



دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش تحقیق در عملیات

پایان نامه کارشناسی ارشد

# یک مدل شبکه‌ی عصبی کارا برای حل مسائل بهینه سازی محدب نما

نگارنده: مریم شفائی نوده

استاد راهنما

دکتر علیرضا ناظمی

بهمن ۱۳۹۵

تقدیم بہ

مادر و در مہربانم

ہمسفر عزیزم

و ہمچنین پسر گلم ارمیا

# تقدیر و تشکر

باسپاس فراوان از استاد گرامیم جناب آقای دکتر علیرضا ناظمی به خاطر راهنمایی های ارزنده ای که در طول نگارش این پایان نامه به من نمودند. از مادر، پدر و خواهرمهربان و همچنین همسر عزیزم به خاطر حمایت های بی دریغشان نهایت تشکر و قدردانی را دارم.

مریم شفائی نوده

بهمن ۱۳۹۵

## تعمدنامه

اینجانب **مریم شفائی نوده** دانشجوی کارشناسی ارشد رشته **ریاضی کاربردی علوم ریاضی** دانشگاه شاهرود، نویسنده پایان نامه با عنوان **یک مدل شبکه‌ی عصبی کارا برای حل مسائل بهینه سازی محدب نما**، تحت راهنمایی **علیرضا ناظمی** متعهد می‌شوم:

- تحقیقات در این پایان نامه 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 پایان نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام “ دانشگاه شاهرود “ یا “ Shahrood University of Technology “ به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده) شده است، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

**مریم شفائی نوده**

**بهمن ۱۳۹۵**

## 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 صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان نامه بدون ذکر منبع مجاز نمی‌باشد.

## چکیده

در این پایان نامه یک شبکه عصبی جدید برای حل مسائل برنامه ریزی درجه دوم بامحدودیت های مساوی و نامساوی پیشنهاد شده است. در مقایسه با شبکه عصبی موجود برای حل چنین مسائل، شبکه عصبی پیشنهاد شده دارای نرون های کم تر و ساختار تک لایه است. همچنین تکنیک های بهینه سازی برای حل مسائل برنامه ریزی محدب نما مورد بررسی قرار گرفته است. یک شبکه عصبی ساده و بازگشتی بر اساس شرایط لازم و کافی بهینه گی، پیشنهاد می شود. ثابت می کنیم که جواب بهینه ی مساله ی بهینه سازی یک نقطه ی تعادل شبکه ی عصبی است و نقطه ی تعادل هم با جواب بهینه مساله مطابقت دارد. شبکه ی عصبی پیشنهادی پایدار سراسری به مفهوم لیاپانوف بوده و همگرایی سراسری به جواب مساله ی بهینه سازی اصلی است. شبیه سازی های عددی برای توضیح همگرایی سراسری شبکه ی عصبی داده شده است. همچنین کاربردهای مدل پیشنهادی در بخش های تجاری و شیمی برای نشان دادن تأثیر شبکه ی عصبی، نشان داده شده اند.

کلمات کلیدی: شبکه ی عصبی بازگشتی، برنامه ریزی محدب نما، همگرایی سراسری، شبکه عصبی، پایداری همگرایی، مساله برنامه ریزی درجه دو، نیمه معین مثبت.



# فهرست مطالب

۱	مروری بر شبکه‌های عصبی مصنوعی	۱
۱	پیشگفتار	۱.۱
۳	مسائلی که می‌توان از شبکه‌های عصبی مصنوعی در حل آنها استفاده کرد	۲.۱
۳	تاریخچه‌ی تکامل شبکه‌ی عصبی	۳.۱
۵	مدل بیولوژیک نرون عصبی	۴.۱
۶	مدل ریاضی نرون عصبی	۵.۱
۷	شبکه‌های عصبی مصنوعی و بهینه‌سازی	۶.۱
۹	تاریخچه‌ی حل مساله‌ی بهینه‌سازی با استفاده از شبکه‌ی عصبی	۷.۱
۱۳	مقدمات	۲
۱۳	مسائل بهینه‌سازی محدب	۱.۲
۱۵	توابع محدب‌نما	۲.۲
۱۶	روش تابع جریمه (پنالتی)	۳.۲
۱۹	سیستم‌های دینامیکی	۴.۲
۱۹	مقدمات	۱.۴.۲
۲۰	نظریه‌ی لیاپانف و پایداری نقطه	۲.۴.۲
۲۳	یک مدل شبکه‌ی عصبی جدید برای حل مسائل برنامه‌ریزی درجه دوم در حالت کلی	۳
۲۵	مدل شبکه‌ی عصبی پیشنهادی	۱.۳
۲۷	تحلیل پایداری	۱.۱.۳
۳۲	مثال عددی	۲.۱.۳
۳۷	حل یک مساله‌ی برنامه‌ریزی غیرخطی محدب نما با یک شبکه عصبی جدید کارا	۴
۳۷	معرفی	۱.۴
۳۸	مدل شبکه‌ی عصبی	۲.۴
۴۰	پایداری و تحلیل همگرایی	۳.۴
۴۳	یک مثال عددی ساده	۴.۴
۴۴	کاربرد در تجارت	۵.۴

۴۷	.....	۶.۴ کاربرد در شیمی
۴۹	.....	۷.۴ نتیجه گیری
۵۱		مراجع
۶۱		واژه‌نامه فارسی به انگلیسی
۶۳		واژه‌نامه انگلیسی به فارسی





# فصل ۱

## مروری بر شبکه‌های عصبی مصنوعی

### ۱.۱ پیشگفتار

شبکه‌های عصبی مصنوعی<sup>۱</sup> پدیده‌ی نسبتاً جدیدی است. هرچند شروع تحقیقات در زمینه‌ی شبکه‌های عصبی مصنوعی به دهه‌ی ۱۹۴۰ میلادی برمی‌گردد، اما گسترش آن در دهه‌ی ۱۹۸۰ صورت پذیرفت [۲].

شبکه‌های مصنوعی شاخه‌ای از هوش مصنوعی است و در واقع تلاشی برای پیاده‌سازی نرون‌های عصبی مغز انسان به صورت مصنوعی می‌باشد. در یک شبکه‌ی عصبی مصنوعی سلول‌های عصبی یا همان نرون‌ها به صورت یک مدل ریاضی بیان می‌شوند و با مرتب کردن سلول‌های عصبی در اشکال مختلف می‌توان برخی از عملکردهای مغز انسان مانند ذخیره‌ی اطلاعات در حافظه، تشخیص الگوها، فرایند یادگیری و ارتباط برقرار کردن بین الگوها را پیاده‌سازی کرد.

مهمترین ویژگی شبکه‌های عصبی مصنوعی قابلیت یادگیری در آنها است. به این مفهوم که این شبکه بر مبنای یک قاعده‌ی مشخص و ثابت برنامه‌ریزی نشده و با گذشت زمان و یا در هر تکرار شبکه وضعیت آن تغییر می‌کند یا به اصطلاح آموزش می‌بیند. مبحث شبکه‌های عصبی مصنوعی مربوط به الگوبرداری از قوه‌ی یادگیری در انسان و پیاده‌سازی آن به صورت الگوریتم‌های کامپیوتری است. بنابراین شبکه‌های عصبی این قابلیت را دارند که کارهای پیچیده‌ای که برای سیستم‌های مبتنی بر قاعده مشکل هستند را فرا بگیرند.

---

<sup>۱</sup>Artificial Network Neural

یک شبکه‌ی عصبی مصنوعی نمی‌تواند در مورد کارهایی که انجام می‌دهد استدلال کند و این یک نقطه ضعف برای شبکه‌ی عصبی مصنوعی محسوب می‌شود. به‌عنوان مثال اگر یک شبکه‌ی عصبی عمل تشخیص چهره را آموخت، نمی‌تواند توضیح دهد که چگونه این کار را انجام داده است. لذا در حال حاضر تحقیقات بر روی دسته‌ای از سیستم‌های هوشمند که مرکب از شبکه‌های عصبی با قابلیت یادگیری و سیستم‌های مبتنی بر قاعده (با قابلیت توضیح و استدلال) است، انجام می‌پذیرد.

از ابتدای دهه‌ی ۱۹۸۰ با معرفی شبکه‌ی هاپفیلد<sup>۲</sup> و حل مساله‌ی فروشنده‌ی دوره‌گرد و برنامه‌ریزی خطی که به کمک آنها موج گسترده‌ای از تحقیقات بر روی شبکه‌های عصبی پس از سال‌ها توقف آغاز شد، بسیاری از مسائل بهینه‌سازی در صنعت و اقتصاد را می‌توان به صورت یک مساله برنامه‌ریزی محدب خطی یا برنامه‌ریزی محدب غیرخطی مدل‌بندی کرد. این امر موجب صرفه‌جویی در واحدهای تجاری و صنعتی می‌شود. امروزه استفاده از این روش‌ها در سایر بخش‌های جامعه رواج یافته است.

حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی نسبت به روش‌های کلاسیک از سرعت بالاتری برخوردار است. اثبات می‌شود که نقاط تعادل در شبکه‌ی عصبی با نقاط بهینه‌ی مساله‌ی اولیه متناظرند و همچنین شبکه به طور سراسری همگراست. لذا به ازای هر نقطه‌ی اولیه، معادله‌ی مسیر شبکه به جواب بهینه‌ی مساله‌ی برنامه‌ریزی محدب همگرا می‌شود. اگرچه شبکه‌های عصبی مساله را به صورت تقریبی حل می‌کنند ولی سرعت محاسباتی بالای آنها این نقص را تا حدی جبران می‌کند. همچنین با تعریف مناسب شبکه‌ی عصبی، می‌توان تقریب‌های بسیار خوبی از جواب مساله به دست آورد.

