

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده علوم ریاضی
گروه ریاضی کاربردی

پایان نامه

برای دریافت درجه کارشناسی ارشد در رشته
ریاضی کاربردی، گرایش تحقیق در عملیات

عنوان

یک شبکه عصبی با کارایی بالا برای حل رده ای از مسائل برنامه ریزی غیر خطی با محدودیت های هیبریدی

استاد راهنما

دکتر علیرضا ناظمی

دانشجو

سمیرا سوخت سرایی

۱۳۹۳

تقدیم به مادر عزیز و پدر مهربانم

به پاس تعبیر عظیم و انسانی شان از کلمه ایثار و از خودگذشتگی شان

به پاس عاطفه سرشار و گرمای امید بخش وجودشان که در این سردترین روزگار ان بهترین پشتیبان من است

به پاس قلب های بزرگشان که فریادرس است و سرگردانی و ترس در پناهشان به شجاعت می گراید

و به پاس محبت های بی دریغشان که هرگز فروکش نمی کند.

و تقدیم به همه آن ها که مرا علم آموختند.

خدایا...

به من زیستنی عطا کن که در لحظه مرگ، بر بی‌ثمیری لحظه‌ای که برای زیستن گذشته است، حسرت نخورم و مردنی عطا کن که بر بیهودگیش، سوگوار نباشم. بگذار تا ان را، خود انتخاب کنم، اما آنچه‌ان که تو دوست می‌داری.

تو می‌دانی و همه می‌دانند شکنجه دیدن، بخاطر تو، زندانی کشیدن بخاطر تو و رنج بردن به پای تو تنها لذت زندگی من است، از شادی توست که من در دل می‌خندم، از امید رهایی توست که برق امید در چشمان خسته‌ام می‌درخشد و از خوشبختی توست که هوای پاک سعادت را در ریه‌هایم احساس می‌کنم. نمی‌توانم خوب حرف بزنم. نیروی شگفتی را که در زیر کلمات ساده و جمله‌های ضعیف و افتاده پنهان کرده‌ام دریاب، دریاب.

تو می‌دانی و همه می‌دانند که زندگی از تحمیل لبخندی بر لبان من، از آوردن برق امیدی در نگاه من، از برانگیختن موج شغفی در دل من، عاجز است.

تو، چگونه زیستن را به من بیاموز، چگونه مردن را خود خواهم آموخت.

به من توفیق تلاش در شکست، صبر در نومیدی، رفتن بی همراه، جهاد بی سلاح، کار بی پاداش، فداکاری در سکوت، دین بی دنیا، مذهب بی عوام، عظمت بی نام، خدمت بی نان، ایمان بی ریا، خوبی بی نمود، گستاخی بی خامی، قناعت بی غرور، عشق بی هوس، تنهایی در انبوه جمعیت، و دوست داشتن بی آنکه دوست بداند، روزی کن.^۱

^۱ مناجاتی از دکتر علی شریعتی

سپاس‌گزاری...

سپاس بیکران خدای را که مرا یاری داد تا در راهی قدم نهادم موفق گردم. در این راه پس از او ستایشگر کسانی هستم که مرا رهنما شدند، از آن جمله استاد راهنمای گرانقدر جناب آقای دکتر علیرضا ناظمی که با کمک‌های بی دریغ، ارزنده و عالمانه خود همواره راهگشای اینجانب در انجام و اتمام پایان نامه بوده‌اند. از زحمات استاد جناب آقای دکتر حجت احسنی طهرانی و دکتر حیدر طوسی‌شاندیز که زحمت داوری این پایان نامه را به عهده داشته‌اند نیز کمال تشکر را دارم. همچنین از پدر و مادر عزیز، خواهر و برادرانم که آرامش روحی و آسایش فکری فراهم نمودند تا با حمایت‌های همه جانبه در محیطی مطلوب، مراتب تحصیلی و نیز پایان نامه درسی را به نحو احسن به اتمام برسانم سپاسگزار می‌نمایم. باشد که این جملات نمایانگر سپاس بی‌پایان من نسبت به محبت بی‌دریغ آنان به شمار آید. و در پایان با تشکر فراوان از همهی دوستان و عزیزانی که در این مدت وجودشان سرمایه‌ی امیدم بود.

از جمله خانم‌ها: سیمین پویا، الهام آشوری، عطیه سابقی و مرضیه مرتضایی
باشد تا روزی بیشتر از اینها بدانیم، بیش از اینها بنویسیم و چیزهایی بخوانیم و بنویسیم که پس از خواندن و نوشتن آن‌ها این حس در ما بیدار شود که انسان‌تر شده‌ایم.

سمیرا سوخت‌سهرابی
۱۳۹۳

تعمدنامه

اینجانب سمیرا سوخت‌سراییی دانشجوی کارشناسی ارشد رشته ریاضی کاربردی دانشکده علوم ریاضی دانشگاه شاهرود، نویسنده پایان‌نامه با عنوان یک شبکه عصبی با کارایی بالا برای حل رده ای از مسائل برنامه ریزی غیر خطی با محدودیت های هیبریدی، تحت راهنمایی دکتر علیرضا ناظمی متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تاکنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه شاهرود متعلق دارد، و مقالات مستخرج با نام “ دانشگاه شاهرود “ یا “ Shahrood University “ به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده) شده است، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

سمیرا سوخت‌سراییی
۱۳۹۳

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

چکیده

با پیشرفت فن‌آوری اطلاعات و ارتباطات و توسعه ارتباط درون سازمانی و بین سازمانی نیاز به استفاده از مدل‌های بهینه‌سازی را برای استفاده منطقی از داده‌ها و اطلاعات فراهم شده گسترش داده است. این مطلب متضمن بزرگ شدن اندازه مسائل بهینه‌سازی که در عمل وجود دارند خواهد بود. در این شرایط لزوم به کارگیری روش‌های کارآمدی که بتوانند با سرعت بالا مسائل بسیار بزرگ را با کیفیت قابل قبول حل کنند بیش از پیش احساس می‌شود.

در چند دهه اخیر روش‌های بهینه‌سازی که بر پایه رویکرد هوش مصنوعی توسعه یافته‌اند، موفقیت‌های چشمگیری در حل مؤثر و کارایی مسائل بهینه‌سازی به دست آورده‌اند.

روش‌هایی چون الگوریتم ژنتیک، جستجوی ممنوع، شبیه‌سازی تبریدی، شبکه عصبی و... قابلیت‌های خود را در حل مسائل بزرگ عملی به خوبی نشان داده‌اند.

امتیازات ویژه‌ی موجود در شبکه‌های عصبی امکان کاربرد آنها را در حوزه وسیعی از تحقیقات فراهم ساخته است. از جمله آن امتیازات می‌توان به امکان یادگیری و بهبود عملکرد براساس داده‌های ورودی اشاره کرد. همچنین امکان انجام محاسبات به صورت موازی در شبکه‌های عصبی امتیاز دیگری است که با توجه به گسترش سخت‌افزارهای موازی، امکان حل مسائل بسیار بزرگ را توسط این رویکرد ممکن می‌سازد.

در این پایان‌نامه یک مدل شبکه عصبی بازگشتی برای حل رده‌ای از مسائل بهینه‌سازی ارائه می‌شود. تحلیل وجود یکتایی، پایداری و همگرایی سراسری جواب‌ها مورد بررسی قرار می‌گیرند و عملکرد روش‌های ارائه شده با به کارگیری چند مثال از مسائل برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی، کنترل بهینه سیستم‌های خطی زمان گسسته نشان داده می‌شود.

در انتها نتایج کار و پیشنهاداتی برای کارهای آتی ارائه می‌دهیم.

لیست مقالات مستخرج از پایان نامه

۱. مقاله اول

۲. مقاله دوم

۳. مقاله سوم

فهرست مطالب

۱	مقدمات و مفاهیم اساسی	۱
۱	۱.۱ مقدمه‌ای بر شبکه‌های عصبی	۱
۱	۱.۱.۱ شبکه‌های عصبی بیولوژیکی	۱
۲	۲.۱.۱ شبکه‌های عصبی مصنوعی	۲
۶	۳.۱.۱ تاریخچه تکامل شبکه عصبی	۶
۷	۴.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی	۷
۹	۲.۱ مقدمه‌ای بر بهینه‌سازی غیرخطی	۹
۱۱	۳.۱ سیستم‌های دینامیکی	۱۱
۱۷	۲ یک مدل شبکه عصبی برای مسائل برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی	۱۷
۱۷	۱.۲ مقدمه	۱۷
۱۸	۲.۲ بیان مسأله	۱۸
۱۹	۳.۲ مدل شبکه عصبی	۱۹
۲۰	۴.۲ خواص همگرایی و پایداری	۲۰
۲۴	۵.۲ شبیه‌سازی‌های عددی	۲۴
۳۳	۳ کنترل بهینه سیستم‌های هیبریدی خطی زمان گسسته با استفاده از یک مدل شبکه عصبی	۳۳
۳۳	۱.۳ مقدمه	۳۳
۳۴	۲.۳ بیان مسأله و ارائه یک مدل شبکه عصبی	۳۴
۳۷	۳.۳ شبیه‌سازی‌های عددی	۳۷
۵۲	آ کد مثال‌های فصل ۲	۵۲
۵۲	۱.آ مثال (۱.۵.۲)	۵۲
۵۳	۲.آ مثال (۲.۵.۲)	۵۳
۵۳	۳.آ مثال (۳.۵.۲)	۵۳
۵۴	۴.آ مثال (۴.۵.۲)	۵۴
۵۵	۵.آ مثال (۵.۵.۲)	۵۵

۵۶	مثال (۶.۵.۲)	۶.آ
۵۷	مثال (۷.۵.۲)	۷.آ
۵۷	مثال (۸.۵.۲)	۸.آ
۵۹		ب کد مثال‌های فصل ۳	
۵۹	مثال (۱.۳.۳)	۱.ب
۶۱	مثال (۲.۳.۳)	۲.ب
۶۲	مثال (۳.۳.۳)	۳.ب
۶۵	مثال (۴.۳.۳)	۴.ب
۶۶	مثال (۵.۳.۳)	۵.ب
۶۹		مراجع	
۷۵		واژه‌نامه فارسی به انگلیسی	
۷۷		واژه‌نامه انگلیسی به فارسی	

فصل ۱

مقدمات و مفاهیم اساسی

۱.۱ مقدمه‌ای بر شبکه‌های عصبی

۱.۱.۱ شبکه‌های عصبی بیولوژیکی

واحد اصلی دستگاه عصبی، سلولی خاص به نام نرون^۱ می‌باشد و بدون شک نحوه کار مغز و راز شعور آدمی در آن نهفته است. نرون‌ها با وجود تفاوت‌های زیاد از نظر اندازه و شکل ظاهری، مشخصه‌های مشترکی دارند. همان‌طور که در شکل ۱.۱ نشان داده شده است از تنه سلولی نرون تعدادی شاخک کوتاه خارج می‌شود که دندریت^۲ نام دارند. دندریت‌ها و تنه سلولی، سیگنال‌ها را از نرون‌های مجاور دریافت می‌کنند و از طریق یک لوله باریک به نام آکسون^۳ به نرون‌های دیگر منتقل می‌گردد.

آکسون در انتهای خود به تعدادی رشته جانبی باریک تقسیم می‌گردد که پایانه‌های آکسونی نام دارد و با دندریت‌های سایر نرون‌ها مرتبط است. ارتباط بین آکسون یک نرون و دندریت نرون دیگر را سیناپس^۴ می‌نامند. یک سیناپس مرکب از پایانه قبل از سیناپسی^۵، شکاف سیناپسی^۶ و پایانه بعد از سیناپسی^۷ می‌باشد. همه سیگنال‌های جمع‌بندی شده، در تنه نرون ترکیب می‌شوند و اگر وسعت سیگنال‌های ترکیب شده به آستانه نرون برسد، مرحله تحریک شدن^۸ فعال می‌شود و یک سیگنال خروجی تولید می‌شود. این سیگنال به صورت یک پالس منفرد یا بخشی از پالس‌ها در یک میزان خاص به موازات آکسون به پایانه‌های سیناپسی انتقال می‌یابد و موجب ترشح موادی به نام عصب-رسانه می‌شود. عصب-رسانه در داخل شکاف سیناپسی پخش می‌شود و نرون بعدی را تحریک می‌کند و به این ترتیب یک سیگنال از

^۱Neuron

^۲Dendrite

^۳Axon

^۴Synapse

^۵Presynaptic Terminal

^۶Synaptic Cleft

^۷Postsynaptic Terminal

^۸Firing

یک نرون به دیگری انتقال می‌یابد. تعداد بسیار زیادی آکسون از نرون‌های مختلف با دندریت‌های یک نرون به این صورت ارتباط برقرار می‌کنند.

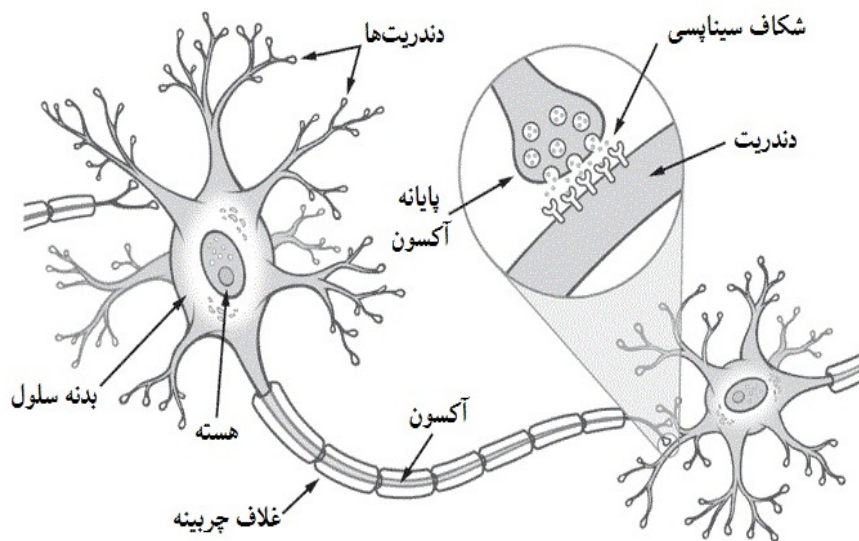
نرون‌ها با توجه به عملکردشان به سه دسته تقسیم می‌شوند:

(۱) **نرون‌های حسی:** این نرون‌ها پیام‌ها را از گیرنده‌ها به دستگاه عصبی مرکزی منتقل می‌کنند. گیرنده سلول خاصی است که در اندام‌های حسی، عضلات، پوست و ... قرار دارد و تغییرات فیزیکی یا شیمیایی را در می‌یابد، این تغییرات را به سیگنال‌های عصبی تبدیل می‌کند.

(۲) **نرون‌های حرکتی:** این نرون‌ها پیام‌ها را از مغز یا طناب نخاعی به اندام‌های عمل‌کننده می‌رسانند.

(۳) **نرون‌های رابط:** این نرون‌ها پیام‌ها را از نرون‌های حسی می‌گیرند و به یک نرون رابط دیگر انتقال می‌دهند و یا پیام‌ها را به یک نرون حرکتی می‌رسانند. نرون‌های رابط فقط در مغز، چشم و طناب نخاعی یافت می‌شوند.

حال منظور از یادگیری عبارت است از تغییری نسبتاً ساختاری در سیناپس (بعد از تحریک)، که این تغییر ساختاری باعث افزایش کارایی سیناپس می‌شود.



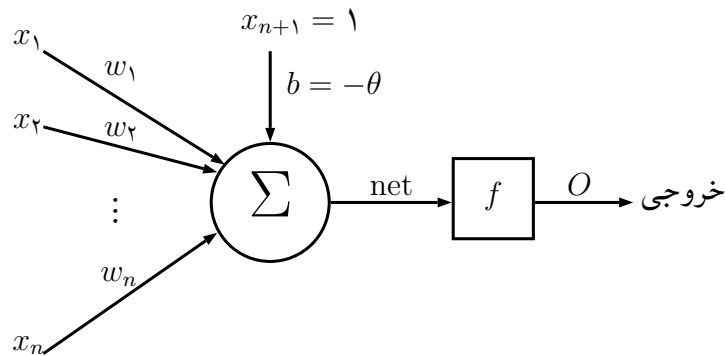
شکل ۱.۱: ساختار یک نرون.

۲.۱.۱ شبکه‌های عصبی مصنوعی

بعد از اشاره‌ای گذرا بر شبکه عصبی بیولوژیکی و نشان دادن گوشه‌ای بسیار کوچک از این جهان مرموز و فوق‌العاده پیچیده به موضوع اصلی یعنی شبکه‌های عصبی مصنوعی می‌پردازیم. شبکه‌های عصبی مصنوعی^۹ مدل‌های سخت افزاری یا نرم افزاری الهام گرفته از ساختار و رفتار نرون‌های بیولوژیکی و

^۹Artificial Neural Networks

سیستم‌های عصبی انسان هستند که شامل عناصر پردازشی (به نام نرون‌ها) و ارتباطات میان آنها با ضرایب (وزن‌ها) اختصاص یافته به ارتباطات است، که ساختار شبکه عصبی را تشکیل می‌دهند. در این پایان‌نامه ما شبکه‌های عصبی مصنوعی را یک مدل ریاضی فرض خواهیم کرد و به‌جای استفاده از عبارت شبکه‌های عصبی مصنوعی به‌طور خلاصه عبارت شبکه‌های عصبی را به‌کار خواهیم برد.



شکل ۲.۱: مدل ریاضی یک نرون.

مدل شبیه‌سازی شده یک نرون، در شکل ۲.۱ نمایش داده شده است. یک نرون در حالت کلی یک مجموعه از $n + 1$ ورودی x_j ($j = 1, \dots, n + 1$) دارد. این ورودی‌ها همان سیگنال‌های عصبی می‌باشند که از محیط یا از یک نرون دیگر فرستاده می‌شوند. w_j ($j = 1, \dots, n + 1$) که روی آکسون‌ها واقع هستند نقش ساختار سیناپسی را ایفا می‌کنند. این‌ها هستند که باید تغییر کنند تا فرآیند یادگیری اتفاق افتد حال این‌که این تغییر بر چه قاعده‌ای استوار باشد بستگی به الگوریتم آموزش یا تعلیم دارد. وزن آخرین ورودی یعنی θ را با b نشان می‌دهند و آن را بایاس^{۱۰} می‌نامند. سیگنال‌های عصبی x_1 و x_2 و ... و x_{n+1} به‌ترتیب در w_1 و w_2 و ... و w_{n+1} ضرب شده و مجموع این حاصل‌ضرب‌ها که در شکل آن را با net نشان می‌دهیم، وارد نرون می‌شوند. در این مرحله نرون همانند یک تابع روی net عمل می‌کند و حاصل کار خود (برد تابع) را از طریق آکسون‌های طرف راست به نرون‌های دیگر می‌فرستد. این تابع را تابع فعال‌سازی^{۱۱} می‌نامند.

البته خاطر نشان می‌کنیم که اگر نرون به‌عنوان تابع فرض شود این تابع از \mathbb{R}^n به \mathbb{R} تعریف می‌گردد. یعنی سیگنال خروجی از نرون در تمام آکسون‌های طرف راست یکسان می‌باشد که روی شکل با O نمایش داده‌ایم. همان‌طور که قبلاً اشاره شد نرون‌ها می‌توانند حسی، ارتباطی و یا حرکتی باشند. اطلاعات از طریق نرون‌های حسی دریافت می‌شود و از طریق نرون‌های ارتباطی انتقال می‌یابد و بالاخره نتیجه کار شبکه عصبی و یا واکنش شبکه عصبی به اطلاعات دریافت شده از طریق نرون‌های حرکتی به خارج منتقل می‌گردد (شکل ۳.۱).

نرون‌های حسی را لایه ورودی، نرون‌های حرکتی را لایه خروجی و نرون‌های ارتباطی را لایه پنهان نیز می‌گویند. لایه مخفی می‌تواند از چندین لایه تشکیل گردد البته در شکل ۳.۱ لایه مخفی را تک لایه

^{۱۰} Bias

^{۱۱} Activation Function

در نظر گرفته شده است.

مشخصه‌های اصلی یک شبکه عصبی که در طراحی آن باید مدنظر قرار گیرند عبارت‌اند از:

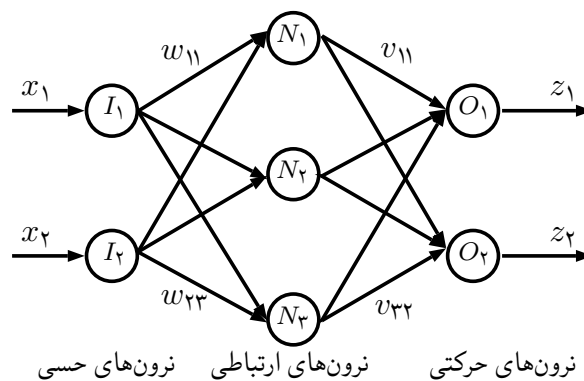
معماری شبکه عصبی: نحوه اتصالات بین نرون‌ها، تعداد آن‌ها و تعداد لایه‌های تشکیل دهنده بخش نرون‌های ارتباطی را معماری شبکه عصبی می‌گوییم.

تابع‌های فعال‌سازی: این‌که چه تابعی روی نرون قرار بگیرد تا بر ورودی‌های نرون اثر کرده و خروجی نرون را تولید کند تابع فعال‌سازی آن نرون گویند. در جدول ۱.۱ تعدادی از مهم‌ترین توابع فعال‌سازی نشان داده شده است.

الگوریتم آموزش: روشی که وزن‌های روی آکسون‌ها بر اساس آن تغییر می‌کنند (w_{ij} ها و v_{jk} ها در شکل ۳.۱) تا خروجی شبکه عصبی به حالت مطلوب درآید را الگوریتم آموزش می‌گوییم.

با توجه به مطالب قبل می‌توان برای شبکه‌های عصبی مصنوعی خصوصیات عمومی زیر را نسبت داد:

۱. پردازش اطلاعات در عناصر بسیار ساده به نام نرون (سلول عصبی) انجام می‌گیرد.
۲. سیگنال‌های عصبی بین سلول‌های عصبی از طریق اتصالات بین آن‌ها مبادله می‌شوند.
۳. به هر خط اتصال (آکسون) یک وزن نسبت داده می‌شود که در شبکه‌های عصبی معمولی این وزن‌ها در سیگنال‌های منتقل شده ضرب می‌گردند.
۴. هر نرون دارای یک تابع فعال‌سازی می‌باشد که این تابع در اکثر موارد غیرخطی است.



شکل ۳.۱: ساختار یک نرون با لایه ورودی، لایه پنهان و لایه خروجی.

شبکه‌های عصبی بازگشتی

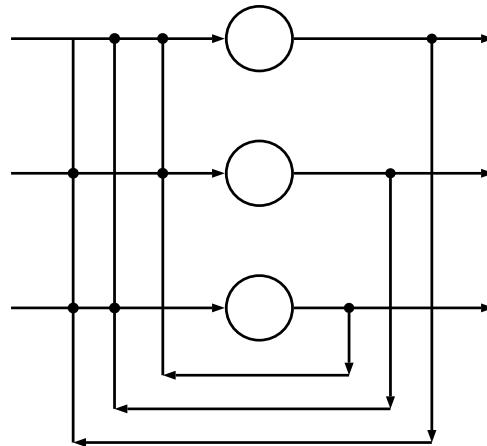
در شبکه‌های عصبی بازگشتی^{۱۲}، حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرونهای همان لایه و یا لایه قبل وجود دارد. چنین شبکه‌ای حافظه‌ای را نگه می‌دارد و حالت بعد نه تنها به

^{۱۲}Feedback (Recurrent) Neural Network

جدول ۱.۱: توابع فعال‌سازی

شکل تابع	تعریف تابع	نام	ردیف
	$f(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$	آستانه‌ای دو مقداره	۱
	$f(x) = \begin{cases} -1 & x < 0, \\ 1 & x \geq 0. \end{cases}$	آستانه‌ای دو مقداره متقارن	۲
	$f(x) = x$	خطی	۳
	$f(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$	آستانه‌ای خطی متقارن	۴
	$f(x) = \begin{cases} 0, & x < 0, \\ x & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$	آستانه‌ای خطی	۵
	$f(x) = \frac{1}{1+e^{-x}}$	سیگموئید	۶

سیگنالهای ورودی بلکه به حالات قبل شبکه نیز وابسته است. این شبکه‌های عصبی معمولاً در مدل ریاضی به صورت معادلات دیفرانسیل مرتبه اول ظاهر می‌شوند.



