هفت نورون خروجی لایه‌ی میانی و شش نورون لایه‌ی خروجی است. در حالت کلی، تعداد کل ورودی‌های یک شبکه با I ورودی، L نورون در لایه‌ی زمینه و O خروجی چنین است:

$$I + L \times L + O \times L \quad (15)$$

در شکل ۳ ساختار یک شبکه‌ی ترکیبی بازگشتی توسعه‌یافته نشان داده شده است. این ساختار ایده‌ی جدیدی است که عملکرد بهتری در آموزش و تعمیم‌دهی دینامیک غیرخطی هواپیما دارد. همانند شبکه‌های قبلی، این شبکه‌ی ترکیبی را می‌توان به معادلات ۱۶ تبدیل کرد.

$$Cn(k+1) = F(Cn(k), U(k))$$

$$Y(k) = G(Cn(k), U(k))$$

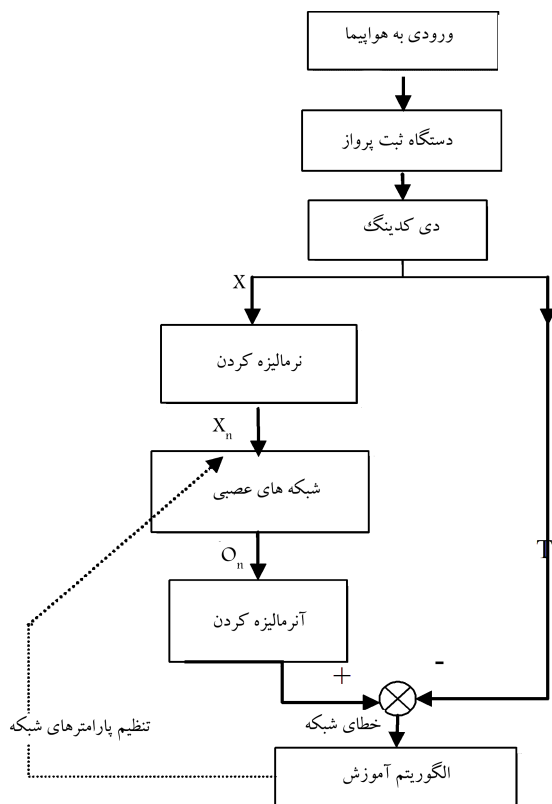
$$Y_i^*(k) = Cn_i(k-1) \quad \forall i = 1, 2, \dots, k \quad (16)$$

می‌گیرد: یافتن وزن‌های شبکه‌ی عصبی، پیدا کردن معماری بهینه، بهینه‌سازی همزمان معماری و وزن‌های شبکه، و تحلیل یک شبکه‌ی عصبی. در روند بهینه‌سازی با الگوریتم ژنتیک معمولی برای یافتن وزن‌های بهینه‌ی شبکه‌ی عصبی، از ارتباط‌های وزنی رو به جلو، بازگشتی، ارتباط‌های خودبازگشتی و بایاس استفاده می‌شود. هر نسل در الگوریتم ژنتیک جمعیتی از اعضاست و هر عضو با یک کروموزوم مشخص می‌شود که ویژگی‌های آن را به صورت کد شده دارد. در شبکه‌های عصبی، کروموزوم‌ها آرایه‌های سطری شامل ژن‌ها (بیت‌ها) هستند که تمام وزن‌ها و بایاس‌های شبکه را در بر می‌گیرد؛ کافی است تمام وزن‌ها و بایاس‌های شبکه در یک رشته به دنبال هم ذخیره شوند. همچنین می‌توان وزن‌های مربوط به ارتباط‌های داخلی و پس‌خور را به کروموزوم‌ها اضافه کرد و در فرایند آموزش دخالت داد. کروموزوم‌ها معمولاً به صورت باینری کد می‌شوند ولی در مسئله‌ی شبکه‌ی عصبی این کار باعث زیاد شدن طول کروموزوم و در نتیجه درست کار نکردن عملگرهای الگوریتم ژنتیک می‌شود، لذا در این تحقیق به منظور اجتناب از افزایش طول کروموزوم، پارامترهای وزنی و بایاس‌ها با اعداد دهگانی بیان شده‌اند.

روند آموزش شبکه‌ی عصبی با داده‌های حاصل از آزمایش پرواز در شکل ۴ به نمایش درآمده است. ابتدا سیگنال ورودی آموزشی به دست آمده از دستگاه ثبت اطلاعات هواپیما که به صورت گسسته گرفته می‌شود بین -۱ و +۱ طبق رابطه‌ی ۲۲ بهنجار ۲۱ می‌شود.

$$X_n = 2 \frac{(X - \min(X))}{\max(X) - \min(X)} - 1 \quad (22)$$

ورودی بهنجار شده به شبکه‌ی عصبی داده می‌شود و خروجی آن طبق روابط مطرح شده برای شبکه‌ها، محاسبه می‌شود. سپس خروجی شبکه طبق رابطه‌ی



شکل ۴. روند به‌کارگیری داده‌ها در الگوریتم آموزش شبکه.

این وزن‌ها باعث می‌شود که در این شبکه تعداد حالت‌های گذشته مورد استفاده، و همین‌طور تعداد نورون‌های موجود در لایه‌ی پنهان قابل تنظیم باشند. این انعطاف‌پذیری برای شبکه ضروری است زیرا اغلب، مدل سیستم‌ها برای شناسایی ناشناخته بوده و شبکه باید برای دریافت خطای کمینه در حین آموزش سازگار شود.

۳. آموزش و تعمیم‌دهی در شبکه‌ی عصبی

آموزش در شبکه‌های عصبی به این معناست که شبکه‌ی اتصالات وزن‌دار خود را طوری تنظیم کند که خروجی شبکه با خروجی مورد نظر یکسان شود. یک شبکه‌ی عصبی زمانی ابزار کمی ارزشمندی است که عملکردش پس از آموزش نیز به همان دقتی باشد که در حین آموزش از خود بروز داده است. بنابراین، تعمیم‌دهی برای شبکه‌های عصبی از اهمیت خاصی برخوردار است. تعمیم‌دهی عبارت است از ارائه‌ی یک خروجی مناسب به‌ازای ورودی‌هایی که شبکه با آن آموزش ندیده است. اغلب الگوریتم‌های یادگیری در شبکه‌های عصبی مبتنی بر روش‌های گرادینتی و نوعاً گونه‌یی از نوع پس‌انتشار خطا هستند که برای یافتن پارامترهای وزنی شبکه مورد استفاده قرار می‌گیرند. اگرچه الگوریتم‌های جست‌وجوی گرادینتی شبکه‌های عصبی را به‌خوبی آموزش می‌دهند و برای داده‌های آموزشی مناسب‌اند، با توجه به پیچیدگی شبکه‌های عصبی و وجود کمینه‌های محلی فراوان، احتمال به دام افتادن در کمینه‌های محلی زیاد است. لذا ممکن است شبکه‌ی عصبی الگوهای اصلی را در طول آموزش به‌خوبی فرا نگیرد و قابلیت آن در تشخیص الگوهای داده از داده‌های جدید قابل اعتماد نباشد؛ به عبارت دیگر ممکن است از تعمیم‌پذیری کمی برخوردار باشد و بر این اساس این‌گونه روش‌ها برای آموزش شبکه‌های عصبی مناسب نیستند. طبق گزارش محققین، در مقایسه با دیگر روش‌های یادگیری، الگوریتم‌های ژنتیک می‌تواند جایگزین روش‌های پس‌انتشار در آموزش شبکه‌های عصبی شود و عملکرد شبکه را بهینه کند.^[۱۹] الگوریتم‌های ژنتیک از جمله روش‌هایی هستند که برای پیدا کردن نقطه‌ی کمینه‌ی سراسری مناسب‌اند. در الگوریتم ژنتیک با رمزگذاری پارامترهای وزنی و بایاس‌های شبکه به صورت باینری یا مقادیر واقعی در قالب کروموزوم، می‌توان جواب بهینه‌ی سراسری را یافت و شبکه را آموزش داد.^[۲۶]