ویژگی‌های یک شبکه‌ی عصبی مطلوب برای حل مسائل بهینه‌سازی را می‌توان این‌گونه بیان کرد که اولاً شبکه دارای نقطه تعادل باشد، یعنی بعد از تکرار متناهی شبکه به نقطه‌ای ثابت همگرا شود و به عبارت دیگر شبکه به طور مجانبی پایدار باشد یعنی به ازای هر نقطه‌ی اولیه دلخواه شبکه به نقطه‌ی تعادل خود همگرا شود. ثانیاً نقاط تعادل شبکه‌ی عصبی متناظر با مقدار دقیق و یا مقدار تقریبی جواب مساله‌ی بهینه‌سازی باشد. اگر مساله‌ی اولیه دارای نقطه‌ی بهینه‌ی یکتا باشد، در این صورت شبکه‌ی عصبی باید به گونه‌ای باشد که نقطه‌ی تعادل آن منحصر به فرد باشد زیرا در غیراین صورت شبکه به نقطه‌ی تعادل نامطلوبی همگرا می‌شود. همچنین باید بتوان برای شبکه‌ی عصبی تابع انرژی به دست آورد که دارای خواص یک تابع لیاپانف<sup>۳</sup> باشد و نقاط مینیمم آن همان نقاط تعادل شبکه‌ی عصبی باشد. در صورتی که تابع انرژی یک تابع لیاپانف باشد در این صورت اطمینان حاصل می‌شود که شبکه به طور مجانبی پایدار و به نقطه تعادل همگراست.

لذا لازم است تا با مفاهیم بهینه‌سازی و همچنین با مفاهیمی مانند سیستم دینامیکی، نقطه ساکن

<sup>۲</sup> John Hopfield

<sup>۳</sup> Liapunov function

به‌طور مجانبی پایدار، تابع انرژی، تابع لیاپانف و غیره آشنا شویم. در فصل دوم به مرور این مفاهیم می‌پردازیم.

در فصل سوم به بررسی و تحلیل یک شبکه‌ی عصبی جدید و همچنین کاربرد آن در حل برنامه‌ریزی درجه دوم می‌پردازیم. همچنین در این فصل مشکلات این روش بررسی می‌شود.

فصل چهارم به توسعه شبکه‌ی عصبی ارائه شده در فصل سوم می‌پردازیم. در این فصل پایداری و همگرایی روش به تفصیل بحث می‌شود. در ادامه‌ی این فصل نرون‌های عصبی و همچنین شبکه‌ی عصبی مصنوعی ارائه شده برای حل برخی مسائل کاربردی بیان می‌شود.

## ۲.۱ مسائلی که می‌توان از شبکه‌های عصبی مصنوعی در حل آنها استفاده کرد

همانطور که ذکر شد شبکه‌ی عصبی مصنوعی شاخه‌ای از هوش مصنوعی است. زمانی که در حل مسائل با حجم انبوهی از داده‌ها مواجه هستیم به گونه‌ای که مهندسی نرم‌افزار قادر به حل آن نیست، می‌توان از هوش مصنوعی به‌عنوان روشی کارآمد استفاده کرد. لذا از شبکه‌های عصبی مصنوعی می‌توان در حل مسائلی که دارای تعداد زیادی داده می‌باشد، استفاده کرد [۴]. به‌عنوان مثال در طراحی نرم‌افزاری برای یک قلم نوری که بتواند هر کلمه‌ی نوشته شده با دست را تایپ کند، باید تعداد بسیار زیادی نمونه خط را جمع‌آوری نمود تا با مقایسه پیکسل به پیکسل آنها، نمونه‌ی استاندارد از خط را به دست آورد. اما در طراحی با استفاده از شبکه‌ی عصبی به تعداد اندکی نمونه نیاز است. زیرا شبکه‌ی عصبی با شناسایی الگوی به کار رفته در نمونه‌های ورودی می‌تواند خود را تعلیم دهد و هر نمونه‌ی دیگر را شناسایی نماید.

یکی دیگر از موارد کاربرد شبکه‌های عصبی مصنوعی مربوط به حالتی است که الگوهای ورودی مدام تغییر شکل می‌دهند و یا دارای نویز می‌باشند. به‌عنوان مثال در تشخیص تصاویری که از وضوح کافی برخوردار نیستند و یا برای حذف اختلال از صدا می‌توان از شبکه‌های عصبی مصنوعی استفاده کرد [۴].

## ۳.۱ تاریخچه‌ی تکامل شبکه‌ی عصبی

در سال ۱۹۴۳ یک نرولوژیست به‌نام وارن مکلوث<sup>۴</sup> به‌همراه یک ریاضیدان به نام والتر پیتس<sup>۵</sup> اولین مدل ریاضی نرون را ارائه دادند [۵۰]. این نرون ساده یک عنصر محاسبه‌گر بود که عناصری به‌عنوان ورودی را در مقادیر ثابتی به‌نام وزن ضرب می‌کرد و آنها را از یک عملگر خطی عبور می‌داد و حاصل، خروجی

<sup>۴</sup> McCulloch Warren

<sup>۵</sup> Pitts Walter

نرون را تشکیل می‌داد. در این مدل وزن‌ها مقادیری ثابت بودند.

در سال ۱۹۴۹ دونالد هب<sup>۶</sup> کتابی درباره‌ی نحوه‌ی یادگیری در مغز انسان نوشت [۲۸]. 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<sup>۷</sup> [۱۹] مدلی از نرون‌ها را به‌طور تصادفی به‌هم وصل کردند و قانون هب را پیاده‌سازی نمودند و نشان دادند که می‌توان دو الگوی ورودی را با این قانون تعلیم و از هم تشخیص داد. با پیشرفت علم کامپیوتر اولین مدل از نرون عصبی مصنوعی توسط ناتانیال روچستر<sup>۸</sup> محقق شرکت IBM در سال ۱۹۵۸ با استفاده از کامپیوتر شبیه‌سازی شد.

روزنبلات<sup>۹</sup> در سال ۱۹۵۸ نرون ساده‌ی مکلوت و پیتس را اصلاح کرد و به آن یک قانون تعلیم اضافه کرد و این نرون را پرسپترون<sup>۱۰</sup> نامید. قانون تعلیم پرسپترون اولین قانون رسمی برای شبکه‌های عصبی است.

در سال ۱۹۵۹ برنارد ویدرو<sup>۱۱</sup> و مارسین هاف<sup>۱۲</sup> از دانشگاه استنفورد مدل دیگری از نرون را ارائه دادند که آدلاین<sup>۱۳</sup> نام‌گذاری شد که مختصر شده‌ی حروف "عنصر خطی تطبیقی"<sup>۱۴</sup> بود 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 مدل‌های<sup>۱۵</sup> ساخته شد که اولین شبکه‌ی عصبی می‌باشد. ویدرو و هاف از شبکه‌ی عصبی مدل‌های برای حذف اکو از خطوط تلفن استفاده نمودند.

سال ۱۹۶۹ آغاز یک دوره رکود برای شبکه‌ی عصبی بود. پاپرت و مینسکی<sup>۱۶</sup> کتابی به نام پرسپترون نوشتند و نشان دادند که اگرچه گیت‌های OR و AND را می‌توان با استفاده از پرسپترون پیاده‌سازی کرد ولی گیت XOR قابل پیاده‌سازی نمی‌باشد و لذا نمی‌توان از شبکه‌ی عصبی برای پردازش اطلاعات استفاده کرد. این دوره‌ی رکود تا دهه‌ی هشتاد ادامه داشت.

سال ۱۹۸۲ تا ۱۹۸۶ را می‌توان دوره‌ی تولد دوباره شبکه‌ی عصبی دانست. در این سال‌ها دو اتفاق مهم در این زمینه رخ داد. اول ارائه شبکه‌ی عصبی بازگشتی و مفهوم تابع انرژی توسط جان هاپفیلد بود

<sup>۶</sup> Hebb Donald

<sup>۷</sup> clark and farly

<sup>۸</sup> Rochester Nathaniel

<sup>۹</sup> Rosenblan Frank

<sup>۱۰</sup> Perceptron

<sup>۱۱</sup> Widrow Bernard

<sup>۱۲</sup> Hoff Marcian

<sup>۱۳</sup> ADELIN

<sup>۱۴</sup> Elements Linnear Adaptive

<sup>۱۵</sup> MADALINE

<sup>۱۶</sup> Minsky and Papert

که در سال ۱۹۸۲ منتشر شد [۲۹]. همچنین دومین اتفاق، حل مساله فروشنده‌ی دوره‌گرد (TSP) [۳۰] توسط هاپفیلد بود. هاپفیلد نشان داد که از این شبکه‌ی بازگشتی می‌توان برای حل مسائل بهینه‌سازی استفاده کرد و افق جدیدی را در شبکه‌های عصبی گشود. اولین تلاش در حل مساله‌ی برنامه‌ریزی خطی در سال ۱۹۸۶ و توسط تانک<sup>۱۷</sup> و هاپفیلد انجام شد [۶۳].

کندی و چا در سال ۱۹۸۸ [۳۵] با اصلاح شبکه‌ی هاپفیلد از آن برای حل مسائل برنامه‌ریزی غیرخطی استفاده کردند که البته این شبکه دارای نقص‌هایی بود زیرا تابع فعالیت شبکه دارای پارامتر بود و همچنین شبکه به تقریبی از جواب مساله همگرا می‌شد.

در سال ۱۹۹۰ رودریگز<sup>۱۸</sup> و همکاران [۶۱] با ایده گرفتن از شبکه‌ی هاپفیلد، کندی و چا و با اصلاح آن شبکه‌ی عصبی‌ای را ارائه دادند که جواب‌های دقیق‌تری از مساله را پیدا می‌کرد.

روش‌هایی که تاکنون بحث شد را می‌توان جزو یک دسته حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی قرار داد و از آن به عنوان روش‌های مبتنی بر گرادیان نام برد. اما روش‌های دیگر حل این مسائل، طراحی شبکه‌ی عصبی بازگشتی با استفاده از روش‌هایی چون روش تصویر، روش مبتنی بر دوگان و روش ضرایب لاگرانژ است.