شکل ۴.۱: شبکه‌های عصبی دو لایه بازگشتی.

۳.۱.۱ تاریخچه تکامل شبکه عصبی

در سال ۱۹۴۳ یک نرولوژیست به نام وارن مک کلاچ^{۱۳} به همراه یک متخصص آمار به نام والتر پیتز^{۱۴} اولین مدل ریاضی نرون را ارائه دادند [۶۵]. این نرون ساده یک عنصر محاسبه‌گر بود که عناصری به نام ورودی را در مقادیر ثابتی به نام وزن ضرب می‌کرد و آنها را از یک عملگر خطی عبور می‌داد و حاصل خروجی نرون را تشکیل می‌داد. در این مدل وزن‌ها مقادیر ثابتی بودند. در سال ۱۹۴۹ دونالد هب^{۱۵} [۲۲] کتابی درباره نحوه یادگیری در مغز انسان نوشت. بر این مبنا قانون یادگیری برای نرون را ارائه داد که به قانون هب معروف است و بعدها در تعلیم شبکه‌های عصبی از آن استفاده شد. در سال ۱۹۵۴ فارلی و کلارک^{۱۶} [۱] مدلی از نرون‌ها را به‌طور تصادفی به هم وصل کردند و قانون هب را پیاده‌سازی نمودند و نشان دادند که می‌توان دو الگوی ورودی را با این قانون تعلیم از هم تشخیص داد. با پیشرفت علم کامپیوتر اولین مدل از نرون عصبی مصنوعی توسط ناتانیال روچستر^{۱۷} محقق شرکت IBM در سال ۱۹۵۸ با استفاده از کامپیوتر شبیه‌سازی شد. روزنبلات^{۱۸} [۳۱، ۳۰] در سال ۱۹۵۸ نرون ساده مک‌کلاچ پیتز را اصلاح کرد و به آن قابلیت یادگیری و سازگاری را اضافه کرد و این نرون را پرسپترون نامید. قانون تعلیم پرسپترون^{۱۹} اولین قانون رسمی برای شبکه‌های عصبی است.

برنارد ویدرو^{۲۰} و مارسین هاف^{۲۱} و همکارانشان یک مدل سه سطحی از نرون‌ها را ابداع کردند که

^{۱۳}Warren McCulloch

^{۱۴}Walter Pitts

^{۱۵}Donald Hebb

^{۱۶}Farley and Clark

^{۱۷}Nathanial Rochester

^{۱۸}Frank Rosenblatt

^{۱۹}Perceptron

^{۲۰}Bernard Widrow

^{۲۱}Marcian Hoff

آدلاین^{۲۲} نام‌گذاری شد و مختصر شده حروف ” عنصر خطی تطبیقی^{۲۳}” بود [۱۲، ۱۳، ۱۴، ۱۵]. و برای آدلاین یک قانون تعلیم بر مبنای مشتق‌گیری بیان نمودند. این قانون تعلیم به $\alpha - LMS$ معروف است. از متصل کردن چند آدلاین به یکدیگر مادالاین^{۲۴} ساخته شد. ویدرو و هاف از شبکه عصبی مادالاین برای حذف اکو از خطوط تلفن استفاده نمودند.

سال ۱۹۶۹ آغاز یک دوره رکود برای شبکه عصبی بود. پاپرت و مینسکی^{۲۵} [۴۶] کتابی به نام پرسپترون نوشتند و نشان دادند که اگرچه گیت‌های AND و OR را می‌توان با استفاده از پرسپترون پیاده‌سازی کرد ولی گیت XOR قابل پیاده‌سازی نمی‌باشد و لذا نمی‌توان از شبکه عصبی برای پردازش اطلاعات استفاده کرد. الگوریتم پس انتشار^{۲۶} پال ورباس^{۲۷} [۵۶] در سال ۱۹۷۴ ارائه شد و بعدها توسط رامل هارت^{۲۸} [۲۴] به‌طور مستقل کشف گردید. این الگوریتم از زمان پیدایش به‌طور گسترده به‌عنوان یک الگوریتم آموزش در شبکه‌های عصبی پیش‌خورد^{۲۹} مورد استفاده قرار گرفته است. سال ۱۹۸۲ تا ۱۹۸۶ را می‌توان تولد دوباره شبکه عصبی دانست. در این سال‌ها دو اتفاق مهم در این زمینه رخ داد. اول ارائه شبکه عصبی بازگشتی و مفهوم تابع انرژی توسط جان هاپفیلد^{۳۰} [۳۴] بود که در سال ۱۹۸۲ منتشر شد. دومین اتفاق، حل مسأله فروشنده دوره‌گرد (TSP) توسط شبکه هاپفیلد بود [۳۵]. هاپفیلد نشان داد که از این شبکه بازگشتی می‌توان برای حل مسائل بهینه‌سازی استفاده کرد و افق جدیدی را در شبکه‌های عصبی مصنوعی گشود.

۴.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی

طیف گسترده‌ای از مسائل مهم در رشته‌های علوم پایه و مهندسی از جمله کنترل بهینه، طراحی ساختمان، پردازش تصویر، تقریب توابع، تحلیل رگرسیون و... قابل تبدیل به مسائل بهینه‌سازی خطی و غیرخطی هستند. در دهه‌های گذشته الگوریتم‌های بهینه‌سازی عددی فراوانی برای حل مسائل بهینه‌سازی خطی و غیرخطی ارائه شده‌اند. از آن‌جا که زمان مورد نیاز برای حل مسائل بهینه‌سازی و به‌خصوص مسائل بهینه‌سازی غیرخطی بسیار وابسته به بعد و ساختار مسأله است بنابراین الگوریتم‌های عددی کارایی کمی از خود نشان می‌دهند. یک رهیافت امیدوارکننده برای حل مسائل بهینه‌سازی با ابعاد بالا استفاده از شبکه‌های عصبی مصنوعی می‌باشد.

در سال ۱۹۸۶ تانک و هاپفیلد [۱۹] اولین شبکه عصبی را برای حل مسائل برنامه‌ریزی خطی معرفی کردند و نشان دادند که وضعیت شبکه در هر تکرار به‌گونه‌ای تغییر می‌کند که تابع انرژی متناظر

^{۲۲} ADELIN

^{۲۳} Adaptive Linear Elements

^{۲۴} MADALINE

^{۲۵} Papert and Minsky

^{۲۶} Back Propagation

^{۲۷} Paul Werbos

^{۲۸} Rumelhart

^{۲۹} Feed Forward

^{۳۰} John Hopfield

با آن به‌طور یکنواخت کاهش می‌یابد تا جایی که به نقطه مینیمم خود می‌رسد و این نقطه مینیمم متناظر با نقطه تعادل شبکه عصبی می‌باشد. آن‌ها این شبکه عصبی را توسط یک مدار الکتریکی پیاده‌سازی کردند. هم‌چنین از این شبکه برای حل مسأله فروشنده دوره‌گرد با ۳۰ شهر استفاده نمودند. این شبکه دارای نقص‌هایی بود به‌خصوص این‌که نقطه تعادل شبکه در شرایط KKT^{۳۱} صدق نمی‌کرد و لذا جواب مطلوبی از مسأله حاصل نمی‌شد. با این وجود کارهای هاپفیلد انگیزه بسیار خوبی را برای محققین به‌وجود آورد تا در این زمینه فعالیت کنند. کندی و چا^{۳۲} [۴۳] با افزودن یک پارامتر جریمه متناهی به شبکه هاپفیلد آن را توسعه دادند و از آن برای حل مسأله برنامه‌ریزی غیرخطی محدب استفاده نمودند. با پارامترهای جریمه این شبکه در یافتن نقطه بهینه دقیق ناتوان است به‌خصوص اگر پارامتر جریمه بزرگ باشد این شبکه به‌سختی عمل می‌کند.

برای جلوگیری از بکار بردن پارامتر جریمه یک شبکه عصبی سوئیچ-خازن^{۳۳} توسط رودریگز-وازکز^{۳۴} و همکاران [۱۰] معرفی شد. هم‌چنین ما و شانبلات^{۳۵} [۱۷] یک شبکه عصبی دو فازی ارائه کردند که فاز اول شبکه مشابه شبکه کندی و چا بود و در فاز دوم آن مسیر شبکه به جواب دقیق مسأله همگرا می‌شد بنابراین این روش جواب‌های دقیق‌تری نسبت به شبکه کندی و چا ارائه می‌داد. مشکل این شبکه در این بود که پایداری فاز دوم شبکه بستگی به انتخاب یک مقدار بزرگ از پارامتر جریمه داشت و لذا اگرچه تأثیر پارامتر جریمه در این روش کاهش یافته بود و جواب‌های حاصل دقیق‌تر بودند اما هنوز مستقل از پارامتر نبود.

ژانگ و کنستانتینیدس^{۳۶} [۶۴] برمبنای روش لاگرانژ یک شبکه عصبی ارائه کردند که کاملاً مستقل از پارامتر جریمه و قادر به حل مسائل غیرخطی بود. نقطه تعادل این شبکه عصبی در شرایط بهینگی مرتبه اول و دوم صدق می‌کرد و هم‌چنین شبکه حاصل همگرا بود. در سال ۱۹۹۳ بوزردوم و پتیسن^{۳۷} [۷] شبکه‌ای را برمبنای گرادیان و روش تصویر و مستقل از پارامتر جریمه ابداع کردند که تنها قادر به حل مسائل درجه دوم با متغیرهای کران‌دار بود. این روش در عمل روش کارایی بود اما نمی‌توانست مسائل کلی برنامه‌ریزی خطی و درجه دوم را حل کند.

زیا^{۳۸} و همکاران [۶۷، ۷۳، ۶۸، ۶۹] چندین مدل را برای حل مسائل بهینه‌سازی غیرخطی با قیود خطی و غیرخطی ارائه کردند. هم‌چنین چندین مدل شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی توسط عفتی و همکاران [۶۱، ۶۳، ۶۲، ۳، ۴۷، ۴۸] ارائه شده است.

^{۳۱}Karush-Kuhn-Tucker

^{۳۲}Kennedy and Chua

^{۳۳}Switched-Capacitor

^{۳۴}Rodriguez-Vazquez

^{۳۵}Maa and Shanblatt

^{۳۶}Zhang and Constantinides

^{۳۷}Bouzerdoum and Pattison

^{۳۸}xia

۲.۱ مقدمه‌ای بر بهینه‌سازی غیرخطی

تعریف ۱.۲.۱. تابع $f : \mathbb{R}^n \rightarrow \mathbb{R}$ روی مجموعه محدب و ناتهی $S \subseteq \mathbb{R}^n$ محدب نامیده می‌شود اگر برای هر $x, y \in S$ و $\alpha \in (0, 1)$ داشته باشیم

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

تابع f روی S اکیداً محدب نامیده می‌شود اگر نامساوی بالا زمانی که $x \neq y$ است، به‌طور اکید برقرار باشد.

یک مسأله برنامه‌ریزی غیرخطی در حالت کلی به‌صورت زیر در نظر گرفته می‌شود:

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0, \\ & h_j(x) = 0, \\ & x \in X \subset \mathbb{R}^n, \end{aligned} \quad (1.1)$$

که $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ، $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($i = 1, \dots, m$) و $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ ($j = 1, \dots, p$) توابع دو بار مشتق پذیر پیوسته هستند. برای تسهیل در نمادگذاری، توابع برداری $g = (g_1, g_2, \dots, g_m)$ و $h = (h_1, h_2, \dots, h_p)$ را معرفی می‌کنیم و (۱.۱) را به‌صورت زیر می‌نویسیم:

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & g(x) \leq 0, \\ & h(x) = 0, \\ & x \in X \subset \mathbb{R}^n, \end{aligned} \quad (2.1)$$

که محدودیت‌های $g(x) \leq 0$ و $h(x) = 0$ ، محدودیت‌های تابعی و محدودیت $x \in X \subset \mathbb{R}^n$ ، محدودیت مجموعه‌ای نامیده می‌شود. هرگاه توابع f و g روی X محدب باشند و h آفین باشد یعنی $h = Ax - b$ ، $A \in \mathbb{R}^{p \times n}$ ، $rank(A) = p$ ($0 \leq p < n$) و $b \in \mathbb{R}^p$ باشد، آن‌گاه (۲.۱) یک مسأله بهینه‌سازی محدب^{۳۹} و در غیر این صورت یک مسأله بهینه‌سازی نامحدب^{۴۰} نامیده می‌شود.

تعریف ۲.۲.۱. نقطه x جواب شدنی^{۴۱} مسأله (۲.۱) نامیده می‌شود هرگاه در محدودیت‌های مسأله (۲.۱) صدق کند. نقطه شدنی x یک نقطه منظم^{۴۲} نامیده می‌شود اگر بردارهای گرادیان $\nabla g_i(x)$ ، $i \in \{i | g_i(x) = 0\}$ و $\nabla h_j(x)$ ، $j = 1, \dots, p$ مستقل خطی باشند.

^{۳۹}Convex Optimization

^{۴۰}Nonconvex Optimization

^{۴۱}Feasible Solution

^{۴۲}Regular Point

قضیه ۳.۲.۱ (شرایط کافی مرتبه دوم^{۴۳} [۲۳]). فرض کنید x^* یک نقطه شدنی برای مسأله (۲.۱) باشد. اگر $\mu \in \mathbb{R}^m$ و $\lambda \in \mathbb{R}^p$ وجود داشته باشند به طوری که

$$\mu \geq 0, \quad (3.1)$$

$$\mu^T g(x^*) = 0, \quad (4.1)$$

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0, \quad (5.1)$$

و ماتریس هسین^{۴۴}

$$\nabla^2 L(x^*, \lambda, \mu) = \nabla^2 f(x^*) + \sum_{i=1}^m \mu_i \nabla^2 g_i(x^*) + \sum_{j=1}^p \lambda_j \nabla^2 h_j(x^*) \quad (6.1)$$

بر زیرفضای

$$M(x^*) = \{d \mid \nabla h(x^*)d = 0, \nabla g_j(x^*)d = 0, \forall j \in J(x^*)\},$$

که در آن

$$J(x^*) = \{j \mid g_j(x^*) = 0, \mu_j > 0\}$$

معین مثبت باشد آن گاه x^* یک نقطه مینیمم محلی اکید^{۴۵} برای مسأله (۲.۱) است.

مسأله دوگان لاگرانژ

مسأله برنامه‌ریزی غیرخطی (۱.۱) را در نظر بگیرید، این مسأله را مسأله اولیه (P) نیز می‌نامیم. مسأله دوگان لاگرانژ^{۴۶} (D) به صورت زیر تعریف می‌شود:

$$\begin{aligned} \max \quad & \theta(u, v) \\ \text{s.t.} \quad & u \geq 0, \end{aligned} \quad (7.1)$$

که در آن

$$\begin{aligned} \theta(u, v) = \inf_x \left\{ f(x) + \sum_{i=1}^m u_i g_i(x) + \sum_{j=1}^p v_j h_j(x), x \in X \right\}, \quad (8.1) \\ u = (u_1, u_2, \dots, u_m)^t, \quad v = (v_1, v_2, \dots, v_p)^t, \end{aligned}$$

u_i -امین مؤلفه از u ، متغیر دوگان^{۴۷} یا ضربگر لاگرانژ^{۴۸} مربوط به قید $g_i(x) \leq 0$ می‌باشد و v_j -امین مؤلفه v ، متغیر دوگان یا ضربگر لاگرانژ مربوط به قید $h_j(x) = 0$ می‌باشد.

^{۴۳}Second-Order Sufficient Conditions

^{۴۴}Hessian Matrix

^{۴۵}Strict Local Minimum Point

^{۴۶}Lagrangian Dual Problem

^{۴۷}Dual Variable

^{۴۸}Lagrangian Multiplier

مسئله دوگان لاگرانژ در شکل برداری به صورت زیر نوشته می‌شود:

$$\begin{aligned} \max \theta(u, v) \\ \text{s.t. } u \geq 0, \end{aligned} \quad (9.1)$$

که در آن

$$\theta(u, v) = \inf_x \{ f(x) + u^t g(x) + v^t h(x), x \in X \},$$

هم چنین $h = (h_1, \dots, h_p) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ و $g = (g_1, \dots, g_m) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ، $f : \mathbb{R}^n \rightarrow \mathbb{R}$ هستند.

قضیه ۴.۲.۱ (دوگانی موضعی^{۴۹} [۲۳]). فرض کنید مسئله

$$\begin{aligned} \min f(x) \\ \text{s.t. } h(x) = 0, \\ x \in X \subset \mathbb{R}^n, \end{aligned} \quad (10.1)$$

که $f, h \in C^1$ و $h(x) \in \mathbb{R}^m$ دارای یک جواب موضعی در x^* با مقدار متناظر r^* و ضریب لاگرانژ v^* است. هم چنین فرض کنید که x^* یک نقطه منظم برای قیود است و هسین لاگرانژ متناظر $\nabla^2 L(x^*, v^*)$ معین مثبت است. در این صورت مسئله دوگان

$$\max \theta(v) \quad (11.1)$$

که در آن

$$\theta(v) = \inf_x \{ f(x) + v^t h(x), x \in X \},$$

دارای یک جواب موضعی در v^* با مقدار متناظر r^* است و x^* جواب متناظر با v^* در تعریف θ است. به سادگی می‌توان نتایج فوق را به مسائلی که علاوه بر قیود تساوی قیود نامساوی هم دارند تعمیم داد [۲۳].

۳.۱ سیستم‌های دینامیکی

تعریف ۱.۳.۱. سیستم دینامیکی زیر را در نظر بگیرید:

$$\dot{x} = f(x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^n, \quad (12.1)$$

که در آن f یک تابع از \mathbb{R}^n به \mathbb{R}^n می‌باشد. x^* یک نقطه تعادل^{۵۰} (۱۲.۱) نامیده می‌شود اگر $f(x^*) = 0$. اگر همسایگی $\Omega^* \subseteq \mathbb{R}^n$ از x^* وجود داشته باشد که $f(x^*) = 0$ و $f(x) \neq 0$ ، $\forall x \in \Omega^* \setminus \{x^*\}$ ، x^* یک نقطه تعادل تنها^{۵۱} نامیده می‌شود.

تعریف ۲.۳.۱ (پایداری به مفهوم لیاپانوف^{۵۲}). فرض می‌کنیم که $x(t)$ یک جواب (۱۲.۱) باشد، نقطه

^{۴۹}Local Duality

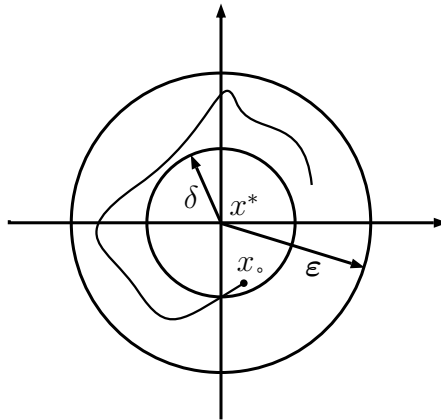
^{۵۰}Equilibrium Point

^{۵۱}Isolated Equilibrium Point

^{۵۲}Stability in the Sense of Lyapunov

تعادل تنها x^* ، پایدار به مفهوم لیاپانوف است اگر برای هر $x_0 = x(t_0)$ و هر $\varepsilon > 0$ ، یک $\delta > 0$ وجود داشته باشد به طوری که اگر $\|x(t_0) - x^*\| < \delta$ آن‌گاه

$$\|x(t) - x^*\| < \varepsilon, \forall t \geq t_0.$$



شکل ۵.۱: پایداری لیاپانوف.

تعریف ۳.۳.۱. سیستم دینامیکی (۱۲.۱) همگرای سراسری^{۵۳} به مجموعه جواب‌های (۱۲.۱) که با Ω^* نمایش داده می‌شود، گفته می‌شود اگر برای هر نقطه آغازین دلخواه، مسیر $x(t)$ در رابطه زیر صدق کند:

$$\lim_{t \rightarrow \infty} \text{dist}(x(t), \Omega^*) = 0,$$

که در آن

$$\text{dist}(x, \Omega^*) = \inf_{y \in \Omega^*} \|x - y\|,$$

و $\|\cdot\|$ نرم l_2 را نشان می‌دهد.

تعریف ۴.۳.۱. سیستم دینامیکی (۱۲.۱) در نقطه تعادل x^* که یکتا می‌باشد، پایدار مجانبی سراسری^{۵۴} نامیده می‌شود اگر x^* به مفهوم لیاپانوف پایدار باشد و در شرط زیر صدق کند:

$$\lim_{t \rightarrow \infty} x(t) = x^*.$$

تعریف ۵.۳.۱. سیستم دینامیکی (۱۲.۱) در نقطه x^* پایداری نمایی سراسری^{۵۵} با آهنگ β نامیده می‌شود اگر برای هر نقطه آغازین دلخواه، مسیر $x(t)$ در رابطه زیر صدق کند:

$$\|x(t) - x^*\| \leq \alpha \|x(t_0) - x^*\| e^{-\beta(t-t_0)}, \forall t \geq t_0,$$

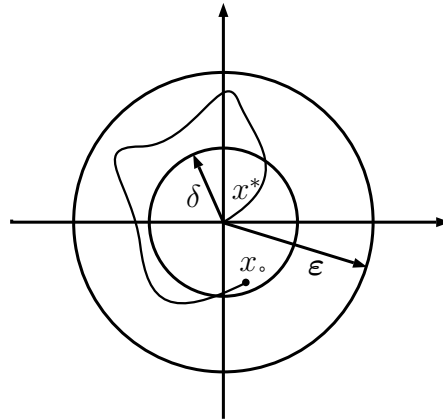
که در آن α و β ثابت‌های مثبت و مستقل از نقاط آغازین هستند.

واضح است که پایداری نمایی سراسری پایداری مجانبی سراسری را نتیجه می‌دهد.

^{۵۳}Globally Convergent

^{۵۴}Globally Asymptotically Stable

^{۵۵}Globally Exponential Stability



شکل ۶.۱: پایداری مجانبی.

تعریف ۶.۳.۱. مجموعه $M \subseteq \mathbb{R}^n$ یک مجموعه تغییر ناپذیر^{۵۶} نسبت به سیستم (۱۲.۱) گفته می‌شود اگر $x(t_0) \in M$ و $t_0 \geq 0$ به ازای هر $x(t) \in M, t \geq t_0$ باشد.

تعریف ۷.۳.۱. فرض می‌کنیم که $\Omega \subset \mathbb{R}^l$ یک مجموعه بسته و محدب باشد که ممکن است بیکران باشد. $P_\Omega(x): \mathbb{R}^l \rightarrow \Omega$ یک عملگر تصویر^{۵۷} نامیده می‌شود و به صورت زیر تعریف می‌شود:

$$P_\Omega(x) = \arg \min_{v \in \Omega} \|x - v\|, \quad (۱۳.۱)$$

که $\|\cdot\|$ نرم l_2 را نشان می‌دهد.