۱.۳. الگوریتم ژنتیک در آموزش شبکه‌ی عصبی

الگوریتم ژنتیک یک شیوه‌ی جست‌وجوی سراسری است که مضامین و عملکردهای خود را از نظریه‌ی تکامل طبیعی الهام گرفته است. یک الگوریتم ژنتیک با جمعیتی از راه حل‌ها، کروموزوم‌ها و فرایند تصادفی آغاز می‌شود و با به‌کارگیری عملگرهای ملهم از فرایندهای موجود در طبیعت به راه حل‌های بهتری دست می‌یابد. عملگرهای مورد استفاده در این الگوریتم عبارت‌اند از: ترکیب، جهش، نخبه‌گزینی و انتخاب تصادفی از میان اعضا. تشکیل جمعیت اولیه به صورت تصادفی است و اپراتورهایی که نسل‌های جدید را می‌سازند دارای عناصر تصادفی‌اند. اعضای جمعیت در هر نسل با هم به رقابت می‌پردازند و آنهایی که برحسب خصوصیات برترشان، قدرت و شایستگی بیشتری برای بقا داشته باشند زنده می‌مانند و به تولید مثل می‌پردازند. این ویژگی باعث می‌شود الگوریتم ژنتیک قادر به بهینه‌سازی در نقاط سراسری باشد.

کاربرد الگوریتم‌های ژنتیک در حوزه شبکه‌های عصبی به چند روش صورت

۲۳ ناپهنجار^{۲۲} شده، با خروجی مطلوب مقایسه می‌شود تا بتوان از رابطه‌ی ۲۴ میانگین مربعات خطا را محاسبه کرد. الگوریتم ژنتیک با داشتن میانگین مربعات خطا پارامترهای شبکه را طی روندی تکراری بهینه می‌کند تا کم‌ترین خطا حاصل و فرایند آموزش تکمیل شود.

$$O = \frac{1}{n} (O_n + 1) (\max(T) - \min(T)) + \min(T) \quad (23)$$

$$MSE = \frac{1}{n_d} \sum_{i=1}^{n_d} (O_i - T_i)^2 \quad (24)$$

تابع شایستگی یا ارزیابی نقش محیطی را ایفا می‌کند که راه حل‌های خوب و بد را تشخیص می‌دهد. در هر نسل، راه حل‌های نسبتاً خوب برای یک مشکل تکثیر شده و برای جایگزینی راه حل‌های نسبتاً بد که ماندگار نیستند تولید مثل می‌کنند. از آنجا که هدف الگوریتم ژنتیک کمینه کردن اختلاف خروجی شبکه‌ی عصبی و خروجی پاسخ دینامیکی سیستم مورد نظر (پاسخ دینامیکی هواپیما) است، معکوس میانگین مربعات خطا (MSE)^{۲۳} معیار مناسبی به‌عنوان تابع شایستگی است:

$$J = MSE = \sum_{i=1}^{n_d} (O_i - T_i)^2 \quad (25)$$

$$Fit = \frac{1}{MSE} \quad (26)$$

که در آن T خروجی مطلوب (که از آزمایش‌های پروازی به دست می‌آید)، O خروجی شبکه‌ی عصبی و n_d تعداد نقاط داده، J تابع هزینه و Fit معیار شایستگی است. معیار شایستگی هر یک از ژن‌ها در جمعیت ژنتیکی معمولاً میانگین مربعات خطای نوروهای خروجی طی یک دوره‌ی آموزشی را شامل می‌شود، و پس از آنکه شبکه‌ی عصبی برای یک دوره‌ی زمانی آموزش دید اندازه‌گیری می‌شود. در هر عضو، هرچه خروجی شبکه و خروجی مطلوب در نقاط داده بیشتری به هم نزدیک‌تر شوند، میانگین مربعات خطا کوچک‌تر و شایستگی آن عضو بالاتر می‌رود.

۲.۳. آموزش شبکه و بهینه‌سازی معماری آن به‌طور همزمان

الگوریتم ژنتیک معمولی در آموزش شبکه‌های عصبی معمولاً به دو صورت کاربرد دارد: پیدا کردن وزن‌های شبکه با معماری ثابت، و بهینه‌سازی معماری شبکه‌ی عصبی. برای تعمیم‌دهی بهتر می‌توان از بهینه‌سازی همزمان معماری و آموزش وزن‌های شبکه‌ی عصبی از الگوریتم ژنتیک اصلاح شده استفاده کرد. از الگوریتم ژنتیک اصلاح شده با نام «الگوریتم بهینه‌سازی همزمان شبکه‌ی عصبی (NNSOA)^{۲۴}» برای پیدا کردن یک معماری صرفه‌جو در شبکه‌ی عصبی استفاده شده است. در تحقیق حاضر، ضمن توسعه‌ی الگوریتم یادشده برای آموزش شبکه‌های عصبی بازگشتی و بازگشتی ترکیبی، از آن در شناسایی و مدل‌سازی دینامیک غیرخطی هواپیما بهره برده‌ایم.

الگوریتم ژنتیک اصلاح شده همانند الگوریتم ژنتیک معمولی است با این تفاوت که در این الگوریتم از یک اپراتور جدید به نام «جهش - ۲» استفاده می‌شود که به‌کمک آن می‌توان به‌طور تصادفی برخی از وزن‌ها را صفر و از شبکه خارج کرد. در این اپراتور برای هر بیت از هر کروموزوم یک عدد تصادفی بین صفر و ۱ براساس یک توزیع یک‌نواخت تولید می‌شود. اگر این عدد از P_{m2} (احتمال عملگر «جهش - ۲») کم‌تر باشد، مقدار پارامتر آن بیت که ممکن است وزن روی ارتباط‌های پیش‌خور یا پس‌خور یا بایاس باشد، صفر می‌شود. پس از تولید نسل جدید به‌منظور تعیین بیت‌های غیر صفر هر عضو (کروموزوم) جست‌وجویی صورت می‌گیرد.

تابع هزینه‌ی رایج در مطالعات مربوط به شبکه‌های عصبی معمولاً تابع کوادراتیک میانگین مربعات خطا (MSE)، ریشه‌ی میانگین مربعات خطا (RMSE)^{۲۵} یا مجموع مربعات خطا (SSE)^{۲۶} است. دلیل انتخاب تابع کوادراتیک از سوی محققین، کارکرد خوب، قابل تشخیص بودن و امکان استفاده از مشتقات مناسب برای بهینه‌سازی است. با توجه به این که در الگوریتم ژنتیک محدودیتی برای مشتق‌گیری وجود ندارد، می‌توان تابع هدف را طوری انتخاب کرد که از افزایش وزن‌های غیرضروری جلوگیری شود. بنابراین در این تحقیق تابع هدف طوری اصلاح شده که با کاهش خطا و حذف وزن‌های اضافی تابع هدف نیز کاسته می‌شود. تابع هدف اصلاح شده تابعی یک‌نواخت نیست و با اضافه یا حذف شدن وزن‌ها ممکن است بی‌قاعده و نامرتب شود. معادله‌ی ۲۷ نشان‌گر تابع هدف اصلاح‌شده‌ی به کار رفته در این تحقیق است. در این تابع معیار شایستگی هزینه‌ی بر تعداد وزن‌های غیر صفر اعمال می‌شود تا الگوریتم را در راستای رسیدن به ساختاری با کم‌ترین ارتباط‌های ممکن تشویق کند.