از سال ۱۹۸۵ تاکنون در زمینه‌ی حل مسائل بهینه‌سازی با استفاده از روش‌های مبتنی بر گرادیان، روش‌های تصویر و روش‌های مبتنی بر دوگان و همچنین روش لاگرانژ مقاله‌های زیادی نوشته شده است.

## ۴.۱ مدل بیولوژیک نرون عصبی

هر نرون از سه بخش جداگانه به نام‌های دندریت<sup>۱۹</sup>، جسم سلولی یا سوما<sup>۲۰</sup> و آکسون<sup>۲۱</sup> تشکیل شده است [۳]. شکل ۱.۱ مدل یک نرون بیولوژیک را نشان می‌دهد.

دندریت‌ها، سیگنال‌ها را از دیگر نرون‌ها دریافت می‌کنند در واقع این سیگنال‌ها یک نمونه از پالس‌های الکتریکی هستند که در طول فاصله سیناپسی<sup>۲۲</sup> عبور و با یک سری از تغییرات شیمیایی دامنه‌ی آنها تغییر می‌کند.

جسم سلولی ابتدا تمام سیگنال‌های به دست آمده یا سیگنال‌های ورودی از دندریت‌ها را با هم جمع

<sup>۱۷</sup>Tank

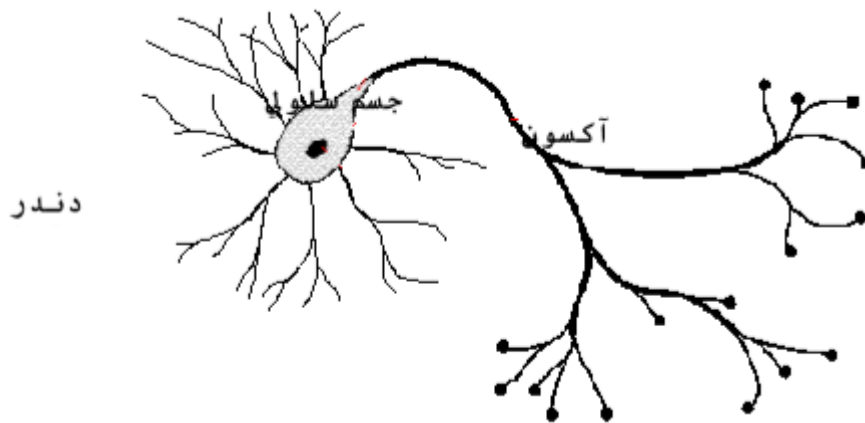
<sup>۱۸</sup>Rodriguez-Vazquez

<sup>۱۹</sup>Dendrit

<sup>۲۰</sup>Soma

<sup>۲۱</sup>Axon

<sup>۲۲</sup>Gap Synaptic



شکل ۱.۱: مدل بیولوژیک نرون عصبی

می‌کند و سپس اگر مجموع سیگنال‌ها بیشتر از حد معین باشد آنگاه سلول تحریک شده یا اصطلاحاً آتش می‌کند. در غیراینصورت عکس‌العملی نشان نمی‌دهد. در هر نرون عصبی دو حافظه موجود است. حافظه‌ی دراز مدت<sup>۲۳</sup> و حافظه‌ی کوتاه مدت<sup>۲۴</sup>. حافظه‌ی دراز مدت از نظر بیولوژیک ترشح هورمون‌ها در فضای سیناپسی است که در طی زمان مقدار ترشح آنها بر اساس تکرار و تجزیه تنظیم می‌شود و در مدل ریاضی در واقع همان مقادیر وزن‌ها می‌باشد. حافظه‌ی کوتاه مدت نیز همان سیگنال‌های ورودی هستند که از محیط اطراف دریافت و مقادیر وزن‌ها را براساس آن در کوتاه مدت تنظیم می‌کند.

## ۵.۱ مدل ریاضی نرون عصبی

در هر شبکه‌ی عصبی مصنوعی اطلاعاتی که قرار است پردازش شود، در محلی که نرون نام دارد قرار می‌گیرد. هر کدام از اطلاعات یا سیگنال‌ها با یک نوع ارتباط به نرون‌های دیگر متصل می‌شود. در هر نرون مصنوعی، اطلاعات با الگوهای ورودی در عددی به نام وزن سیناپسی<sup>۲۵</sup> ضرب می‌شود. وزن اولین ورودی که مقدار آن همواره ۱- است را بایاس<sup>۲۶</sup> می‌نامند [۲]. مجموع حاصلضرب ورودی‌ها در وزن‌های متناظرشان توسط تابع تحریک<sup>۲۷</sup> به یک مقدار دیگر نگاشته می‌شود.

مدل توصیف شده اولیه‌ی نرون یا همان مدل مکلوث و پیتس است که تابع تحریک آن به نام محدودکننده‌ی سخت<sup>۲۸</sup> معروف است [۲].

در این مدل  $net = \sum x_i w_i - \theta$  و  $\theta$  وزن اولین نرون است که به‌عنوان یک آستانه‌ی تحریک<sup>۲۹</sup> عمل می‌کند. اگر  $\sum x_i w_i$  از مقدار  $\theta$  بیشتر باشد خروجی ۱ است و در غیراینصورت ۰- می‌باشد.

<sup>۲۳</sup>Memoty Term Long

<sup>۲۴</sup>MemotyTerm Short

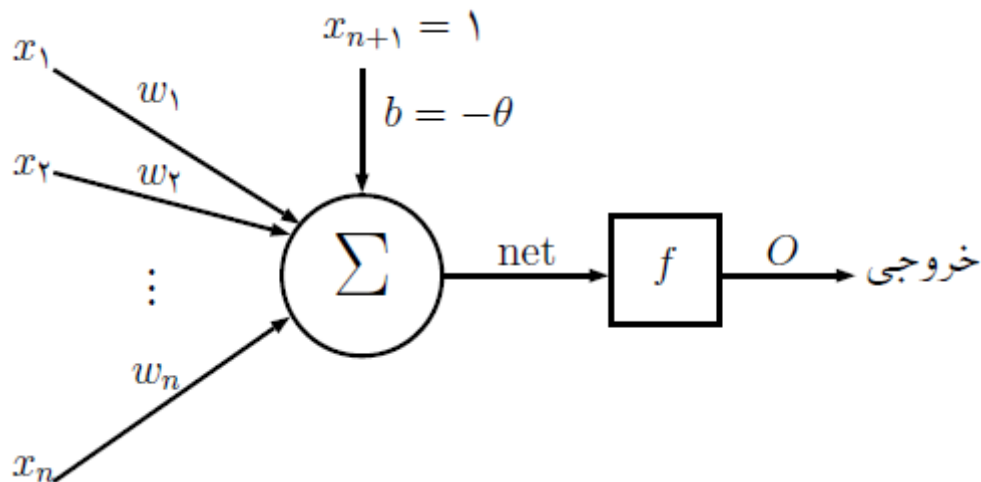
<sup>۲۵</sup>Weight Synaptic

<sup>۲۶</sup>Bias

<sup>۲۷</sup> Function Activation

<sup>۲۸</sup>Hard Limit

<sup>۲۹</sup>Threshold



شکل ۲.۱: مدل ریاضی نرون عصبی

در زیر دو نوع از توابع محدودکننده‌ی سخت معرفی می‌شوند.

$$y = f(n) = \begin{cases} -1 & n < 0 \\ 1 & n > 0 \end{cases}$$

$$y = f(n) = \begin{cases} 0 & n < 0 \\ 1 & n > 0 \end{cases}$$

در حالت کلی تابع فعالیت یک نرون می‌تواند با توابع دیگری اعم از خطی یا غیرخطی جایگزین شود. بنا به نوع تابع فعالیت که برای نرون در نظر می‌گیریم، خروجی نرون می‌تواند گسسته و یا پیوسته باشد. به‌عنوان مثال در شبکه‌ی هاپفیلد تابع فعالیت هر نرون یک تابع به‌طور یکنواخت صعودی و به‌طور یکنواخت مشتق‌پذیر است که اصطلاحاً یک تابع سیگموئید<sup>۳۰</sup> یا  $S$ -شکل<sup>۳۱</sup> یا لجستیک<sup>۳۲</sup> نامیده می‌شود. نمونه‌ای از تابع سیگموئید در زیر آورده می‌شود و شکل ۳.۱ نمودار آن را نشان می‌دهد.

$$f(u) = \frac{1}{1 + \exp(-u)}$$

## ۶.۱ شبکه‌های عصبی مصنوعی و بهینه‌سازی

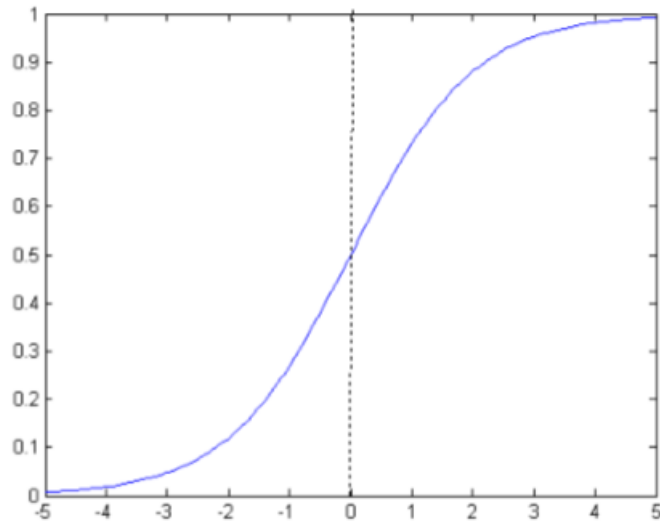
برای حل مسائل بهینه‌سازی از شبکه‌های عصبی بازگشتی استفاده می‌شود. شکل ۴.۱ نمونه‌ای از یک شبکه بازگشتی را نشان می‌دهد [۳].