تعریف ۸.۳.۱. تابع $F: E \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ روی E در شرط لیب شیتز صدق می‌کند (پیوسته لیب شیتز^{۵۸} است) اگر ثابت L وجود داشته باشد به طوری که برای هر دو نقطه $x, y \in E$ داشته باشیم:

$$\|F(x) - F(y)\| \leq L \|x - y\|.$$

تابع F پیوسته لیب شیتز محلی^{۵۹} روی E نامیده می‌شود اگر برای نقطه $x_0 \in E$ یک ε -همسایگی مانند $N_\varepsilon(x_0) \subset E$ و یک ثابت L_0 وجود داشته باشد به طوری که برای هر دو نقطه $x, y \in N_\varepsilon(x_0)$ داشته باشیم:

$$\|F(x) - F(y)\| \leq L_0 \|x - y\|.$$

قضیه ۹.۳.۱. تابع $F: E \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ را در نظر بگیرید. اگر $F \in C^1(E)$ ، آنگاه F روی E لیب شیتز محلی است.

□

برهان. به [۴۱] مراجعه شود.

تعریف ۱۰.۳.۱. نگاشت $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$

^{۵۶}Invariant Set

^{۵۷}Projection Operator

^{۵۸}Lipschitz Continuous

^{۵۹}Locally Lipschitz Continuous

(آ) یکنوا نامیده می‌شود اگر برای هر دو نقطه $x, y \in \mathbb{R}^n$ داشته باشیم:

$$\langle x - y, F(x) - F(y) \rangle \geq 0.$$

(ب) اکیداً یکنوا نامیده می‌شود اگر برای هر دو نقطه متمایز $x, y \in \mathbb{R}^n$ داشته باشیم:

$$\langle x - y, F(x) - F(y) \rangle > 0.$$

(پ) قویاً یکنوا نامیده می‌شود اگر یک ثابت $\beta > 0$ وجود داشته باشد که برای هر دو نقطه $x, y \in \mathbb{R}^n$ داشته باشیم:

$$\langle x - y, F(x) - F(y) \rangle \geq \beta \|x - y\|^2.$$

که $\langle \cdot, \cdot \rangle$ ضرب داخلی را نشان می‌دهد.

تعریف ۱۱.۳.۱. اگر $g: \Omega \subset \mathbb{R}^l \rightarrow \mathbb{R}$ ، آن‌گاه هر مجموعه غیر تهی به شکل زیر

$$L(r) = \{u \in \Omega \mid g(u) \leq r, r \in \mathbb{R}\}$$

یک مجموعه سطح^{۶۰} از g نامیده می‌شود.

تعریف ۱۲.۳.۱. یک ماتریس $n \times n$ ، $M(x)$ که عناصر آن m_{ij} ، $i = 1, \dots, n$ و $j = 1, \dots, n$ توابعی هستند که روی مجموعه $S \subset \mathbb{R}^n$ تعریف شده‌اند، روی S نیمه معین مثبت^{۶۱} نامیده می‌شود اگر

$$v^T M(x) v \geq 0, \forall v \in \mathbb{R}^n, \forall x \in S.$$

ماتریس $M(x)$ روی S معین مثبت قوی^{۶۲} نامیده می‌شود اگر عددی مثبت مانند α وجود داشته باشد به قسمی که:

$$v^T M(x) v \geq \alpha \|v\|^2, \forall v \in \mathbb{R}^n, \forall x \in S.$$

اگر $\gamma(x)$ کوچکترین مقدار ویژه ماتریس $M(x)$ باشد سه حالت زیر را می‌توان نتیجه گرفت:

۱. $M(x)$ روی S نیمه معین مثبت است اگر و فقط اگر به ازای هر $x \in S$ ، $\gamma(x) \geq 0$.

۲. $M(x)$ روی S معین مثبت^{۶۳} است اگر و فقط اگر به ازای هر $x \in S$ ، $\gamma(x) > 0$.

۳. $M(x)$ روی S قویاً معین مثبت است اگر و فقط اگر به ازای هر $x \in S$ ، $\gamma(x) \geq \alpha > 0$.

تعریف ۱۳.۳.۱. فرض می‌کنیم $V: \mathbb{R}^n \rightarrow \mathbb{R}$ یک تابع به‌طور پیوسته مشتق پذیر باشد. $V(x)$ به‌طور شعاعی بی‌کران گفته می‌شود اگر

$$\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty.$$

^{۶۰} Level Set

^{۶۱} Positive Semidefinite

^{۶۲} Strongly Positive Definite

^{۶۳} Positive Definite

لم ۱۴.۳.۱ ([۳۹]). فرض کنیم که $g : \Omega_1 \subset \mathbb{R}^l \rightarrow \mathbb{R}$ باشد که در آن Ω_1 بی‌کران است. آن‌گاه همه مجموعه‌های سطح g کراندار می‌باشند اگر و فقط اگر

$$\lim_{k \rightarrow \infty} g(u^k) = +\infty,$$

که در آن $\{u^k\} \subset \Omega_1$ و $\lim_{k \rightarrow \infty} \|u^k\| = +\infty$.

لم ۱۵.۳.۱ ([۳۹]). فرض کنیم که $g : \Omega_1 \subset \mathbb{R}^l \rightarrow \mathbb{R}$ روی مجموعه بسته Ω_1 پیوسته باشد آن‌گاه همه مجموعه‌های سطح g کراندار می‌باشند اگر و فقط اگر مجموعه مینیمم‌کننده‌های g غیرتهی و کراندار باشند.

لم ۱۶.۳.۱ ([۳۹]). فرض کنیم که $U : D \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$ روی مجموعه محدب و باز D مشتق‌پذیر باشد تابع $g : \mathbb{R}^m \rightarrow \mathbb{R}^m$ را به صورت زیر تعریف می‌کنیم:

$$g(x) = \int_0^1 (x - x_0)^T U(x_0 + t(x - x_0)) dt$$

اگر ماتریس ژاکوبین تابع $U(x)$ برای هر $x \in D$ متقارن باشد آن‌گاه $\nabla g(x)^T = U(x)$.

قضیه ۱۷.۳.۱ (قضیه نقطه ثابت بروئر^{۶۴} [۳۹]). تابع پیوسته $f : \Omega \rightarrow \Omega$ ، که Ω یک مجموعه فشرده و محدب می‌باشد حداقل یک نقطه ثابت دارد.

قضیه ۱۸.۳.۱ ([۳۹]). فرض کنیم که تابع $F : K \subset \mathbb{R}^l \rightarrow \mathbb{R}^l$ روی K به طور پیوسته مشتق‌پذیر باشد، آن‌گاه F روی K یکنوا (اکیداً) است اگر و فقط اگر ماتریس ژاکوبین آن $\nabla F(x)$ برای هر $x \in K$ نیمه‌معین مثبت (معین مثبت) باشد.

قضیه ۱۹.۳.۱ ([۳۹]). فرض کنیم که تابع $F(x)$ روی K به طور پیوسته مشتق‌پذیر باشد و ژاکوبین $F(x)$ قویاً معین مثبت باشد آن‌گاه $F(x)$ قویاً یکنوا می‌باشد.

قضیه ۲۰.۳.۱ ([۴۱]). در سیستم دینامیکی (۱۲.۱) فرض کنیم که f یک تابع پیوسته از \mathbb{R}^n به \mathbb{R}^n باشد، آن‌گاه برای هر $t_0 \geq 0$ و $x_0 \in \mathbb{R}^n$ ، یک جواب محلی $x(t)$ به ازای $t \in [t_0, \tau)$ که $\tau > t_0$ وجود دارد. علاوه بر این اگر f در x_0 پیوسته لیپ شیتز باشد آن‌گاه جواب یکتاست و اگر f در \mathbb{R}^n پیوسته لیپ شیتز باشد آن‌گاه τ می‌تواند تا $+\infty$ توسعه داده شود.

تعریف ۲۱.۳.۱. اگر یک جواب محلی تعریف شده در بازه $[t_0, \tau)$ نتواند به یک جواب محلی روی یک بازه بزرگتر $[t_0, \tau_1)$ که $\tau_1 > \tau$ می‌باشد گسترش یابد، آن‌گاه یک جواب ماکسیمال نامیده می‌شود و بازه $[t_0, \tau)$ ، بازه ماکسیمال وجود جواب نامیده می‌شود. بازه ماکسیمال وجود جواب مربوط به x_0 اغلب به صورت $[t_0, \tau(x_0))$ تعریف می‌شود.

قضیه ۲۲.۳.۱ ([۴۰]). فرض کنیم $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ یک تابع پیوسته باشد. اگر $x(t)$ ، $t \in [t_0, \tau(x_0))$ یک جواب ماکسیمال باشد و $\tau(x_0) < +\infty$ آن‌گاه

$$\lim_{t \rightarrow \tau(x_0)} \|x(t)\| = +\infty.$$

^{۶۴}Brouwer's Fixed Point Theorem

قضیه ۲۳.۳.۱ (قضیه پایداری لیاپانوف [۳۳]). فرض کنید که $x = 0$ یک نقطه تعادل سیستم (۱۲.۱) باشد و $V: \mathbb{R}^n \rightarrow \mathbb{R}$ یک تابع به طور پیوسته مشتق پذیر باشد به طوری که:

$$V(0) = 0. \quad ۱.$$

$$V(x) > 0, \quad x \in \mathbb{R}^n \setminus \{0\} \quad \text{به ازای هر } ۲.$$

$$\dot{V}(x) \leq 0, \quad x \in \mathbb{R}^n \setminus \{0\} \quad \text{به ازای هر } ۳.$$

آنگاه $x = 0$ نقطه پایداری سیستم خواهد بود.

قضیه ۲۴.۳.۱ (قضیه پایداری مجانبی [۳۳]). تحت شرایط قضیه (۲۳.۳.۱) اگر $V(\cdot)$ در شرایط زیر صدق کند:

$$V(0) = 0. \quad ۱.$$

$$V(x) > 0, \quad x \in \mathbb{R}^n \setminus \{0\} \quad \text{به ازای هر } ۲.$$

$$\dot{V}(x) < 0, \quad x \in \mathbb{R}^n \setminus \{0\} \quad \text{به ازای هر } ۳.$$

آنگاه $x = 0$ پایدار مجانبی خواهد بود.

قضیه ۲۵.۳.۱ (قضیه پایداری مجانبی سراسری [۳۳]). تحت شرایط قضیه ۲۴.۳.۱ اگر $V(\cdot)$ به طور شعاعی بی کران باشد آنگاه نقطه $x = 0$ پایدار مجانبی سراسری است.

قضیه ۲۶.۳.۱ (اصل تغییرناپذیری لاسال^{۶۵} [۳۳]). فرض کنیم V یک تابع به طور پیوسته مشتق پذیر باشد، همچنین فرض کنیم:

۱. $M \subset \mathbb{R}^n$ یک مجموعه فشرد و پایدار نسبت به جواب سیستم (۱۲.۱) باشد.

$$\dot{V}(x) \leq 0, \quad x \in M \quad \text{به ازای هر } ۲.$$

$$\dot{V}(x) = 0 \quad \text{مجموعه همه نقاط } M \text{ باشد به طوری که } ۳.$$

۴. N بزرگترین مجموعه پایدار در E باشد.

آنگاه به ازای هر $x(t_0) \in M$ که $t_0 \geq 0$ می باشد داریم:

$$\lim_{t \rightarrow \infty} \text{dist}(x(t), N) = 0.$$

^{۶۵}LaSalle Principle of Invariance

فصل ۲

یک مدل شبکه عصبی برای مسائل برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی

۱.۲ مقدمه

مسئله‌ی بهینه‌سازی در طیف وسیعی از کاربردهای علمی و مهندسی از جمله پردازش سیگنال، شناسایی سیستم، طراحی صافی، تقریب توابع، تجزیه و تحلیل رگرسیون و... بکار برده می‌شوند [۴۴، ۴۹، ۵۳، ۵۷، ۱۱]. در بسیاری از مسائل بهینه‌سازی عملی مثل برنامه‌ریزی سیستم‌های قدرت و مسیریابی سیستم‌های ارتباط از راه دور، تعداد متغیرهای تصمیم‌گیری و محدودیت‌ها معمولاً بسیار زیاد هستند. چالش اصلی زمانی بوجود می‌آید که یک روش بهینه‌سازی در مقیاس بزرگ باید عملکرد یک مسئله بهینه‌سازی زمان حقیقی^۱ را بهینه‌سازی کند. برای چنین کاربردهایی، روش‌های بهینه‌سازی کلاسیک ممکن است با توجه به چند بعدی بودن مسئله در زمان محاسبات مناسب نباشد. یک راه بسیار امیدوارکننده، روش‌های بهینه‌سازی بر مبنای شبکه‌های عصبی مصنوعی برای حل این نوع از مسائل است. شبکه‌ی عصبی از مک‌کلاچ^۲ و پژوهش‌های پیشگامانه‌ی نیم قرن پیش توسط پیتز^۳ ناشی می‌شود. از آن زمان به بعد مدل‌های شبکه عصبی متعددی ایجاد شده است. در سال‌های اخیر، شبکه‌های عصبی زمان پیوسته‌ی بسیاری برای بهینه‌سازی مقید [۲۱، ۷۱، ۷۰، ۵۸، ۳۸] وجود دارد که با استفاده از پارامتر پنالته [۸، ۵۴، ۷۲] ایجاد شده است. در این فصل، یک شبکه عصبی زمان پیوسته‌ای را برای حل یک مسئله برنامه‌ریزی درجه دوم با محدودیت‌های هیبریدی پیشنهاد می‌کنیم. ثابت می‌کنیم که مدل ارائه شده پایدار به مفهوم لیاپانوف و همگرای سراسری به جواب مسئله اصلی است. در نهایت، نتایج شبیه‌سازی و مثال‌های به کار رفته، کارایی شبکه عصبی پیشنهادی را کاملاً تایید می‌کند.

^۱Real time

^۲McCulloch

^۳Pitts

۲.۲ بیان مسأله

مسأله‌ی برنامه‌ریزی غیرخطی زیر با محدودیت هیبرید را در نظر بگیرید:

$$\text{minimize } f(x) \quad (۱.۲)$$

subject to

$$Ax = b, \quad (۲.۲)$$

$$Hx \leq d, \quad (۳.۲)$$

$$x \in Q, \quad (۴.۲)$$

که در آن $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ محدب و به طور پیوسته دوبار مشتق‌پذیر است، $A \in \mathbb{R}^{l \times n}$ ، $H \in \mathbb{R}^{m \times n}$ ، $d \in \mathbb{R}^m$ ، $b \in \mathbb{R}^l$ ، $Q = \{x \in \mathbb{R}^n : \alpha_i \leq x_i \leq \beta_i \quad (i = 1, \dots, n)\}$ و β_i ها می‌توانند $+\infty$ و α_i ها می‌توانند $-\infty$ باشند. فرض کنید

$$\left\{ \begin{array}{l} D = \text{diag}(1, 1, \dots, 1) \in \mathbb{R}^{n \times n}, \\ E = \begin{pmatrix} H \\ D \\ -D \end{pmatrix} \in \mathbb{R}^{(m+2n) \times n}, \\ \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \in \mathbb{R}^n, \\ \alpha = \begin{pmatrix} -\alpha_1 \\ \vdots \\ -\alpha_n \end{pmatrix} \in \mathbb{R}^n, \\ g = \begin{pmatrix} d \\ \beta \\ \alpha \end{pmatrix} \in \mathbb{R}^{m+2n}, \end{array} \right.$$

در اینصورت مسأله (۱.۲)–(۴.۲) به صورت زیر تبدیل می‌شود

$$\text{minimize } f(x) \quad (۵.۲)$$

subject to

$$Ax = b, \quad (۶.۲)$$

$$Ex \leq g. \quad (۷.۲)$$

قضیه ۱.۲.۲. [۴۲] $x^* \in \mathbb{R}^n$ یک جواب بهینه (۵.۲)-(۷.۲) است اگر و فقط اگر $u^* \in \mathbb{R}^{m+l}$ و $v^* \in \mathbb{R}^l$ وجود داشته باشند به طوری که $(x^{*T}, u^{*T}, v^{*T})^T$ در شرایط کروش-کان-تاکر زیر صدق کند:

$$\begin{cases} u \geq 0, & Ex - g \leq 0, & u^T(Ex - g) = 0, \\ \nabla f + E^T u + A^T v = 0, \\ Ax = b. \end{cases} \quad (۸.۲)$$

x^* یک نقطه KKT (۵.۲)-(۷.۲) نامیده می‌شود و زوج $(u^{*T}, v^{*T})^T$ بردار ضرایب لاگرانژ متناظر با x^* نامیده می‌شوند.

قضیه ۲.۲.۲. [۴۲] x^* یک جواب بهینه (۵.۲)-(۷.۲) است اگر و فقط اگر x^* یک نقطه KKT از (۵.۲)-(۷.۲) باشد.

۳.۲ مدل شبکه عصبی

فرض کنید $x(\cdot)$ ، $u(\cdot)$ و $v(\cdot)$ متغیرهای وابسته به زمان باشند. هدف ساختن یک سیستم دینامیکی زمان-پیوسته است که بر نقطه KKT مساله (۵.۲)-(۷.۲) منطبق خواهد شد. یک مدل شبکه عصبی برای حل (۵.۲)-(۷.۲) و دوگان آن پیشنهاد می‌کنیم که معادله دینامیکی آن به صورت زیر تعریف می‌شود.

$$\frac{dx}{dt} = -(\nabla f + E^T(u + Ex - g)^+ + A^T v), \quad (۹.۲)$$

$$\frac{du}{dt} = (u + Ex - g)^+ - u, \quad (۱۰.۲)$$

$$\frac{dv}{dt} = Ax - b, \quad (۱۱.۲)$$

با نقطه اولیه $(x_0^T, u_0^T, v_0^T)^T$ که

$$(u + Ex - g)^+ = ([(u + Ex - g)_1]^+, [(u + Ex - g)_2]^+, \dots, [(u + Ex - g)_m]^+)^T,$$

$$[(u + Ex - g)_k]^+ = \max\{ (u + Ex - g)_k, 0 \}, \quad k = 1, 2, \dots, m.$$

برای سادگی فرض کنید، $y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ و D^* را به عنوان مجموعه جواب‌های بهینه (۵.۲)-(۷.۲) و دوگانش در نظر گرفته و تعریف می‌کنیم

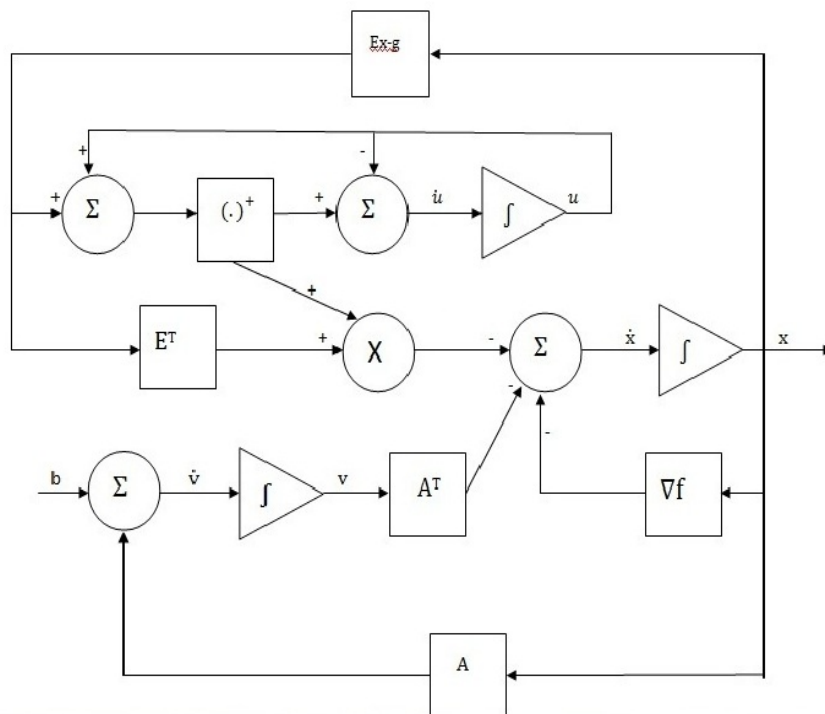
$$\Psi(y) = \begin{pmatrix} -(\nabla f + E^T(u + Ex - g)^+ + A^T v) \\ (u + Ex - g)^+ - u \\ Ax - b \end{pmatrix}. \quad (۱۲.۲)$$

با به کارگیری یک عامل تنظیم‌کننده، مدل شبکه عصبی (۹.۲)–(۱۱.۲) می‌تواند به صورت زیر نوشته شود:

$$\frac{dy}{dt} = \eta \Psi(y), \quad (۱۳.۲)$$

$$y(t_0) = y_0, \quad \eta > 0, \quad (۱۴.۲)$$

که در آن η یک پارامتر مقیاس^۴ است و آهنگ همگرایی شبکه عصبی (۱۳.۲) و (۱۴.۲) را نشان می‌دهد. برای سادگی تحلیل، فرض می‌کنیم $\eta = 1$ باشد. نحوه اجرای سخت‌افزاری شبکه عصبی (۱۳.۲) و (۱۴.۲) در شکل (۱.۲) نشان داده می‌شود.



شکل ۱.۲: نمودار جعبه‌ای برای شبکه عصبی (۱۳.۲)–(۱۴.۲)

۴.۲ خواص همگرایی و پایداری

قضیه ۱.۴.۲. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل شبکه عصبی (۱۳.۲) و (۱۴.۲) باشد، آنگاه x^* یک نقطه KKT مساله (۵.۲)–(۷.۲) است. برعکس، اگر $x^* \in \mathbb{R}^n$ یک جواب بهینه مساله (۵.۲)–(۷.۲) باشد، آنگاه $u^* \in \mathbb{R}^{m+2n}$ و $v^* \in \mathbb{R}^l$ وجود دارند به طوری که $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ یک نقطه تعادل شبکه عصبی پیشنهادی (۱۳.۲) و (۱۴.۲) است.

^۴Scale parameter

برهان. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل (۱۳.۲) و (۱۴.۲) باشد آنگاه $\frac{dx^*}{dt} = 0$ و $\frac{du^*}{dt} = 0$ و $\frac{dv^*}{dt} = 0$.

به سادگی نتیجه می‌شود که

$$\nabla f + E^T(u^* + Ex^* - g)^+ + A^T v^* = 0, \quad (15.2)$$

$$(u^* + Ex^* - g)^+ - u^* = 0, \quad (16.2)$$

$$Ax^* - b = 0. \quad (17.2)$$

واضح است که $(u^* + Ex^* - g)^+ = u^*$ اگر و فقط اگر

$$u^* \geq 0, \quad Ex^* - g \leq 0, \quad u^{*T}(Ex^* - g) = 0. \quad (18.2)$$

بعلاوه، با جایگذاری (۱۵.۲) در (۱۵.۲) داریم

$$\nabla f + E^T u^* + A^T v^* = 0. \quad (19.2)$$

از (۱۶.۲)-(۱۹.۲) می‌توان دید که $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ در شرایط KKT صدق می‌کند. اثبات عکس قضیه واضح است. \square

لم ۲.۴.۲. برای هر نقطه اولیه $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ ، یک جواب پیوسته منحصر به فرد $y(t) = (x(t)^T, u(t)^T, v(t)^T)^T$ برای سیستم (۱۳.۲) و (۱۴.۲) وجود دارد.

