



یافتن جواب‌های بهین دسته‌ای از مسائل بهینه‌سازی پارامتری بر حسب مقادیر پارامتر با استفاده از شبکه‌های عصبی چندلایه

کبری محمدصلاحي، فرزین مدرس خیابانی^{*}، نیما آذر میر شتربانی

گروه ریاضی، دانشگاه آزاد اسلامی، تبریز، ایران

دبیر مسئول: سهراب عفتی

تاریخ پذیرش: ۱۴۰۰/۷/۱۶

تاریخ دریافت: ۱۴۰۰/۴/۱۹

چکیده: در این مقاله، مسائل بهینه‌سازی پارامتری مورد بررسی قرار گرفته‌اند. در یک مساله بهینه‌سازی پارامتری فرض می‌شود $\lambda \in \mathbb{R}^n$ بردار پارامترها و $x^* \in \mathbb{R}^n$ جواب بهین متناظر با λ باشد. هدف این تحقیق مشخص کردن تابعی مانند ψ است که $\psi(\lambda) = x^*$ جواب بهین متناظر با λ باشد. برای این کار ابتدا به‌ازای هر مقدار λ ، جواب بهین متناظر محاسبه می‌شود. بدین ترتیب یک مجموعه از داده‌های آموزشی متشکل از پارامترها و مقادیر بهین آنها به‌دست می‌آید. یک شبکه عصبی چندلایه را توسط داده‌های آموزشی، آموزش داده و در نتیجه عملکرد ψ در یک دامنه معلوم توسط این شبکه عصبی مشخص می‌شود. در واقع تابع ψ به‌ازای هر مقدار از پارامتر (پارامترها)، جواب بهین متناظر را توسط شبکه چندلایه آموزش دیده مشخص می‌کند. در نهایت چند مثال برای بررسی کارایی روش ارائه می‌شود.

واژه‌های کلیدی: بهینه‌سازی پارامتری، شبکه‌های عصبی چندلایه، شبکه‌های عصبی بازگشتی، بهینه‌سازی بدون مشتق.

رده‌بندی ریاضی: 90C31; 92-XX

۱ مقدمه

در مدل‌سازی‌های مختلف علمی، با بهینه‌سازی پارامتری مواجه می‌شویم. اولین مطالعه در حوزه بهینه‌سازی پارامتری توسط مان در سال ۱۹۵۳، [۱۱] انجام شد. تحقیقات انجام شده در بهینه‌سازی پارامتری از جنبه‌های مختلف خطی یا غیرخطی و تک‌پارامتری یا چندپارامتری حائز اهمیت و قابل تفکیک‌اند. اکثر مسائل بهینه‌سازی پارامتری از مدل‌سازی یک پدیده طبیعی حاصل می‌شود. ژو و همکاران در تحقیق خود، یک تکنیک کنترل محدودیت تکاملی را برای بهینه‌سازی پارامتری از یک مدل ایمونوترابی سرطان ارائه داده‌اند. در این تحقیق، یک مساله بهینه‌سازی چندپارامتری مقید که از فرمول‌بندی مدل ایمونوترابی سرطان سینه بر اساس اصول بیولوژیکی حاصل شده، مورد بررسی قرار گرفته است [۲۹]. بزگان و همکاران، یک روش تقریبی از مرتبه کاملاً چندجمله‌ای برای حل دسته‌ای از مسائل بهینه‌سازی پارامتری ارائه

^{*} نویسنده مسئول مقاله

کرده‌اند [۶]. بر اساس این تحقیق، روش تقریبی برای حل مساله بهینه‌سازی پارامتری خطی به کار رفته است که در آن مقدار پارامتر در خط حقیقی نامنفی تغییر می‌کند، مساله ناپارامتری در زمان چندجمله‌ای قابل حل باشد و مقادیر حاصل از جواب‌های شدنی برای تابع هدف، مقادیر نامنفی و صحیح با یک کران بالای قابل محاسبه از مرتبه چندجمله‌ای باشند. نوع دیگری از مسائل چندپارامتری زمانی حاصل می‌شوند که دو تصمیم‌گیرنده وجود داشته باشد. در این دسته از مسائل بهینه‌سازی تصمیم‌گیری در دو سطح انجام شده و در نتیجه مساله بهینه‌سازی دو هدفی حاصل می‌شود. در [۲۳] الگوریتم‌های جدیدی برای به دست آوردن جواب‌های دقیق و جامع دو دسته از مسائل برنامه‌ریزی دو هدفی ارائه شده است. دسته اول مسائل برنامه‌ریزی خطی دو هدفی با متغیرهای صحیح و غیرصحیح بوده و دسته دوم، مسائل برنامه‌ریزی درجه دوم محدب دو هدفی با متغیرهای صحیح و غیرصحیح می‌باشند. این دو دسته از مسائل بهینه‌سازی چندپارامتری تبدیل می‌شوند. یک الگوریتم تقریبی برای حل دسته‌ای از مسائل بهینه‌سازی پارامتری خطی در [۵] ارائه شده است. بر اساس این تحقیق، برای تعدادی از مسائل بهینه‌سازی پارامتری شامل نسخه‌های پارامتری مساله کوتاه‌ترین مسیر، مساله واگذاری و مساله جریان کم‌ترین هزینه، یک تابع خطی تک‌ه‌ای به‌ازای هر مقدار پارامتر، مقدار هدف بهین را نظیر می‌کند. در نتیجه هر الگوریتم برای حل چنین مسائلی یک تعداد از جواب‌های فوق چندجمله‌ای (متحنی که دارای چند نقطه است که شیب در آن نقاط تغییر می‌یابد) حاصل می‌کند. ضمانت تقریب به دست آمده از این روش برای حل مسائل پارامتری، وابسته به ضمانت تقریب الگوریتم استفاده شده برای مساله ناپارامتری متناظر است. هوان و همکاران در [۱۰] نشان داده‌اند که برای برقراری پایداری رابینسون و خاصیت شبه لیپ شیتزی در نقاط ایستایی از یک مساله بهینه‌سازی مقید پارامتری چگونه مشتقات مرتبه اول و دوم تابع هدف و قیدها با هم ترکیب می‌شوند. در تحقیق کاربردی دیگری، روش ماشین کاری تخلیه الکتریکی برش سیم، برای حل مسائل بهینه‌سازی پارامتری به کار برده شده است [۸]. این روش یک مدل‌سازی از ماشین کاری بدون تماس دست است که برای مواد سخت از جمله تیتانیوم، نیکل و آلیاژهای مختلف استفاده می‌شود. در این مدل‌سازی مساله بهینه ایجاد شده دارای چهار پارامتر به نام‌های زمان شروع (T_{on})، زمان پایان (T_{off})، جریان (I) و سرعت سیم (WS) است. ثابت می‌شود که پارامترهای بهین در این مدل‌سازی عبارتند از:

$$T_{on} = 13 \mu s, T_{off} = 10 \mu s, I = 1 A, WS = 850 rpm.$$

جانانان و همکاران در تحقیق جدیدی یک مدل‌سازی از سازگاری بال‌های هواپیما نسبت به تغییرات شرایط پرواز به صورت یک مساله برنامه‌ریزی پارامتری انجام داده‌اند [۱۴]. چالش اصلی در این تحقیق این است که شکل بهین بال چگونه تغییر کند که در هر شرایط بهینگی برقرار شود. تکنیک‌های بهینه‌سازی مهندسی مرسوم قادرند این مساله را تحت یک شرط حل کنند اما بهینه‌سازی پارامتری موجب می‌شود دسته‌ای از مسائل بهینه‌سازی به‌طور هم‌زمان مورد بررسی قرار گیرند. مسائل بهینه‌سازی دوگان-تباهیده تک پارامتری توسط یک الگوریتم دنبال‌کننده مسیر اصلاح‌گر پیش‌گو در [۲۷] مورد بررسی قرار گرفته‌اند. در این مرجع، هر تکرار الگوریتم در طول مسیر شامل سه قسمت است. اول، یک گام اصلاح‌گر نیوتن برای متغیرهای اولیه و دوگان که از حل یک دستگاه معادلات خطی به دست می‌آیند، دوم، یک گام پیش‌گو برای متغیرهای اولیه و دوگان که به صورت جوابی از یک مساله برنامه‌ریزی درجه دوم پیدا می‌شوند و قسمت سوم، یک گام پرش برای متغیرهای دوگان است که به صورت جوابی از یک مساله برنامه‌ریزی خطی پیدا می‌شوند.

از جمله تحقیقات دیگر، در حوزه کاربرد بهینه‌سازی پارامتری، در بهینه‌سازی برداری می‌توان به [۱۵، ۲۱] و در بهینه‌سازی تصادفی به [۴] اشاره کرد. در مرجع [۲] ابزارهای جبری ارائه شده است که این ابزارها به تشکیل بهینه‌سازی پارامتری چندجمله‌ای کمک کرده و قادر به حل مسائل کنترل بهینه است و این مسائل در مدل‌های کنترل پیش‌بین ظاهر می‌شوند. در [۱۶] روش بهینه‌سازی پارامتری برای یادگیری ماشین به کار رفته است. در این روش، مساله کلاس‌بندی نقاط یک صفحه به دو مجموعه مجزا می‌تواند به صورت یک برنامه‌ریزی پارامتری مدل‌سازی شود. کاربرد دیگری از بهینه‌سازی پارامتری، استفاده از این مدل در علم شیمی است، طوری که قسمتی از تحقیق [۱] توسط بهینه‌سازی پارامتری حل می‌شود.