$$J = \sum_{i=1}^{n_d} (O_i - T_i)^2 + C \sqrt{\frac{\sum_{i=1}^{n_d} (O_i - T_i)^2}{n_d}} \quad (27)$$

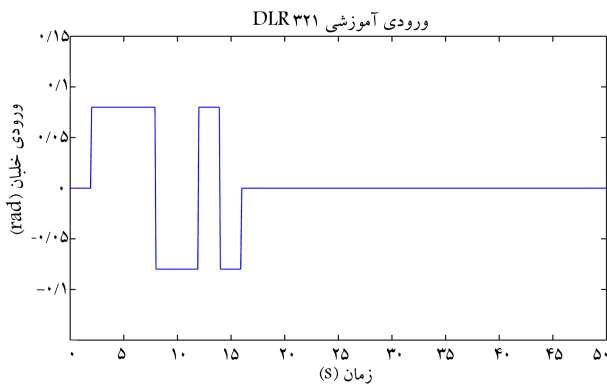
$$Fit = \frac{1}{J} \quad (28)$$

در تابع ۲۷، C تعداد وزن‌های غیر صفر است که به‌صورت یک هزینه‌ی دینامیکی یعنی متغیر با روند آموزش، اعمال شده است تا از تسلط آن بر معیار شایستگی جلوگیری شود. در صورتی که C را به‌صورت یک ضریب ثابت در معیار شایستگی قرار دهیم مشاهده می‌شود که در روند آموزش، مقدار MSE به مرور کاهش می‌یابد. در نتیجه، اهمیت خطای خروجی شبکه در معیار شایستگی از بین می‌رود و الگوریتم به سمت کوچک کردن شبکه (کاهش C) پیش می‌رود و آموزش نمی‌بیند. بنابراین، ضریب تأثیر تعداد وزن‌های غیر صفر باید متناسب با MSE کاهش یابد. در الگوریتم بهینه‌سازی همزمان شبکه‌ی عصبی (NNSOA)، مجذور میانگین مربعات خطا (RMSE) به‌عنوان ضریب هزینه بر وزن‌های غیر صفر انتخاب شده است. این انتخاب به‌صورت آزمون و خطا بوده و اثباتی برای بهینگی آن وجود ندارد. بدین ترتیب هرچه مجذور میانگین مربعات خطا کاهش یابد، یعنی فرایند آموزش به مراحل آخر نزدیک شود، سهم هزینه بر وزن‌های غیر صفر ($C \times RMSE$) نیز کم‌رنگ‌تر می‌شود.

در انتخاب عدد احتمال برای عملگر «جهش - ۲» (P_{m2}) باید دقت کافی اعمال شود. اگر این احتمال به‌اندازه کافی کوچک نباشد، وزن‌های زیادی در شروع آموزش حذف شده و شبکه هیچ‌گاه به خطای مطلوب نمی‌رسد. از طرف دیگر، اگر احتمال عملگر «جهش - ۲» بسیار کوچک باشد، الگوریتم NNSOA کارایی خود را در رسیدن به صرفه‌جویی شبکه از دست می‌دهد، زیرا عملگر «جهش - ۲» فرصت کافی برای حذف وزن‌های اضافی در اختیار نخواهد داشت. در این مطالعه احتمال مناسب برای عملگر «جهش - ۲» که به روش سعی و خطا به دست آمده مقدار 10^{-2} است. استفاده از این الگوریتم سبب کاهش ۴۰ درصدی تعداد ارتباط‌های وزنی ولی افزایش مدت زمان آموزش و تعداد نسل‌های لازم شد.

۴. آماده‌سازی داده‌ها

با توجه به این که اعمال ورودی‌هایی مانند نویز سفید و سیگنال چیرپ^{۲۷} در پرواز واقعی قدری مشکل است، برای بررسی و مقایسه‌ی قابلیت تعمیم‌دهی شبکه‌های



شکل ۵. ورودی ۳۲۱۱ برای آموزش شبکه.

را تحریک کند. از طرفی از دید خلبان، ورودی انتخابی باید آسان و قابل تکرار باشد و سبب درگیر شدن با مانورهای پیچیده یا تغییرات کلی شرایط اولیه پرواز نشود. ورودی‌های مناسب برای هواپیما به منظور داده برداری عبارت‌اند از: ورودی استاندارد چندپله‌یی DLR۳۲۱۱ [۳۱] و ورودی جاروب فرکانسی ۲۹، ۳۰ [۳۲]. در این تحقیق از ورودی چندپله‌یی و ورودی سینوسی به‌طور جداگانه برای هر یک از سطوح فرامین استفاده شده است. برای تولید داده به‌کمک شبیه‌سازی پرواز از ورودی رادر در بخش عرضی و الویتور در بخش طولی هواپیما به شکل ورودی چندپله‌یی ۳۲۱۱ (شکل ۵) استفاده شده است. در این حالت می‌توان ورودی را به شکل لاپلاس بیان کرد:

$$\delta_R(S) = \frac{0.1e^{-1S} - 0.2e^{-8S} + 0.2e^{-12S} - 0.2e^{-14S} + 0.2e^{-16S}}{S} \quad (30)$$

با قرار دادن $\delta_E(s)$ یا $\delta_R(s)$ در روابط مربوط به حالت‌های مختلف هواپیما و محاسبه‌ی لاپلاس معکوس، مقادیر $\beta(t)$ ، $\psi(t)$ و $\phi(t)$ در بخش عرضی و $U(t)$ ، $\alpha(t)$ و $\theta(t)$ در بخش طولی به دست می‌آید. با گسسته‌سازی $\delta_R(t)$ و $\delta_E(s)$ با گام زمانی مناسب دو آرایه‌ی سطری از داده‌ها به دست خواهد آمد که داده‌های آموزشی را تشکیل می‌دهند. گام زمانی انتخابی نیز نباید آن‌قدر بزرگ باشد که همواری 3° پاسخ از بین برود، یا چنان کوچک انتخاب شود که تعداد نقاط داده‌های آموزشی افزایش یابد و همگرایی فرایند آموزش کند شود.

۶. نتایج شناسایی

سه ساختار مختلف شبکه‌ی المان، جردن و شبکه‌ی بازگشتی ترکیبی پیشنهادی، هر یک با معماری عادی و بهینه (الگوریتم یادگیری ژنتیک معمولی و الگوریتم ژنتیک اصلاح شده) مورد آموزش قرار گرفته‌اند. در جدول ۱ مقادیر پارامترهای مورد استفاده در الگوریتم ژنتیک معمولی و ژنتیک اصلاح شده ارائه شده است. این پارامترها در محدوده‌ی مناسب برای بهینه‌سازی وزن‌های شبکه‌ی عصبی قرار دارند و با آزمون و خطا تعیین شده‌اند. با توجه به این که سرعت آموزش شبکه یکسان نیست، فرایند آموزش در دو حالت مورد استفاده قرار گرفته است: حالت اول با تعداد تولید نسل

جدول ۱. پارامترهای الگوریتم ژنتیک.

تعداد جمعیت	شکاف نسل‌ها	احتمال جهش ۱	نرخ جهش ۱	احتمال جهش ۲
۵۰	٪۸۰	۰/۳	۰/۰۱	۰/۰۰۰۱

اصلی و بهینه، علاوه بر داده‌ی واقعی از داده‌های شبیه‌سازی شده با ورودی‌های مختلف نیز استفاده می‌شود. داده‌های تولید شده، حاصل شبیه‌سازی شش درجه آزادی هواپیما جنگنده F-۴D در شرایط کروز است. داده‌های واقعی نیز از داده‌های ثبت شده در آزمایشات پروازی یک هواپیما جنگنده‌ی نسل چهارم با قابلیت مانور بالا به دست آمده است. در این تحقیق از جنگنده‌ی مذکور با نام Xcraft یاد می‌شود.