برهان. به سادگی می‌توان تحقیق کرد که $(u + Ex - g)^+ - u, \nabla f + E^T(u + Ex - g)^+ + A^T v$ و $Ax - b$ بر روی مجموعه باز $\bar{D} \subseteq \mathbb{R}^{3n+m+l}$ پیوسته لیپ شیتز محلی هستند. بر اساس قضیه وجود و یکتایی جواب یک معادله دیفرانسیل (۲۰.۳.۱)، شبکه عصبی (۱۳.۲) و (۱۴.۲) یک جواب پیوسته منحصر به فرد $y(t), t \in [t_0, \tau)$ برای $\tau > t_0$ وقتی که $\tau \rightarrow \infty$ دارد. \square

لم ۳.۴.۲. ماتریس ژاکوبین $\nabla \Psi(y)$ از نگاشت Ψ که در (۱۲.۲) تعریف شده است یک ماتریس نیمه معین منفی است.

برهان. بدون کاسته شدن از کلیت، فرض کنید که $0 < p < m$ به طوری وجود داشته باشد که

$$(u + Ex - g)^+ = ((u + Ex - g)_1, (u + Ex - g)_2, \dots, (u + Ex - g)_p, \underbrace{0, 0, \dots, 0}_{m+2n-p})^T.$$

با یک محاسبه ساده، نشان داده می‌شود

$$\nabla \Psi(y) = \begin{pmatrix} -(\nabla^v f + (E^p)^T E^p) & -(E^p)^T & -A^T \\ E^p & \mathcal{Z}_{(m+2n) \times (m+2n)} & O_{(m+2n) \times l} \\ A & O_{l \times (m+2n)} & O_{l \times l} \end{pmatrix},$$

که در آن O بیانگر یک ماتریس صفر است،

$$E^p = \begin{pmatrix} U_{p \times n} \\ O_{(m+2n-p) \times n} \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \dots \\ \dots \\ E_p \\ O_{1 \times n} \\ O_{1 \times n} \\ \dots \\ \dots \\ O_{1 \times n} \end{pmatrix},$$

و

$$\mathcal{Z}_{(m+2n) \times (m+2n)} = \begin{pmatrix} O_{p \times p} & O_{p \times (m+2n-p)} \\ O_{(m+2n-p) \times p} & -I_{(m+2n-p) \times (m+2n-p)} \end{pmatrix}.$$

از [۵۹] می‌بینیم که $(E^p)^T E^p$ یک ماتریس نیمه معین مثبت است. $\nabla^2 f$ نیمه معین مثبت است، چون f محدب است. بعلاوه، واضح است که ماتریس $\mathcal{Z}_{(m+2n) \times (m+2n)}$ نیمه معین منفی است. طبق مطالب بیان شده، می‌توانیم نتیجه بگیریم که ماتریس ژاکوبین $\nabla \Psi(y)$ یک ماتریس نیمه معین منفی است.

اگر $p = m + 2n$ ، یعنی،

$$(u + Ex - g)^+ = ((u + Ex - g)_1, (u + Ex - g)_2, \dots, (u + Ex - g)_{m+2n})^T,$$

آنگاه

$$\nabla \Psi(y) = \begin{pmatrix} -(\nabla^2 f + E^T E) & -E^T & -A^T \\ E & O_{(m+2n) \times (m+2n)} & O_{(m+2n) \times l} \\ A & O_{l \times (m+2n)} & O_{l \times l} \end{pmatrix}.$$

مشابه حالت قبل، به سادگی ثابت می‌شود که $\nabla \Psi(y)$ یک ماتریس نیمه معین منفی است. نهایتاً، اگر $p = 0$ ، یعنی $(u + Ex - g)^+ = (\underbrace{0, 0, \dots, 0}_{m+2n})^T$ ، آنگاه داریم:

$$\nabla \Psi(y) = \begin{pmatrix} -\nabla^2 f & O_{n \times (m+2n)} & -E^T \\ O_{(m+2n) \times n} & -I_{(m+2n) \times (m+2n)} & O_{(m+2n) \times l} \\ A & O_{l \times (m+2n)} & O_{l \times l} \end{pmatrix}.$$

در این حالت همچنین به سادگی می‌توان نیمه معین منفی بودن ماتریس $\nabla \Psi(y)$ را تصدیق کرد. این برهان را کامل می‌کند. \square

حال نتایج اصلی را به شرح ذیل بیان می‌کنیم.

قضیه ۴.۴.۲. مدل شبکه عصبی پیشنهادی در (۱۳.۲) و (۱۴.۲) پایدار لیاپانوف و همگرایی سراسری به $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ است، که جواب بهینه (۵.۲) - (۷.۲) است.

برهان. تابع $E : \mathbb{R}^{n+m+l} \rightarrow \mathbb{R}$ را به صورت زیر در نظر بگیرید:

$$E(y) = \|\Psi(y)\|^2 + \frac{1}{\gamma} \|y - y^*\|^2. \quad (20.2)$$

از (۱۲.۲)، می‌بینیم که

$$\frac{d\Psi}{dt} = \frac{\partial \Psi}{\partial y} \frac{dy}{dt} = \nabla \Psi(y) \Psi(y).$$

محاسبه مشتق $E(t)$ به همراه جواب $y(t)$ از شبکه عصبی (۱۳.۲) و (۱۴.۲) داریم

$$\frac{dE(y(t))}{dt} = \left(\frac{d\Psi}{dt}\right)^T \Psi + \Psi^T \left(\frac{d\Psi}{dt}\right) + (y - y^*)^T \frac{dy(t)}{dt} \quad (21.2)$$

$$= \Psi^T (\nabla \Psi(y)^T + \nabla \Psi(y)) \Psi + (y - y^*)^T \Psi(y). \quad (22.2)$$

با به کارگیری لم (۳.۴.۲)، به دست می‌آوریم

$$\Psi^T(y) (\nabla \Psi(y)^T + \nabla \Psi(y)) \Psi(y) \leq 0. \quad (23.2)$$

بعلاوه، از تعریف (۱۰.۳.۱) و قضیه (۱۸.۳.۱)، داریم

$$(y - y^*)^T (\Psi(y) - \Psi(y^*)) = (y - y^*)^T \Psi(y) \leq 0.$$

بنابراین

$$\frac{dE(y(t))}{dt} \leq 0. \quad (24.2)$$

این بدین معنی است که شبکه عصبی (۱۳.۲) و (۱۴.۲) پایدار به مفهوم لیاپانوف است. چون

$$E(y_0) \geq E(y) \geq \frac{1}{\gamma} \|y - y^*\|^2, \quad (25.2)$$

پس $y(t)$ کراندار است. فرض کنید $\Lambda = \{y \in \mathbb{R}^{n+m+l} | E(y) \leq E(y_0)\}$. با استفاده از قضیه لسال داریم $\{(x(t)^T, u(t)^T, v(t)^T)^T \rightarrow M, t \rightarrow \infty\}$ که M بزرگترین مجموعه تغییرناپذیر در $K = \{(x(t)^T, u(t)^T, v(t)^T)^T | \frac{dE(y(t))}{dt} = 0\}$ است. از (۱۳.۲)، (۱۴.۲) و (۲۴.۲) طبق [۹] داریم:

$$\frac{dx}{dt} = \frac{du}{dt} = \frac{dv}{dt} = 0 \Leftrightarrow \frac{dE(y(t))}{dt} = 0.$$

بنابراین $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T \in D^*$ و $M \subseteq K \subseteq D^*$.

در نتیجه، شبکه عصبی پیشنهادی در (۱۳.۲) و (۱۴.۲) همگرایی سراسری به یک نقطه تعادل $\bar{y} = (\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ است، که جواب بهینه (۵.۲) - (۷.۲) است. □

یک نتیجه از قضیه (۴.۴.۲) به صورت زیر بیان می‌شود:

نتیجه ۵.۴.۲. اگر $D^* = \{(x^{*T}, u^{*T}, v^{*T})^T\}$ باشد، آنگاه شبکه عصبی (۱۳.۲) و (۱۴.۲) برای حل (۵.۲)–(۷.۲) پایدار مجانبی سراسری به نقطه تعادل یکتای $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ است.

۵.۲ شبیه‌سازی‌های عددی

به منظور اثبات کارایی و موثر بودن شبکه عصبی پیشنهادی، در این بخش، چندین مثال را توسط شبکه عصبی (۱۳.۲) و (۱۴.۲) آزمایش می‌کنیم. شبیه‌سازی توسط نرم افزار مطلب^۵ ۷ اجرا می‌شود. حل کننده معادله دیفرانسیل معمولی ode45 می‌باشد. دقت جواب مثال‌های زیر توسط نرم افزار لینگو^۶ ۱۱ احراز شده است.

مثال ۱.۵.۲

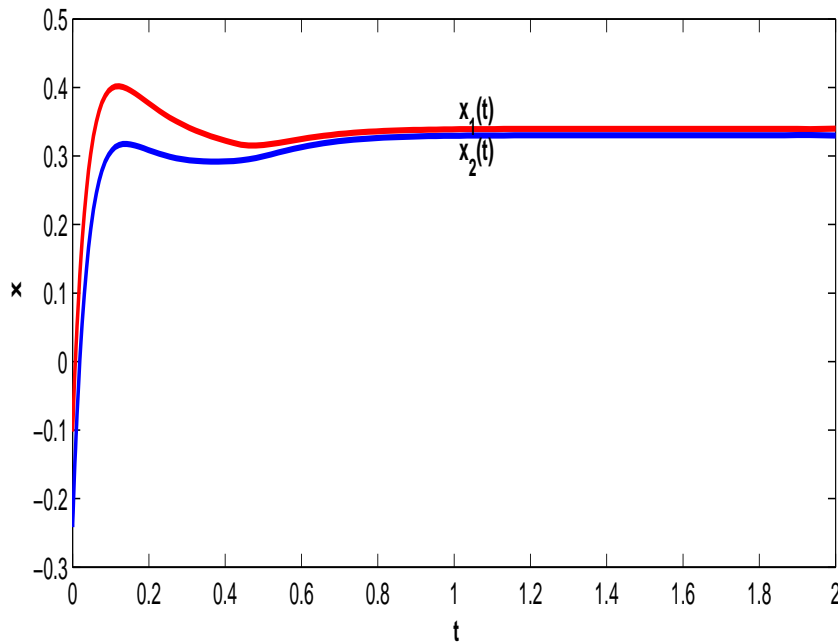
$$\text{minimize } f(x) = 0.4x_2 + x_1^2 + x_2^2 - x_1x_2 + \frac{1}{30}x_1^3$$

$$\text{subject to } \begin{cases} x_1 \geq 0, & x_2 \geq 0, & x_1 + 0.5x_2 \geq 0.4, \\ 0.5x_1 + x_2 \geq 0.5. \end{cases}$$

جواب بهینه $x^* = (0.3395, 0.3303)^T$ است. شکل (۲.۲) نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند.

^۵Matlab

^۶Lingo



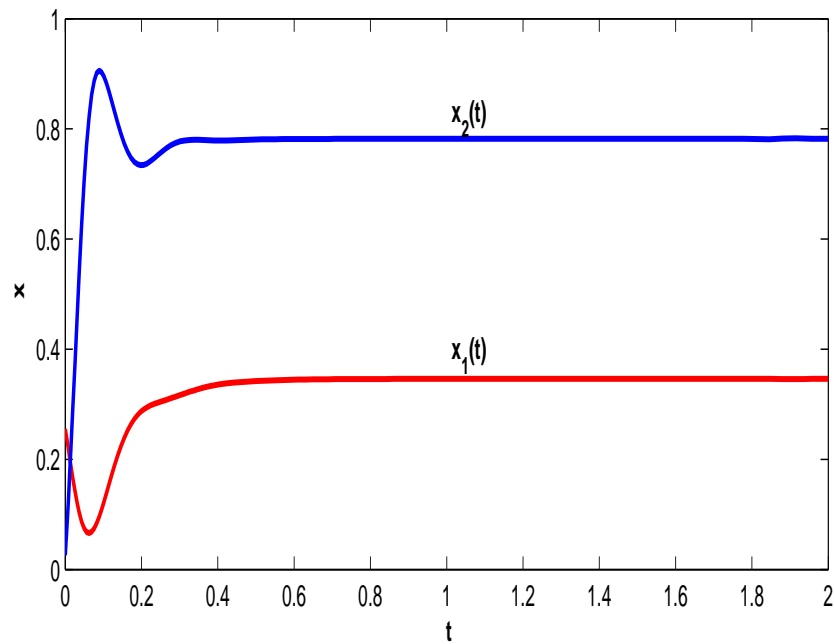
شکل ۲.۲: رفتارهای گذرای x_1 و x_2 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۱.۵.۲).

مثال ۲.۵.۲

$$\text{minimize } f(x) = \frac{1}{4}x_1^4 + 0.5x_1^2 + \frac{1}{4}x_2^4 + 0.5x_2^2 - 0.9x_1x_2$$

$$\text{subject to } \begin{cases} -x_1 - x_2 \geq -2, \\ x_1 - x_2 \geq -2, \\ x_1 - 3x_2 = -2, \\ 0 \leq x \leq 1. \end{cases}$$

جواب بهینه $x^* = (0.3460, 0.7822)^T$ است. شکل (۳.۲) نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند.



شکل ۳.۲: رفتارهای گذرای x_2 و x_1 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۲.۵.۲).

مثال ۳.۵.۲

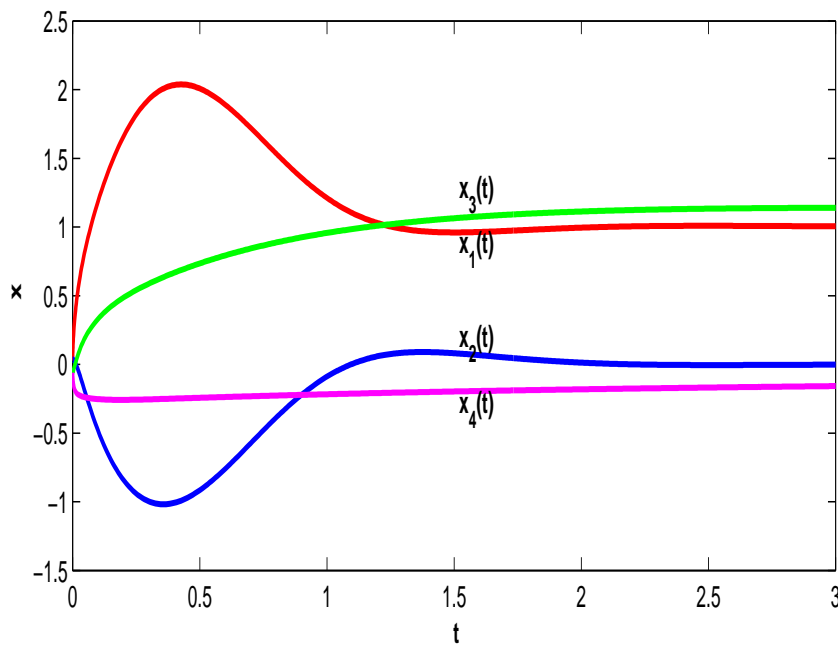
$$\begin{aligned} \text{minimize } f(x) = & \frac{3}{4}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) \\ & + 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_4 \end{aligned}$$

$$\text{subject to } \begin{cases} x_1 + x_2 = 1, \\ x_3 + x_4 = 1, \\ x \in Q. \end{cases}$$

که در آن $Q = \{x \in \mathbb{R}^4 \mid 0 \leq x_i \leq 1 \quad \forall i = 1, 2, 3, 4\}$.

جواب بهینه $x^* = (1/0026, -0/0014, 1/1413, -0/1615)^T$ است. شکل (۴.۲) نشان می‌دهد

که مسیرها همگرا به جواب بهینه هستند.



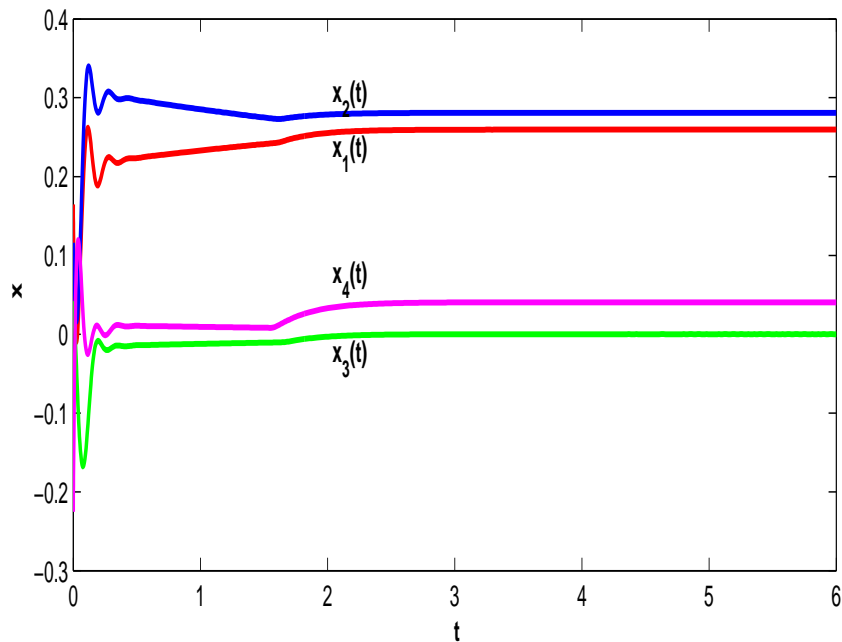
شکل ۴.۲: رفتارهای گذرای x_1, x_2, x_3 و x_4 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۳.۵.۲).

مثال ۴.۵.۲

$$\text{minimize } f(x) = 0.4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{x_1^3}{30}$$

$$\text{subject to } \begin{cases} x_1 + x_2 - x_4 = 0.5, \\ x_1 + 0.5x_2 - x_3 = 0.4, \\ x \geq 0. \end{cases}$$

جواب بهینه $x^* = (0.2595, 0.2809, 0, 0.404)^T$ است. شکل (۵.۲) نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند.



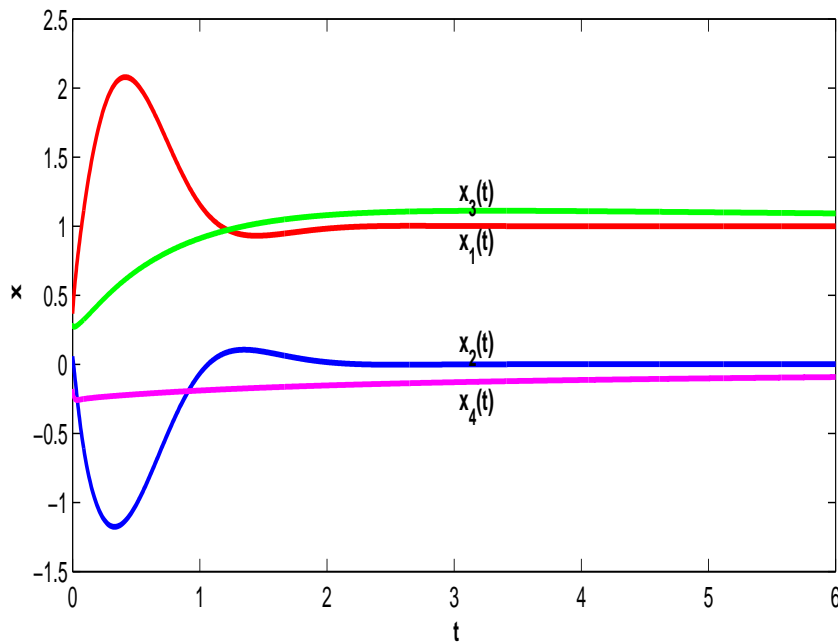
شکل ۵.۲: رفتارهای گذرای x_1, x_2, x_3, x_4 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۴.۵.۲).

مثال ۵.۵.۲

$$\begin{aligned} \text{minimize } f(x) = & \frac{3}{4}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) \\ & + 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_4 \end{aligned}$$

$$\text{subject to } \begin{cases} x_1 + x_2 = 1, \\ x_3 + x_4 = 1, \\ 0.1 \leq x_1 \leq 1.0, & 0 \leq x_2 \leq 1.0, \\ 0 \leq x_3 \leq 1.0, & 0.1 \leq x_4 \leq 1.0. \end{cases}$$

جواب بهینه $x^* = (1.0001, 0, -0.0001, 1.0003)^T$ است. شکل (۶.۲) نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند.



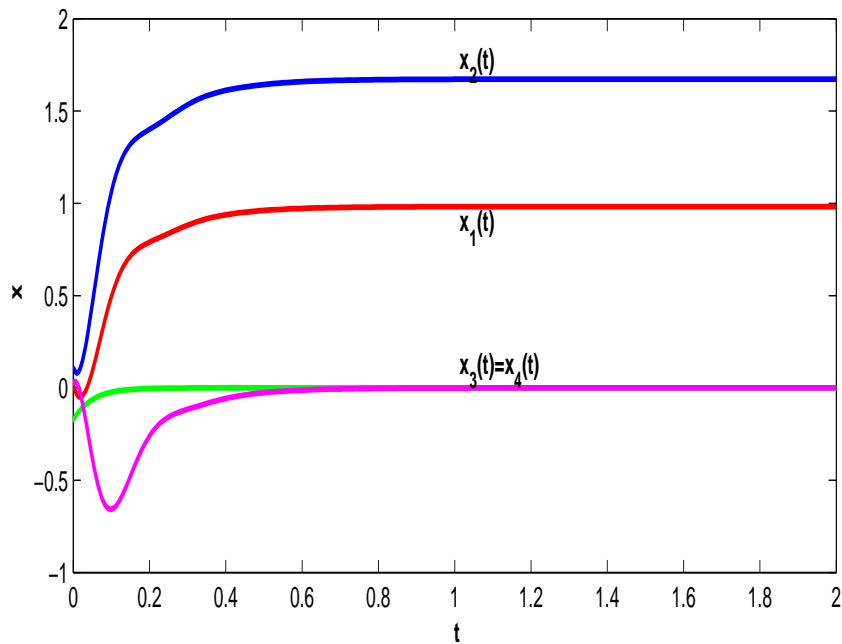
شکل ۶.۲: رفتارهای گذرای x_1, x_2, x_3 و x_4 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۵.۵.۲).

مثال ۶.۵.۲

$$\text{minimize } f(x) = 0.4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{1}{30}x_1^3$$

$$\text{subject to } \begin{cases} -x_1 + x_2 - x_3 \leq 2, \\ 3x_1 + x_2 - x_3 - x_4 \leq 18, \\ \frac{1}{4}x_1 + x_2 - x_4 = 2, \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases}$$

جواب بهینه $x^* = (0.9820, 1.6727, 0, 0)^T$ است. شکل (۷.۲) همگرایی مسیرها را به جواب بهینه نشان می‌دهد.



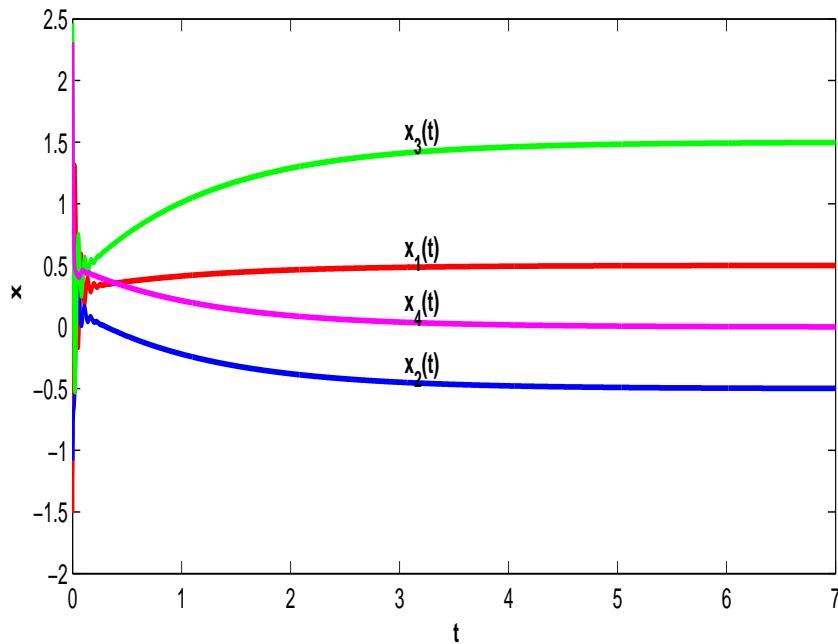
شکل ۷.۲: رفتارهای گذرای x_1, x_2, x_3 و x_4 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۶.۵.۲).