<sup>۳۰</sup> Sigmoid

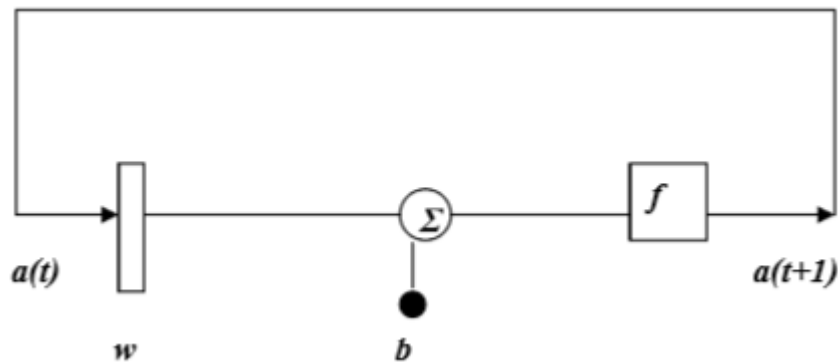
<sup>۳۱</sup> S-shap

<sup>۳۲</sup> Logistic





شکل ۳.۱: نمونه‌ای از یک تابع سیگموئید



شکل ۴.۱: یک واحد از شبکه‌ی عصبی بازگشتی

همانطور که در شکل ملاحظه می‌شود، خروجی شبکه  $a(t+1)$  برابر ورودی شبکه در یک واحد زمانی عقب‌تر است. یک شبکه‌ی بازگشتی باید مقدار اولیه در لحظه‌ی  $t = 0$  مشخص باشد این مقدار اولیه را با  $a(0)$  نشان می‌دهیم. اگر بردار ورودی  $p$  نشان دهنده‌ی شرایط اولیه‌ی شبکه باشد، آنگاه شبکه‌ی عصبی بازگشتی را می‌توان توسط سیستم زیر نشان داد:

$$\begin{cases} a(t+1) = f(\omega(a(t)) + b), \\ a(0) = p. \end{cases}$$

سیستم فوق را می‌توان به‌عنوان یک سیستم دینامیکی در نظر گرفت زیرا وضعیت آن در هر تکرار و با گذشت زمان تغییر می‌کند.

شبکه‌ی عصبی بازگشتی که برای حل مسأله‌ی بهینه‌سازی استفاده می‌شود باید شبکه‌ای همگرا باشد. همگرایی شبکه را می‌توان این‌طور بیان کرد که با گذشت زمان خروجی شبکه به مقدار ثابتی همگرا می‌شود. این نقطه‌ی ثابت را نقطه‌ی تعادل شبکه‌ی عصبی می‌نامیم. این نقطه‌ی تعادل باید یک

نقطه‌ی تعادل پایدار باشد.

## ۷.۱ تاریخچه‌ی حل مساله‌ی بهینه‌سازی با استفاده از شبکه‌ی عصبی

در سال ۱۹۸۶، تانک و هاپفیلد [۶۳] شبکه‌ی عصبی بازگشتی را برای حل مساله‌ی برنامه‌ریزی خطی معرفی کردند و نشان دادند که وضعیت شبکه در هر تکرار به گونه‌ای تغییر می‌کند که تابع انرژی متناظر با آن به‌طور یکنواخت کاهش می‌یابد تا جایی که به نقطه‌ی مینیمم خود می‌رسد و این نقطه‌ی مینیمم متناظر با نقطه‌ی تعادل شبکه‌ی عصبی می‌باشد. آنها این شبکه را توسط یک مدار الکترونیکی پیاده‌سازی کردند. همچنین از این شبکه برای حل مساله‌ی فروشنده دوره‌گرد با ۳۰ شهر استفاده نمودند [۳۰]. این شبکه دارای نقص‌هایی بود به ویژه اینکه نقطه‌ی تعادل شبکه در شرایط K.K.T صدق نمی‌کرد و لذا جواب مطلوبی از مساله حاصل نمی‌شد. با این وجود کارهای هاپفیلد انگیزه‌ی بسیار خوبی را برای محققین به‌وجود آورد تا در این زمینه فعالیت کنند [۳۵]–[۳۹]. کندی وچا [۳۵] با ایده گرفتن از شبکه‌ی هاپفیلد آن را بهبود دادند و از آن برای حل مساله‌ی برنامه‌ریزی غیرخطی محدب استفاده نمودند. آنها اثبات کردند که تابع انرژی در این مدل یک تابع لیاپانف می‌باشد و به همین دلیل شبکه‌ی عصبی به‌طور سراسری همگرا است. همچنین ثابت کردند که مجموعه نقاط تعادل شبکه متناظر با نقاط مینیمم مساله‌ی برنامه‌ریزی غیرخطی می‌باشد.

ما و شانبلات<sup>۳۳</sup> [۴۶] ثابت کردند که نقطه‌ی تعادل در شبکه‌ی کندی و چا شامل شرایط بهینه‌گی K.K.T است. همچنین نشان دادند که به دلیل وجود پارامتر جریمه در شبکه‌ی کندی و چا جواب‌های دقیق زمانی حاصل می‌شود که پارامتر جریمه به سمت بی‌نهایت میل کند که در عمل چنین چیزی ممکن نیست و لذا این روش جواب‌های تقریبی مساله را ارائه می‌کند که با افزایش پارامتر جریمه تقریب بهتری از جواب بهینه مساله حاصل می‌شود.

برای جلوگیری از پارامتر جریمه، ما و شانبلات یک شبکه‌ی عصبی دو فازی را ارائه دادند [۴۵] که فاز اول شبکه مشابه شبکه‌ی کندی و چا بود و در فاز دوم آن مسیر شبکه به جواب دقیق مساله همگرا می‌شد. لذا این روش جواب‌های دقیق‌تری نسبت به شبکه‌ی کندی و چا ارائه می‌داد. مشکل این روش در این بود که پایداری فاز دوم شبکه بستگی به انتخاب یک مقدار بزرگ از پارامتر جریمه داشت و لذا اگرچه تاثیر پارامتر جریمه در این روش کاهش یافته بود و جواب‌های حاصل دقیق‌تر بودند اما هنوز مستقل از پارامتر نبود.

ژانگ و کنستانتینید<sup>۳۴</sup> [۸۴] بر مبنای روش لاگرانژ شبکه‌ای عصبی را پیاده‌سازی کردند که کاملاً مستقل از پارامتر جریمه بود و قادر به حل مسائل غیرخطی بود. نقطه‌ی تعادل این شبکه‌ی عصبی در شرایط بهینه‌گی درجه اول و دوم صدق می‌کرد و همچنین شبکه‌ی حاصل به‌طور سراسری همگرا بود.

<sup>۳۳</sup> Shanblat and Maa

<sup>۳۴</sup> Constantinides and Zhang

در سال ۱۹۹۳ بوزردوم و پتیسن<sup>۳۵</sup> شبکه‌ای را بر مبنای گرادیان و روش تصویر<sup>۳۶</sup> و مستقل از پارامتر جریمه ابداع کردند که تنها قادر به حل مسائل درجه دو با متغیرهای کراندار بود [۷]. این روش در عمل روش کارایی بود اما نمی‌توانست مسائل کلی برنامه‌ریزی خطی و درجه دو را حل کند.

با استفاده از روش تصویر دوگان، اکسیا<sup>۳۷</sup> و همکاران [۷۵]–[۷۶] شبکه‌ی عصبی جدیدی را برای حل مسائل خطی و غیرخطی ارائه دادند و ثابت کردند که شبکه به‌طور سراسری به جواب مساله‌ی اولیه و دوگان همگرا می‌شود. ویژگی بارز این روش در این بود که به‌طور همزمان مساله‌ی اولیه و دوگان را حل می‌کرد. همچنین مزیت دیگر این روش در این بود که سیستم دینامیکی شبکه‌ی عصبی فاقد پارامتر پنالتی بود.

وانگ و همکاران نیز مدل‌های مختلفی برای مسائل مشتق‌ناپذیر و نامحدب ارائه دادند.

در ادامه به برخی از کارهای انجام شده در زمینه‌ی شبکه‌های عصبی و بهینه‌سازی در داخل کشور اشاره می‌شود.

در سال ۲۰۰۵ بایمانی و عفتی [۱۳] شبکه‌ی عصبی بازگشتی مبتنی بر دوگان را برای حل مسائل درجه دو و مسائل غیرخطی محدب ارائه نمودند. در این روش نقطه‌ی تعادل شبکه‌ی عصبی در شرایط قضیه دوگان صدق کرده و لذا این شبکه قادر بود تا به‌طور همزمان مساله‌ی اولیه و دوگان را حل کند.

در سال‌های ۲۰۰۵، ۲۰۰۶ و ۲۰۰۷ مالک و همکاران [۴۹] با ساده‌تر کردن مدل اکسیا، مدل‌های دیگری از شبکه‌ی عصبی مبتنی بر دوگان ارائه دادند که به‌طور سراسری به جواب مساله‌ی اولیه و دوگان همگرا می‌شد. این روش نسبت به روش اکسیا از سرعت همگرایی بیشتری برخوردار بود.

همچنین عفتی و همکاران [۱۸] در سال ۲۰۰۷ با استفاده از روش مبتنی بر تصویر، شبکه‌ی عصبی بهبود یافته‌ای را برای حل مسائل برنامه‌ریزی محدب ارائه نمودند. در این روش ابتدا مساله‌ی برنامه‌ریزی محدب به یک مساله‌ی تصویر متناظر تبدیل شده که جواب آن همان جواب مساله‌ی برنامه‌ریزی اولیه است و سپس سیستم دینامیکی برای حل مساله‌ی تصویر ارائه می‌شود. نقطه‌ی تعادل این سیستم دینامیکی جواب مساله‌ی تصویر و متناظراً جواب مساله‌ی اولیه خواهد بود.

از دیگر کارهای انجام شده می‌توان به مقاله‌ی رنجبر و عفتی اشاره نمود [۱۶] که در آن از شبکه‌ی عصبی بازگشتی برای حل مساله‌ی ماکزیمم جریان استفاده شده است. در این روش ابتدا مساله‌ی

<sup>۳۵</sup>Pattison and Bouzerdoum

<sup>۳۶</sup>Method Projection

<sup>۳۷</sup>Xia

ماکزیمم جریان به یک مساله‌ی برنامه‌ریزی خطی تبدیل می‌شود و سپس مساله‌ی برنامه‌ریزی خطی با یک شبکه‌ی عصبی مناسب حل می‌شود. نقطه‌ی تعادل شبکه جواب مساله‌ی ماکزیمم جریان خواهد بود.