۱.۴. تولید داده به‌کمک معادلات خطی شده‌ی هواپیما

معادلات خطی‌شده‌ی هواپیما به‌صورت توابع تبدیل در فضای لاپلاس در دسترس است. [۳۰] با قرار دادن مشتقات ابعادی پایداری [۳۱] برای سرعت و ارتفاع خاص، توابع تبدیل را می‌توان برحسب ورودی در بخش عرضی و الویتور در بخش طولی به دست آورد. برای مثال تابع تبدیل زاویه‌ی حمله‌ی هواپیما F-۴D برحسب ورودی الویتور عبارت است از:

$$\frac{\alpha(S)}{\delta_E(S)} = \frac{-5S^2 - 34475S^2 - 424s - 141}{876S^2 + 99878S^2 + 707476S^2 + 271S - 1173} \quad (29)$$

با تبدیل این توابع از فضای لاپلاس به فضای حالت و حل آن‌ها (شبیه‌سازی) می‌توان داده‌های لازم برای آموزش شبکه‌ی عصبی را تولید کرد. این داده‌ها عبارت خواهد بود از: مقادیر تغییرات زوایای حمله، سرش، سرعت‌های خطی و زوایای اوایلر.

۲.۴. داده‌های ثبت شده در پرواز

برای به دست آوردن داده‌های تجربی از یک هواپیما جنگنده‌ی دوموتوره استفاده شده است. آزمایش هواپیما در شرایط هوای آرام بدون باد و توربالانس با سرعت و ارتفاع معلوم صورت گرفته است. خلبان ورودی مناسب را به هر یک از ورودی‌های کنترلی هواپیما -- سه زاویه‌ی سطوح اصلی کنترلی (الویتور، ایلان و رادر) -- وارد می‌کند. اعمال ورودی، در طول آزمایش پروازی برای داده برداری، در شرایط تریم در یک ارتفاع و سرعت خاص توسط خلبان صورت گرفته است. معمولاً با توجه به قابلیت مانور بالا و چابکی هواپیما، اندکی تغییرات در سرعت و ارتفاع به‌هنگام آزمایش دیده می‌شود. در این نوع هواپیما، فرامین کنترلی دارای ارتباط داخلی با یکدیگر هستند؛ لذا زمانی که تنها یکی از فرامین به هواپیما وارد می‌شود، در دیگر سطوح کنترلی نیز اندکی تغییرات وجود خواهد داشت.

داده‌های زیادی روی نوار مغناطیسی دستگاه ثبت اطلاعات پرواز (FDR) ۲۸ ثبت می‌شود. پارامترهای مورد استفاده برای شناسایی دینامیک هواپیما عبارت‌اند از: سرعت V ؛ سرعت واقعی V_T ؛ ارتفاع فشاری H_p ؛ ارتفاع فشاری H_R ؛ ارتفاع نسبی n_x ؛ شتاب طولی n_y ؛ شتاب عرضی n_z ؛ شتاب عمودی ψ ؛ یاو؛ زاویه‌ی رول؛ θ زاویه‌ی پیچ؛ α زاویه‌ی حمله؛ δ_E زاویه‌ی تغییرات سکان افقی؛ δ_R زاویه‌ی تغییرات رادر؛ δ_A زاویه‌ی تغییرات ایلرون؛ $\delta_{R_{th}}$ مقدار حرکت پدال؛ $\delta_{A_{th}}$ حرکت استیک در جهت رول؛ $\delta_{E_{th}}$ حرکت استیک در جهت پیچ؛ M عدد ماخ.

۵. سیگنال ورودی مناسب برای آموزش

نوع سیگنال تحریک مناسب برای جمع‌آوری داده‌های شناسایی در فرایند آموزش و تعمیم‌دهی شبکه بسیار اهمیت دارد. در اشیاء پرنده از جمله هواپیما، سیگنال ورودی به کنترل فرامین باید چنان صورت بگیرد که مدهای دینامیکی مختلف وسیله‌ی پرنده

متفاوت و حالت دوم با فرض تعداد تولید نسل یکسان، اما مقدار میانگین مربعات خطای یکسان در نظر گرفته شده است.

با توجه به این که بعضی از شبکه‌ها سریع‌تر آموزش می‌بینند، برای ارزیابی کارایی دو شبکه با دو روش آموزش، صرف‌نظر از تعداد نسل تولید شده، آموزش تا اندازه‌ی ادامه خواهد یافت که میزان میانگین مربعات خطای هر دو شبکه به یک میزان کاهش یابد. در این صورت تعمیم‌دهی شبکه با داده‌های جدید مقایسه می‌شود. برای بررسی و مقایسه‌ی قابلیت تعمیم‌دهی شبکه‌های اصلی و بهینه، هر دو شبکه با ورودی‌های جدیدی مانند موج مربعی 3^1 (ترکیبی از چند ورودی پله) و ورودی نویز سفید و سیگنال چیرپ 3^2 (موجی سینوسی با فرکانس متناوب) تحریک شده‌اند. البته چون پیاده‌سازی این ورودی‌های در پرواز واقعی قدری مشکل است، برای ارزیابی شبکه از داده‌های شبیه‌سازی شده استفاده شده است. میانگین مربعات خطای دو شبکه نیز به‌ازای ورودی‌های مختلف در جدول ۲ ارائه شده است. نتایج این جدول

جدول ۲. مقایسه‌ی میانگین مربعات خطای شبکه‌ی اصلی و بهینه.

ورودی	MSE شبکه‌ی	MSE شبکه‌ی
	اصلی	بهینه
ورودی ۳۲۱۱ (داده‌های آموزشی)	1×10^{-2}	1×10^{-2}
موج مربعی (داده‌های غیرآموزشی)	$4/163 \times 10^{-2}$	$3/318 \times 10^{-2}$
ورودی نویز (داده‌های غیرآموزشی)	$2/528 \times 10^{-2}$	$2/432 \times 10^{-2}$
ورودی چیرپ (داده‌های غیرآموزشی)	$2/764 \times 10^{-2}$	$3/475 \times 10^{-2}$

نشان می‌دهد که شبکه‌ی که با الگوریتم ژنتیک اصلاح شده آموزش دیده است نسبت به داده‌های جدید دارد.

در جدول ۳ میزان کاهش میانگین مربع خطای داده‌های آموزشی با تولید نسل متفاوت، برای سه شبکه‌ی جردن، شبکه‌ی المان اصلاح شده و شبکه‌ی ترکیبی پیشنهادی، با دو روش آموزش الگوریتم ژنتیک معمولی و الگوریتم ژنتیک اصلاح شده ارائه شده است. چنان که مشاهده می‌شود، روند آموزش با الگوریتم آموزش ژنتیک معمولی سریع‌تر از الگوریتم ژنتیک اصلاح شده است. همچنین این جدول نشان می‌دهد که شبکه‌ی ترکیبی پیشنهادی سریع‌تر آموزش می‌بیند.

در جدول ۴ میزان کاهش میانگین مربعات خطای داده‌های اعتباربخشی برای سه شبکه‌ی جردن، شبکه‌ی المان اصلاح شده و شبکه‌ی ترکیبی پیشنهادی، با دو روش آموزش الگوریتم ژنتیک معمولی و الگوریتم ژنتیک اصلاح شده برای ورودی‌های جدید ارائه شده است. چنان که مشاهده می‌شود شبکه‌ی آموزش دیده با الگوریتم ژنتیک اصلاح شده نسبت به سایر شبکه‌ها از خطای تعمیم‌دهی کم‌تری برخوردار است. چنان که از این جدول برمی‌آید، آموزش بیش از حد باعث می‌شود که خطای شبکه در تعمیم‌دهی برای ورودی‌های جدید بیشتر شود. بنابراین برای مقایسه‌ی دقیق‌تر الگوریتم ژنتیک معمولی با الگوریتم ژنتیک اصلاح شده، نتایج آموزش و تعمیم‌دهی شبکه‌ی بازگشتی با ورودی یکسان با داده‌های واقعی برای تولید نسل‌های مختلف در جدول ۵ و ۶ ارائه شده است. از نتایج این جداول مشخص است که کاهش خطا در شبکه‌ی ترکیبی پیشنهادی نیز با روش آموزش الگوریتم ژنتیک اصلاح شده بیشتر از روش اول است.