مثال ۷.۵.۲.

$$\text{minimize } f(x) = 3x_1^2 + 3x_2^2 + 4x_3^2 + 5x_4^2 + 3x_1x_2 + 5x_1x_3 + x_2x_4 - 11x_1 - 5x_4$$

$$\text{subject to } \begin{cases} 3x_1 - 3x_2 - 2x_3 + x_4 = 0, \\ 4x_1 + x_2 - x_3 - 2x_4 = 0, \\ -x_1 + x_2 \leq -1, \\ -2 \leq 3x_1 + x_3 \leq 4. \end{cases}$$

جواب بهینه $x^* = (0.4994, -0.4980, 1.4964, 0.0016)^T$ است. همگرایی مسیرها به x^* در شکل (۸.۲) نشان داده می‌شود.



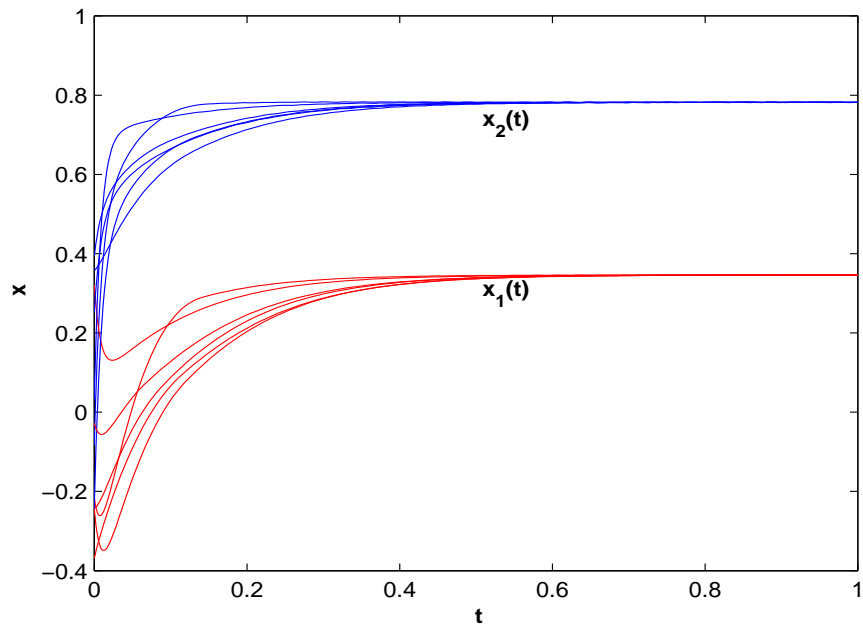
شکل ۸.۲: رفتارهای گذرای x_1, x_2, x_3, x_4 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۷.۵.۲).

مثال ۸.۵.۲

$$\text{minimize } f(x) = \frac{1}{4}x_1^4 + \frac{1}{2}x_1^2 + \frac{1}{4}x_2^4 + \frac{1}{2}x_2^2 - 0.9x_1x_2$$

$$\text{subject to } \begin{cases} x_1 + x_2 \leq 2, \\ -x_1 + x_2 \leq 2, \\ x_1 - 3x_2 \leq -2, \\ x_1, x_2 \geq 0. \end{cases}$$

جواب بهینه $x^* = (0.3462, 0.7814)^T$ است. شکل (۹.۲) نشان می‌دهد که مسیرها همگرا به جواب بهینه هستند.



شکل ۹.۲: رفتارهای گذرای x_2 و x_1 از شبکه عصبی (۱۳.۲) و (۱۴.۲) در مثال (۸.۵.۲).

فصل ۳

کنترل بهینه سیستم‌های هیبریدی خطی زمان گسسته با استفاده از یک مدل شبکه عصبی

۱.۳ مقدمه

اخیرا فن آوری‌های نو سبب گرایش قابل توجهی در بررسی فرایندهای ترکیبی پیوسته و گسسته با اشاره به سیستم‌های هیبریدی شده است. سیستم‌های هیبریدی در جامع ترین حالت با تعامل مدل‌های زمان پیوسته (با معادلات تفاضلی و دیفرانسیلی توصیف می‌شود) و قواعد منطق سیستم‌های پیشامد گسسته (مثلا با استفاده از منطق زمانی، ماشین‌های حالت متناهی یا قواعد دیگر توصیف می‌شود) و مولفه‌های گسسته مشخص می‌شوند (کلید روشن و خاموش یا سوپاپ‌ها، چرخ دنده‌ها یا انتخاب‌گرهای سرعت و...) . چنین سیستم‌هایی بین حالت‌های عملیاتی مختلفی که هر قسمت توسط قوانین دینامیکی خاصی اداره می‌شود (سوییچ) تغییر می‌کند. تغییر از یک حالت به حالت دیگر به وسیله متغیرهای آستانه‌ای مشخص، با گذشت دوره زمانی معین یا به وسیله ورودی‌های خارجی انجام می‌شود [۵۵]. مفاهیم اصلی می‌تواند در [۵۵، ۵۰، ۵۱، ۴، ۳۶، ۳۷] یافت شود. روش‌های مختلف برای تجزیه و تحلیل و طراحی کنترل کننده‌ها برای سیستم‌های هیبریدی در طول چند سال گذشته مطالعه شده است [۲۵، ۱۶، ۵۲، ۳۷، ۴]. در میان آن‌ها رده‌ی کنترل کننده‌های بهینه یکی از مهم‌ترین بررسی‌ها است. این مطالعات تا حد زیادی با مدل‌های هیبریدی پذیرفته شده، در فرمول‌بندی مساله کنترل بهینه و در روش بکار رفته برای حل آن متفاوت است. در اینجا بر روی مدل‌های هیبریدی خطی زمان گسسته تمرکز می‌کنیم. مدل‌های زمان گسسته می‌تواند بسیاری از فرایندها را توصیف کند مثل سیستم‌های گسسته زمانی خطی با خطی سازی قطعه‌ای ثابت، سیستم‌های گسسته زمانی خطی با متغیرهای حالت و ورودی گسسته، سیستم‌های سوییچ که رفتار دینامیکی آن‌ها به وسیله تعداد متناهی از مدل‌های گسسته زمانی خطی همراه با مجموعه‌ای از قوانین منطقی برای سوییچ بین مدل‌های موجود توصیف می‌شود، تقریب سیستم‌های گسسته زمانی یعنی همراه با خطی سازی‌های چندگانه در نقاط عملیاتی مختلف. برای رده‌ی سیستم‌های هیبریدی زمان گسسته خطی می‌توانیم قانون کنترل بهینه را بدون شبکه بندی فضای حالت،

مشخص و محاسبه کنیم. این راه حل برای مسائل کنترل بهینه سیستم‌های هیبریدی زمان گسسته اولین بار توسط سنتج^۱ در [۲۵] مطرح شد. ماین^۲ در سخنرانی عمومی خود در کنفرانس کنترل ۲۰۰۱ اروپا [۲۰]، خصوصیات بسیار جذابی از راه حل بازخورد حالت را برای مسائل کنترل بهینه سیستم‌های هیبرید خطی با معیارهای عملکرد براساس نرم‌های درجه دوم و خطی ارائه داد. قانون کنترل بهینه زمان گسسته می‌تواند با استفاده از برنامه‌ریزی چندپارامتری بطور موثر محاسبه شود. بویژه در [۲۷]، الگوریتم جدیدی پیشنهاد شده است که معادله هامیلتون - ژاکوبی - بلمن را با استفاده از حل کننده چند پارامتری ساده حل می‌کند. در همکاری با شرکت‌ها و موسسات مختلف، نتایجی که در [۲۷] توصیف می‌شود برای طیف وسیعی از مسائل [۳۲، ۱۸، ۴، ۲۸، ۶، ۵، ۲۶، ۴۵، ۶۰] بکاربرده شده است.

در این فصل از پایان‌نامه، ابتدا یک مسأله کنترل بهینه زمان گسسته با تابع هدف درجه دوم، قیود خطی و متغیرهای کراندار معرفی می‌کنیم. با تبدیل مسأله بیان شده به یک مسأله برنامه‌ریزی درجه دوم استاندارد بیان شده در فصل دوم، یک مدل شبکه عصبی کارا برای حل آن معرفی می‌کنیم. در انتها با ارائه چندین مثال عددی، کارایی مدل ارائه شده را بررسی می‌کنیم.

۲.۳ بیان مسأله و ارائه یک مدل شبکه عصبی

در این بخش سیستم گسسته که توسط معادله وضعیت زیر بیان می‌شود مورد بررسی قرار می‌گیرد

$$x_{k+1} = A_k x_k + B_k u_k, \quad (1.3)$$

که در آن $A_k \in \mathbb{R}^{n \times n}$ و $B_k \in \mathbb{R}^{n \times m}$

$$x_k = \begin{pmatrix} x_{1k} \\ \vdots \\ x_{nk} \end{pmatrix},$$

$$u_k = \begin{pmatrix} u_{1k} \\ \vdots \\ u_{mk} \end{pmatrix},$$

$$p_{ik} \leq x_{ik} \leq q_{ik}, \quad (i = 1, \dots, n, k = 0, \dots, N),$$

و

$$s_{jk} \leq u_{jk} \leq t_{jk}, \quad (j = 1, \dots, m, k = 0, \dots, N).$$

مسأله یافتن مقدار بهینه $(x_k^{*T}, u_k^{*T})^T$ به ترتیبی است که تابع هدف زیر حداقل شود مد نظر است:

$$J = \frac{1}{2} x_N^T H x_N + \frac{1}{2} \sum_{k=0}^{N-1} [x_k^T Q x_k + u_k^T R u_k], \quad (2.3)$$

^۱Sontag

^۲Mayne

که در آن H و Q ماتریس‌های $n \times n$ متقارن حقیقی نیمه معین مثبت می‌باشند. \mathbb{R} ماتریس $m \times m$ متقارن حقیقی معین مثبت و N عدد طبیعی (صحیح و مثبت) است. مسأله (۱.۳) با تابع هدف (۲.۳)، یک مسأله برنامه‌ریزی محدب درجه دوم است. بنابراین برای حل آن از مدل شبکه عصبی ارائه در فصل قبل استفاده می‌کنیم. تعریف کنید

$$D_k = \text{diag}(1, 1, \dots, 1) \in \mathbb{R}^{n \times n},$$

$$q_k = (q_{1k}, \dots, q_{nk})^T \in \mathbb{R}^n,$$

$$p_k = (p_{1k}, \dots, p_{nk})^T \in \mathbb{R}^n,$$

$$E_k = \text{diag}(1, \dots, 1) \in \mathbb{R}^{m \times m},$$

$$s_k = (s_{1k}, \dots, s_{mk})^T \in \mathbb{R}^m,$$

$$t_k = (t_{1k}, \dots, t_{mk})^T \in \mathbb{R}^n.$$

بنابراین مسأله (۱.۳) با تابع هدف (۲.۳) را می‌توان بصورت زیر نوشت:

$$\text{minimize } f(x) = \frac{1}{2} x_N^T H x_N + \frac{1}{2} \sum_{k=0}^{N-1} [x_k^T Q x_k + u_k^T R u_k] \quad (۳.۳)$$

$$\text{subject to } \begin{cases} x_{k+1} = A_k x_k + B_k u_k, & k = 0, \dots, N-1, \\ D_k x_k - q_k \leq 0, \\ -D_k x_k + p_k \leq 0, \\ E_k u_k - t_k \leq 0, \\ -E_k u_k + s_k \leq 0. \end{cases} \quad (۴.۳)$$

برای حل مسأله بهینه‌سازی (۳.۳)-(۴.۳) یک مدل شبکه عصبی به صورت زیر پیشنهاد می‌کنیم:

$$\frac{dx_k}{dt} = -[Qx_k + Hx_k + D_k^T(v_k + D_k x_k - q_k)^+ - D_k^T(w_k - D_k x_k - p_k)^+ + A_k^T r_{1k}],$$

$$(k = 0, \dots, N),$$

$$\frac{du_k}{dt} = -[E_k^T(y_k + E_k u_k - t_k)^+ - E_k^T(z_k - E_k u_k - s_k)^+ + B_k^T r_{2k}], \quad (k = 0, \dots, N-1),$$

$$\frac{dv_k}{dt} = (v_k + D_k x_k - q_k)^+ - v_k, \quad v_k \in \mathbb{R}^n, (k = 0, \dots, N),$$

$$\frac{dw_k}{dt} = (w_k - D_k x_k + p_k)^+ - w_k, \quad w_k \in \mathbb{R}^n, (k = 0, \dots, N),$$

$$\frac{dy_k}{dt} = (y_k + E_k u_k - t_k)^+ - y_k, \quad y_k \in \mathbb{R}^m, (k = 0, \dots, N-1),$$

$$\frac{dz_k}{dt} = (z_k - E_k u_k - s_k)^+ - z_k, \quad z_k \in \mathbb{R}^m, (k = 0, \dots, N-1),$$

$$\frac{dr_k}{dt} = (A_k x_k + B_k u_k - x_{k+1}), \quad r_k \in \mathbb{R}^n, (k = 0, \dots, N-1).$$

تعریف کنید

$$X = \begin{pmatrix} x_0 \\ \vdots \\ x_N \end{pmatrix}, \quad U = \begin{pmatrix} u_0 \\ \vdots \\ u_{N-1} \end{pmatrix}, \quad V = \begin{pmatrix} v_0 \\ \vdots \\ v_N \end{pmatrix}$$

$$W = \begin{pmatrix} w_0 \\ \vdots \\ w_N \end{pmatrix}, \quad Y = \begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}, \quad Z = \begin{pmatrix} z_0 \\ \vdots \\ z_{N-1} \end{pmatrix}$$

$$R = \begin{pmatrix} r_0 \\ \vdots \\ r_{N-1} \end{pmatrix}, \quad r_k = \begin{pmatrix} r_{\setminus k} \\ r_{\sphericalangle k} \end{pmatrix},$$

$$y = \begin{pmatrix} X \\ U \\ V \\ W \\ Y \\ Z \\ R \end{pmatrix},$$

$$\Psi(y) = \begin{pmatrix} -[Qx_k + Hx_k + D_k^T(v_k + D_k x_k - q_k)^+ - D_k^T(w_k - D_k x_k - p_k)^+ + A_k^T r_{\setminus k}] \\ -[E_k^T(y_k + E_k u_k - t_k)^+ - E_k^T(z_k - E_k u_k - s_k)^+ + B_k^T r_{\sphericalangle k}] \\ (v_k + D_k x_k - q_k)^+ - v_k \\ (w_k - D_k x_k + p_k)^+ - w_k \\ (y_k + E_k u_k - t_k)^+ - y_k \\ (z_k - E_k u_k - s_k)^+ - z_k \\ (A_k x_k + B_k u_k - x_{k+1}) \end{pmatrix}.$$

با به کارگیری یک عامل تنظیم‌کننده η ، مدل شبکه عصبی (۱.۳) و (۲.۳) می‌تواند به صورت زیر نوشته شود:

$$\frac{dy}{dt} = \eta \Psi(y), \quad (۵.۳)$$

$$y(t_0) = y_0, \quad \eta > 0. \quad (۶.۳)$$

تحلیل پایداری و همگرایی مدل شبکه عصبی (۵.۳) و (۶.۳) دقیقاً مشابه فصل قبل است که در اینجا از ذکر آن صرف نظر می‌کنیم.

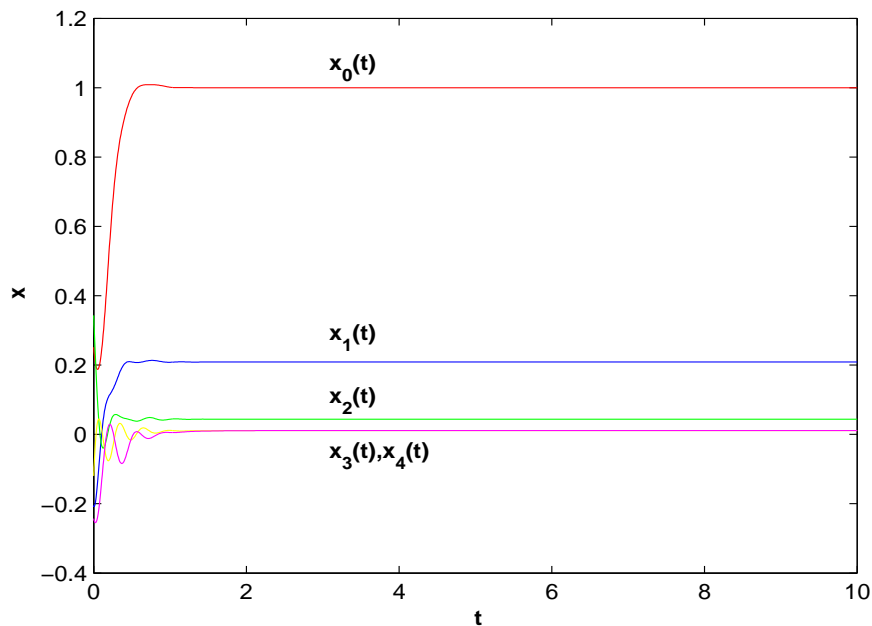
۳.۳ شبیه سازی های عددی

به منظور اثبات کارایی و موثر بودن شبکه عصبی پیشنهادی، در این بخش، ما چندین مثال را توسط شبکه عصبی (۵.۳) و (۶.۳) آزمایش می‌کنیم. شبیه سازی توسط نرم افزار مطلب^۳ اجرا می‌شود. حل کننده معادله دیفرانسیل معمولی ode45 می‌باشد. دقت جواب مثال‌های زیر توسط نرم‌افزار لینگو^۴ احراز شده است.

مثال ۱.۳.۳.

$$\text{minimize } \left(\frac{1}{4}\right)x_4^2 + \frac{1}{4} \sum_{k=0}^3 1/5 x_k^2 + u_{1k}^2 + 2u_{2k}^2$$

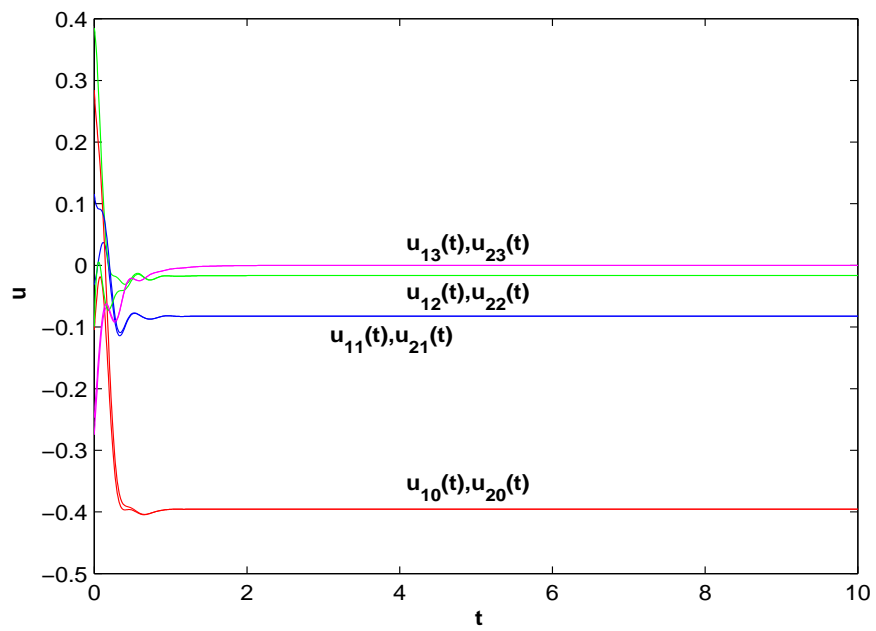
$$\text{subject to } \begin{cases} x_{k+1} = x_k + u_{1k} + u_{2k}, & k = 0, 1, 2, 3, \\ x_0 = 1. \end{cases}$$



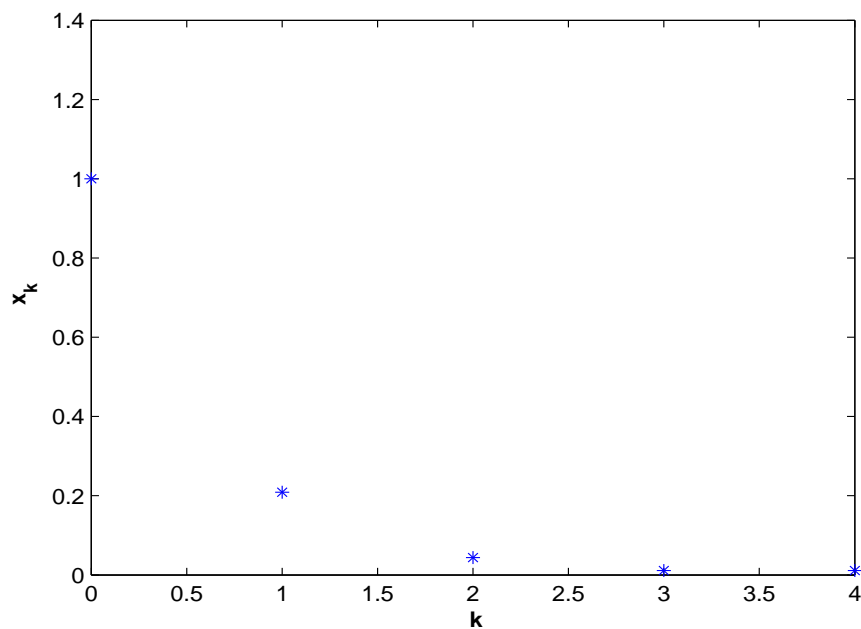
شکل ۱.۳: رفتارهای گذرای x_0, x_1, x_2, x_3 و x_4 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۱.۳.۳).