برای حل کامل یک مساله بهینه‌سازی پارامتری باید جواب‌های بهین مساله (مقادیر بهین متغیرها) نسبت به تغییرات مجاز پارامتر (پارامترها) مشخص شود. این تحلیل در برخی حالت‌ها به دلیل غیرخطی بودن مساله مشکل خواهد بود [۱۹، ۲۸]. در هر حال می‌توان با تعیین مقادیر مجاز پارامترها و تعیین حوزه‌های خاصی برای آنها، جواب بهین مسائل پارامتری در هر حوزه را یافت [۳]. برای آشنایی با روش‌های تحلیلی در حل مسائل بهینه‌سازی پارامتری به [۱۳] مراجعه شود.

در تحقیق حاضر، مسائل بهینه‌سازی چندپارامتری مورد ارزیابی قرار گرفته‌اند. به این صورت که نمونه‌هایی از دامنه مشخص برای پارامترها اختیار کرده و سپس توسط روش‌های موجود به‌ازای هر مقدار پارامتر جواب بهین را به دست می‌آوریم. در نتیجه یک مجموعه از داده‌های آموزشی حاصل می‌شود. مجموعه داده‌های آموزشی یک تابع جدولی برای درونیایی تابع ψ است. برای درونیایی این تابع در نقاط به دست آمده از یک شبکه عصبی چندلایه استفاده می‌کنیم. لازم به ذکر است که روش‌های درونیایی مرسوم برای درونیایی ψ نمی‌توانند مورد استفاده قرار گیرند. زیرا دامنه این تابع ممکن است یک فضای m بعدی باشد و درونیایی در ابعاد بالاتر مشکلات خاص خود را دارد. در نتیجه پس از آموزش کامل شبکه چندلایه عملکرد تابع ψ به‌ازای هر مقدار از پارامترها، بدون حل مساله بهینه‌سازی در دست است. توجه داریم که یافتن ضابطه ψ نیز کاری مشکل و گاهی ناممکن است.

در ادامه، سازماندهی مطالب به این شرح است: در فصل دوم مساله کلی را بیان کرده و سپس روش جمع‌آوری داده‌های آموزشی ارائه می‌شود. روش‌های جمع‌آوری داده‌ها نسبت به نوع مساله بهینه‌سازی حاصل شده متفاوت خواهد بود. سه روش شبکه یانگ گائو، شبکه عفتی و روش تابع جریمه روش‌های موجود در تحقیقات مختلف است که برای حل مسائل بهینه‌سازی مختلف بیان شده‌اند. در فصل سوم بحث

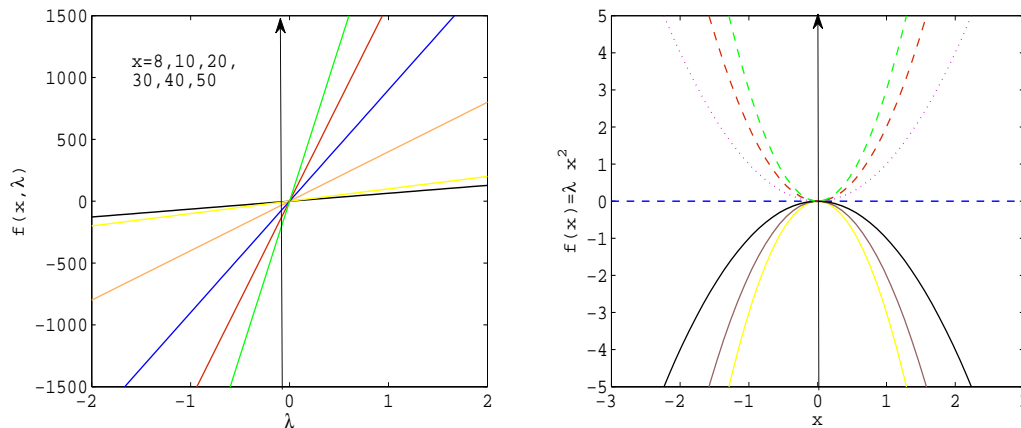
همگرایی روش بیان شده است. فصل چهارم مثال‌های عددی ارائه می‌شود و نهایتاً فصل پنجم نتیجه‌گیری خواهد بود.

۲ روش حل

در این فصل ابتدا مساله اصلی و روش پیشنهادی را بیان کرده و سپس روش‌های جمع‌آوری داده‌های آموزشی بیان می‌شوند.

۱.۲ بیان مساله و روش پیشنهادی

قبل از بیان مساله اصلی، تفاوت پارامتر و متغیر در یک مساله بهینه‌سازی توضیح داده می‌شود. تابع پارامتری $f(x, \lambda) = \lambda x^2$ را در نظر می‌گیریم که در آن $x, \lambda \in \mathbb{R}$ ($\lambda \neq 0$). اگر λ پارامتر بوده و x متغیر فرض شود آنگاه نمودار f سهمی‌های مختلف است که رئوس آنها در مبدا مختصات‌اند و اگر x پارامتر و λ متغیر باشد آنگاه نمودار حاصل خطوط با شیب مثبت‌اند. این دو حالت در شکل ۱ دیده می‌شوند.



شکل ۱: نمودارهای تابع پارامتری $f(x, \lambda) = \lambda x^2$ در حالتی که λ پارامتر یا متغیر است.

مساله بهینه‌سازی زیر را در نظر می‌گیریم:

$$\begin{aligned} \text{Min } f(x, \lambda) \\ x \in \Omega(x, \lambda) \end{aligned} \quad (1.2)$$

که در آن $x \in \mathbb{R}^n$ و $\Omega(x, \lambda)$ مجموعه‌ای از قیود برحسب متغیر x و پارامتر λ است و این مجموعه، مجموعه شدنی مساله است. به‌عنوان یک مثال، مساله بهینه‌سازی پارامتری زیر را در نظر می‌گیریم [۲]:

$$\begin{aligned} \text{Min } \lambda^2 x^2 - 2\lambda(1-\lambda)x \\ \Omega(x, \lambda) = \{x, \lambda \in \mathbb{R} | \lambda \geq 0, x \geq 0\} \end{aligned} \quad (2.2)$$

که در آن λ با شرایط ذکر شده پارامتر و x متغیر است. واضح است که جواب بهین برحسب پارامتر λ به‌صورت زیر است:

$$x^*(\lambda) = \begin{cases} [0, +\infty) & \lambda = 0 \\ \frac{1-\lambda}{\lambda} & 0 < \lambda < 1 \\ 0 & \lambda \geq 1. \end{cases} \quad (3.2)$$

توجه داریم که به‌ازای $\lambda = 0$ ، هر عدد حقیقی نامنفی جواب بهین است. در واقع رابطه (۳.۲) به‌ازای $\lambda > 0$ ضابطه تابع $\psi(\lambda)$ است و ضابطه این تابع در حالت کلی خصوصاً زمانی که تعداد پارامترها زیاد باشد قابل محاسبه نخواهد بود. هدف اصلی یافتن تابع ψ است به‌طوری‌که به‌ازای مقادیر مشخصی از بردار پارامتر λ ، جواب بهین x^* به‌فرم $x^* = \psi(\lambda)$ حاصل شود. برای این کار لازم است به‌ازای مقادیری از

پارامتر λ ، جواب‌های بهین متناظر محاسبه شوند و بدین ترتیب یک مجموعه از پارامترها و نقاط بهین متناظر با آنها به صورت $(\lambda, x^*(\lambda))$ به دست می‌آید؛ این مجموعه را داده‌های آموزشی می‌نامیم. مساله اصلی یافتن ضابطه تابع ψ در یک دامنه مشخص است. روش پیشنهادی برای حل مساله (۱.۲) به شرح زیر است:

فرض می‌کنیم جواب‌های بهین مساله (۱.۲) نسبت به تغییرات مقادیر پارامترها، تابعی پیوسته باشد. هدف این است که تابعی مانند

$$\begin{aligned} \psi : \mathbb{R}^m &\rightarrow \mathbb{R}^n \\ \psi(\lambda) &= x^* \end{aligned} \quad (۴.۲)$$

را مشخص کنیم که در آن بردار λ بردار شامل پارامترها بوده (مقادیر هر یک از پارامترها در یک بازه محدوداند) و x^* جواب بهین مساله بهینه‌سازی متناظر با پارامترهای λ است. واضح است که یافتن ضابطه تابع ψ به لحاظ ریاضی در عمل مشکل است. زمانی که $m = n = ۱$ ، آنگاه ضابطه ψ از روش‌های درونیایی در آنالیز عددی به صورت‌های مختلف قابل محاسبه است. اما وقتی که مقادیر m و n اعدادی بزرگتر از یک باشند آنگاه عمل یافتن ضابطه ψ کاری مشکل‌تر خواهد بود. در این حالت شبکه عصبی چندلایه قادر خواهد بود توسط خاصیت یادگیری خودش ضابطه تابع ψ را آموزش دیده و در اختیار ما قرار دهد. فرض کنیم $\lambda \in \mathbb{R}^m$ مقداری برای پارامترهای یک مساله بوده و $x^*(\lambda)$ جواب بهین مساله متناظر با λ باشد. اگر به تعداد کافی از این نمونه‌ها داشته باشیم آنگاه یک شبکه چندلایه قادر است رفتار تابع ψ را آموزش دیده و در نتیجه به ازای هر مقدار از پارامترهای مساله جواب بهین، بدون حل مجدد مساله در دست خواهد بود. مجموعه این نمونه‌ها را داده‌های آموزشی نامیده و به صورت زیر در نظر گرفته می‌شوند:

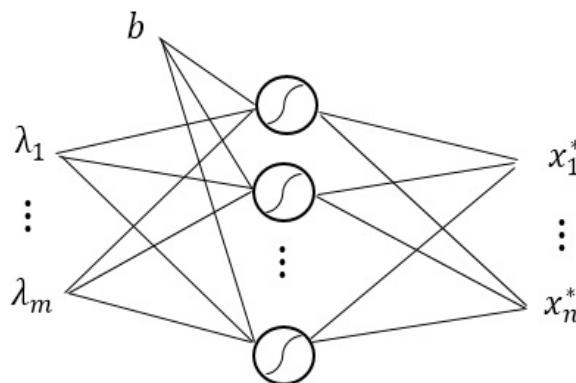
$$\mathcal{B} = \{(\lambda, x^*(\lambda)) \mid \lambda \text{ پارامتر و } x^* \text{ جواب بهین متناظر با آن است}\} \quad (۵.۲)$$

لازم است برای محاسبه اعضای مجموعه \mathcal{B} به ازای یافتن هر عضو، یک مساله بهینه‌سازی به طور کامل حل شود. حل دقیق هر یک از این مسائل اهمیت زیادی دارد، زیرا وجود خطا در این داده‌های آموزشی نهایتاً موجب بیش‌تر شدن خطا در برآورد کلی می‌شود.

اولین ایده برای حل مسائل بهینه‌سازی حاصل، در یافتن اعضای مجموعه \mathcal{B} ، استفاده از نرم‌افزارهای موجود است. برای این کار نرم‌افزارهای زیادی از جمله *Lindo*، *Lingo* [۱۷]، جعبه ابزار *Cplex*، *MATLAB* و چندین نرم‌افزار دیگر وجود دارد. به طور مثال در [۲۲، ۲۳] کارایی جعبه ابزار *MATLAB* برای حل مسائل بهینه‌سازی پارامتری توضیح داده شده است. صرف نظر از نرم‌افزارها، روش مهم و موثر دیگر برای حل مسائل بهینه‌سازی حاصل‌شده از جای‌گذاری پارامترها، استفاده از شبکه‌های عصبی بازگشتی (*RNN*) است [۱۸]. با یک برنامه‌نویسی دقیق می‌توان ابتدا توسط یک *RNN* داده‌های آموزشی را جمع‌آوری کرده سپس آنها را برای یادگیری یک شبکه چندلایه به کار گرفت. هدف اصلی در این مقاله، مشخص کردن تابع ψ توسط یک شبکه است که به ازای مقادیر دلخواه پارامتر، مساله بهینه‌سازی مجدداً حل نشود و جواب بهین متناظر با پارامتر داده‌شده سریعاً قابل محاسبه باشد.

پس از آماده شدن مجموعه \mathcal{B} قانون یادگیری شبکه *MLP* به دو روش می‌تواند انجام شود. یکی استفاده از قانون پس‌انتشار خطا و دیگری استفاده از جعبه ابزار *MATLAB* است. البته این دو روش در یک راستا عمل کرده و هر دو از قانون پس‌انتشار خطا استفاده می‌کنند زیرا نوع یادگیری، یادگیری با ناظر است.

شبکه عصبی *MLP* تابع ψ را درونیایی می‌کند. ساختار شبکه *MLP* با m ورودی و n خروجی در شکل ۲ نشان داده شده است. در شبکه شکل ۲، فقط یک لایه پنهان وجود دارد. بنابراین قضایای موجود خواهیم دید که این نوع شبکه برای تقریب هر تابع پیوسته مانند ψ کافی است. این شبکه را با علامت $(m - N - n)$ نشان خواهیم داد که در آن m تعداد ورودی‌ها، N تعداد نرون‌ها و n تعداد خروجی‌ها است.



شکل ۲: ساختار شبکه *MLP* برای آموزش تابع ψ .

۲.۲ روش‌های جمع‌آوری داده‌های آموزشی

مرحله مهم در فرآیند حل مساله بهینه‌سازی پارامتری که توسط یادگیری شبکه MLP انجام می‌شود جمع‌آوری داده‌های آموزشی است. اگر در این مرحله داده‌ها دارای خطا باشند، این خطا در یادگیری شبکه هم وجود خواهد داشت و در نهایت جواب‌ها دارای خطا خواهند بود. در واقع شبکه MLP هر آنچه را که به‌عنوان داده آموزشی می‌بیند، یاد می‌گیرد. استفاده از نرم‌افزارهای آماده برای جمع‌آوری داده‌ها دارای یک ایراد بزرگ است. علی‌رغم دقت قابل قبول نرم‌افزارها، وقتی یک مساله بهینه‌سازی توسط یک نرم‌افزار داده‌های آموزشی را تولید کند، از آنجایی که غالباً تعداد آنها بیش از ۵۰ عدد است، برای انتقال آنها به محیط دیگر جهت یادگیری شبکه MLP لازم است ارتباط بین دو نرم‌افزار ایجاد شود و این در برخی موارد غیر ممکن است. به‌طور مثال، آیا بین خروجی‌های نرم‌افزار $Lingo$ یا $Cplex$ با نرم‌افزار $MATLAB$ به راحتی می‌توان ارتباط ایجاد کرد؟

در نتیجه، برای جمع‌آوری داده‌های آموزشی و یادگیری شبکه MLP بهتر است از یک نرم‌افزار استفاده شود. چون یادگیری در محیط $MATLAB$ است، لذا جمع‌آوری داده‌ها نیز بهتر است در آنجا انجام شود. موثرترین ابزار برای این کار استفاده از RNN ها است. در واقع به‌ازای هر یک از مقادیر پارامترها لازم است یک مساله بهینه‌سازی حل شود. خوشبختانه براساس تحقیقات انجام‌شده برای انواع مختلف مسائل بهینه‌سازی، RNN مناسبی وجود دارد. به‌طور مثال ممکن است مساله پارامتری به‌ازای هر مقدار پارامتر یک برنامه‌ریزی خطی ایجاد کند. در این صورت RNN باید بتواند این دسته از مسائل را حل کند. همچنین ممکن است مساله بهینه‌سازی ایجادشده درجه دوم با قيود خطی باشد. چند نمونه از RNN های استفاده‌شده در این تحقیق به شرح زیرند.

۱.۲.۲ شبکه یانگ گائو

یکی از RNN های موثر، شبکه یانگ [۳۰] است که در آن مسائل بهینه‌سازی به فرم زیر حل شده است:

$$\begin{aligned} & \text{Min } f(x) \\ & \text{S.t. : } \begin{cases} Ax = b \\ x \geq 0, \end{cases} \end{aligned} \quad (۶.۲)$$

در مساله (۶.۲)، $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ تابعی پیوسته و مشتق‌پذیر است و از \mathbb{R}^n به \mathbb{R} محدب است. همچنین $b \in \mathbb{R}^m$ و $rank(A) = m$ ، $0 < m < n$ برای حل مساله (۶.۲) به‌صورت زیر است:

$$\begin{cases} \frac{dx}{dt} = -(I - P)(\nabla f(x) - (\mu - x)^+) - Q(Ax - b) \\ \frac{d\mu}{dt} = -\mu + (\mu - x)^+, \end{cases} \quad (۷.۲)$$

که در آن $P = A^T(AA^T)^{-1}A$ ، $Q = A^T(AA^T)^{-1}$ و I ماتریس همانی $n \times n$ است. همچنین $\mu \in \mathbb{R}^n$ ، $\mu^+ = \max\{0, \mu_i\}$ و $(\mu_1^+, \dots, \mu_n^+)^T$ شبکه (۷.۲) را شبکه یانگ می‌نامیم. در تحقیق یانگ ثابت می‌شود که شبکه یانگ از نظر لیپانوف پایدار است.