برای مقایسه‌ی دقیق‌تر شبکه‌های مختلف، میزان مربعات خطای هر یک از

جدول ۳. مقدار میانگین خطای شبکه در مرحله‌ی آموزش با تولید نسل‌های مختلف با ورودی یکسان.

شبکه‌ی عصبی	۵۰۰ Gen	۱۰۰۰ Gen	۲۰۰۰ Gen	۳۰۰۰ Gen	۴۰۰۰ Gen
شبکه‌ی جردن با آموزش الگوریتم NGA	۳,۷۰۴۲	۲,۹۴۰۳	۲,۲۹۰۳	۱,۲۳۶۱	۰,۹۷۴۰
شبکه‌ی جردن با آموزش MGA	۳,۹۰۲۲	۳,۱۴۲۰	۲,۳۷۳۰	۱,۱۰۲۱	۰,۸۳۲۱
شبکه‌ی المان با آموزش NGA	۲,۸۴۰۱	۲,۴۰۷۹	۱,۴۲۲۱	۰,۷۸۶۹	۰,۵۶۱۸
شبکه‌ی المان با آموزش MGA	۲,۹۱۷۷	۲,۹۲۱۸	۱,۳۳۰۷	۰,۶۰۳۶	۰,۴۸۵۷
شبکه‌ی ترکیبی پیشنهاد شده با آموزش NGA	۰,۸۷۵۲	۰,۳۰۰۱	۰,۲۱۳۳	۰,۱۶۶۹	۰,۱۴۵۸
شبکه‌ی ترکیبی پیشنهاد شده با آموزش MGA	۰,۹۳۷۵	۰,۳۱۰۷	۰,۲۱۴۸	۰,۱۲۸۹	۰,۱۱۳۷

جدول ۴. مقدار میانگین خطا شبکه در مرحله‌ی تعمیم دهی با epoch های مختلف با ورودی یکسان.

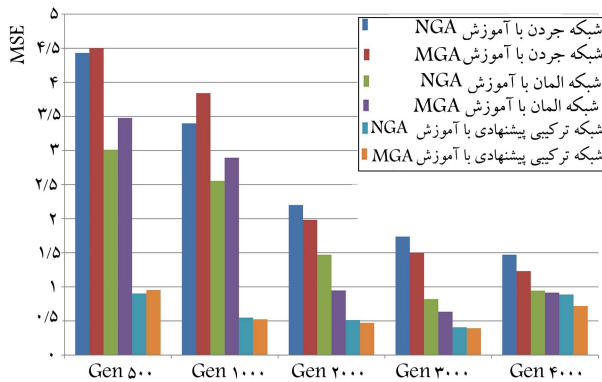
شبکه‌ی عصبی	۵۰۰ Gen	۱۰۰۰ Gen	۲۰۰۰ Gen	۳۰۰۰ Gen	۴۰۰۰ Gen
شبکه‌ی جردن با آموزش NGA	۴,۴۳۱۱	۳,۳۹۸۰	۲,۲۰۰۱	۱,۷۳۶۶	۱,۴۷۱۰
شبکه‌ی جردن با آموزش MGA	۴,۵۰۱۱	۳,۸۴۱۰	۱,۹۸۳۰	۱,۵۰۱۱	۱,۲۳۰۱
شبکه‌ی المان با آموزش NGA	۳,۰۱۱۶	۲,۵۵۴۵	۱,۴۷۱۱	۰,۸۲۱۰	۰,۹۴۴۸
شبکه‌ی المان با آموزش MGA	۳,۴۷۷۰	۲,۸۹۴۲	۰,۹۴۶۶	۰,۶۳۵۱	۰,۹۱۴۹
شبکه‌ی ترکیبی پیشنهادی با آموزش NGA	۰,۹۰۳۶	۰,۵۴۷۸	۰,۵۱۳۰	۰,۴۰۷۱	۰,۸۸۸۴
شبکه‌ی ترکیبی پیشنهادی با آموزش MGA	۰,۹۵۳۹	۰,۵۲۳۴	۰,۴۷۰۰	۰,۳۹۳۱	۰,۷۱۸۴

جدول ۵. مقدار میانگین خطای شبکه در مرحله‌ی آموزش با epoch های مختلف با ورودی یکسان.

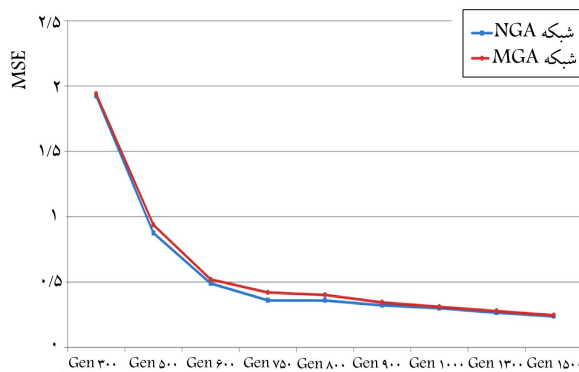
شبکه‌ی عصبی	Gen ۳۰۰	Gen ۵۰۰	Gen ۶۰۰	Gen ۷۵۰	Gen ۸۰۰	Gen ۹۰۰	Gen ۱۰۰۰	Gen ۱۳۰۰	Gen ۱۵۰۰
شبکه‌ی ترکیبی پیشنهادی NGA	۱,۹۲۲۸	۰,۸۷۵۲	۰,۴۸۹۶	۰,۳۶۰۴	۰,۳۵۹۳	۰,۳۲۱۵	۰,۳۰۰۱	۰,۲۶۵۵	۰,۲۳۷۸
شبکه‌ی ترکیبی پیشنهادی MGA	۱,۹۴۰۲	۰,۹۳۷۵	۰,۵۲۰۲	۰,۴۲۰۶	۰,۴۰۱۸	۰,۳۴۴۰	۰,۳۱۰۷	۰,۲۷۸۲	۰,۲۴۶

جدول ۶. مقدار میانگین خطا شبکه در مرحله‌ی تعمیم‌دهی با تولید نسل‌های مختلف با ورودی یکسان.

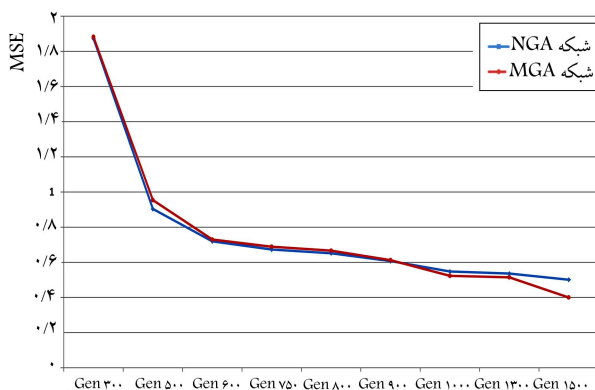
Gen	Gen	Gen	Gen	Gen	Gen	Gen	Gen	Gen	شبکه‌ی عصبی
۱۵۰۰	۱۳۰۰	۱۰۰۰	۹۰۰	۸۰۰	۷۵۰	۶۰۰	۵۰۰	۳۰۰	
۰٫۵۲۱۱	۰٫۵۳۵۹	۰٫۵۴۷۸	۰٫۶۰۷۰	۰٫۶۵۱۸	۰٫۶۷۲۸	۰٫۷۱۹۶	۰٫۹۰۳۶	۱٫۸۷۲۶	شبکه‌ی ترکیبی پیشنهادی NGA
۰٫۴۸۰۱	۰٫۵۱۴۷	۰٫۵۲۳۴	۰٫۶۱۲۱	۰٫۶۶۲۸	۰٫۶۸۸۷	۰٫۷۲۹۴	۰٫۹۵۳۹	۱٫۸۸۲۷	شبکه‌ی ترکیبی پیشنهادی MGA



شکل ۷. مقدار میانگین مربعات خطا سه نوع شبکه در مرحله تعمیم‌دهی با تولید نسل مختلف با ورودی یکسان.