^۳Matlab

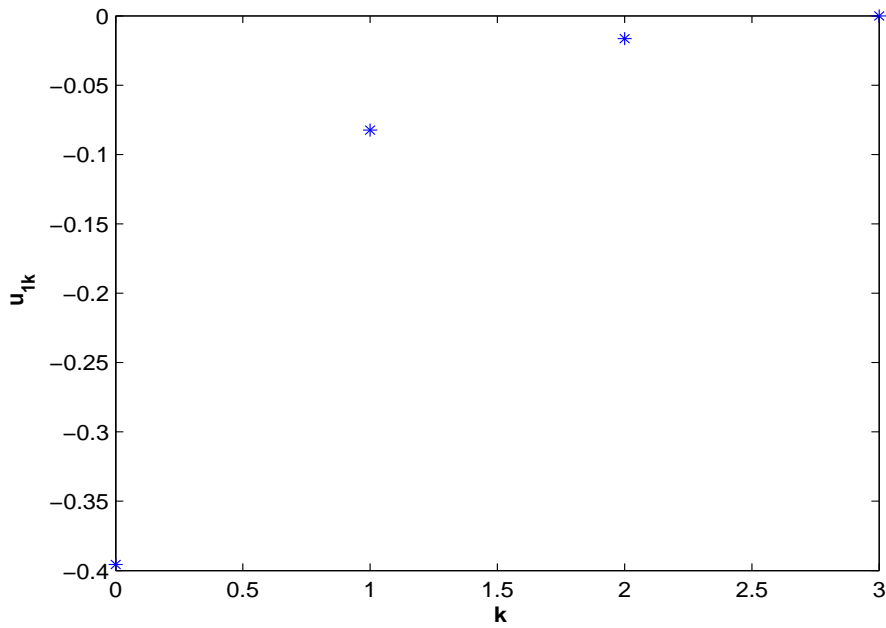
^۴Lingo



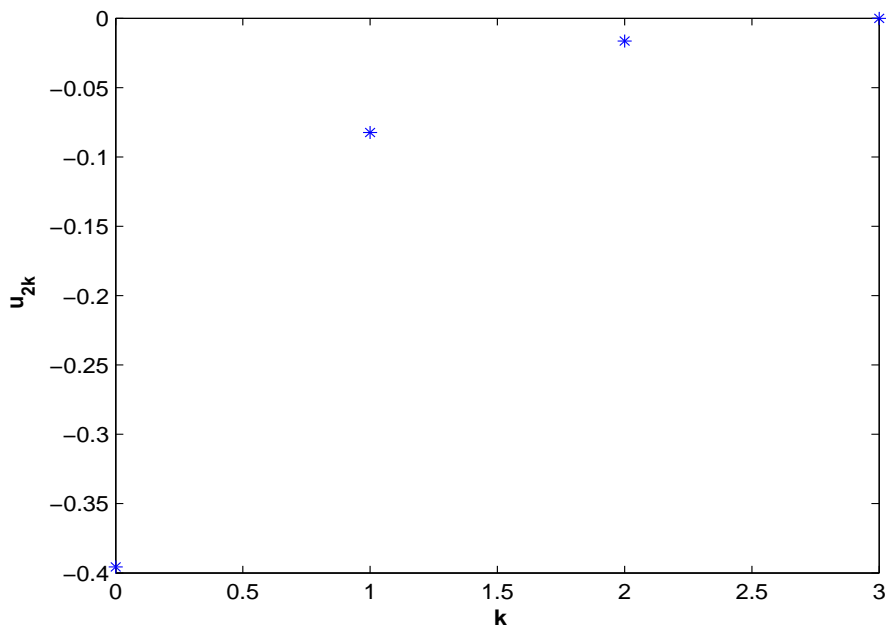
شکل ۲.۳: رفتارهای گذرای $u_{۲۳}, \dots, u_{۲۰}$ و $u_{۱۳}, \dots, u_{۱۰}$ از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۱.۳.۳).



شکل ۳.۳: رفتارهای گذرای زمان گسسته $x_۴, x_۳, x_۲, x_۱, x_۰$ از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۱.۳.۳).



شکل ۴.۳: رفتارهای گذرای زمان گسسته u_{10}, u_{11}, u_{12} و u_{13} از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۱.۳.۳).

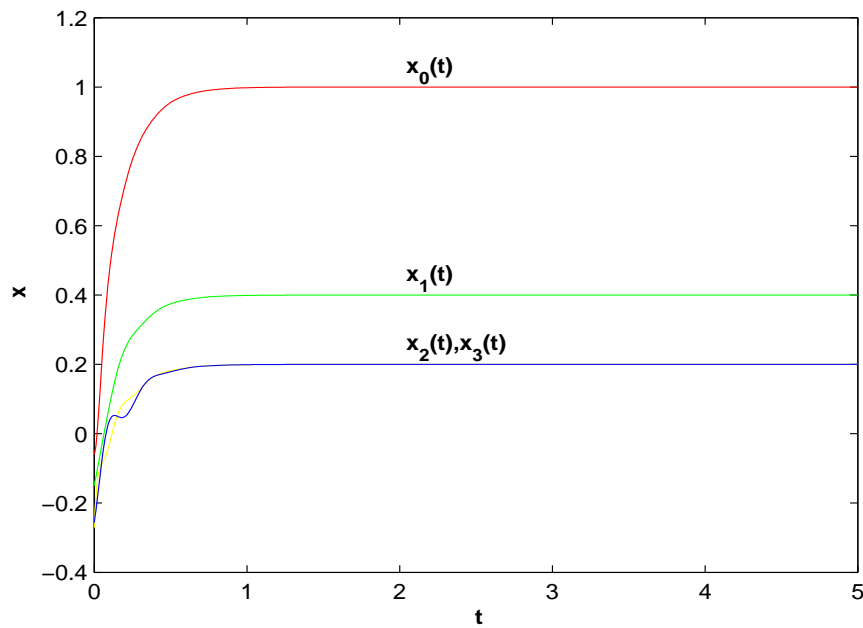


شکل ۵.۳: رفتارهای گذرای زمان گسسته u_{20}, u_{21}, u_{22} و u_{23} از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۱.۳.۳).

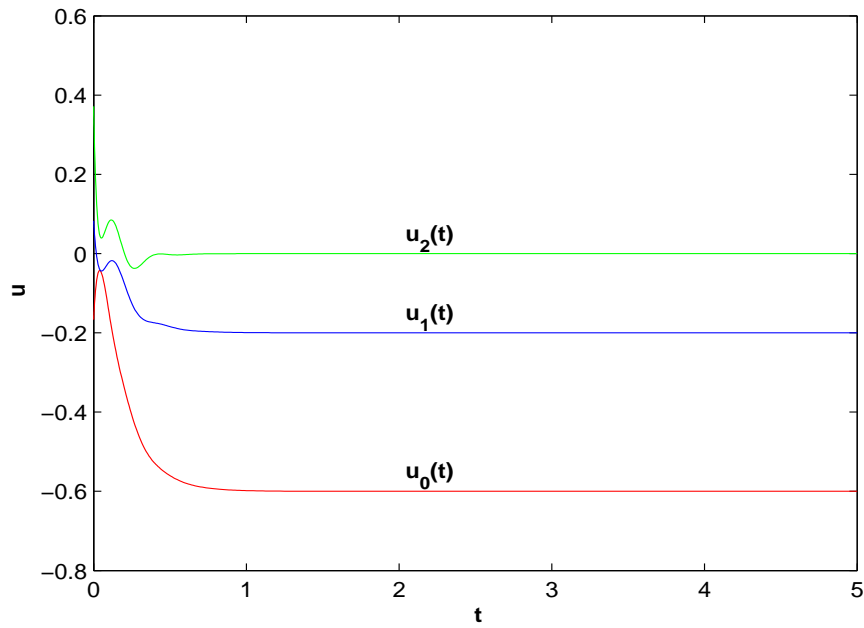
مثال ۲.۳.۳.

$$\text{minimize } \sum_{k=0}^2 [u_k^2 + x_k^2]$$

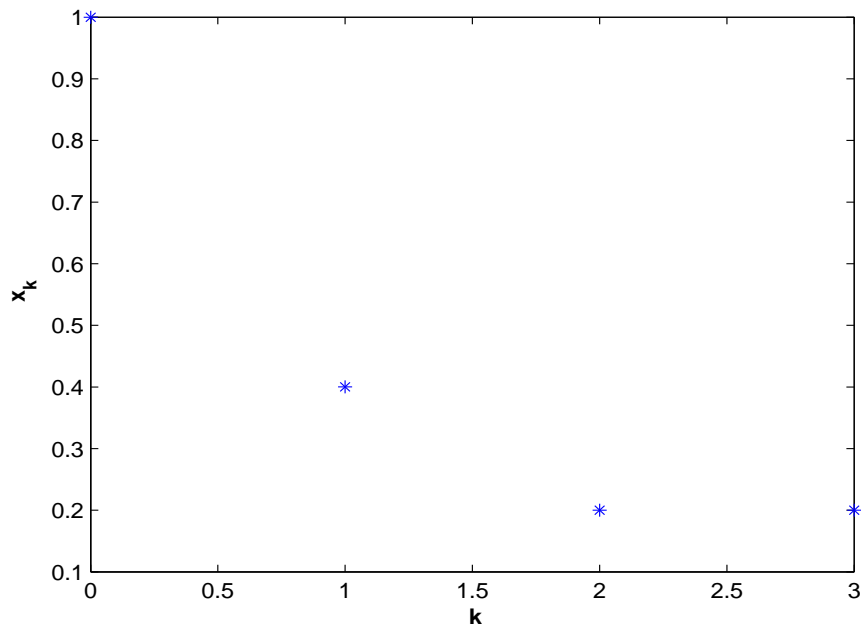
$$\text{subject to } \begin{cases} x_{k+1} = x_k + u_k, & k = 0, 1, 2, \\ x_0 = 1. \end{cases}$$



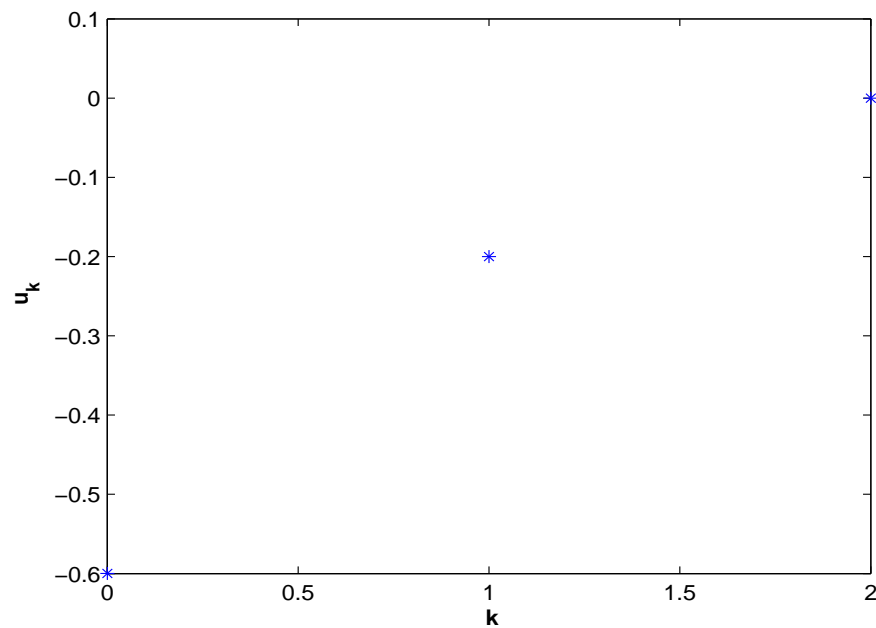
شکل ۶.۳: رفتارهای گذرای x_0 ، x_1 ، x_2 و x_3 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۲.۳.۳).



شکل ۷.۳: رفتارهای گذرای u_2 و u_1 و u_0 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۲.۳.۳).



شکل ۸.۳: رفتارهای گذرای زمان گسسته x_3 و x_2 ، x_1 ، x_0 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۲.۳.۳).

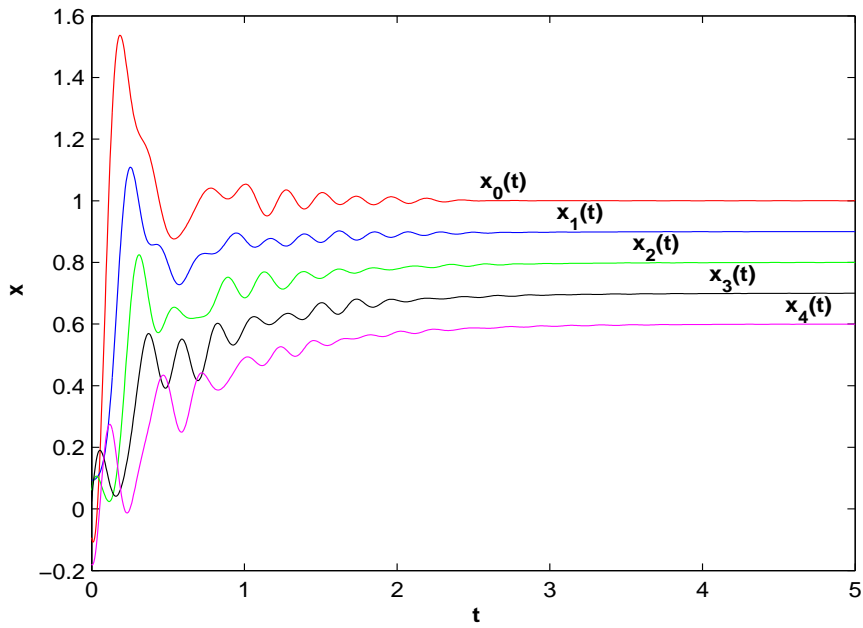


شکل ۹.۳: رفتارهای گذرای زمان گسسته u_0, u_1, u_2 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۲.۳.۳).

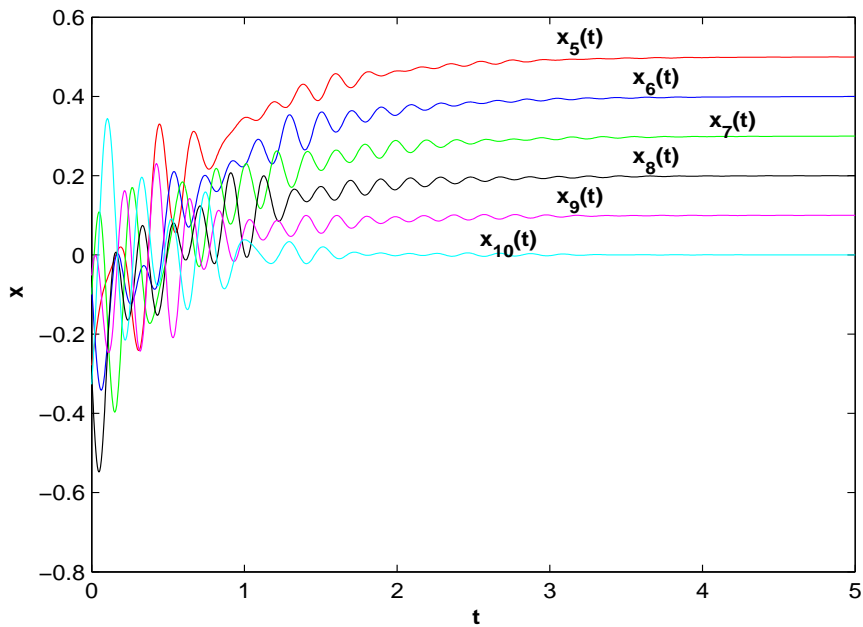
مثال ۳.۳.۳.

$$\text{minimize } \frac{1}{2} \sum_{k=0}^9 u_k^2$$

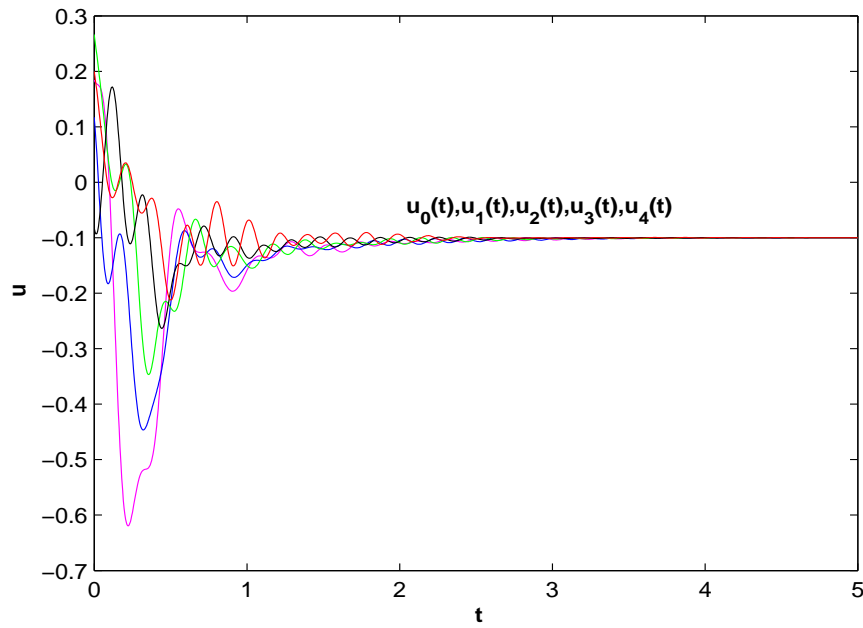
$$\text{subject to } \begin{cases} x_{k+1} = x_k + u_k, & k = 0, \dots, 9, \\ x_0 = 1, & x_{10} = 0. \end{cases}$$



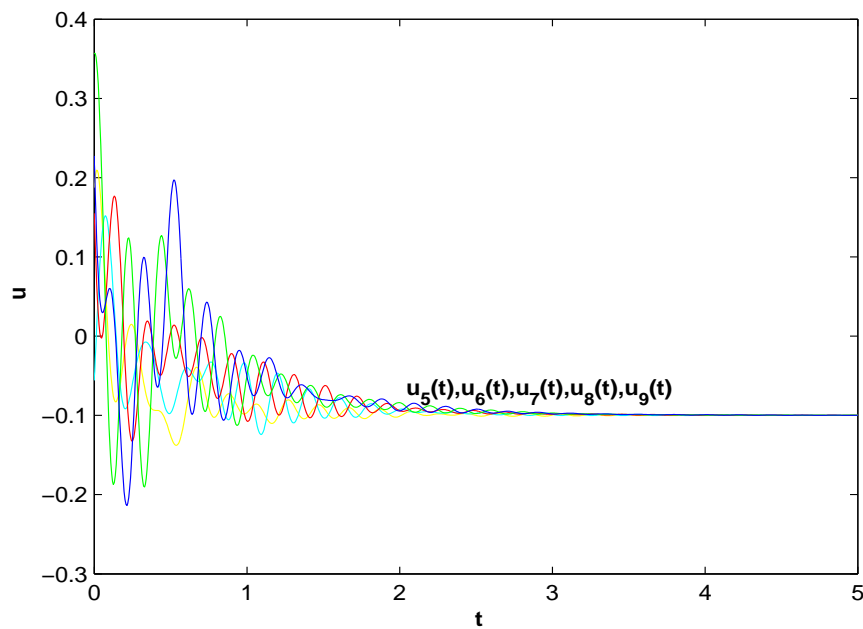
شکل ۱۰.۳: رفتارهای گذرای x_0, x_1, x_2, x_3, x_4 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۳.۳).



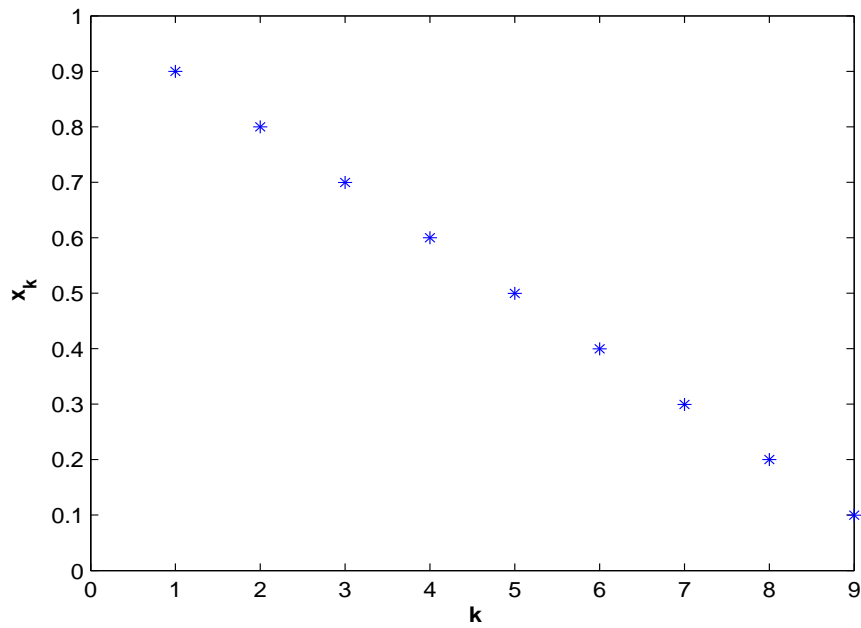
شکل ۱۱.۳: رفتارهای گذرای $x_5, x_6, x_7, x_8, x_9, x_{10}$ از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۳.۳).



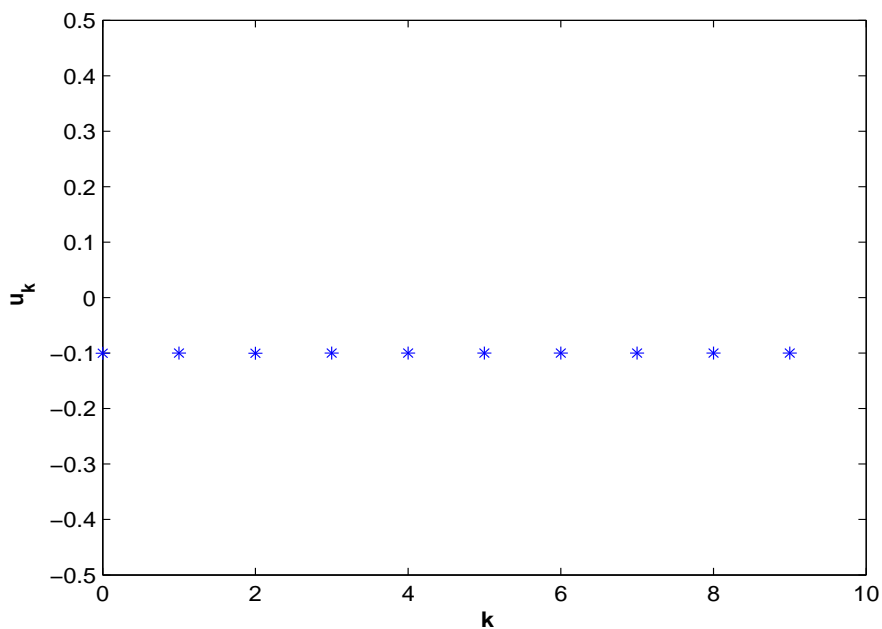
شکل ۱۲.۳: رفتارهای گذرای u_0, u_1, u_2, u_3, u_4 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۳.۳).



شکل ۱۳.۳: رفتارهای گذرای u_5, u_6, u_7, u_8, u_9 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۳.۳).



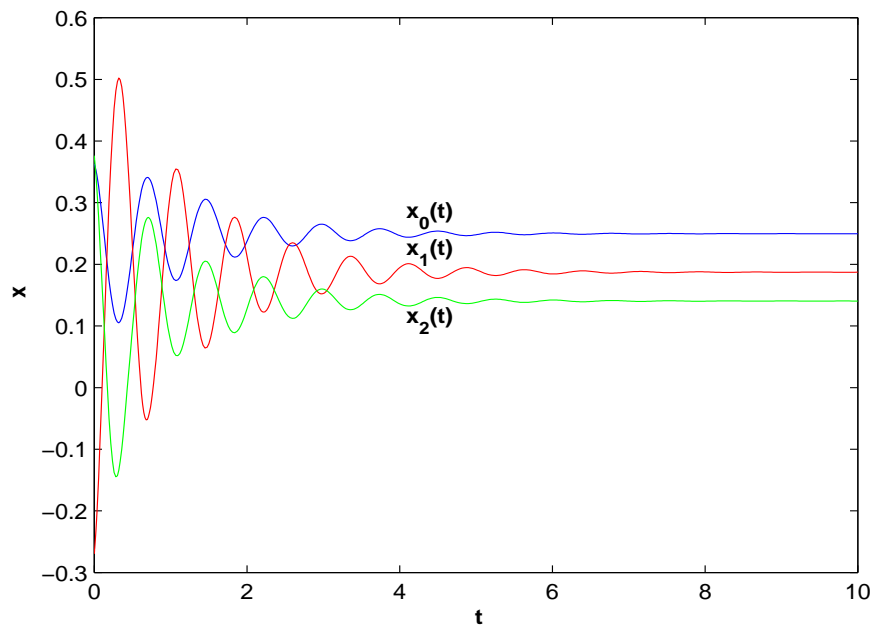
شکل ۱۴.۳: رفتارهای گذرای زمان گسسته x_1, \dots, x_9 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۲.۳).