۲.۲.۲ شبکه عفتی

عفتی و همکارش یک RNN برای حل مسائل پارامتری زیر طراحی کرده‌اند [۲۵]:

$$\begin{aligned} & \text{Min } f(x, \lambda) \\ & \text{S.t. : } \begin{cases} h_i(x, \lambda) \leq 0, \quad i = 1, \dots, m \\ \lambda \in [\alpha, \beta], \end{cases} \end{aligned} \quad (۸.۲)$$

که در آن α, β اعداد ثابت‌اند، $x \in \mathbb{R}^n$ و توابع $f(x, \lambda)$ و $h_i(x, \lambda)$ نسبت به مولفه اول محدب‌اند. RNN برای حل مساله بهینه‌سازی حاصل به‌ازای هر مقدار پارامتر λ به‌صورت زیر است:

$$x'(t, \lambda) = -\frac{\partial f(x(t), \lambda)}{\partial x} - K \sum_{i=1}^m h_i^+(x(t), \lambda) \frac{\partial h_i(x(t), \lambda)}{\partial x}, \quad (۹.۲)$$

که در آن:

$$\frac{\partial f(x(t), \lambda)}{\partial x} = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T, \quad \frac{\partial h_i(x(t), \lambda)}{\partial x} = \left(\frac{\partial h_i}{\partial x_1}, \dots, \frac{\partial h_i}{\partial x_n} \right)^T.$$

نقش K در شبکه (۹.۲) به‌عنوان تنظیم‌کننده سرعت همگرایی است. در اکثر RNN ها با بزرگ شدن مقدار K سرعت همگرایی بیش‌تر می‌شود و این پارامتر به‌نام شتاب‌دهنده شناخته شده است. شبکه عفتی پایدار لیپانوف است.

۳.۲.۲ روش تابع جریمه

در جمع‌آوری داده‌های آموزشی به ازای هر مقدار از بردار پارامتر یک مساله بهینه‌سازی باید حل شود. گاهی ممکن است مساله بهینه‌سازی حاصل محدب نبوده و یا هموار نباشد و ما نتوانیم از ابزاری مانند گرادیان استفاده کنیم. در چنین شرایطی بهتر است از روش مینیمم‌سازی توابع جریمه استفاده کنیم. یکی از روش‌های موثر در این زمینه، به شرح زیر است. مساله بهینه‌سازی زیر را در نظر می‌گیریم:

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{S.t} \quad & g(x) \leq 0, \end{aligned} \quad (10.2)$$

که در آن، $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ، $x \in \mathbb{R}^n$ یک تابع دلخواه است و $g(x) = (g_1(x), \dots, g_m(x))^T$ نیز توابع دلخواه‌اند. اگر x^* جواب بهین مساله (۱۰.۲) و M یک کران پایین برای $f(x^*)$ فرض شود ($f(x^*) \geq M$)، آنگاه تابع جریمه زیر را در نظر می‌گیریم:

$$V(x, M) = (f(x) - M)^2 + K \sum_{i=1}^m (g_i^+(x))^2. \quad (11.2)$$

در این رابطه، $g^+(x) = \max\{0, g(x)\}$ و K ضریب جمله جریمه است و با افزایش آن سرعت همگرایی افزایش می‌یابد. اکنون جواب بهین مساله

$$\text{Min}_x V(x, M). \quad (12.2)$$

را با x_1 نشان داده و یک مقدار جدید برای M از رابطه

$$M_{k+1} = M_k + \sqrt{V(x_1, M_k)}, \quad (13.2)$$

به دست می‌آوریم. با جای‌گذاری مقدار جدید M در (۱۲.۲) و حل مجدد آن به x_2 می‌رسیم. با ادامه این تکرارها به دو دنباله $\{x_k\}$ و $\{M_k\}$ می‌رسیم. لازم‌به‌ذکر است که برای یافتن جواب بهین مساله (۱۲.۲) می‌توان از روش‌های بهینه‌سازی نامقید مانند تندترین کاهش (SD) یا روش $BFGS$ استفاده کرد. اما اگر تابع V ناهموار باشد، می‌توان از روش‌های بدون مشتق استفاده کرد. یکی از روش‌های موثر و بدون مشتق روش بهینه‌سازی نلد-مید است [۲۶]. ثابت می‌شود دنباله $\{x_k\}$ به x^* و دنباله $\{M_k\}$ به $f(x^*)$ میل می‌کنند (تمرین ۱۰ از فصل ۱۲ [۹]). در این تحقیق برای به دست آوردن کمینه موجود در (۱۲.۲) از روش بدون مشتق نلد-مید استفاده شده است. روش تابع جریمه، از نوع RNN نبوده و برای مسائل بهینه‌سازی با شرایط خاص مانند نامحدب و ناهموار نیز مناسب است.

ملاحظه ۱.۲. پس از جمع‌آوری داده‌های آموزشی، جواب‌های بهین مساله اصلی به‌ازای برخی از مقادیر پارامترها در دست است اما برای این که مدیر پروژه بتواند جواب‌های بهین را به ازای هر مقدار دلخواه پارامتر بدون حل مساله اصلی مشاهده کند لازم است تابع ψ توسط یک شبکه چندلایه آموزش ببیند. در نتیجه پس از جمع‌آوری داده‌ها و آموزش شبکه MLP سه هدف زیر برآورده می‌شود.

(۱) می‌توان به ازای هر مقدار دلخواه پارامتر در محدوده تعریف شده، جواب بهین و مقدار بهین تابع هدف را محاسبه کرد.

(۲) با رسم نمودارهای مناسبی می‌توان جواب بهین را به ازای تمام مقادیر پارامتر مشاهده کرده و تحلیل حساسیت دقیق انجام داد. واضح است که اگر تعداد پارامترها بیش از یک مورد باشد آنگاه می‌توان با ثابت نگه داشتن تعدادی از آن‌ها جواب‌های بهین را بر حسب یک یا دو مورد آنها به صورت نموداری مشاهده کرد.

(۳) در این مقاله، فرض بر این است که جواب‌های بهین، چندگانه نباشند؛ زیرا شبکه MLP در فرآیند یادگیری به ازای هر ورودی فقط یک خروجی را می‌تواند آموزش ببیند (دقیقاً مانند توابع ریاضی). همچنین فرض بر این است که ψ تابعی پیوسته باشد. البته اگر چنین نباشد می‌توان با بررسی داده‌های آموزشی و کلاس‌بندی آنها باز هم عمل یادگیری را انجام داد که در این تحقیق به این مساله پرداخته نشده است.

۳ همگرایی

در این بخش، همگرایی و پایداری روش‌های استفاده شده بررسی می‌شوند. حل مسائل پارامتری در این مقاله، از دو قسمت تشکیل شده است. قسمت اول جمع‌آوری داده‌های آموزشی است. در این قسمت ابزارهای مورد استفاده یا نرم‌افزارهایند و یا از RNN ها استفاده شده است. در مورد RNN ها با توجه به تحقیقات ارائه‌شده پایداری لیاپانوف و نیز همگرایی آنها با توجه به مراجع ذکر شده محرزند. در قسمت دوم، ابزار اصلی مورد استفاده، شبکه‌های MLP اند. دو قضیه در زیر نشان می‌دهند که یک شبکه MLP مطابق با شکل ۲ فقط با یک لایه پنهان و تعداد کافی نرون در لایه پنهان قادر به تقریب هر تابع پیوسته با هر دقتی است. این موضوع توانایی شبکه‌های MLP در تقریب توابع را نشان می‌دهد.

قضیه ۱.۳. به ازای هر تابع پیوسته $f: [0, 1]^n \rightarrow \mathbb{R}$ ، توابعی مانند $g: \mathbb{R} \rightarrow \mathbb{R}$ و $s_i: [0, 1] \rightarrow [0, 1]$ ($i = 0, \dots, 2n$) و ثابت‌های v_j ($j = 1, \dots, n$) وجود دارند به طوری که:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2n} g \left[\sum_{j=1}^n v_j s_i(x_j) \right]. \quad (1.3)$$

□

اثبات. برای اثبات به مرجع [۷] مراجعه شود.

قضیه ۲.۳. به ازای هر تابع پیوسته $f: [0, 1]^n \rightarrow \mathbb{R}$ ، تابع کران‌دار $s: \mathbb{R} \rightarrow \mathbb{R}$ و هر $\varepsilon > 0$ ، عدد طبیعی H و ثابت‌های حقیقی b_i, v_i, w_{ij} ($i = 1, \dots, H, j = 1, \dots, n$) وجود دارند به طوری که:

$$\left| f(x_1, \dots, x_n) - \sum_{i=1}^H v_i s \left[\left(\sum_{j=1}^n w_{ij} x_j \right) + b_i \right] \right| < \varepsilon. \quad (2.3)$$

□

اثبات. برای اثبات به مرجع [۱۲] مراجعه شود.

دو قضیه ۱.۳ و ۲.۳ وجود و یکتایی شبکه چندلایه را جهت تقریب توابع پیوسته نشان می‌دهند، زیرا معادلات ریاضی شبکه چندلایه مذکور در شکل ۲ دقیقاً می‌تواند برابر با ضابطه تابع f در (۱.۳) باشد. در نتیجه دیده می‌شود که این نوع شبکه برای درونیابی‌های از ابعاد بالاتر مناسب بوده و مزیت آن نسبت به روش‌های مرسوم آنالیز عددی این است که وقتی ابعاد متغیر ورودی یعنی n بزرگ می‌شود، آنگاه پیچیدگی یادگیری شبکه تغییر نمی‌کند و همان قانون یادگیری که در ابعاد پایین به کار می‌رفت همچنان قابل استفاده است.