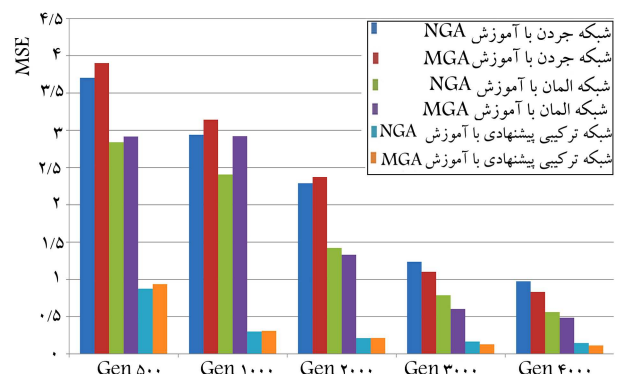
شکل ۸. مقدار میانگین مربعات خطا شبکه‌ی ترکیبی پیشنهادی در مرحله آموزش با تولید نسل مختلف با ورودی یکسان.



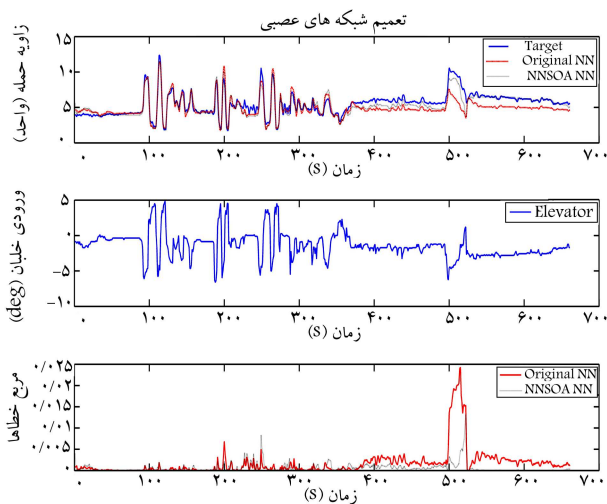
شکل ۹. مقایسه‌ی کاهش میزان خطا شبکه‌ی ترکیبی پیشنهادی با دو روش آموزش در مرحله تعمیم‌دهی با تولید نسل مختلف.

شبکه‌ها با تولید نسل‌های متفاوت به صورت نمودار در شکل ۶ و ۷ نمایش داده شده است. با مقایسه‌ی این نمودارها مشخص می‌شود که شبکه‌ی ترکیبی پیشنهادی زودتر آموزش می‌بیند و در تعمیم‌دهی خطای کم‌تری دارد. همچنین این نمودارها نشان می‌دهند که روند آموزش الگوریتم ژنتیک اصلاح شده کندتر است اما خطای تعمیم‌دهی آن کم‌تر است. چنان که در شکل ۶ مشاهده می‌شود شبکه‌ها با تولید نسل ۴۰۰۰ کم‌ترین مقدار خطا را دارند؛ اما با مقایسه‌ی نتایج تعمیم‌دهی در شکل ۷ مشخص می‌شود که خطای بعضی شبکه‌ها با ۲۰۰۰ تولید نسل و بعضی با ۳۰۰۰ تولید نسل کم‌تر از هنگامی است که شبکه با ۴۰۰۰ تولید نسل آموزش می‌بیند. لذا شبکه برای این که کم‌ترین مقدار خطا را داشته باشد نباید بیش از حد آموزش داده شود. البته معیار مشخصی برای این که نشان دهد چه مقدار شبکه باید آموزش داده شود وجود ندارد. این مقدار تنها با آزمون و خطا و از نتیجه‌ی تعمیم‌دهی مناسب شبکه به دست می‌آید. در شکل ۸ و ۹، روند کاهش خطای شبکه ترکیبی پیشنهادی نسبت به افزایش تولید نسل‌های مختلف با دو روش آموزش، برای داده‌های آموزشی و اعتبارسنجی نشان داده شده است. چنان که در شکل ۸ نشان داده شده است، هر دو شبکه به یک میزان آموزش داده شده، اما در شبکه‌ی بهینه‌شده میزان کاهش خطا در تعمیم‌دهی بیشتر است.

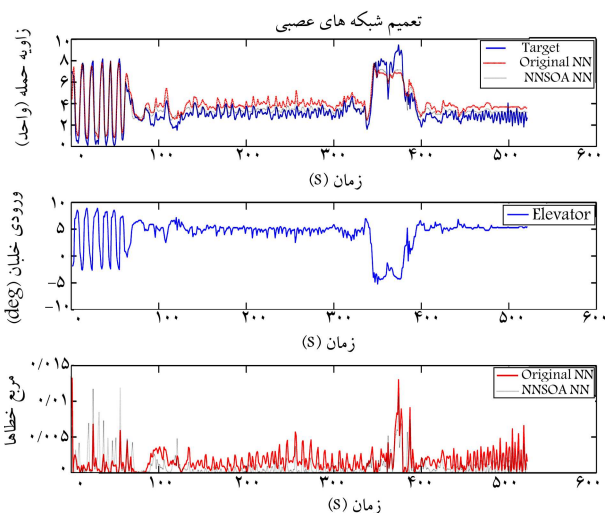
در شکل ۱۰ تا ۱۴، نتایج عملکرد شبکه برای داده‌های آموزش و تعمیم‌دهی برای ورودی‌های مختلف نشان داده شده است. برای داده‌های تجربی، آموزش با ورودی چندپله‌یی و سینوسی برای مدت ۱۲۰ ثانیه انجام شده است. برای تعمیم‌دهی شبکه با یک ورودی جدید به مدت ۵۰۰ ثانیه خروجی شبکه با داده‌های واقعی پرواز مورد مقایسه قرار گرفته است. هر یک از این شکل‌ها شامل سه بخش است: بخش اول مربوط به مقایسه‌ی خروجی شبکه‌ی اصلی و شبکه‌ی بهینه و با خروجی مطلوب است. بخش وسط نشان‌گر ورودی اعمالی به شبکه‌ها، و بخش سوم نیز نشان‌گر خطای خروجی دو شبکه است. با مقایسه‌ی این شکل‌ها درمی‌یابیم که



شکل ۱۰. مقدار میانگین مربعات خطا سه نوع شبکه در مرحله آموزش با تولید نسل مختلف با ورودی یکسان.



شکل ۱۳. مقایسه‌ی نتایج تعمیم‌دهی شبکه با داده‌های تجربی استخراج شده از تست پرواز با ورودی جدید.