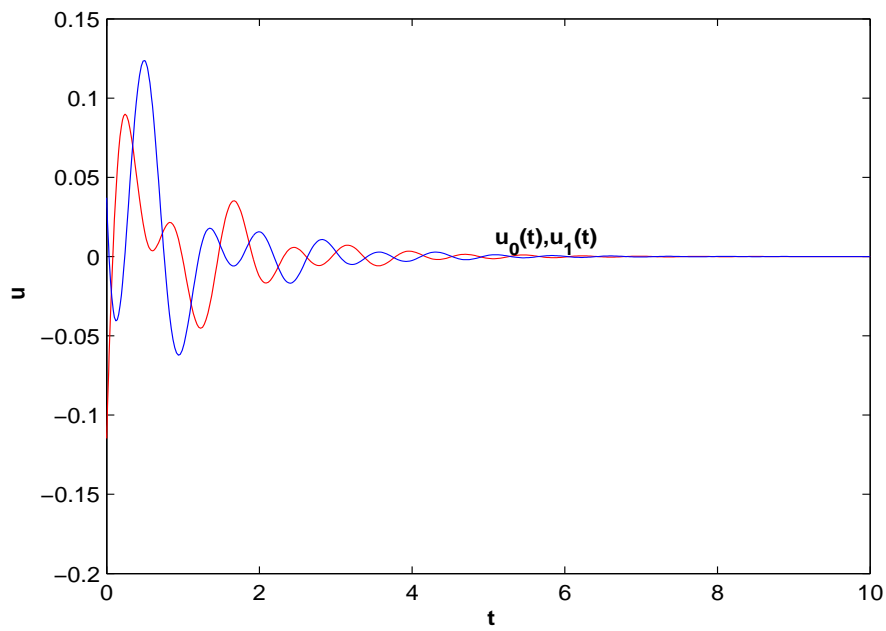
شکل ۱۵.۳: رفتارهای گذرای زمان گسسته u_0, \dots, u_9 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۳.۲.۳).

مثال ۴.۳.۳.

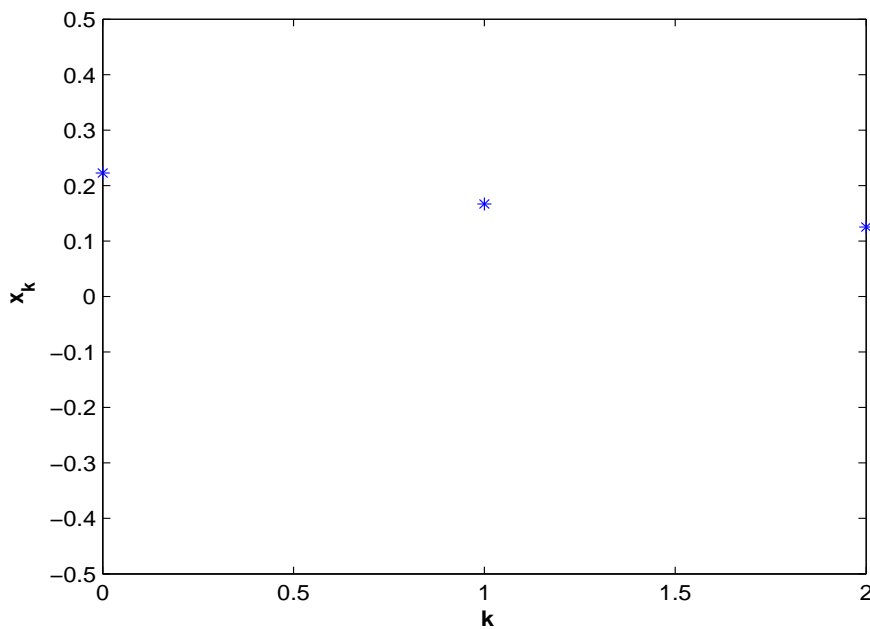
$$\begin{aligned} & \text{minimize} \quad \sum_{k=0}^1 u_k^2 \\ & \text{subject to} \quad \begin{cases} x_{k+1} = 0.75x_k + u_k, & k = 0, 1, \\ 0 \leq x_2 \leq 2, \\ 0 \leq x_k \leq 6, \\ -1 \leq u_k \leq 1. \end{cases} \end{aligned}$$



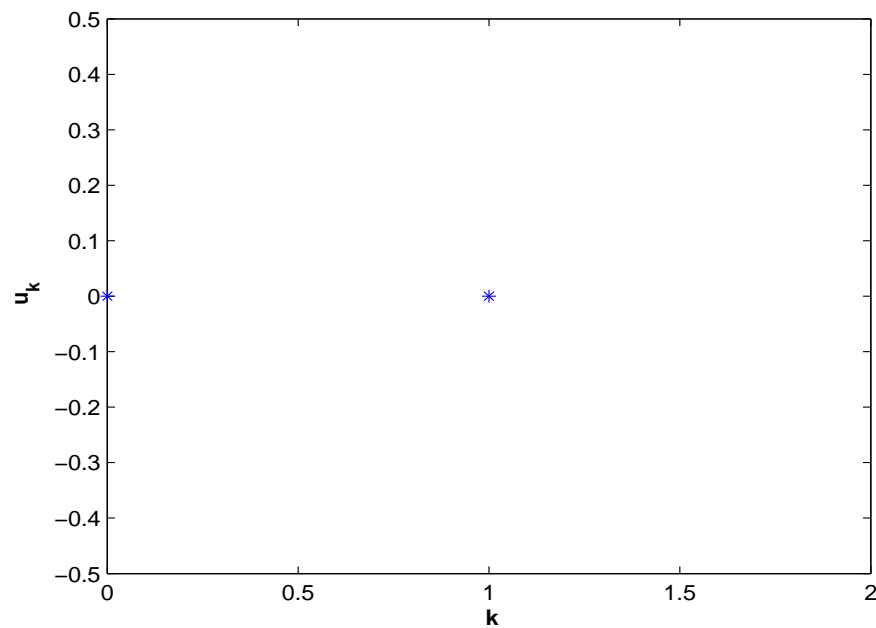
شکل ۱۶.۳: رفتارهای گذرای x_0 ، x_1 و x_2 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۴.۳.۳).



شکل ۱۷.۳: رفتارهای گذرای زمان گسسته u_0 و u_1 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۴.۳.۳).



شکل ۱۸.۳: رفتارهای گذرای زمان گسسته x_0 ، x_1 و x_2 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۴.۳.۳).

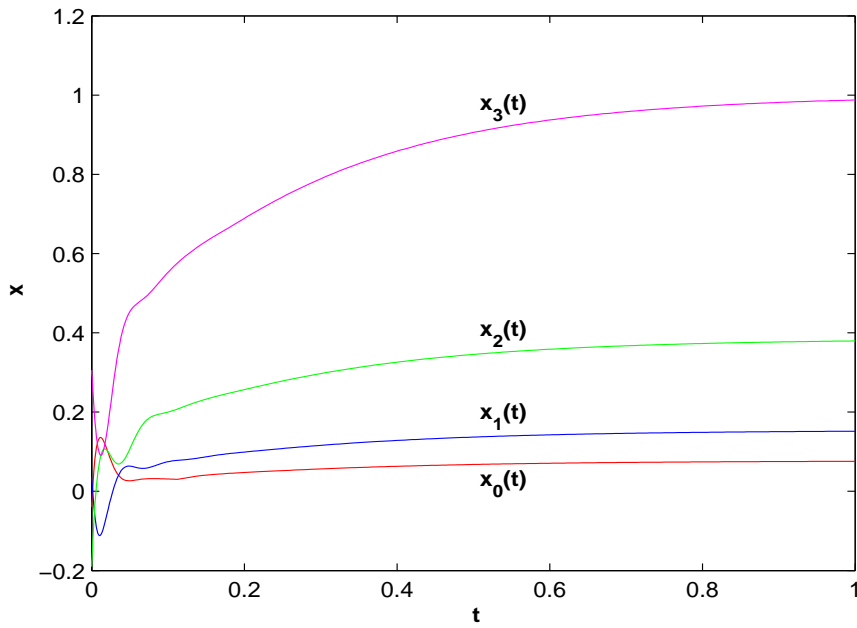


شکل ۱۹.۳: رفتارهای گذرای زمان گسسته u_0 و u_1 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۴.۳.۳).

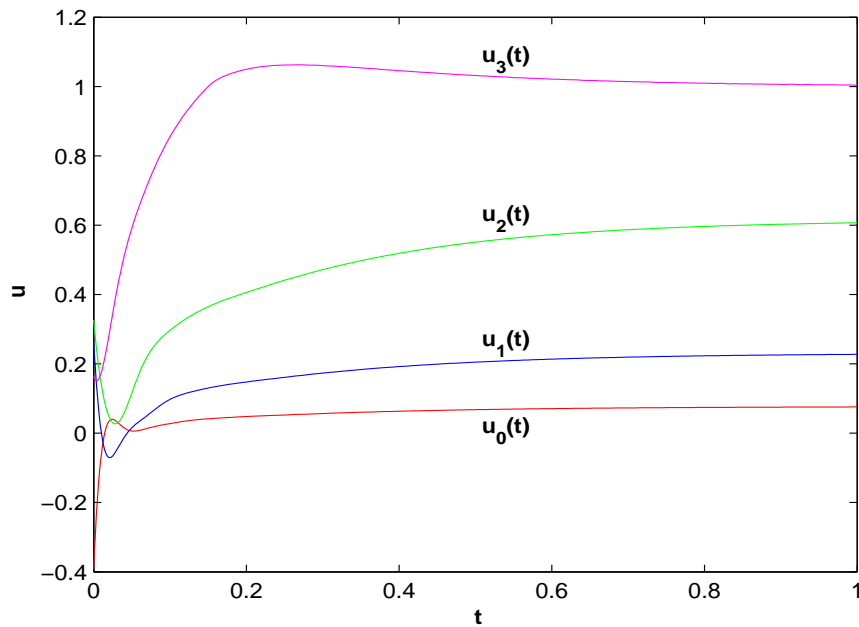
مثال ۵.۳.۳

$$\text{minimize } \left(\sum_{k=0}^3 [x_k^2 + u_k^2] + 2(x_4 - 1)^2 \right)$$

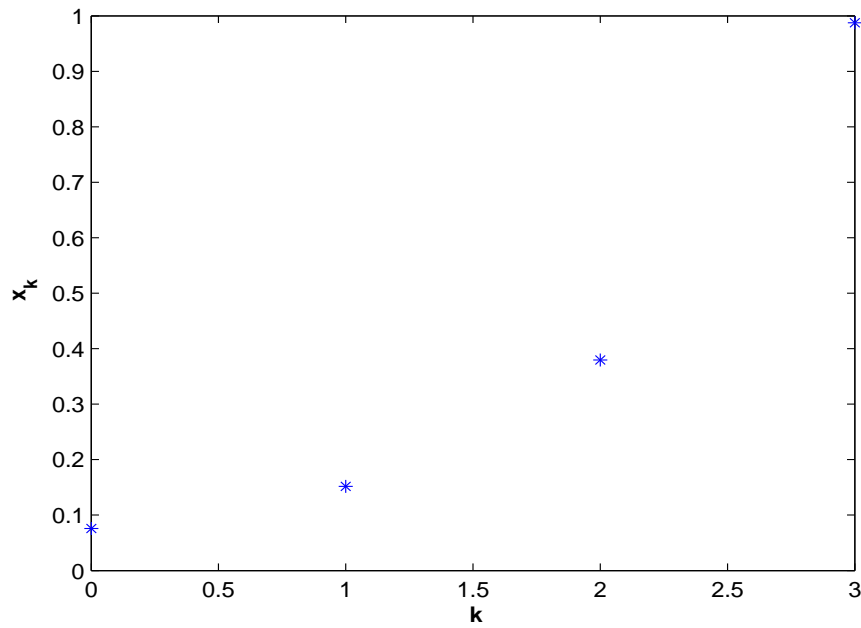
$$\text{subject to } \begin{cases} x_{k+1} = x_k + u_k, \\ 0 \leq x_k \leq 3, \quad -1 \leq u_k \leq 1, \quad x_4 = 2, \\ k = 0, 1, 2, 3. \end{cases}$$



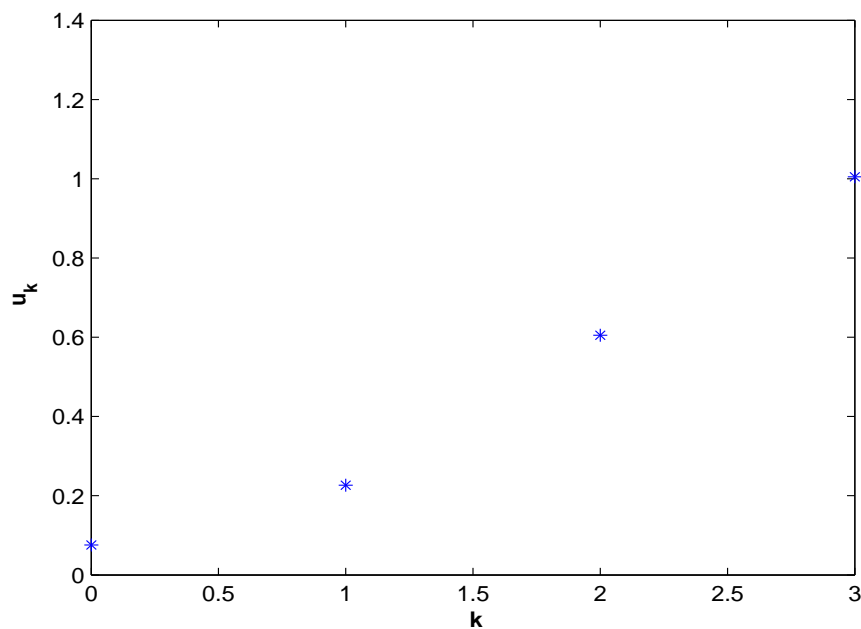
شکل ۲۰.۳: رفتارهای گذرای x_0, x_1, x_2 و x_3 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۵.۳.۳).



شکل ۲۱.۳: رفتارهای گذرای u_0, u_1, u_2 و u_3 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۵.۳.۳).



شکل ۲۲.۳: رفتارهای گذرای زمان گسسته x_0, x_1, x_2, x_3 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۵.۳.۳).



شکل ۲۳.۳: رفتارهای گذرای زمان گسسته u_0, u_1, u_2, u_3 از شبکه عصبی (۵.۳) و (۶.۳) در مثال (۲.۳.۳).

نتایج و پیشنهادهایی برای کارهای آینده

در این پایان‌نامه ابتدا مروری بر شبکه‌های عصبی داشته و با استفاده از یک مدل شبکه عصبی کارا به حل مسائل برنامه‌ریزی غیرخطی با محدودیت‌های هیبریدی و کنترل بهینه سیستم‌های خطی زمان گسسته پرداختیم. همچنین پایداری و همگرایی سراسری به جواب بهینه را مورد بررسی قرار دادیم. از جمله مزیت‌های مدل ارائه شده می‌توان به موارد زیر اشاره کرد:

- حل مسأله رگرسیون
- حل مسأله ربات چند انگشتی
- حل مسأله پردازش تصویر
- حل مسائل بهینه‌سازی سبد سهام
- استفاده از مدل پیشنهادی برای حل مسأله طبقه بندی
- تعمیم مدل پیشنهادی برای داده‌های رگرسیون با ابعاد بالاتر
- حل مسأله رگرسیون ریدج^۵ با مدل ارائه شده
- مدل ذکر شده همگرایی تضمین شده دارد
- این مدل نیاز به هیچ‌گونه پارامتر جریمه ندارد
- با وجود سادگی مدل، در رده بسیاری از مسائل بهینه‌سازی محدب قابل استفاده است
- این مدل هیچگونه وابستگی به نقطه شروع نداشته و این نقطه می‌تواند خارج از ناحیه شدنی و نقطه‌ای تصادفی باشد
- این مدل می‌تواند قادر به حل مسائل وابسته به زمان و مسائل با ابعاد بزرگ باشد

^۵Ridge regression

پیوست آ

کد مثال‌های فصل ۲

آ.۱ مثال (۱.۵.۲)

```
w=.4
```

```
a = unifrnd (-w , w , [1,6]);  
[t,x]=ode45('s100',[0 2],a);  
a=unifrnd(-w,w,[1,6]);  
plot(t,x(:,1),'r');  
hold on;  
plot(t,x(:,2),'b');  
hold on;  
  
text(1,0.36,'\bf x_1(t)','FontSize',10);  
text(1,0.3,'\bf x_2(t)','FontSize',10);  
  
xlabel('\bf t')  
ylabel('\bf x')
```

٢.آ. مثال (٢.٥.٢)

w=.4

```
a = unifrnd (-w , w , [1,9]);
[t,x]=ode45('t100',[0 2],a);
a=unifrnd(-w,w,[1,9]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;

text(1,0.39,'\bf x_1(t)','FontSize',10);
text(1,0.82,'\bf x_2(t)','FontSize',10);

xlabel('\bf t')
ylabel('\bf x')
```

٣.آ. مثال (٣.٥.٢)

w=.3

```
a = unifrnd (-w , w , [1,14]);
[t,x]=ode45('d20',[0 3],a);
a=unifrnd(-w,w,[1,14]);
plot(t,x(:,1),'r');
```

```

hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'m');
hold on;

text(1.5,0.84,'\bf x_1(t)','FontSize',10);
text(1.5,0.16,'\bf x_2(t)','FontSize',10);
text(1.5,1.24,'\bf x_3(t)','FontSize',10);
text(1.5,-0.32,'\bf x_4(t)','FontSize',10);

xlabel('\bf t')
ylabel('\bf x')

```

آ. ۴ مثال (۴.۵.۲)

```
w=.4
```

```

a = unifrnd (-w , w , [1,10]);
[t,x]=ode45('e8',[0 6],a);
a=unifrnd(-w,w,[1,10]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'m');

```



```

hold on;

text(2,0.23,'\bf x_1(t)','FontSize',10);
text(2,0.3,'\bf x_2(t)','FontSize',10);
text(2,-0.03,'\bf x_3(t)','FontSize',10);
text(2,0.07,'\bf x_4(t)','FontSize',10);

xlabel('\bf t')
ylabel('\bf x')

```

٥.٥.٢ مثال

```
w=.4
```

```

a = unifrnd (-w , w , [1,14]);
[t,x]=ode45('e9',[0 6],a);
a=unifrnd(-w,w,[1,14]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'m');
hold on;

text(3,0.86,'\bf x_1(t)','FontSize',10);
text(3,0.12,'\bf x_2(t)','FontSize',10);
text(3,1.21,'\bf x_3(t)','FontSize',10);

```

```
text(3,-0.3,'\bf x_4(t)','FontSize',10);
```

```
xlabel('\bf t')
```

```
ylabel('\bf x')
```

آ.۶ مثال (۶.۵.۲)

```
w=.3
```

```
a = unifrnd (-w , w , [1,11]);
```

```
[t,x]=ode45('e10',[0 2],a);
```

```
a=unifrnd(-w,w,[1,11]);
```

```
plot(t,x(:,1),'r');
```

```
hold on;
```

```
plot(t,x(:,2),'b');
```

```
hold on;
```

```
plot(t,x(:,3),'g');
```

```
hold on;
```

```
plot(t,x(:,4),'m');
```

```
hold on;
```

```
text(1,0.87,'\bf x_1(t)','FontSize',10);
```

```
text(1,1.75,'\bf x_2(t)','FontSize',10);
```

```
text(1,0.1,'\bf x_3(t)=x_4(t)','FontSize',10);
```

```
xlabel('\bf t')
```

```
ylabel('\bf x')
```

۷.آ. مثال (۷.۵.۲)

w=.3

```

a = unifrnd (-w , w , [1,9]);
[t,x]=ode45('e12',[0 7],a);
a=unifrnd(-w,w,[1,9]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'m');
hold on;

text(3,0.6,'\bf x_1(t)','FontSize',10);
text(3,-0.34,'\bf x_2(t)','FontSize',10);
text(3,1.55,'\bf x_3(t)','FontSize',10);
text(3,0.16,'\bf x_4(t)','FontSize',10);

xlabel('\bf t')
ylabel('\bf x')
```

۸.آ. مثال (۸.۵.۲)

```

p=0;
while p<=5;
w=.4
```

```
a = unifrnd (-w , w , [1,7]);
[t,x]=ode45('s200',[0 1],a);
a=unifrnd(-w,w,[1,7]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
p=p+1;
end
text(0.5,0.3,'\bf x_1(t)','FontSize',10);
text(0.5,0.73,'\bf x_2(t)','FontSize',10);

xlabel('\bf t')
ylabel('\bf x')
```

پیوست ب

کد مثال‌های فصل ۳

ب.۱ مثال (۱.۳.۳)

w=.4

```
a = unifrnd (-w , w , [1,18]);
[t,x]=ode45('e1',[0 10],a);
a=unifrnd(-w,w,[1,18]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'y');
hold on;
plot(t,x(:,5),'m');
hold on;
text(3,1.06,'\bf x_0(t)','FontSize',10);
text(3,0.28,'\bf x_1(t)','FontSize',10);
text(3,0.1,'\bf x_2(t)','FontSize',10);
```

```
text(3,-0.06,'\bf x_3(t)=x_4(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf x')

[m,n]=size(x)
x=[x(m,1),x(m,2),x(m,3),x(m,4),x(m,5)];
k=[0,1,2,3,4];
plot(k,x,'*');
xlabel('\bf k')
ylabel('\bf x_k')

u=[x(m,6),x(m,7),x(m,8),x(m,9)];
k=[0,1,2,3];
plot(k,u,'*');
xlabel('\bf k')
ylabel('\bf u_{1k}')

u=[x(m,10),x(m,11),x(m,12),x(m,13)];
k=[4,5,6,7];
plot(k,u,'*');
xlabel('\bf k')
ylabel('\bf u_{2k}')

plot(t,x(:,6),'r');
hold on;
plot(t,x(:,7),'b');
hold on;
plot(t,x(:,8),'g');
hold on;
plot(t,x(:,9),'m');
hold on;
plot(t,x(:,10),'r');
hold on;
plot(t,x(:,11),'b');
```

```

hold on;
plot(t,x(:,12),'g');
hold on;
plot(t,x(:,13),'m');
hold on;
text(4,-0.36,'\bf u_{10}(t)=u_{20}(t)', 'FontSize',10);
text(3,-0.12,'\bf u_{11}(t)=u_{21}(t)', 'FontSize',10);
text(4,-0.05,'\bf u_{12}(t)=u_{22}(t)', 'FontSize',10);
text(4,0.03,'\bf u_{13}(t)=u_{23}(t)', 'FontSize',10);
xlabel('\bf t')
ylabel('\bf u')

```

ب.٢. مثال (٢.٣.٣)

w=.4

```

a = unifrnd (-w , w , [1,11]);
[t,x]=ode45('e2',[0 5],a);
a=unifrnd(-w,w,[1,11]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'g');
hold on;
plot(t,x(:,3),'y');
hold on;
plot(t,x(:,4),'b');
hold on;
text(2,1.05,'\bf x_0(t)', 'FontSize',10);
text(2,0.45,'\bf x_1(t)', 'FontSize',10);
text(2,0.25,'\bf x_2(t)=x_3(t)', 'FontSize',10);