۴ مثال‌های عددی

در این بخش، مثال‌هایی از حالات مختلف بهینه‌سازی پارامتری بیان می‌شوند تا مراحل جمع‌آوری داده‌ها و یادگیری شبکه MLP بررسی شود. برنامه‌ها در نرم‌افزار $MATLAB$ اجرا شده و برای دریافت پاسخ سیستم در RNN ‌ها از روش عددی رانگ-کوتای مرتبه ۴ استفاده شده است. برنامه‌ها در محیط نرم‌افزار $MATLAB(2008)$ و در یک لپ‌تاپ با حافظه $RAM = 2GB$ و پردازنده $Intel(R), Core(TM)2$ اجرا شده‌اند. در همه مثال‌ها برای آموزش شبکه از جعبه ابزار $MATLAB$ و توسط دستور $newff$ استفاده شده است.

مثال ۱.۴. به عنوان اولین مثال، مساله (۲.۲) را در نظر می‌گیریم. در این مساله داده‌های آموزشی را به ازای مقادیر مختلف $\lambda \in (0, 4]$ توسط روش جریمه جمع‌آوری می‌کنیم. برای این کار یک بار بازه را به ۱۰ قسمت مساوی و بار دیگر به ۵۰ قسمت مساوی تقسیم می‌کنیم. لازم به ذکر است که در این روش، برای کمینه‌سازی تابع $V(x, M)$ از روش بهینه‌سازی نامقید نلدر-مید استفاده شده است. پس از مشخص شدن مجموعه داده‌های آموزشی از یک شبکه $(1 - 8 - 1)$ برای یادگیری داده‌ها استفاده می‌کنیم. تعداد سیکل‌ها ۲۰۰۰ بوده و نتایج حاصل از یادگیری شبکه MLP یکبار با ۱۰ داده آموزشی و بار دیگر با ۵۰ داده آموزشی به همراه جواب دقیق در شکل ۳ نشان داده شده است.

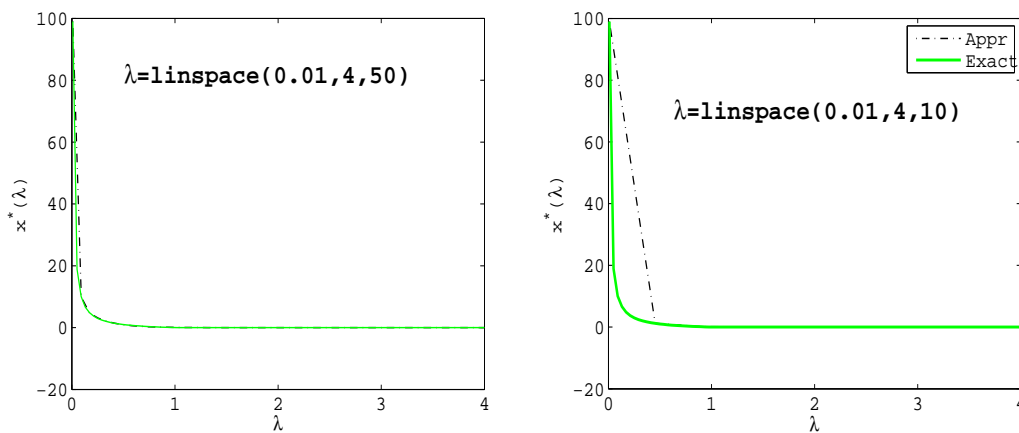
مثال ۲.۴. مساله بهینه‌سازی تک پارامتری زیر را در نظر می‌گیریم [۲۵]:

$$\begin{aligned} \text{Min} \quad & -x_1^2 - 3x_2^2 \\ \text{S.t.} \quad & \begin{cases} x_1 + x_2 \leq 6 - \lambda \\ -x_1 + 2x_2 \leq 6 + \lambda \\ x_1, x_2 \geq 0, \lambda \in [0, 6]. \end{cases} \end{aligned} \quad (1.4)$$

با استفاده از روش سیمپلکس [۲۰] جواب بهینه عبارت است از:

$$\forall \lambda \in [0, 2], x^*(\lambda) = \begin{bmatrix} 2 - \lambda \\ 4 \end{bmatrix}, z^*(\lambda) = -\lambda^2 + 4\lambda - 52,$$

$$\forall \lambda \in [2, 6], x^*(\lambda) = \begin{bmatrix} 0 \\ 6 - \lambda \end{bmatrix}, z^*(\lambda) = -3\lambda^2 + 36\lambda - 108.$$



شکل ۳: جواب‌های بهین مثال ۱.۴ برحسب پارامتر $\lambda \in (0, 4]$ با تعداد نقاط آموزشی ۱۰ (سمت راست) و ۵۰ (سمت چپ).

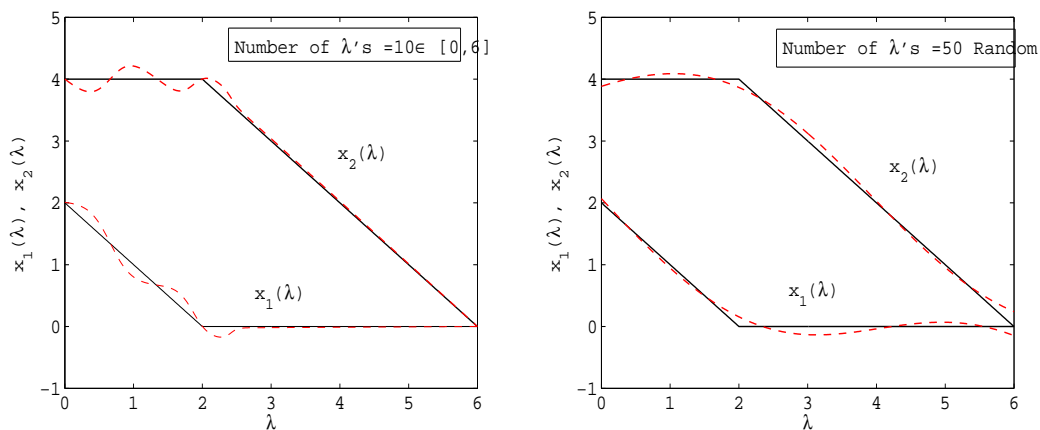
برای جمع‌آوری داده‌ها از شبکه عفتی استفاده می‌کنیم. این شبکه برای به‌دست آوردن جواب بهین به‌ازای هر مقدار پارامتر به‌صورت زیر است.

$$x'(t, \lambda) = \begin{bmatrix} x_1(t, \lambda) \\ x_2'(t, \lambda) \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 6x_2 \end{bmatrix} + \begin{bmatrix} -k \sum_{i=1}^4 h_i^+(x, \lambda) \frac{\partial h_i(x, \lambda)}{\partial x_1} \\ -k \sum_{i=1}^4 h_i^+(x, \lambda) \frac{\partial h_i(x, \lambda)}{\partial x_2} \end{bmatrix}. \quad (2.4)$$

ابتدا به تعداد ۱۰ داده آموزشی در بازه $[0, 6]$ توسط دستور $\text{linspace}(0, 6, 10)$ انتخاب می‌شود. میانگین مجموع مربعات خطا برای x_1 و x_2 پس از انتخاب داده‌ها آموزشی به صورت زیر است.

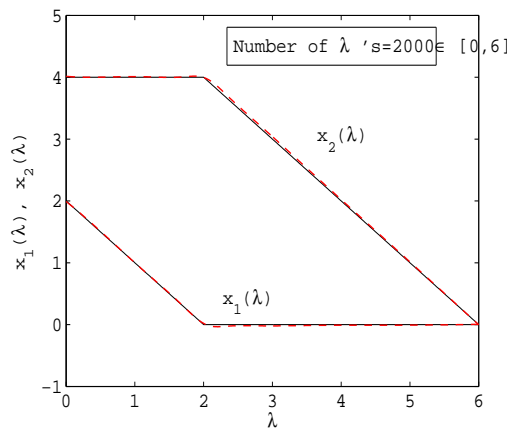
$$SSE(x_1) = 0.0031, \quad SSE(x_2) = 0.0062.$$

با انتخاب شبکه MLP به‌صورت $(1 - 10 - 2)$ فرآیند آموزش را شروع می‌کنیم. پس از ۸۰۰ سیکل شبکه MLP آماده ارائه جواب بهین برحسب مقادیر مختلف پارامتر است. نمودار تغییرات جواب بهین بر حسب پارامتر به ازای افراز بازه $[0, 6]$ به ۱۰ و بار دیگر به ۵۰ قسمت مساوی در شکل ۴ دیده می‌شود.



شکل ۴: جواب بهین مثال ۲.۴ برحسب ۱۰ مقدار برای پارامتر (سمت چپ) و ۵۰ مقدار برای پارامتر (سمت راست).

نهایتاً با انتخاب ۲۰۰۰ نقطه تصادفی به صورت $Rand(1, 2000) * 6$ از بازه $[0, 6]$ داده‌های آموزشی جمع‌آوری می‌شوند. مجموع مربعات خطا برای x_1 و x_2 به ترتیب $2.62E - 004$ و $5.17E - 004$ هستند. در RNN ، پاسخ سیستم توسط روش عددی رانگ-کوتای مرتبه ۴ به‌دست آمده و طول گام برابر 0.0005 انتخاب شده است. در اکثر مثال‌ها نیز مقدار K ضریب مربوط به سرعت همگرایی در RNN برابر ۱۰۰۰ در نظر گرفته شده است. شبکه MLP برای یادگیری داده‌ها به‌صورت $(2 - 12 - 1)$ بوده و نمودار شکل ۵ جواب‌های بهین بر حسب مقادیر پارامتر را نشان می‌دهد.