در سال‌های ۲۰۱۱ تا ۲۰۱۵، ناظمی و همکاران [۵۳]–[۵۶] یک مدل کارا برای حل مسائل ماکزیمم جریان – کوتاهترین مسیر، برنامه‌ریزی هندسی – برنامه‌ریزی درجه دوم و غیرخطی ارائه دادند.



# فصل ۲

## مقدمات

در این بخش برخی از مفاهیم و قضایای بهینه‌سازی که در این پایان‌نامه از آن استفاده شده، آورده شده است.

### ۱.۲ مسائل بهینه‌سازی محدب

در این بخش مفاهیمی از مسائل بهینه‌سازی محدب و شرایط لازم و کافی برای حل آن از [۸۸] آورده شده است.

تعریف ۱.۱.۲. مجموعه‌ی محدب<sup>۱</sup>.  $X \subset \mathbb{R}^n$  را یک مجموعه‌ی محدب گوئیم هرگاه:

$$\forall a, b \in X \Rightarrow [a, b] \subset X,$$

که در آن

$$[a, b] = \{x \in \mathbb{R}^n \mid x = \lambda a + (1 - \lambda)b, 0 \leq \lambda \leq 1\}.$$

تعریف ۲.۱.۲. تابع محدب<sup>۲</sup>. فرض کنید  $X \subset \mathbb{R}^n$  یک مجموعه‌ی محدب و غیرتهی باشد. در

---

<sup>۱</sup> Set Convex

<sup>۲</sup> Function Convex

این صورت تابع  $f : X \rightarrow \mathbb{R}$  را یک تابع محدب گوئیم هرگاه :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \quad x, y \in X,$$

که در آن  $0 \leq \lambda \leq 1$ .

**تعریف ۳.۱.۲. مساله برنامه ریزی محدب<sup>۳</sup>.** مساله برنامه ریزی محدب به صورت زیر تعریف می شود:

$$\begin{aligned} (CP) \quad & \min f(x) \\ \text{s.t.} \quad & \\ & g_i(x) \leq 0, \quad i = 1, \dots, m, \\ & AX = b, \\ & x \in X, \end{aligned} \tag{1.2}$$

که در آن  $x \in \mathbb{R}^n$  متغیر تصمیم گیری مساله و  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ،  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  و  $f$  و  $g_i$  توابع محدب و  $X$  نیز یک مجموعه ی محدب است. همچنین  $A = [a_{ij}]_{p \times n}$  و  $b = [b_i]_{p \times 1}$  و مشتق توابع  $f$  و  $g_i$  پیوسته می باشد.

**تعریف ۴.۱.۲. مساله برنامه ریزی درجه دو<sup>۴</sup>.** مساله برنامه ریزی درجه دو به شکل زیر تعریف می شود:

$$\begin{aligned} (QP) \quad & \min f(x) = \frac{1}{2}x^T Qx + D^T x \\ \text{s.t.} \quad & \\ & Ex \leq d, \\ & Ax = b. \end{aligned} \tag{2.2}$$

**قضیه ۵.۱.۲.** اگر در مساله برنامه ریزی درجه دو  $Q$  ماتریس نیمه معین مثبت باشد، آنگاه مساله ی درجه دوی  $(QP)$  یک مساله ی برنامه ریزی محدب خواهد بود.

**تعریف ۶.۱.۲. نقطه ی منظم<sup>۵</sup>.** مساله ی برنامه ریزی زیر را در نظر بگیرید:

$$\begin{aligned} (P) \quad & \min f(x) \\ \text{s.t.} \quad & \\ & g_i(x) \leq 0 \quad i = 1, \dots, m, \\ & h_j(x) = 0 \quad j = 1, \dots, p. \end{aligned}$$

<sup>۳</sup> Programming Convex

<sup>۴</sup> Programming Quadratic

<sup>۵</sup> Point Regular

مجموعه‌ی  $A(x)$  را به شکل زیر تعریف می‌کنیم:

$$A(x) = \{i \mid g_i(x) = 0\}.$$

در این صورت نقطه‌ی  $x^*$  که در محدودیت‌های  $h_j(x) = 0$  و  $g_i(x) \leq 0$  صدق می‌کند را یک نقطه‌ی منظم گویند هرگاه  $\nabla h_j(x^*)$  و  $\nabla g_i(x^*)$  مستقل خطی باشند که در آن  $i \in A(x^*)$ .

**قضیه ۷.۱.۲.** شرط لازم بهینه‌گی درجه اول (شرایط K.K.T) فرض کنید  $x^*$  یک نقطه شدنی و همچنین نقطه‌ی مینیمم برای مساله‌ی  $(P)$  باشد و علاوه بر آن  $x^*$  نقطه‌ی منظم محدودیت‌ها نیز باشد، در این صورت بردارهایی مانند  $\lambda \in E^m$  و  $\mu \in E^p$  وجود دارند به طوری که:

$$\begin{cases} \nabla f(x^*) + \nabla h(x^*)^T \lambda + \nabla g(x^*)^T \mu = 0, \\ \mu \geq 0, \\ \mu^T g(x^*) = 0. \end{cases} \quad (3.2)$$

**قضیه ۸.۱.۲.** برای مساله‌ی برنامه‌ریزی محدب (۱.۲)، شرایط قضیه‌ی ۷.۱.۲ لازم و کافی هستند.

## ۲.۲ توابع محدب‌نما

در این بخش انواع توابع محدب‌نما را معرفی نموده و قضایای مربوطه را ارائه می‌دهیم. مطالب این بخش از [۸۸] جمع‌آوری شده است.

**تعریف ۱.۲.۲.** [۷۱]. تابع  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  پیوسته‌ی لیپ شیتز روی مجموعه  $\Omega$  گفته می‌شود هرگاه ثابت  $L$  وجود داشته باشد به طوری که

$$\forall x, y \in \Omega, \quad \|F(x) - F(y)\| \leq L\|x - y\|$$

که در آن  $\|\cdot\|$  نرم اقلیدسی از  $\mathbb{R}^n$  را نشان می‌دهد.

**تعریف ۲.۲.۲.** تابع مشتق‌پذیر  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  روی مجموعه‌ی محدب  $\Omega$  محدب نما گفته می‌شود، هرگاه برای هر جفت  $x, y \in \Omega$ ،

$$\nabla f(x)(y - x) \geq 0 \implies f(y) \geq f(x)$$

**لم ۳.۲.۲.** [۷۲]. فرض کنید تابع مشتق‌پذیر  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  و روی مجموعه‌ی  $\Omega$  محدب‌نما باشد، در این صورت  $x^* \in \Omega$  در رابطه‌ی

$$\nabla f(x^*)(x - x^*) \geq 0, \forall x \in \Omega$$

صدق می‌کند اگر و فقط اگر  $x^*$  نقطه‌ی مینیممی از تابع  $f(x)$  باشد.



**تعریف ۴.۲.۲.** [۷۳]. تابع  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  روی مجموعه‌ی  $\Omega$  شبه یکنواخت گفته می‌شود هرگاه برای هر  $x, y \in \Omega$ ،

$$F^T(x)(y-x) \geq 0 \implies F^T(y)(y-x) \geq 0$$

**تعریف ۵.۲.۲.** [۷۳]. تابع  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  روی مجموعه‌ی  $\Omega$  شبه یکنواخت اکید گفته می‌شود هرگاه ثابت  $r$  وجود داشته باشد، به طوری که برای هر  $x, y \in \Omega$ ،

$$F^T(x)(y-x) \geq 0 \implies F^T(y)(y-x) \geq r\|x-y\|^2$$

**لم ۶.۲.۲.** [۷۴]. یک تابع مشتق پذیر محدب نما است اگر و فقط اگر گرادیانش یک نگاشت شبه یکنواخت باشد.

## ۳.۲ روش تابع جریمه (پنالتی)

در این بخش روش تابع جریمه (پنالتی)<sup>۶</sup> را معرفی نموده و چند نمونه از آن را ارائه می‌دهیم. مطالب این بخش از [۸۸] آورده شده است.

مساله‌ی زیر را در نظر بگیرید:

$$(P1) \quad \min f(x)$$

s.t

$$x \in S,$$

که در آن  $f$  تابعی پیوسته روی  $\mathbb{R}^n$  و  $S$  یک مجموعه‌ی قیدی در  $\mathbb{R}^n$  است. در بسیاری از کاربردها  $S$  به طور ضمنی به وسیله‌ی تعدادی قیود تعریف می‌شود.

روش تابع جریمه بر این ایده مبتنی است که به جای مساله‌ی  $(P1)$ ، یک مساله‌ی نامقید به شکل زیر را در نظر بگیریم:

$$(P2) \quad \min f(x) + cP(x),$$

که در آن  $c$  یک ثابت مثبت و  $P$  تابعی روی  $\mathbb{R}^n$  است که دارای خواص زیر می‌باشد:

(۱)  $P(x)$  پیوسته است.

(۲) به ازای هر  $x \in \mathbb{R}^n$ ،  $P(x) \geq 0$ .

(۳)  $P(x) = 0 \iff x \in S$

<sup>۶</sup>Penalty function method

ملاحظه می‌شود که به ازای مقادیر بزرگ  $c$ ، نقطه‌ی مینیمم مساله‌ی  $(P_2)$  در ناحیه‌ای که  $P$  کوچک است قرار می‌گیرد. بنابراین انتظار می‌رود که نقاط جواب متناظر به ناحیه‌ی شدنی  $S$  میل کنند و منوط بر نزدیک بودن،  $f$  را مینیمم‌سازی کنند. زمانی که  $c$  به بی‌نهایت میل می‌کند، نقطه‌ی جواب مساله‌ی جریمه به یک جواب مساله‌ی مقید میل می‌کند.