```

```

xlabel('\bf t')
ylabel('\bf x')

[m,n]=size(x)
x=[x(m,1),x(m,2),x(m,3),x(m,4)];
k=[0,1,2,3];
plot(k,x,'*');
xlabel('\bf k')
ylabel('\bf x_k')

u=[x(m,5),x(m,6),x(m,7)];
k=[0,1,2];
plot(k,u,'*');
xlabel('\bf k')
ylabel('\bf u_k')

plot(t,x(:,5),'r');
hold on;
plot(t,x(:,6),'b');
hold on;
plot(t,x(:,7),'g');
hold on;
text(2,-0.56,'\bf u_0(t)','FontSize',10);
text(2,-0.16,'\bf u_1(t)','FontSize',10);
text(2,0.04,'\bf u_2(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf u')

```

ب.۳ مثال (۳.۳.۳)

w=.4


```
a = unifrnd (-w , w , [1,33]);
[t,x]=ode45('e3',[0 5],a);
a=unifrnd(-w,w,[1,33]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'k');
hold on;
plot(t,x(:,5),'m');
hold on;
plot(t,x(:,6),'r');
hold on;
plot(t,x(:,7),'b');
hold on;
plot(t,x(:,8),'g');
hold on;
plot(t,x(:,9),'k');
hold on;
plot(t,x(:,10),'m');
hold on;
plot(t,x(:,11),'c');
hold on;
text(2.5,1.05,'\bf x_0(t)','FontSize',10);
text(3,0.95,'\bf x_1(t)','FontSize',10);
text(3.5,0.85,'\bf x_2(t)','FontSize',10);
text(4,0.75,'\bf x_3(t)','FontSize',10);
text(4.5,0.65,'\bf x_4(t)','FontSize',10);
text(3,0.54,'\bf x_5(t)','FontSize',10);
text(3.5,0.44,'\bf x_6(t)','FontSize',10);
```

```

text(4,0.33,'\bf x_7(t)','FontSize',10);
text(3.5,0.24,'\bf x_8(t)','FontSize',10);
text(3,0.14,'\bf x_9(t)','FontSize',10);
text(2.5,0.03,'\bf x_{10}(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf x')

```

```

[m,n]=size(x)
x=[x(m,1),x(m,2),x(m,3),x(m,4),x(m,5),x(m,6),
    x(m,7),x(m,8),x(m,9),x(m,10),x(m,11)];
k=[0,1,2,3,4,5,6,7,8,9,10];
plot(k,x,'*');
axis([0 9 0 1]);
xlabel('\bf k')
ylabel('\bf x_k')

```

```

u=[x(m,12),x(m,13),x(m,14),x(m,15),x(m,16),
    x(m,17),x(m,18),x(m,19),x(m,20),x(m,21)];
k=[0,1,2,3,4,5,6,7,8,9];
plot(k,u,'*');
axis([0 10 -0.5 0.5]);
xlabel('\bf k')
ylabel('\bf u_k')

```

```

plot(t,x(:,12),'m');
hold on;
plot(t,x(:,13),'b');
hold on;
plot(t,x(:,14),'g');
hold on;
plot(t,x(:,15),'k');
hold on;
plot(t,x(:,16),'r');
hold on;

```

```

plot(t,x(:,17),'y');
hold on;
plot(t,x(:,18),'c');
hold on;
plot(t,x(:,19),'r');
hold on;
plot(t,x(:,20),'g');
hold on;
plot(t,x(:,21),'b');
hold on;
text(2,-0.05,'\bf u_0(t)=u_1(t)=u_2(t)=u_3(t)=u_4(t)', 'FontSize',10);
text(2,-0.07,'\bf u_5(t)=u_6(t)=u_7(t)=u_8(t)=u_9(t)', 'FontSize',10);
xlabel('\bf t')
ylabel('\bf u')

```

ب.٤. مثال (٤.٣.٣)

w=.4

```

a = unifrnd (-w , w , [1,17]);
[t,x]=ode45('e4',[0 10],a);
a=unifrnd(-w,w,[1,17]);
plot(t,x(:,1),'b');
hold on;
plot(t,x(:,2),'r');
hold on;
plot(t,x(:,3),'g');
hold on;
text(4,0.28,'\bf x_0(t)', 'FontSize',10);
text(4,0.22,'\bf x_1(t)', 'FontSize',10);

```

```

text(4,0.11,'\bf x_2(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf x')

[m,n]=size(x)
x=[x(m,1),x(m,2),x(m,3)];
k=[0,1,2];
plot(k,x,'*');
axis([0 1 -0.5 0.5]);
xlabel('\bf k')
ylabel('\bf x_k')

u=[x(m,4),x(m,5)];
k=[0,1];
plot(k,u,'*');
axis([0 2 -0.5 0.5]);
xlabel('\bf k')
ylabel('\bf u_k')

plot(t,x(:,4),'r');
hold on;
plot(t,x(:,5),'b');
hold on;
text(5,0.01,'\bf u_0(t)=u_1(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf u')

```

ب.۵ مثال (۵.۳.۳)

w=.4

```

a = unifrnd (-w , w , [1,30]);
[t,x]=ode45('e11',[0 1],a);
a=unifrnd(-w,w,[1,30]);
plot(t,x(:,1),'r');
hold on;
plot(t,x(:,2),'b');
hold on;
plot(t,x(:,3),'g');
hold on;
plot(t,x(:,4),'m');
hold on;
text(0.5,0.02,'\bf x_1(t)','FontSize',10);
text(0.5,0.19,'\bf x_2(t)','FontSize',10);
text(0.5,0.4,'\bf x_3(t)','FontSize',10);
text(0.5,0.97,'\bf x_4(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf x')

[m,n]=size(x)
x=[x(m,1),x(m,2),x(m,3),x(m,4)];
k=[1,2,3,4];
plot(k,x,'*');
xlabel('\bf k')
ylabel('\bf x(k)')

u=[x(m,6),x(m,7),x(m,8),x(m,9)];
k=[1,2,3,4];
plot(k,u,'*');
xlabel('\bf k')
ylabel('\bf u(k)')

plot(t,x(:,6),'r');
hold on;

```

```
plot(t,x(:,7),'b');
hold on;
plot(t,x(:,8),'g');
hold on;
plot(t,x(:,9),'m');
hold on;
text(0.5,0.01,'\bf u_1(t)','FontSize',10);
text(0.5,0.26,'\bf u_2(t)','FontSize',10);
text(0.5,0.62,'\bf u_3(t)','FontSize',10);
text(0.5,1.08,'\bf u_4(t)','FontSize',10);
xlabel('\bf t')
ylabel('\bf u')
```

مراجع

- [۱] ضمیری، آذر. شبکه‌های عصبی مصنوعی و بهینه‌سازی. پایان‌نامه کارشناسی ارشد، دانشکده علوم ریاضی، دانشگاه فردوسی مشهد، آبان ۱۳۸۷.
- [۲] قماش‌لی لنگرودی، عباس. حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی تصویر. پایان‌نامه کارشناسی ارشد، دانشکده علوم پایه، دانشگاه تربیت معلم سبزوار، آبان ۱۳۸۳.
- [۳] مقدس، محمد، عفتی، سهراب، و اسحاق‌نژاد، محمد. یک شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی نامحدوب. در ششمین کنفرانس بین‌المللی انجمن ایرانی تحقیق در عملیات، تهران، ایران، اردیبهشت ۱۳۹۲.
- [4] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
- [5] A. Bemporad, N. Giorgetti, I.V. Kolmanovsky, and D. Hrovat. Hybrid modeling and control of a direct injection stratified charge engine. In *Symposium on Advanced Automotive Technologies*, ASME International Mechanical Engineering Congress and Exposition, New Orleans, Louisiana, November 2002.
- [6] A. Bemporad, P. Borodani, and M. Mannelli. Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, HSCC 2003, volume 2623 of *Lecture Notes in Computer Science*, pages 81–96. Springer Verlag, 2003.
- [7] A. Bouzerdoum and T. R. Pattison. Neural network for quadratic optimization with bound constraints. *IEEE Trans. Neural Netw.*, 4(2):293–304, 1993.
- [8] A. Malek, H.G. Oskoei. Primal-dual solution for the linear programming problems using neural networks, *Appl. Math. Comput.* 169 (2005) 451–471.
- [9] A. R. Nazemi. Solving general convex nonlinear optimization problems by an efficient neurodynamic model, *engineering Applications of Artificial Intelligence* 26, 685-696, (2013).
- [10] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J.L. Huertas and E. Sanchez Sinencio. Nonlinear switched capacitor ‘neural’ networks for optimization problems. *IEEE Trans. Circuits Syst.*, 37(3):384–398, 1990.

-
- [11] B.S. He, L. Z. Liao. Improvements of some projection methods for monotone nonlinear variational inequalities, *J. Optimiz. Theory Applicat.* 112 (1) (2002) 111–128.
- [12] B. Widrow. Generalization and information storage in networks of adaline ‘neurons’ in *Self Organizing Systems*, pp. 435–461. Spartan Books, 1959.
- [13] B. Widrow and M. E. Hoff. Adapting switching circuits. in 1960 ire wescon Convention Record, no. 4, pp. 96–104, New York, NY, 1960. ire.
- [14] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proc. IEEE*, 78(9):1415–1442, 1990.
- [15] B. Widrow and R. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. *Computer*, 21(3):25–39, 1988.
- [16] C.J. Tomlin, J. Lygeros, and S.S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceeding of IEEE*, 88:949–970, July 2000.
- [17] C.Y. Maa and M. A. Schanblatt, A two-phase optimization neural network. *IEEE Trans. Neural Netw.*, 3(6):1003–1009, 1992.
- [18] D. Mignone. Control and Estimation of Hybrid Systems via Mathematical Optimization. Dr.sc. tech. thesis, Automatic Control Laboratory - ETH, Zurich, 2002. <http://control.ee.ethz.ch>.
- [19] D. Tank and J. J. Hopfield. Simple ‘neural’ optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. Circuits Syst.*, 33(5):533–541, 1986.
- [20] D.Q. Mayne. Constrained optimal control. In *European Control Conference, Plenary Lecture*, Porto, Portugal, September 2001.
- [21] D.R. Han, H.K. Lo. New alternating direction method for a class of nonlinear variational inequality problems, *J. Optimiz. Theory Applicat.* 112 (3) (2002) 549–560.
- [22] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, Inc., New York, 1949.
- [23] D. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 3rd ed. , 2008.
- [24] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning representations by backpropagating errors. *Nature*, 323:533–536, 1986.
- [25] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automatic Control*, 26(2):346–358, April 1981.

- [26] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. A hybrid approach to traction control. In A. Sangiovanni-Vincentelli and M.D. Di Benedetto, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 162–174. Springer Verlag, 2001.
- [27] F. Borrelli, M. Baoti, A. Bemporad, M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* 41: 1709 – 1721, 2005.
- [28] F.D. Torrisi and A. Bemporad. HYSDEL — a tool for generating computational hybrid models. *IEEE Transactions on Control Systems Technology*, 2003, to appear.
- [29] F. Giannessi, and A. Maugeri. *Variational Analysis And Applications. Nonconvex Optimization and Its Applications*. Springer, 2005.
- [30] F. Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books, 1962.
- [31] F. Rosenblatt. *Neurocomputing: foundations of research*. chap. The perception: a probabilistic model for information storage and organization in the brain, pp. 89–114. MIT Press, Cambridge, MA, USA, 1988.
- [32] G. Ferrari-Trecate, E. Gallestey, A. Stothert, P. Letizia, M. Spedicato, M. Morari, and M. An toine. Modeling and control of co-generation power plants: A hybrid system approach. In C. J. Tomlin and M. R. Greenstreet, editors, *Proc. 5th International Workshop on Hybrid Systems: Computation and Control*, *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
- [33] H. K. Khalil. *Nonlinear Systems*. Prentice Hall PTR, 3rd ed. , 2002.
- [34] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.*, 79(8):2554–2558, 1982.
- [35] J. J. Hopfield and D. W. Tank. “neural” computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.
- [36] J. Lygeros, D.N. Godbole, and S. Sastry. A game theoretic approach to hybrid system design. In R. Alur and T. Henzinger, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, 1996.
- [37] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.
- [38] J. Wang, Q. Hu, D. Jiang. A lagrangian neural network for kinematics control of redundant robot manipulators, *IEEE Trans. Neural Network* 10 (5) (1999) 1123–1132.

- [39] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2000.
- [40] J. Zabczyk. *Mathematical control theory: an introduction*. Systems & control. Birkhäuser, 1992.
- [41] L. Perko. *Differential Equations and Dynamical Systems*. Springer, 3rd ed. , 2001.
- [42] M. S.,Bazaraa,H. D., Sherali and C. M, Shetty. *Nonlinear Programming*. John Wiley & Sons, Inc., 3rd ed. , 2005.
- [43] M. P. Kennedy and L. O. Chua. Neural networks for nonlinear programming. *IEEE Trans. Circuits Syst.*, 35(5):554–562, 1988.
- [44] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [45] M. Baotic, M. Vasak, M. Morari, and N. Peric. Hybrid theory based optimal control of electronic throttle. In *Proc. American Control Conf.*, Denver, Colorado, USA, June 2003.
- [46] M. L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. Mit Press, 1972.
- [47] M. Moghaddas and S. Effati. A novel recurrent neural network based on ncp function for solving convex nonlinear optimization problems. in *The 5 th Iranian Conference on Applied Mathematics*, Hamadan, Iran, September 2013. Bu-Ali Sina University.
- [48] M. Moghaddas and M. Eshaghnejad. A recurrent neural network for solving non-convex non linear optimization problem. in *The 44 th Annual Iranian Mathematics Conference*, Mashhad, Iran, August 2013. Ferdowsi University of Mashhad.
- [49] M.S. Bazaraa, C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons Inc., New York: Wiley, 1990.
- [50] M.S. Branicky. *Studies in hybrid systems: modeling, analysis, and control*. PhDthesis,LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA, 1995.
- [51] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory.*IEEE Trans. Automatic Control*, 43(1):31–45, 1998.

- [52] M.S. Branicky and G. Zhang. Solving hybrid control problems: Level sets and behavioral program ming. In Proc. American Control Conf., Chicago, Illinois USA, June 2000.
- [53] M. Fukushima, Equivalent differentiable optimization problems and decent methods for asymmetric variational inequality problems, Math. Program. 53 (1992) 99–110.
- [54] M. Yashtini, A. Malek, Solving complementarity and variational inequalities problems using neural networks, Appl. Math. Comput. 190 (1) (2007) 216–230.
- [55] P.J. Antsaklis. A brief introduction to the theory and applications of hybrid systems. Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications, 88(7):879–886, July 2000.
- [56] P. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
- [57] P.T. Harker, J.S. Pang, Finite-dimensional variational inequality and nonlinear Complementarity problems: A survey of theory algorithms and applications, Math. Program. 48B (1990) 161–220.
- [58] Q. Tao, J.D. Cao, M.S. Xue, H. Qiao, A high performance neural network for solving nonlinear programming problems with hybrid constraints, Phys. Lett. A 288 (2) (2001) 88–94.
- [59] R. K. Miller, A. N. Michel. Ordinary Differential Equations, Academic Press, New York, 1982.
- [60] R. M. obus, M. Baotic, and M. Morari. Multi-object adaptive cruise control. In O. Maler and A. Pnueli, editors, Hybrid Systems: Computation and Control, HSCC 2003, volume 2623 of Lecture Notes in Computer Science, pages 359–374. Springer Verlag, 2003.
- [61] S. Effati and M. A. Baymani. new nonlinear neural network for solving convex nonlinear programming problems. Appl. Math. and Comput., 168(2):1370 – 1379, 2005.
- [62] S. Effati, A. Ghomashi and M. Abbasi. A novel recurrent neural network for solving mlcps and its application to linear and quadratic programming. Asia-Pacific Journal of Operational Research, 28(4):523–541, November 2011.
- [63] S. Effati, A. Ghomashi, and A. R. Nazemi. Application of projection neural network in solving convex programming problems. Appl. Math. Comput., 188(2):1103 – 1114, 2007.
- [64] S. Zhang and A.G. Constantinides. Lagrange programming neural networks. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., 39(7):441–452, 1992.

-
- [65] W. McCulloch, and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943.
- [66] X. Xu and P. J. Antsaklis. Results and perspectives on computational methods for optimal control of switched systems. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control, HSCC 2003*, volume 2623 of *Lecture Notes in Computer Science*, pages 540–556. Springer Verlag, 2003.
- [67] Y. Xia, G. Feng and J. Wang. A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. *IEEE Trans. Neural Netw.*, 19(8):1340–1353, 2008.
- [68] Y. Xia and J. Wang. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 51(7):1385–1394, 2004.
- [69] Y. Xia, and J. Wang. A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Trans. Neural Netw.*, 16(2):379–386, 2005.
- [70] Y.S. Xia, J. Wang. A general methodology for designing globally convergent optimization neural networks, *IEEE Trans. Neural Network* 9 (6), 1331–1343, 1998.
- [71] Y.S. Xia. A new neural network for solving linear and quadratic programming problems, *IEEE Trans. Neural Networks* 7 (6), 1544–1547, 1996.
- [72] Y. Xia, J. Wang. A Recurrent neural network for solving nonlinear convex programs subject to linear constraints, *IEEE Trans. Neural Network* 16, 379–386, 2005.
- [73] Y. Xia, H. Leung and J. Wang. A projection neural network and its application to constrained optimization problems. *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl*, 49(4):447–458, 2002.

واژه‌نامه فارسی به انگلیسی

LaSalle Principle of Invariance	اصل تغییرناپذیری لسال
Axon	آکسون
Convex Optimization	بهینه‌سازی محدب
Nonconvex Optimization	بهینه‌سازی نامحدب
Postsynaptic Terminal	پایانه بعد از سیناپسی
Presynaptic Terminal	پایانه قبل از سیناپسی
Globally Asymptotically Stable	پایدار مجانبی سراسری
Stability in the Sense of Lyapunov	پایداری به مفهوم لیاپانوف
Globally Exponential Stability	پایداری نمایی سراسری
Back Propagation	پس انتشار
Feed Forward	پیش خورد
Lipschitz Continuous	پیوسته لیپ شیتز
Locally Lipschitz Continuous	پیوسته لیپ شیتز محلی
Activation Function	تابع فعال‌سازی
Firing	تحریک شدن
Feasible Solution	جواب شدنی
Dendrite	دندریت
Local Duality	دوگانی موضعی
Discrete time	زمان گسسته
Hybrid systems	سیستم‌های هیبریدی
Switched-Capacitor	سوئیچ-خازن
Synapse	سیناپس
Neural Networks	شبکه‌های عصبی
Feedback Neural Network	شبکه‌های عصبی بازگشتی
Artificial Neural Networks	شبکه‌های عصبی مصنوعی

Second-Order Sufficient Conditions	شرایط کافی مرتبه دوم
Synaptic Cleft	شکاف سیناپسی
Lagrangian Multiplier	ضرب‌گر لاگرانژ
Projection Operator	عملگر تصویر
Brouwer's Fixed Point Theorem	قضیه نقطه ثابت بروئر
Hessian Matrix	ماتریس هسین
Dual Variable	متغیر دوگان
Invariant Set	مجموعه تغییرناپذیر
Level Set	مجموعه سطح
Lagrangian Dual Problem	مسئله دوگان لاگرانژ
Linear Complementarity Problem	مسئله مکمل خطی
Nonlinear Complementarity Problem	مسئله مکمل غیرخطی
Positive Definite	معین مثبت
Strongly Positive Definite	معین مثبت قوی
Variational Inequalities	نامساوی‌های وردشی
Neuron	نرون
Equilibrium Point	نقطه تعادل
Isolated Equilibrium Point	نقطه تعادل تنها
Regular Point	نقطه منظم
Strict Local Minimum Point	نقطه مینیمم محلی اکید
Strictly Monotone Map	نگاشت اکیداً یکنوا
Strongly Monotone Map	نگاشت قویاً یکنوا
Monotone Map	نگاشت یکنوا
Positive Semidefinite	نیمه معین مثبت
Globally Convergent	همگرای سراسری

واژه‌نامه انگلیسی به فارسی

Activation Function	تابع فعال‌سازی
Artificial Neural Networks	شبکه‌های عصبی مصنوعی
Axon	آکسون
Back Propagation	پس انتشار
Brouwer's Fixed Point Theorem	قضیه نقطه ثابت بروئر
Convex Optimization	بهینه‌سازی محدب
Dendrite	دندریت
Discrete time	زمان گسسته
Dual Variable	متغیر دوگان
Equilibrium Point	نقطه تعادل
Feasible Solution	جواب شدنی
Feedback Neural Network	شبکه‌های عصبی بازگشتی
Feed Forward	پیش‌خورد
Firing	تحریک شدن
Globally Asymptotically Stable	پایدار مجانبی سراسری
Globally Convergent	همگرای سراسری
Globally Exponential Stability	پایداری نمایی سراسری
Hessian Matrix	ماتریس هسین
Hybrid systems	سیستم‌های هیبریدی
Invariant Set	مجموعه تغییرناپذیر
Isolated Equilibrium Point	نقطه تعادل تنها
Lagrangian Dual Problem	مسأله دوگان لاگرانژ
Lagrangian Multiplier	ضرب‌گر لاگرانژ
LaSalle Principle of Invariance	اصل تغییرناپذیری لاسال
Level Set	مجموعه سطح

Linear Complementarity Problem	مسأله مکمل خطی
Lipschitz Continuous	پیوسته لیپ شیتز
Local Duality	دوگانی موضعی
Locally Lipschitz Continuous	پیوسته لیپ شیتز محلی
Monotone Map	نگاشت یکنوا
Neural Networks	شبکه‌های عصبی
Neuron	نرون
Nonconvex Optimization	بهینه‌سازی نامحدب
Nonlinear Complementarity Problem	مسأله مکمل غیرخطی
Positive Definite	معین مثبت
Positive Semidefinite	نیمه معین مثبت
Postsynaptic Terminal	پایانه بعد از سیناپسی
Presynaptic Terminal	پایانه قبل از سیناپسی
Projection Operator	عملگر تصویر
Regular Point	نقطه منظم
Second-Order Sufficient Conditions	شرایط کافی مرتبه دوم
Stability in the Sense of Lyapunov	پایداری به مفهوم لیاپانوف
Strict Local Minimum Point	نقطه مینیمم محلی اکید
Strictly Monotone Map	نگاشت اکیداً یکنوا
Strongly Monotone Map	نگاشت قویاً یکنوا
Strongly Positive Definite	معین مثبت قوی
Switched-Capacitor	سوئیچ-خازن
Synapse	سیناپس
Synaptic Cleft	شکاف سیناپسی
Variational Inequalities	نامساوی‌های وردشی

Aabstract

This involves enlarging the size of the optimization problems that exist in practice. The necessary conditions of efficiency in the use of techniques that enable high-speed, very large problems solved with acceptable quality can be felt more than.

Recently methods of optimization based on artificial intelligence approaches have been developed remarkable success in solving optimization problems efficiently acquired. Methods such as Genetic Algorithms, Tabu Search, refrigeration simulation and neural networks, their ability to solve large problems have good action. Special rates available on the possible application of neural networks in a wide range of research has provided. It points to the possibility of learning and performance improvement based on the input data point. It also allows parallel computations in a neural network is another advantage of the parallel hardware, enabling very large problems by this approach possible.

In this thesis, we tried a model of recursive neural network is presented to solve optimization problems in the traces. Analysis of uniqueness, stability and convergence of global solutions are examined and the performance of the proposed method using several examples of nonlinear programming problems with hybrid constraints and optimal control of discrete time linear hybrid systems is shown.

Finally, we provide conclusions and recommendations for future work.



Shahrood University
Faculty Of Mathematical Sciences

Dissertation Submitted in Partial
Fulfillment of The Requirements For The
Degree of Master of Science in
Applied Mathematics

**A high performance neural network for
solving a class of nonlinear programming
problems with hybrid constraints**

Supervisor

Dr.Nazemi

by

Samira Sukhtsaraei

2015