شکل ۵: جواب‌های بهین مثال ۲.۴ بر حسب ۲۰۰۰ مقدار پارامتر در بازه از ۰ تا ۶.

مثال ۳.۴. برنامه‌ریزی دو پارامتری و خطی زیر را در نظر می‌گیریم:

$$\begin{aligned}
 & \text{Min} \quad (1 - \lambda_1)x_1 + x_2 - 4x_3 \\
 & \text{S.t.} : \begin{cases} x_1 + x_2 + (2 + \lambda_2)x_3 \leq 9 \\ (1 + \lambda_1)x_1 + x_2 - x_3 \leq 2 \\ -x_1 + x_2 + (1 + \lambda_2)x_3 \leq 4 \\ x_1 \geq 0, x_2 \geq 0, \lambda_1 \in [0, 1], \lambda_2 \in [-1, 1]. \end{cases} \quad (3.4)
 \end{aligned}$$

در این مساله، داده آموزشی را توسط نرم افزار *Lingo* جمع‌آوری می‌کنیم. با توجه به بازه مجاز برای هر پارامتر، داده‌ها برای پارامترها به صورت زیر اختیار می‌شوند:

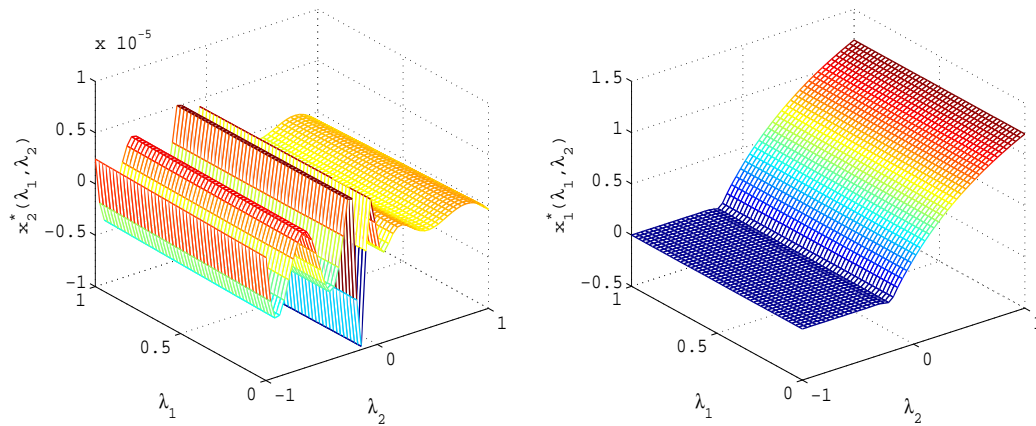
$$\lambda_1 = 0 : 0.1 : 1, \quad \lambda_2 = -1 : 0.1 : 1.$$

در واقع مقادیر مربوط به پارامترها از ناحیه $[0, 1] \times [-1, 1]$ انتخاب می‌شوند. طبیعی است که با افزایش تعداد پارامترها، تعداد داده‌های آموزشی نیز افزایش یافته، در نتیجه زمان اجرا نیز زیاد می‌شود. جواب بهین به ازای $\lambda_1 = \lambda_2 = 0$ عبارت است از: $x^*(0, 0) = (\frac{13}{3}, 0, \frac{1}{3})^T$. اکنون در یک تحلیل حساسیت خواسته شده است که تغییرات جواب بهین نسبت به تغییرات یک واحدی (افزایش و کاهش) در ضریب تکنولوژیکی متغیر x_3 در قیود اول و سوم را بررسی کرده و همچنین اثر کاهش یک واحدی در ضریب متغیر x_1 در تابع هدف و نیز افزایش یک واحدی ضریب تکنولوژیکی x_1 در قید دوم بر جواب بهین بررسی شود.

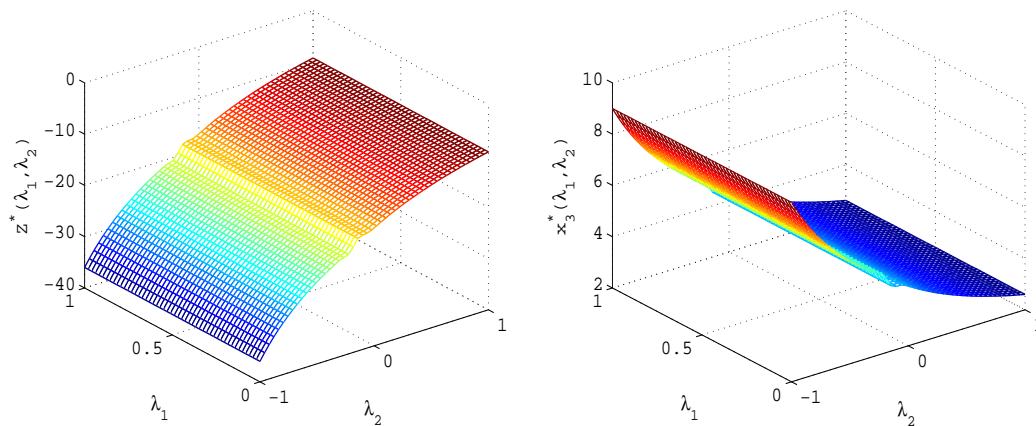
برای آموزش شبکه *MLP* توجه داریم که شبکه به صورت $(3 - 16 - 2)$ است. پس از ۵۰۰۰ سیکل، و تکمیل آموزش، نوبت به تحلیل جواب‌ها می‌رسد. شبکه *MLP* قادر است به‌ازای مقادیر دلخواه پارامترها از بازه‌های مشخص شده، جواب بهین را حاصل کند. اگر تغییرات موردی مد نظر باشد با همین شبکه *MLP* به هدف رسیده‌ایم. اما اگر نیاز باشد که بررسی x_1^*, x_2^*, x_3^* و یا Z^* نسبت به تغییرات هم‌زمان پارامترها انجام شود، باید نمودارهای مجزایی رسم شود. چون در این مساله تابع ψ یک تابع از فضای دو بعدی به سه بعدی است در نتیجه مجبوریم تغییرات تک تک موارد را نسبت به تغییرات پارامترها جداگانه رسم کنیم. در این راستا، نمودار هر یک از موارد x_1^*, x_2^*, x_3^* و یا Z^* نسبت به تغییرات پارامترها در اشکال ۶ و ۷ دیده می‌شوند.

در شکل ۶ توجه داریم که با تغییر مقادیر پارامتر λ_1 در بازه $[0, 1]$ و ثابت ماندن پارامتر دوم، مقدار جواب بهین x_1^* تغییر نیافته و ثابت است (سمت راست) اما با زیاد شدن پارامتر دوم از -1 تا نزدیکی صفر و ثابت ماندن پارامتر اول، مقدار x_1^* صفر بوده و از آن به بعد افزایش می‌یابد. نمودار موجود در سمت چپ شکل ۶ بیان می‌کند که به‌ازای تغییرات هر دو پارامتر، جواب بهین x_1^* تقریباً صفر باقی مانده و تغییر نمی‌کند. توجه داریم که نوسانات دیده شده در این شکل مقادیری در حد ده هزارم و صد هزارم بوده و ناشی از خطای محاسباتی است. اکنون به نمودار سمت راست شکل ۷ توجه می‌کنیم. با توجه به این نمودار می‌توان گفت به ازای هر مقدار λ_2 و تمام مقادیر پارامتر λ_1 جواب بهین x_2^* ثابت است و به ازای هر مقدار از λ_1 و با افزایش λ_2 جواب بهین x_2^* کاهش می‌یابد. از نمودار سمت چپ همین شکل نتیجه می‌گیریم که به‌ازای هر مقدار از λ_2 و تمام مقادیر λ_1 مقدار بهین تابع هدف ثابت می‌ماند و به ازای هر مقدار λ_1 و با افزایش مقادیر λ_2 آنگاه مقدار بهین تابع هدف نیز افزایش می‌یابد.

لازم به ذکر است که چون متغیرهای x_1^* و x_2^* در جدول نهایی سیمپلکس پایه‌ای‌اند تغییر ضرایب این متغیرها و تحلیل حساسیت آنها به ازای هر مقدار از پارامترها از روش‌های معمولی، مستلزم حل مجدد مساله است. در نتیجه ارزش یادگیری شبکه چندلایه در چنین مسائلی مشهود است.



شکل ۶: تغییرات مقادیر بهین x_1^* و x_2^* نسبت به تغییرات پارامترها در مثال ۳.۴.



شکل ۷: تغییرات مقادیر بهین x_3^* و Z^* نسبت به تغییرات پارامترها در مثال ۳.۴.