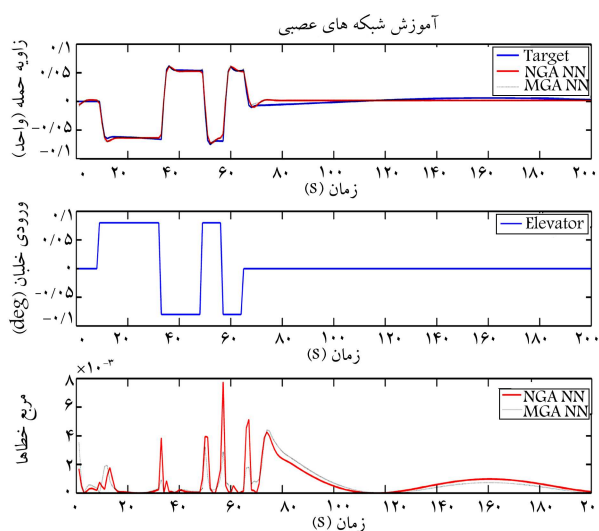


شکل ۱۴. مقایسه‌ی نتایج تعمیم‌دهی شبکه با داده‌های تجربی استخراج شده از تست پرواز با ورودی جدید.

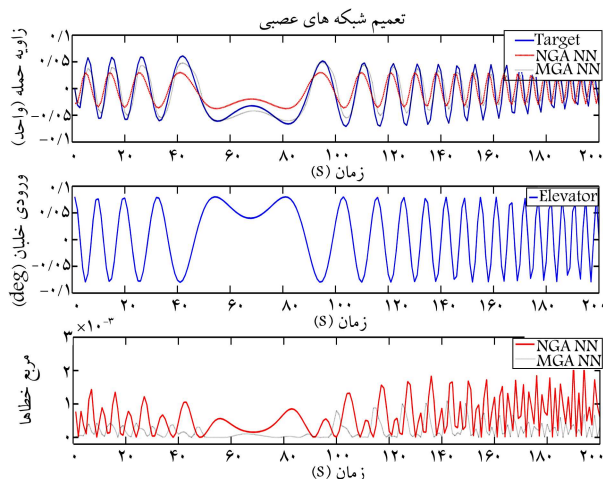
اگر شبکه با الگوریتم ژنتیک اصلاح شده مورد آموزش قرار گیرد نتایج بهتری در تعمیم‌دهی داده‌های جدید خواهیم داشت.

۷. نتیجه‌گیری

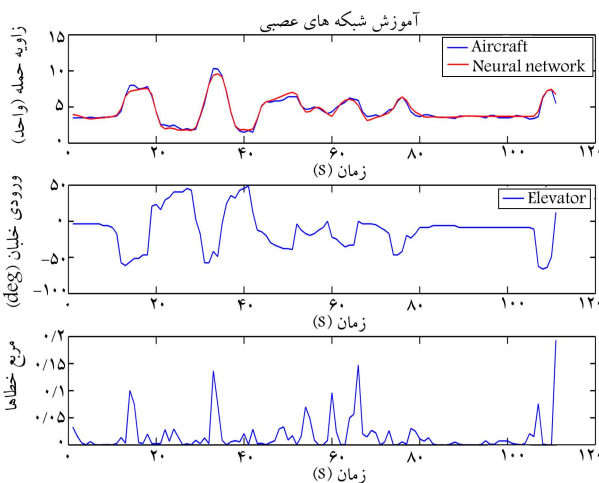
در این نوشتار با اعمال الگوریتم تکاملی برای بهینه‌سازی همزمان پارامترهای شبکه‌ی عصبی در سه سطح ارتباط‌های وزنی، معماری و قوانین یادگیری، شبکه‌ی عصبی صرفه‌جو معرفی شد که از پتانسیل خوبی برای شناسایی و مدل‌سازی دینامیک غیرخطی هواپیما با قابلیت مانور بالا برخوردار است. برای بهبود تعمیم‌دهی شبکه‌ی عصبی برای مدل‌سازی و شناسایی دینامیک هواپیما از الگوریتم ژنتیک معمولی و الگوریتم ژنتیک اصلاح شده برای بهینه‌سازی، و آموزش سه نوع ساختار شبکه‌ی بازگشتی المان اصلاح شده شبکه‌ی ترکیبی بازگشتی پیشنهادی و شبکه‌ی جردن مورد استفاده و مقایسه قرار گرفت. نتایج آموزش و تعمیم‌دهی با این دو روش یادگیری نشان می‌دهد که شبکه‌های عصبی توانایی شناسایی و مدل‌سازی دینامیک غیرخطی



شکل ۱۵. مقایسه‌ی نتایج آموزش شبکه با داده‌های شبیه‌سازی شده با ورودی چندپله‌یی.



شکل ۱۱. مقایسه‌ی نتایج تعمیم‌دهی شبکه با داده‌های شبیه‌سازی شده با ورودی چیرپ.



شکل ۱۲. مقایسه‌ی نتایج آموزش شبکه با داده‌های تجربی استخراج شده از تست پرواز با ورودی چندپله‌یی.

لایه‌ی میانی؛

$\delta_E, \delta_A, \delta_R$: ورودی‌های هواپیما یعنی مقدار تغییر زوایای رادر، شهپر و بالابر؛

u', v', w' : شتاب‌های خطی هواپیما در محور بدنی؛

u, v, w : سرعت‌های اغتشاشی خطی هواپیما؛

U : سرعت رو به جلو (در راستای x)؛

$I_{xx}, I_{yy}, I_{zz}, I_{xz}$: ماتریس گشتاور اینرسی؛

α, β : زاویه سرش جانبی، زاویه حمله؛

ψ, φ, θ : زاویه پیچ، زاویه رول، زاویه باو؛

p, q, r : سرعت زاویه‌یی حول سه محور؛

P_m, P_{m^2} : احتمال برای عملگر جهش - ۲، احتمال جهش ژنتیکی؛

T, O : بردار خروجی شبکه‌ی عصبی و بردار خروجی مطلوب شبکه‌ی عصبی؛

X_n, X : بردارهای متغیر حالت و متغیر حالت بهنجار شده؛

n_d : تعداد نقاط داده؛

$L_{\delta R}, L_{\delta A}, L_r, L_q, L_p, L_\beta$: ضرایب ابعادی تغییرات گشتاور رول با زاویه سرش

جانبی، نرخ زاویه رول، نرخ زاویه پیچ، نرخ زاویه باو، زاویه ایلران و زاویه رادر؛

$N_{\delta R}, N_{\delta A}, N_r, N_q, N_p, N_{T\beta}, N_\beta$: ضریب ابعادی تغییرات گشتاور باو با

زاویه سرش جانبی، نرخ زاویه رول، نرخ زاویه پیچ، نرخ زاویه باو، زاویه ایلران و زاویه

رادر باو؛

$Y_{\delta R}, Y_{\delta A}, Y_r, Y_q, Y_p, Y_\beta$: ضریب ابعادی تغییرات نیرو در راستای y با زاویه

سرش جانبی، نرخ زاویه رول، نرخ زاویه پیچ، نرخ زاویه باو، زاویه ایلران و زاویه رادر باو؛

s : عملگر لاپلاس.

هواپیما را دارند و همچنین نشان داده شده که بهینه‌سازی همزمان معماری شبکه، تعمیم‌دهی مناسب‌تری نسبت به روش آموزش با الگوریتم ژنتیک معمولی دارد، اگرچه زمان آموزش با این روش طولانی‌تر خواهد شد. با استفاده از نتایج این تحقیق می‌توان برای وسایل پرنده که مدل آیرودینامیکی آن‌ها در دسترس نیست، یا تهیهی آن وقت و هزینه‌ی زیادی می‌طلبد، از طریق ثبت داده‌های آزمایش‌های پروازی و آموزش مناسب شبکه‌ی عصبی مدل دینامیکی هواپیما را شناسایی کرد و از آن مدل در شبیه‌ساز پروازی استفاده کرد.