مثال‌های متداولی از توابع پنالتی مانند تابع پنالتی درجه دو،  $P$ -نرم، هبر و لجستیک وجود دارند. تابع درجه دو (Quadratic):

$$a) P(x) = \frac{1}{p}x^2$$

تابع  $P$ -نرم:

$$b) P(x) = \frac{1}{p}|x|^p$$

هبر (Huber):

$$c) P(x) = \begin{cases} \frac{1}{p}x^2 & \text{for } |x| \geq \beta \\ \beta|x| - \frac{\beta^2}{p} & \text{for } |x| < \beta \end{cases}$$

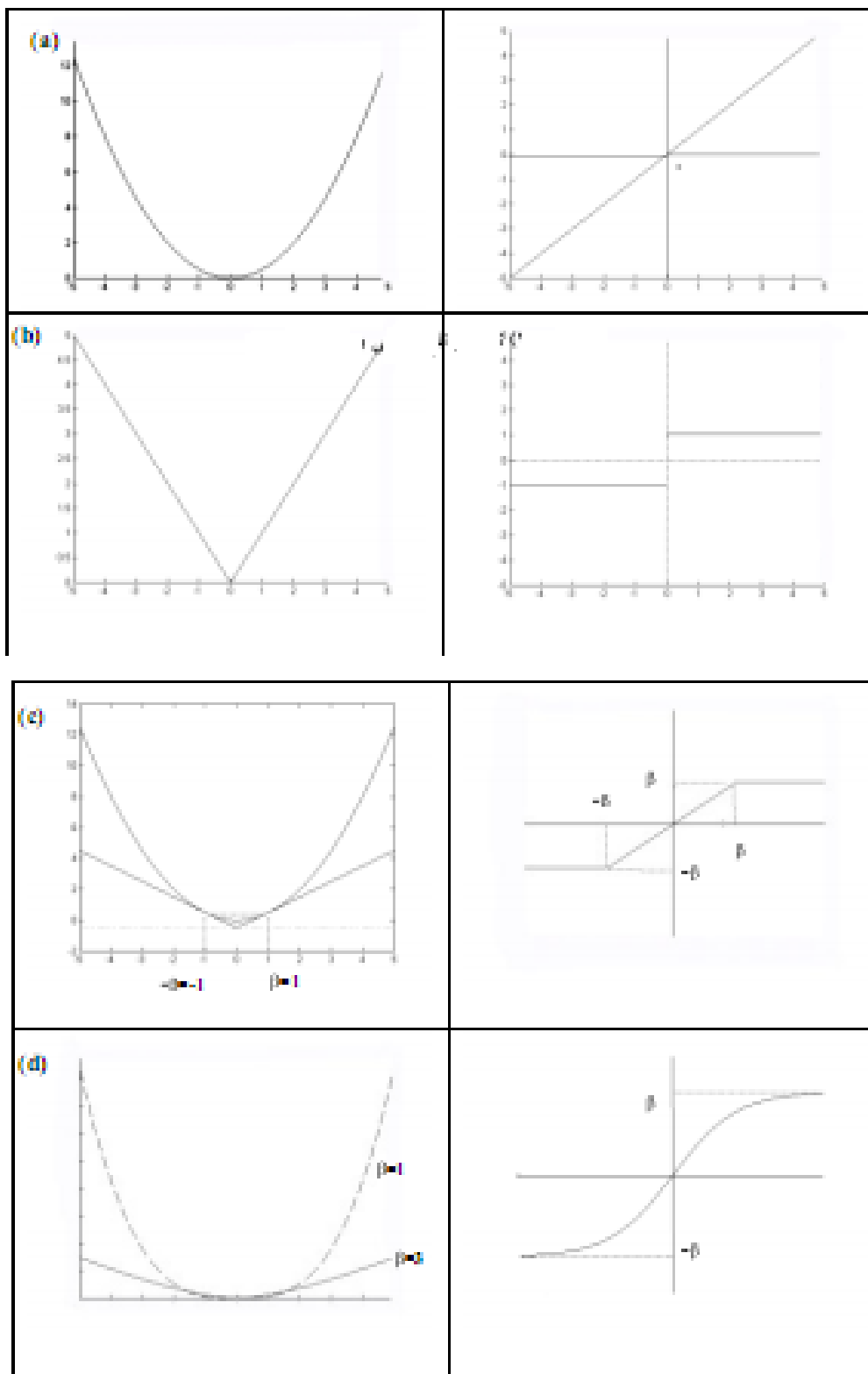
لجستیک (Logistic):

$$d) P(x) = \beta^2 \text{Lncosh}\left(\frac{x}{\beta}\right) \quad \beta > 0$$

شکل ۱.۲ نمودار این توابع پنالتی را در سمت چپ و نمودار مشتق آنها را در سمت راست جدول نشان می‌دهد.

نمودار تابع پناهی

نمودار مشتق تابع پناهی



شکل ۱.۲: نمودار تابع پناهی و نمودار تابع مشتق

## ۴.۲ سیستم‌های دینامیکی

در این بخش سیستم‌های دینامیکی<sup>۷</sup> را معرفی نموده و قضیه‌ی اساسی وجود و یگانگی وجود جواب مسأله‌ی مقدار اولیه و قضایای مربوط به پایداری را ارائه می‌نماییم.

### ۱.۴.۲ مقدمات

دستگاه معادله‌ی دیفرانسیل به شکل زیر را در نظر بگیرید:

$$\frac{dx}{dt} = f(x(t), t), \quad (۴.۲)$$

که در آن  $x(t) = [x_i(t)]^T$ ،  $t \geq 0$  بردار وضعیت یا حالت<sup>۸</sup> یا حرکت<sup>۹</sup> نامیده می‌شود. چنین دستگاهی یک دستگاه وابسته به زمان یا غیرخودگردان<sup>۱۰</sup> است. به این معنی که متغیر  $t$  به‌طور آشکار در ضابطه‌ی  $f$  ظاهر می‌شود.

اگر  $f_i$  به‌طور صریح به  $t$  وابسته نباشد به عبارتی

$$\frac{dx}{dt} = f(x(t)) \quad (۵.۲)$$

در این صورت سیستم را خودگردان<sup>۱۱</sup> یا مستقل از زمان نامند.

به چنین دستگاه معادله‌ی دیفرانسیلی که بردار وضعیت آن با تغییر زمان تغییر می‌کند، یک سیستم دینامیکی گفته می‌شود. در ادامه‌ی از نماد  $\dot{x}(t)$  برای نمایش مشتق  $x$  نسبت به زمان استفاده می‌کنیم.

**قضیه ۱.۴.۲. (قضیه‌ی اساسی وجود و یگانگی وجود جواب مسأله‌ی مقدار اولیه) [۵۲].** فرض کنید  $E$  زیرمجموعه‌ی باز از  $\mathbb{R}^n$  و شامل نقطه‌ی  $x_0$  باشد و فرض کنید هر  $f_i$  تابعی به‌طور پیوسته روی  $\mathbb{R}^n$  مشتق‌پذیر است یا به عبارتی  $f_i \in C^1(\mathbb{R})$ ، در این صورت به ازای هر  $a > 0$  که  $a \in \mathbb{R}$ ، مسأله‌ی مقدار اولیه‌ی زیر روی بازه‌ی  $[-a, a]$  جواب یگانه‌ی  $x(t)$  دارد:

$$\begin{cases} \dot{x}(t) = f(x(t)), \\ x(t_0) = x_0. \end{cases}$$

**تعریف ۲.۴.۲. نقطه‌ی تعادل [۵۲].** جواب سیستم دینامیکی (۵.۲) را نقطه‌ی تعادل، نقطه‌ی ساکن یا نقطه‌ی بحرانی می‌نامیم و با  $x^e$  نشان می‌دهیم، هرگاه  $f(x^e) = 0$ . لذا نقطه‌ی بحرانی جوابی از سیستم  $\dot{x}(t) = f(x(t))$  است که مستقل از زمان می‌باشد.

<sup>۷</sup>Dynamical Systems

<sup>۸</sup>State Vector

<sup>۹</sup>Motion

<sup>۱۰</sup>Nonautonomous

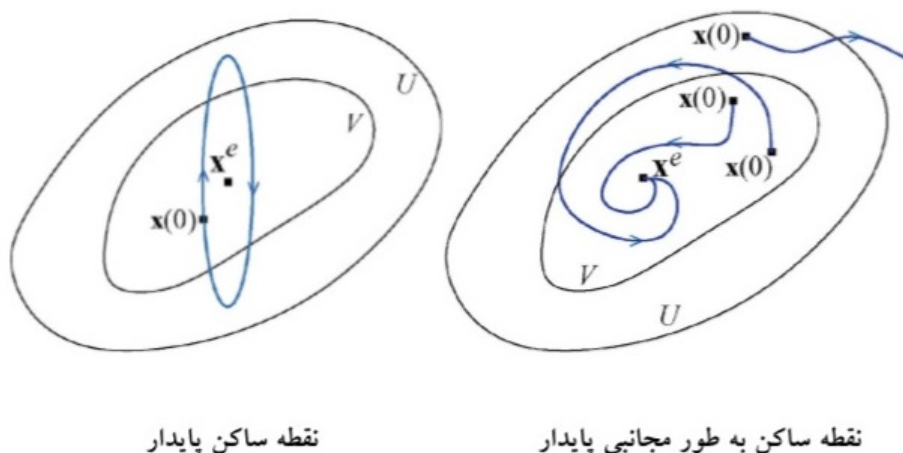
<sup>۱۱</sup>Autonomous

<sup>۱۲</sup>equilibrium

تعریف ۳.۴.۲. پایداری<sup>۱۳</sup> [۵۲]. نقطه ساکن  $x^e$ ، نقطه‌ی پایدار نامیده می‌شود هرگاه

$$\forall \varepsilon \geq 0 \quad \exists \delta > 0 \quad s.t \quad \|x(t_0) - x^e\| < \delta \implies \|x(t) - x^e\| < \varepsilon \quad \forall t \geq t_0.$$

تعریف ۴.۴.۲. پایداری مجانبی<sup>۱۴</sup> [۵۲]. نقطه ساکن  $x^e$  به‌طور مجانبی پایدار است هرگاه پایدار باشد و  $\lim_{t \rightarrow \infty} x(t) = x^e$ . شکل ۲.۲ مفهوم ساکن پایدار و نقطه‌ی ساکن به‌طور مجانبی پایدار را روشن تر می‌کند.