مثال ۴.۴. برنامه‌ریزی سه پارامتری و غیرخطی زیر را در نظر می‌گیریم [۳]:

$$\begin{aligned} \text{Min} \quad & x_1^* + x_2^* + \lambda_1 x_1 + \lambda_2 x_2 \\ \text{S.t.} : \quad & \begin{cases} x_1 + x_2 = b \\ x_1 \geq 0, x_2 \geq 0, \lambda_1, \lambda_2 \in \mathbb{R}, b \in [0, +\infty). \end{cases} \end{aligned} \tag{۴.۴}$$

این مساله در مرجع [۳] به‌طور تحلیلی حل شده و برای آزمودن جواب‌های حاصل از شبکه MLP ، جواب تحلیلی را به‌طور خلاصه بیان می‌کنیم. ابتدا فرض می‌شود $x_1^* = 0$. در این حالت مجموعه جواب‌های بهین بر حسب پارامترها را با I_1 نشان داده، مجموعه I_2 برای $x_1^* = 0$ ، مجموعه I_3 را برای $x_1^* = x_2^* = 0$ و مجموعه I_4 را برای حالت $(x_1^*, x_2^*) \neq (0, 0)$ تعریف می‌کنیم. بنابراین

$$I_1 = \{(\lambda_1, \lambda_2, b)^T \in \mathbb{R}^3 \mid \lambda_1 - \lambda_2 + 2b \geq 0, 0 \leq b < \infty\}, \tag{۵.۴}$$

$$I_2 = \{(\lambda_1, \lambda_2, b)^T \in \mathbb{R}^3 \mid \lambda_1 - \lambda_2 + 2b \leq 0, 0 \leq b < \infty\}, \tag{۶.۴}$$

$$I_3 = \{(\lambda_1, \lambda_2, b)^T \in \mathbb{R}^3 \mid \lambda_1, \lambda_2 \in \mathbb{R}, b = 0\}, \tag{۷.۴}$$

$$I_4 = \{(\lambda_1, \lambda_2, b)^T \in \mathbb{R}^3 \mid \lambda_1 - \lambda_2 + 2b \geq 0, \lambda_1 - \lambda_2 - 2b \leq 0, 0 \leq b < \infty\}, \tag{۸.۴}$$

در نتیجه جواب تحلیلی بهین برای متغیرها بر حسب پارامترها عبارت است از:

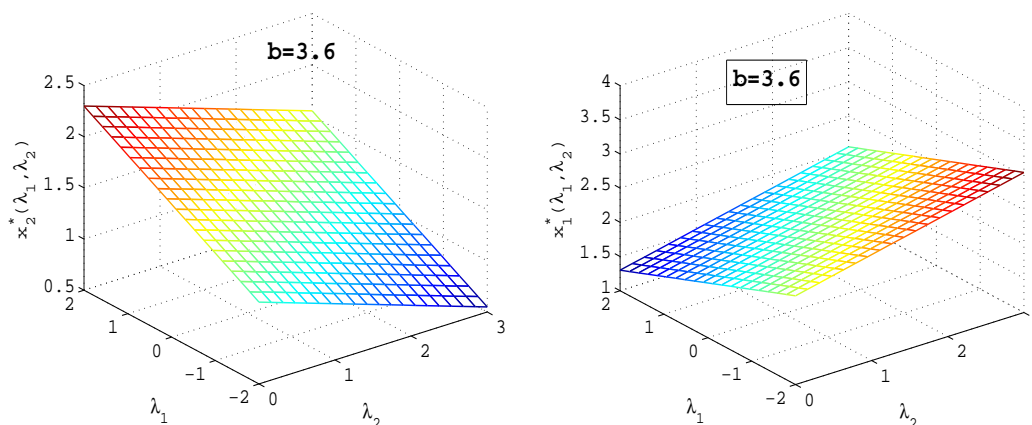
$$x^*(\lambda_1, \lambda_2, b) = \begin{cases} (0, b)^T & (\lambda_1, \lambda_2, b) \in I_1 \\ (b, 0)^T & (\lambda_1, \lambda_2, b) \in I_2 \\ (0, 0)^T & (\lambda_1, \lambda_2, b) \in I_3 \\ (-\frac{\lambda_1}{4} + \frac{\lambda_2}{4} + \frac{b}{4}, \frac{\lambda_1}{4} - \frac{\lambda_2}{4} + \frac{b}{4})^T & (\lambda_1, \lambda_2, b) \in I_4 \end{cases} \quad (9.4)$$

توجه داریم که حل تحلیلی مسائل پارامتری امری مشکل و زمان‌بر است، خصوصاً زمانی که تعداد متغیرها بیش‌تر شده و یا مساله غیرخطی باشد به مراتب مشکل‌تر خواهد بود.

برای جمع‌آوری داده‌های آموزشی با توجه ماهیت مساله و قیود آن از شبکه یانگ استفاده می‌کنیم. با توجه به این‌که در حل عددی، بازه مربوط به پارامتر باید محدود باشد، در جمع‌آوری داده‌ها مقادیر b به صورت زیر اختیار می‌شود.

$$b \in \{3, 3.5, 4, 4.5, 5\}$$

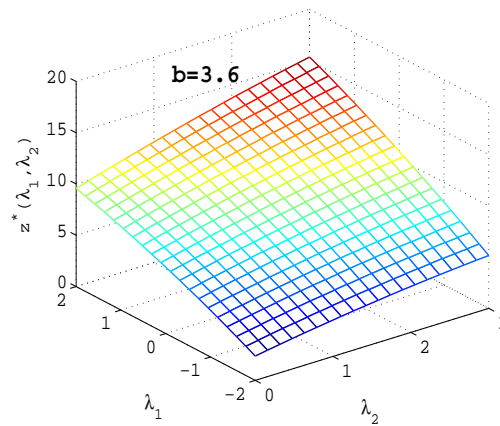
همچنین مقادیر پارامترهای λ_1 و λ_2 از بازه $[-2, 2] \times [0, 3]$ اختیار می‌شوند. در این مثال شبکه MLP به صورت $(2 - 3^0 - 3)$ است. سه پارامتر و دو متغیر وجود دارد و پس از یافتن مقادیر بهین برای متغیرها می‌توانیم مقدار بهین تابع هدف را با توجه به آنها به دست آوریم. اما با این‌که جواب بهین مساله اصلی بعد از آموزش شبکه MLP به ازای هر یک از مقادیر پارامترها در دست است اما دیدن همزمان تمام تغییرات متغیرها بر حسب پارامترها مقدور نیست. در نتیجه در هر مرحله، مقدار پارامتر b را ثابت نگه داشته و هر یک از مقادیر x_1^* و x_2^* را جداگانه بر حسب تغییرات پارامترهای (λ_1, λ_2) رسم می‌کنیم. همچنین چون مقادیر بهین متغیرها در دست است پس می‌توان تغییرات مقدار بهین تابع هدف را نیز بر حسب تغییرات پارامترهای (λ_1, λ_2) رسم کرد. به عنوان نمونه با فرض $b = 3.6$ ، تغییرات x_1^* ، x_2^* و z^* نسبت به تغییرات پارامترهای (λ_1, λ_2) در شکل‌های ۸ و ۹ رسم شده است.



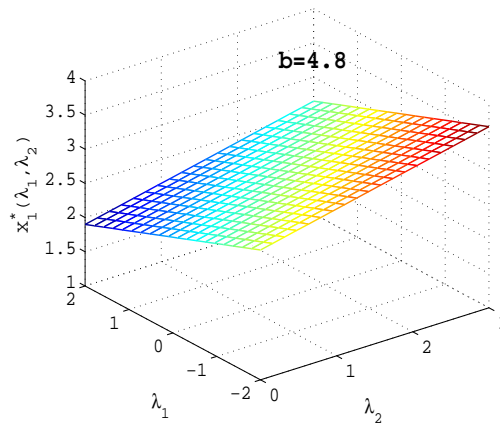
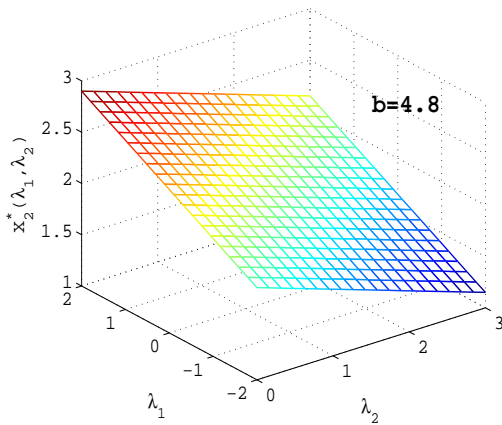
شکل ۸: تغییرات x_1^* و x_2^* نسبت به تغییرات (λ_1, λ_2) در مثال ۴.۴ به ازای $b = 3.6$

براساس این اشکال می‌توان نتیجه گرفت، با افزایش λ_2 و کاهش λ_1 مقدار x_1^* افزایش می‌یابد. همچنین با افزایش λ_1 و کاهش λ_2 مقدار x_2^* افزایش می‌یابد و نهایتاً با افزایش هر دو پارامتر λ_1 و λ_2 مقدار بهین تابع هدف افزایش می‌یابد. این نتیجه، برای تحلیل حساسیت فوق‌العاده اهمیت دارد.