فهرست علائم

U, Y_1^1, Y_n^1 : بردار خروجی، لایه‌ی پنهان شبکه و بردار ورودی فرایند یا سیستم دینامیکی یا ورودی شبکه‌ی عصبی؛

$b^1, b^2, JW^1, CW^1, CW^2, LW, IW$: ماتریس‌های وزن‌های به ترتیب بین

لایه‌ی ورودی و پنهان، بین لایه‌ی پنهان و خروجی، بین لایه‌ی زمینیه و خروجی

و ماتریس وزن‌های اتصال‌های خود بازگشتی در لایه‌ی زمینیه، ماتریس وزن خود

بازخورد لایه‌ی زمینیه خروجی به پنهان، ماتریس بایاس‌های لایه‌ی پنهان و خروجی،

ماتریس وزن خود بازخورد لایه‌ی زمینیه خروجی به پنهان؛

W : ماتریس وزن‌های شبکه‌ی عصبی؛

Cn, Cy : بردار ورودی نورون‌های زمینیه در لایه‌ی خروجی و میانی؛

$purlin, tan\ sin$: تابع تبدیل تانژانت سیگموئید، تابع تبدیل خطی لایه‌ی خروجی

پانویس‌ها

1. coupled
2. quasi-linearization
3. nonlinear autoregressive with exogenous (NARX)
4. nonlinear output error
5. neural network autoregressive moving average with exogenous
6. pruning
7. decremental algorithm
8. incremental algorithm
9. error backpropagation
10. inconsistent
11. unpredictable
12. overfitting
13. modified genetic algorithm
14. parsimonious
15. normal genetic algorithm
16. feedforward
17. recurrent neural networks
18. feedforward multilayer perceptron
19. Elman
20. Jordan
21. normalize
22. unnormalize
23. mean squared error
24. neural network simultaneous optimization algorithm (NNSOA)

25. root mean squared error (RMSE)
26. sum squared error (SSE)
27. Chirp
28. flight data recorder (FDR)
29. sweep frequency
30. smoothness
31. square wave
32. Chirp Signal

منابع (References)

1. Ljung, L., *System Identification: Theory for the User*, 2nd, ed., PTR Prentice Hall, Upper saddle River, New Jersey (1999).
2. Nelles, O., *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, New York (2001).
3. Tischler, M.B. and Remple, R.K., *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples*, AIAA, Blacksburg, USA (2006).
4. Klein, V. and Morelli, E.A., *Aircraft System Identification: Theory and Practice*, AIAA Education Series, Reston, VA (2006).

5. Astrom, K.J. and Eykhoff, P.E. "System identification-a survey", *Automatica*, **7**, pp. 123-162 (1971).
6. Kalaba, R. and Spingarn, K., *Control, Identification and Input Optimization*, Plenum press (1982).
7. Kuzta, B. "Modeling and identification of dynamic systems", Van Nostrand Reinhold Company (1983).
8. Leontaritis, I.J. and Billings, S.A. "Input-output parametric models for non-linear systems, Part I: Deterministic non-linear systems", *Int. J. Contr.*, **41**(2), pp. 303-328 (1985).
9. Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P-Y., Hjalmarsson, H. and Juditsky, A. "Nonlinear black-box modeling in system identification: A unified overview", *Automatica*, **31**(12), pp. 1691-1724 (1995).
10. Saghafi, F. and Heravi, B.M. "A new approach to the modeling of highly maneuver aircrafts using neural networks", *Proc. 25th International Council of the Aeronautical Sciences (ICAS2006)*, Hamburg, Germany (2006).
11. Buskey, G., Wyeth, G. and Roberts, J. "Autonomous helicopter hover using an artificial neural network", *Proceedings of the 2001 ICRA IEEE International Conference on Robotics and Automation, Seoul, Korea*, **2**, pp. 1635-1640 (May 2001).
12. Tee, K.P., Sam Ge, S. and Tay, F.E.H. "Adaptive neural network control for helicopters in vertical flight", *IEEE Transactions on Control Systems Technology*, **16**(4), pp. 753-762 (July 2008).
13. Kechang, Q. and Jian, L. "Control method of dynamic inversion with neural network used for multi-variable nonlinear flight control system", IEEE, Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on 22-23, pp. 1-4 (May 2010).
14. Marwaha, M., Valasek, J. and Singla, P. "GLOMAP approach for nonlinear system identification of aircraft dynamics using flight data", AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, Hawaii, pp. 1-19 (18-21 August 2008).
15. Suresh, S., Kannan, N., Omkar, S.N. and Mani, V. "Non-linear lateral command control using neural network for F-16 aircraft", 2005 American Control Conference, pp. 8-10 (June 2005).
16. Saghafi, F. and Heravi, B.M. "Identification of aircraft dynamics using neural network simultaneous optimization algorithm", *Proceedings of the European Modeling and Simulation Conference (ESM 2005)*, Porto, Portugal (October 2005).
17. Qianqian, R. and Yi, F. "Flight dynamics identification of a helicopter in hovering based on flight data", IEEE International Conference on Artificial Intelligence and Computational Intelligence (2010).
18. Boëly, N. and Mihaela Botez, R., "New approach for the identification and validation of a nonlinear F/A-18 model by use of neural networks", *IEEE Transactions on Neural Networks*, **21**(11), pp.1759-1765 (2010).
19. Hamdy Kassem, A. "Efficient neural network modeling for flight and space dynamics simulation", *Hindawi Publishing Corporation International Journal of Aerospace Engineering*, Article ID 247294 (2011).
20. Zipfel, P.H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Inc., Rensselaer, Virginia (2000).
21. Bebis, G., Georgiopoulos, M. and Kasparis, T. "Coupling weight elimination with genetic algorithms to reduce network size and preserve generalization", *Neurocomputing*, **17**, pp. 167-194 (1997).
22. Castillo, P.A., Merelo, J.J., Prieto, A., Rivas, V. and Romero, G. "G-Prop: Global optimization of multilayer perceptrons using GAs", *Neurocomputing*, **35**, pp. 149-163 (2000).
23. Sexton, R.S., Dorsey, R.E. and Johnson, J.D. "Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation", *Decision Support Systems*, **22**, pp. 171-185 (1998).
24. Sexton, R.S., Dorsey, R.E. and Sikander, N.A. "Simultaneous optimization of neural network function and architecture algorithm", *Decision Support Systems*, **36**, pp. 283-296 (2004).
25. Sexton, R.S., Dorsey, R.E. and Johnson, J.D. "Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing", *European Journal of Operational Research*, **114**, pp. 589-601 (1999).
26. Pham, D.T. and Karaboga, D. "Training elman and Jordan networks for system identification using genetic algorithms", *Artificial Intelligence in Engineering*, **13**, pp. 107-117 (1999).
27. Elman, J.L. "Finding structure in time, state vectors were not zeroed between words", *Cognitive Science*, **14**, pp. 179-211 (1990).
28. Jordan, M. "Serial order: A parallel distributed processing approach: Technical report", Institute for Cognitive Science, University of California, USA (1986).
29. Schaffer, J.D., Whitley, D. and Eshelman, L.J., *Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art*, COGANN- 92. Combinations of Genetic Algorithms and Neural Networks. Los Alamitos, CA: IEEE Computer Society Press, pp. 1-37 (1992).
30. Roskam J., *Airplane Flight Dynamics and Automatic Flight Controls*, Roskam Aviation and Engineering Corporation, First Edition, USA (1979).
31. Stevens, B.L. and Lewis, F.L., *Aircraft Control and Simulation*, John Wiley and Sons Inc (1992).
32. Plaetschke, E., Mulder, J.A. and Breeman, J.H., "Flight test results of five input signals for aircraft parameter estimation", *Proceedings of the 6th IFAC Symposium on Identification and System Parameter Estimation*, Washington, D.C., **2**, pp. 1149-1154 (January 1982).