شکل ۲.۲: تفاوت میان نقطه‌ی ساکن پایدار و نقطه‌ی ساکن به‌طور مجانبی پایدار

## ۲.۴.۲ نظریه‌ی لیاپانف و پایداری نقطه

روش لیاپانف یکی از بهترین روش‌هایی است که در مورد تحلیل پایداری سیستم‌های دینامیکی می‌شناسیم. این روش برای سیستم‌های خطی و غیرخطی از هر مرتبه‌ای قابل استفاده است. این روش شرط کافی برای پایداری مجانبی نقطه‌ی ساکن یک سیستم دینامیکی غیرخطی را می‌دهد. به عبارتی بدون حل معادلات وضعیت سیستم، می‌توان پایداری نقطه‌ی ساکن سیستم غیرخطی را بررسی کرد.

لیاپانف پیشنهاد کرد که ممکن است توابعی وابسته به مسیر وجود داشته باشند که خواص آنها تعیین می‌کند که آیا مسیر به یک نقطه‌ی ساکن همگرا می‌شود یا خیر. چنین تابعی به‌عنوان تابع انرژی برای سیستم دینامیکی در نظر گرفته می‌شود که به تابع لیاپانف نیز معروف است.

قضیه‌ی لیاپانف به‌طور مختصر بیان می‌کند که نقطه‌ی ساکن یا بحرانی یک سیستم دینامیکی به‌طور مجانبی پایدار است هرگاه یک تابع لیاپانف در همسایگی این نقطه وجود داشته باشد.

<sup>۱۳</sup> stability

<sup>۱۴</sup> asymptotic stability

**تعریف ۵.۴.۲.** [۵۲] یک تابع لیاپانف یا تابع انرژی تابعی مانند  $E(x(t))$  است که در شرایط زیر صدق کند:

- (۱)  $E(x(t)) \in C^1$  و مشتقات جزئی مرتبه‌ی اول آن همگی پیوسته باشند، به عبارتی
- (۲)  $E(x(t)) \geq 0$  به ویژه در همسایگی باز نقطه‌ی ساکن  $x^e$ ،  $E(x(t)) > 0$  و  $E(x^e) = 0$
- (۳) مشتق تابع  $E(x(t))$  نسبت به زمان منفی است، به عبارتی:

$$\frac{dE(x^e)}{dt} = [\nabla_{x(t)} E(x(t))]^T \cdot \dot{x}(t) = [\nabla_{x(t)} E(x(t))]^T \cdot f(x(t)) < 0, x \in \Omega - \{x^e\}$$

و در نقطه‌ی ساکن  $x^e$  داریم  $\frac{dE(x^e)}{dt} = 0$ .

**قضیه ۶.۴.۲.** [۵۲] نقطه‌ی ساکن سیستم  $\dot{x}(t) = f(x(t))$  پایدار است اگر یک تابع لیاپانف متناظر با این سیستم وجود داشته باشد.

**قضیه ۷.۴.۲.** [۵۲] نقطه‌ی ساکن سیستم  $\dot{x}(t) = f(x(t))$  به طور سراسری پایدار است اگر یک تابع لیاپانف متناظر با این سیستم وجود داشته باشد و در یک همسایگی نقطه‌ی ساکن  $x^e$ ،  $\frac{dE(x(t))}{dt} < 0$ .

ملاحظه می‌شود که اگر  $\frac{dE(x(t))}{dt} \leq 0$  باشد، آنگاه  $x^e$  پایدار است ولی الزاماً به‌طور مجانبی پایدار

نیست.

اگر متناظر با یک سیستم غیرخطی یک تابع لیاپانف یا تابع انرژی وجود داشته باشد، آنگاه با افزایش  $t$  مقدار تابع انرژی کاهش پیدا می‌کند تا جایی که به کمترین میزان خود برسد. این نقطه‌ی مینیمم تابع انرژی در واقع همان نقطه‌ی ساکن برای سیستم دینامیکی است و سیستم در این نقطه به حالت تعادل یا پایدار خود می‌رسد و خروجی آن با گذشت زمان تغییر نمی‌کند. لذا می‌توان نتیجه گرفت که به جای محاسبه‌ی مستقیم نقطه‌ی ساکن برای سیستم دینامیکی، کافی است مینیمم تابع انرژی متناظر با آن را محاسبه کنیم و یا برعکس مینیمم تابع انرژی در نقطه‌ی ساکن سیستم دینامیکی اتفاق می‌افتد که از این نتیجه در مباحث بعدی برای حل مساله‌ی برنامه‌ریزی محدب استفاده می‌کنیم.



## فصل ۳

# یک مدل شبکه‌ی عصبی جدید برای حل مسائل برنامه ریزی درجه دوم در حالت کلی

در این بخش یک مدل شبکه‌ی عصبی بازگشتی برای حل مسائل برنامه ریزی درجه دوم محدب معرفی می‌شود. سپس یکتایی، پایداری و همگرایی مدل شبکه‌ی عصبی ارائه شده بررسی و اثبات می‌گردد و در نهایت با مثالی عددی کارایی مدل ارائه شده نشان داده می‌شود.

بحث و بررسی در مورد کاربرد شبکه‌های عصبی (شبکه‌های عصبی مصنوعی) در بهینه‌سازی از اوایل سال ۱۹۸۰ میلادی آغاز شده است. نتایج پژوهش‌های انجام شده از آن زمان تاکنون، مواردی چون برنامه‌ریزی خطی، برنامه‌ریزی درجه دوم، برنامه‌ریزی هندسی و برنامه‌ریزی غیرخطی را در برمی‌گیرد. ایده‌ی اصلی در استفاده از شبکه‌های عصبی مصنوعی برای مسائل بهینه‌سازی استفاده از یک تابع انرژی (نامنفی) و یک سیستم دینامیکی است که این دو بیان‌کننده‌ی مدل‌های شبکه‌های عصبی مصنوعی متناظر مسائل بهینه‌سازی هستند. سیستم دینامیکی بیان شده معمولاً یک دستگاه معادلات دیفرانسیل غیرخطی مرتبه‌ی اول است. انتظار می‌رود که برای یک نقطه‌ی آغازین نقطه‌ی پایداری دستگاه معادلات دیفرانسیل غیرخطی به‌دست آمده، جواب بهینه‌ی مساله‌ی بهینه‌سازی اصلی باشد. بنابراین برای یک مدل متناظر با مسائل بهینه‌سازی، اصل اساسی استفاده از شبکه‌های عصبی در اینگونه مسائل به‌صورت زیر بیان می‌شود:

”برای یک نقطه‌ی آغازین دلخواه، نقطه‌ی تعادل دستگاه معادلات دیفرانسیل غیرخطی به‌دست آمده جواب بهینه‌ی مساله‌ی اصلی است و برعکس.”



مدل‌های مطرح شده‌ی متناظر مسائل مختلف بهینه‌سازی را می‌توان به دو قسمت مدل‌های دوگانی و مدل‌های جریمه‌ای تقسیم نمود. نظریه‌ی دوگانی و توابع جریمه‌ای دو نظریه‌ی بسیار مهم در مسائل بهینه‌سازی هستند که اکثر این مسائل بر مبنای این دو روش کلاسیک قابل حل هستند. در نظریه‌ی توابع جریمه‌ای معمولاً از مدل‌های گرادیانی برای معرفی مدل مورد نظر استفاده می‌شود، ولی در نظریه‌ی دوگانی از مدل‌های اولیه-دوگان برای حل این مسائل استفاده می‌شود. اگر بتوان متناظر با هر مساله‌ی بهینه‌سازی، روش مشخصی را ارائه نمود که آن روش برای حل آن مساله شرایط لازم و کافی را برآورده سازد، آنگاه می‌توان متناظر با آن روش، یک مدل شبکه‌ی عصبی برای مساله‌ی مورد نظر بسازیم.

در دهه‌های گذشته، شبکه‌های عصبی به طور گسترده‌ای برای بهینه‌سازی بررسی شده‌اند. برای اولین بار هاپفیلد و تانک شبکه‌های عصبی را برای مسائل برنامه‌ریزی خطی پیشنهاد دادند [۳۰] و [۶۳]. کار آنها توسط بسیاری از محققان جهت بررسی سایر مدل‌های شبکه‌ی عصبی برای حل مسائل برنامه‌ریزی الهام گرفته شده است. کندی و چو [۳۵] یک شبکه‌ی عصبی برای حل مسائل برنامه‌ریزی غیرخطی ارائه دادند. مشخص شده است که مدل شبکه‌ی عصبی شامل پارامترهای جریمه متناهی است و تنها جواب‌های تقریبی را تولید می‌کند. برای جلوگیری از استفاده از پارامترهای جریمه، در سال‌های اخیر روش‌های دیگر بسیاری ارائه شده است، [۱۱]، [۲۱]، [۶۴]، [۶۹]، [۸۱] و [۸۴] را ببینید.