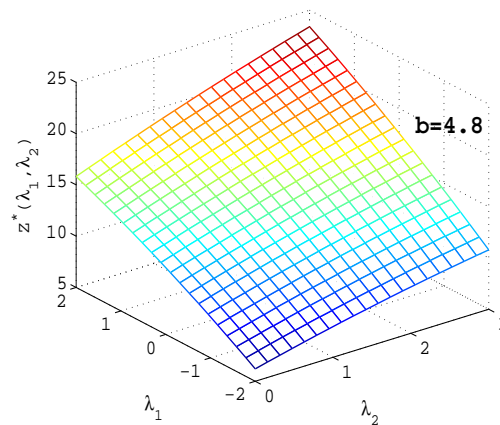
از شکل‌های ۱۰ و ۱۱ نتیجه می‌شود که با افزایش مقدار b (گاهی به معنی موجودی انبار) تغییرات متغیرها و نیز تابع هدف همانند حالت قبل است اما مقادیر عددی، نسبت به حالت قبل تغییر کرده‌اند.



شکل ۹: تغییرات z^* نسبت به تغییرات (λ_1, λ_2) در مثال ۴.۴ به ازای $b = ۳/۶$.



شکل ۱۰: تغییرات z^* نسبت به تغییرات (λ_1, λ_2) در مثال ۴.۴ به ازای $b = ۴/۸$.



شکل ۱۱: تغییرات z^* نسبت به تغییرات (λ_1, λ_2) در مثال ۴.۴ به ازای $b = ۴/۸$.

۵ نتیجه گیری

در این مقاله، مسائل بهینه‌سازی پارامتری توسط آموزش شبکه عصبی چندلایه حل شده است. در اکثر موارد پس از جمع‌آوری دقیق داده‌های آموزشی، اگر تعداد آنها کافی باشد نموداری از تغییرات جواب، بر حسب تغییرات پارامتر را در اختیار داریم. اما به‌ازای مقادیری از پارامتر که در

داده‌های آموزشی وجود ندارند نمی‌توانیم جواب بهین جدید را به دست آوریم و برای رسیدن به آنها مجدداً باید مساله را حل کنیم. در نتیجه مزیت مهم شبکه MLP این است که بعد از آموزش آن، به ازای هر مقدار از پارامترهای داده‌شده، جواب بهین و نیز مقدار بهین تابع هدف بدون حل مجدداً مساله در دست خواهد بود. این نتیجه موضوع تحلیل حساسیت را در برنامه‌ریزی خطی و غیرخطی هموار می‌کند. در ادامه این تحقیق می‌توان حالاتی را در نظر گرفت که تغییرات جواب بهین، نسبت به تغییرات مقادیر پارامترها پیوسته نباشد. مانند مثال ۱ در تحقیق [۲۵].

فهرست منابع

- [1] A. Azizi and M. Masdarian and A. Hassanzadeh and Z. Bahri and T. Niedoba and A. Surowiak, Parametric optimization in rougher Flotation performance of a sulfidized mixed Copper, Ore. Minerals **4** (5) (2020) 1–19.
- [2] A. Fotiou and A. P. Rostalski and P. A. Parrilo and M. Morari, Parametric optimization and optimal control using algebraic geometry methods, International Journal of Control. **79** (11) (2007) 1340–1358.
- [3] B. Bank and J. Guddat and D. Klatte and B. Kummer and K. Tammer, *Non-linear Parametric Optimization*, Springer Fachmedien Wiesbaden GmbH, 1993.
- [4] B. Bereanu, On stochastic linear programming II: distribution problems non stochastic matrix, Review Roumain Mathematical Pures Applications **11** (1966) 713–725 (II).
- [5] C. Bazgan and A. Herzel and S. Ruzika and C. Thielen and D. Vanderpooten, An approximation algorithm for a general class of parametric optimization problems, Journal of Combinatorial Optimization, (2020) 1–31.
- [6] C. Bazgan and A. Herzel and S. Ruzika and C. Thielen and D. Vanderpooten, An FPTAS for a general class of parametric optimization problems, The German Academic Exchange Service **186** (2019) 25–37.
- [7] D. A. Sprecher, On the structure of continuous functions of several variables, Transactions American Mathematical Society **115** (1965) 340–355.
- [8] D. Devarasiddappa and M. Chandrasekaran and R. Arunachalam, Experimental investigation and parametric optimization for minimizing surface roughness during WEDM of Ti6AL4V alloy using modified TLBO algorithm, Journal of the Brazilian Society of Mechanical Sciences and Engineering **42** (128) (2020) 1–18.
- [9] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading MA: Addison Wesley, Second Edition, 1984.
- [10] D. T. K. Huyen and J. C. Yao and N. D. Yen, The stationary point set map in general parametric optimization problems, Set-Valued and Variational Analysis, (2020) 1–23.
- [11] E. Simons, *Note on Parametric Linear Programming*, INFORMS **8** (3) (1962) 355–358.
- [12] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems **2** (4) (1989) 304–314.
- [13] J. Guddat and F. Guerra Vasquez and H. Th. Jongen, *Parametric Optimization: Singularities, Path-following and Jumps*, Springer Fachmedien Wiesbaden GmbH, 1990.

- [14] J. M. Weaver-Rosen and P. B. C. Leal and D. J. Hartl and R. J. Malak Jr., Parametric optimization for morphing structure design: application to morphing wings adapting to changing flight conditions, *Structural and Multidisciplinary Optimization* **62 (6)** (2020) 2995–3007.
- [15] J. M. Weaver-Rosen and R. J. Malak Jr., Efficient parametric optimization for expensive single objective problems, *Journal of Mechanical Design* **143 (3)** (2021) 1–9.
- [16] K. P. Bennett and E. J. Bredensteiner, A parametric optimization method for machine learning, *Informatics Journal on Computing*. **9 (3)** (1997) 311–318.
- [17] L. Schrage, *Optimization Modelling with LINGO*, LINDO Systems Inc., 2003.
- [18] M. M. Gupta and L. Jin and N. Homma, *Static and Dynamic Neural Networks*, IEEE, A John Wiley & Sons, INC. Publication, 2003.
- [19] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming Theory and Applications*, Wiley and Sons New York, 1990.
- [20] M. S. Bazaraa and J. Jarvis and D. Sherali, *Linear Programming and Network Flows*, John Wiley and Sons, 1992.
- [21] M. Zeleny, *Linear Multiobjective Programming*, Springer-Verlog, Berlin-Heidelberg, New York, 1974 (III).
- [22] R. Oberdieck and N. A. Diangelakis and M. M. Papatheasiou and I. Nascu and N. E. N. Pistikopoulos, PoP-Parametric optimization toolbox, *Industrial & Engineering Chemistry Research* **55 (33)** (2016) 8979–8991.
- [23] S. Avraamidou and E. N. Pistokopoulos, A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems, *Computers and Chemical Engineering*, **125** (2019) 98–113.
- [24] S. Avraamidou and E. N. Pistokopoulos, B-POP: Bi-level parametric optimization Toolbox, *Computers and Chemical Engineering* **122 (4)** (2018) 193–202.
- [25] S. Effati and M. Jafarzadeh, A new nonlinear neural network for solving a class of constrained parametric optimization problems, *Applied Mathematics and Computation* **186 (1)** (2007) 814–819.
- [26] S. S. Rao, *Engineering Optimization, Theory and Practice*, Purdue University, West Lafayette, Indiana, 1996.
- [27] V. Kungurtsev and J. Jaschke, A predictor-corrector path-following algorithm for double-degenerate parametric optimization problems, *Society for Industrial and Applied Mathematics* **27 (1)** (2017) 538–564.
- [28] W. M. Haddad and V. Chelboina, *Nonlinear Dynamical Systems and Control*, Princeton University Press, Princeton & Oxford, 2008.
- [29] W. Xu and J. X. Xu and D. He and K. C. Tan, An evolutionary constraint-handling technique for parametric optimization of a cancer immunotherapy model, *IEEE Transactions on Emerging Topics in Computational Intelligence* **3 (2)** (2018) 151–162.
- [30] Y. Yang and Y. Gao, A new neural network for solving nonlinear convex programs with linear constraints, *Neurocomputing* **74** (2011) 3079–3083.



Finding Optimal Solutions to a Class of Parametric Optimization Problems in Terms of Parameter Values by using Multilayer Neural Networks

Kobra Mohammadsalahi, Farzin Modarres Khiyabani[†], Nima Azarmir

Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Communicated by: Sohrab Effati

Received: 2021/7/10

Accepted: 2021/10/8

Abstract: In this paper, parametric optimization problems are investigated. In a parametric optimization problem we assume $\lambda \in \mathbb{R}^n$ is the vector of the parameters and x^* is the optimal answer corresponding to it. The purpose of this paper is to determine a function such as ψ so that we have $\psi(\lambda) = x^*$. To do this, first for each λ , the corresponding optimal answer is calculated. In this way, a set of data bases consisting of parameters and the corresponding optimal values are obtained. A multilayer network of data base is trained to obtain the function ψ in a domain. In fact, the function ψ for each value of the parameter specifies the corresponding answer by the trained multilayer network. Finally, we conduct several numerical examples to test our method.

Keywords: Parametric Optimization, Multilayer Neural Networks, Recurrent Neural Networks, Free Derivative Optimizatiuon.



©2021 Shahid Chamran University of Ahvaz, Ahvaz, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0 license) (<http://creativecommons.org/licenses/by-nc/4.0/>).

[†]Corresponding author.

E-mail addresses: salahi763@gmail.com (K. Mohammadsalahi), iauasrb082@gmail.com (F. M. Khiyabani), azarmir_nim@yahoo.com (N. Azarmir Shotorbani).