مسائل برنامه‌ریزی درجه دوم در طیف گسترده‌ای از کاربردهای علمی و مهندسی، تحلیل رگرسیون، پردازش سیگنال، بازسازی تصویر، کنترل ربات و غیره به وجود می‌آیند. همچنین، برخی مسائل بهینه‌سازی غیرخطی معمولاً با یک مسئله‌ی برنامه‌ریزی درجه دوم تقریب زده می‌شوند. به طور کلی، مسئله‌ی برنامه‌ریزی درجه دوم می‌تواند به شکل زیر نوشته شود

$$\text{Min } \frac{1}{2}x^T Wx + c^T x$$

s.t

$$Ax = b,$$

$$x \geq 0.$$

بسیاری از شبکه‌های عصبی برای مسئله‌ی فوق ارائه شده‌اند. برای مثال، برخی شبکه‌های عصبی اولیه-دوگان ارائه شدند [۹]، [۱۳]، [۶۷] و [۶۸]. به منظور ساده‌سازی ساختار شبکه‌ی عصبی دوگان، شبکه‌ی عصبی دوگان ساده شده معرفی شد [۴۰] و [۳۳]. با استفاده از قضیه‌ی تصویر، چندین شبکه‌ی عصبی تصویری برای حل مسئله‌ی برنامه‌ریزی درجه دوم توسعه داده شدند [۱۷]، [۲۳]، [۲۲]، [۲۴]، [۳۱] و [۴۱]، [۴۲]، [۴۷]، [۸۰] و [۸۳] که به طور کلی به جواب‌های بهینه‌ی دقیق همگرا بودند. در [۴۱] و [۴۲]، ليو وانگ چند شبکه‌ی عصبی بازگشتی تک‌لایه را برای حل مسائل برنامه‌ریزی درجه دوم پیشنهاد دادند. شبکه‌های عصبی بازگشتی تک‌لایه پیچیدگی ساختاری ساده‌تری نسبت به سایر شبکه‌های عصبی مانند شبکه‌ی لاگرانژی و شبکه‌ی تصویری دارند.

تا به حال، تنها چند شبکه‌ی عصبی برای حل مسائل برنامه‌ریزی درجه دوم زیر با محدودیت‌های تساوی و نامساوی توسعه یافته است.

$$\begin{aligned} \text{Min} \quad & \frac{1}{2}x^T W x + c^T x \\ \text{s.t} \quad & \\ & Ax = b, \\ & Bx \leq d. \end{aligned} \tag{۱.۳}$$

که در آن  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ ،  $W$  یک ماتریس  $n \times n$  نیمه معین مثبت متقارن حقیقی،  $d \in \mathbb{R}^p$  و  $b \in \mathbb{R}^m$ ،  $B \in \mathbb{R}^{p \times n}$ ،  $\text{rank}(A) = m$  ( $0 < m < n$ )،  $A \in \mathbb{R}^{m \times n}$ ،  $c \in \mathbb{R}^n$  ژانگ و وانگ [۸۵] و شیا و همکاران [۷۰] با محدودیت‌های تساوی و نامساوی یک طرفه همانند حالات خاص محدودیت‌های کران دوطرفه عمل کردند. نشان دادن کران پایین  $-\infty$  یا تا کران  $+\infty$  نیاز به اعداد به اندازه کافی بزرگ دارد که برای طراحی شبکه‌های عصبی نامساعد است. عفتی و ناظمی [۱۵] برای حل این مسئله با شرایط کاراش-کوهن-تاکر ( $KKT$ ) یک شبکه‌ی عصبی پیشنهاد دادند. با این حال، این شبکه‌ی عصبی دارای نورون‌های بیشتری است و نیاز به شرایط همگرایی قوی‌تری دارد.

با انگیزه‌ی مباحث فوق، هدف این پایان نامه توسعه‌ی یک شبکه‌ی عصبی جدید برای حل (۱.۳) است که متغیرهای حالت کمتر، پیچیدگی ساختاری کمتر و شرایط همگرایی قوی‌تری دارد.

### ۱.۳ مدل شبکه‌ی عصبی پیشنهادی

مسالهی برنامه‌ریزی درجه دوم محدب به صورت کلی زیر را در نظر بگیرید:

$$\begin{aligned} \text{Min} \quad & \frac{1}{2}x^T W x + c^T x \\ \text{s.t} \quad & \\ & Ax = b \\ & Bx \leq d, \end{aligned} \tag{۲.۳}$$

که در آن  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ ،  $W \in \mathbb{R}^{n \times n}$  یک ماتریس نیمه معین مثبت متقارن،  $d \in \mathbb{R}^p$  و  $b \in \mathbb{R}^m$ ،  $B \in \mathbb{R}^{p \times n}$ ،  $\text{rank}(A) = m$  ( $0 < m < n$ )،  $A \in \mathbb{R}^{m \times n}$ ،  $c \in \mathbb{R}^n$  است.

می‌توان مدل شبکه‌ی عصبی زیر را برای به دست آوردن جواب بهینه‌ی مسئله‌ی (۲.۳) به صورت زیر به کار برد [۷۹]:

$$\begin{cases} \frac{dx}{dt} = -(I - P)[Wx + c + B^T(y + Bx - d)^+] - Q(Ax - b) \\ \frac{dy}{dt} = -\frac{1}{\tau}y + \frac{1}{\tau}(y + Bx - d)^+, \end{cases} \tag{۳.۳}$$

که در آن

$$P = A^T(AA^T)^{-1}A \quad Q = A^T(AA^T)^{-1}$$

$$(y + Bx - d)^+ = ([y + Bx - d]_1^+, [y + Bx - d]_2^+, \dots, [y + Bx - d]_m^+)$$

$$[y + Bx - d]_k^+ = \max\{(y + Bx - d)_k, 0\} \quad k = 1, 2, \dots, m$$

لم ۱.۱.۳. [۷۹]  $x^*$  جواب بهینه‌ی (۲.۳) است اگر و تنها اگر  $y^* \geq 0$  موجود باشد به طوری که در شرایط زیر صدق کند:

$$(I - P)[Wx^* + c + B^T y^*] + Q(Ax^* - b) = 0 \quad (۴.۳)$$

$$(y^* + Bx^* - d)^+ - y^* = 0 \quad (۵.۳)$$

برهان. اگر  $x^*$  جواب بهینه‌ی (۲.۳) باشد، مطابق [۸۸] شرایط KKT برای مسائل بهینه‌سازی محدب،  $(x^*, \lambda^*, y^*)$  و  $y^* \geq 0$  موجودند به طوری که در معادلات زیر صدق می‌کنند.

$$Wx^* + c + A^T \lambda^* + B^T y^* = 0 \quad (۶.۳)$$

$$Ax^* = b \quad (۷.۳)$$

$$d - Bx^* \geq 0, \quad y^* \geq 0, \quad (y^*)^T(d - Bx^*) = 0 \quad (۸.۳)$$

به آسانی دیده می‌شود که (۵.۳) و (۸.۳) معادل هستند. به این معنی که

$$d - Bx^* \geq 0, \quad y^* \geq 0, \quad (y^*)^T(d - Bx^*) = 0 \iff (y^* + Bx^* - d)^+ - y^* = 0$$

در ادامه، نشان می‌دهیم که جواب‌های (۴.۳) و جواب‌های (۶.۳) و (۷.۳) معادلند. از (۶.۳) داریم

$$-A(Wx^* + c + A^T \lambda^* + B^T y^*) = 0$$

با جمع کردن معادله‌ی بالا با معادله‌ی (۷.۳) داریم

$$Ax^* - A(Wx^* + c) - AA^T \lambda^* - AB^T y^* = b$$

بنابراین

$$A^T \lambda^* = A^T(AA^T)^{-1}(Ax^* - b) - A^T(AA^T)^{-1}(Wx^* + c + B^T y^*) \quad (۹.۳)$$

با جایگذاری (۹.۳) در معادله‌ی (۶.۳) داریم

$$[I - A^T(AA^T)^{-1}A](Wx^* + c + B^T y^*) + A^T(AA^T)^{-1}(Ax^* - b) = 0$$

با قرار دادن  $P = A^T(AA^T)^{-1}A$  و  $Q = A^T(AA^T)^{-1}$  می‌توان معادله‌ی فوق را به صورت زیر نوشت

$$(I - P)[Wx^* + c + B^T(y^* + Bx^* - d)^+] + Q(Ax^* - b) = 0$$

برعکس، اگر  $y^*$  موجود باشد به طوری که  $(x^*, y^*)^T$  در معادله‌ی (۴.۳) صدق کند با ضرب  $A$  در طرفین معادله‌ی (۴.۳) داریم

$$A(I - P)[Wx^* + c + B^T(y^* + Bx^* - d)^+] + AQ(Ax^* - b) = 0$$

توجه کنید که  $AQ = I$  و  $A(I - P) = 0$  پس داریم  $Ax^* - b = 0$ . در نتیجه از (۴.۳) داریم  $(I - P)(Wx^* + c + B^T y^*) = 0$ . با قرار دادن  $\lambda^* = -(AA^T)^{-1}A(Wx^* + c + B^T y^*)$  داریم

$$\begin{aligned} Wx^* + c + A^T \lambda^* + B^T y^* &= Wx^* + c + B^T y^* - A^T (AA^T)^{-1} A(Wx^* + c + B^T y^*) \\ &= (I - P)(Wx^* + c + B^T y^*) \\ &= 0 \end{aligned}$$

□ پس حکم ثابت شده است.

ملاحظه ۲.۱.۳. با استفاده از لم ۱.۱.۳، به آسانی می‌بینیم که  $x^*$  جواب بهینه‌ی مساله‌ی (۲.۳) است اگر و تنها اگر وجود داشته باشد  $y^* \geq 0$  به طوری که  $(x^*, y^*)^T$  نقطه‌ی تعادل شبکه‌ی عصبی ارائه شده در (۳.۳) باشد. بنابراین، زمانی که شبکه‌ی عصبی به نقطه‌ی تعادل همگرا است، مسیر حرکت  $x(t)$  به جواب بهینه‌ی مساله‌ی (۲.۳) همگرا می‌باشد.

### ۱.۱.۳ تحلیل پایداری

در این بخش نشان خواهیم داد که مدل شبکه‌ی عصبی (۳.۳) همگرای سراسری است.

لم ۳.۱.۳. [۷۹] فرض کنید  $(x^*, y^*)^T \in \Omega^e$  یک نقطه‌ی تعادل شبکه‌ی عصبی (۳.۳) باشد و

$$\begin{aligned} V(x, y) &= \frac{1}{\gamma} (x - x^*)^T W (x - x^*) - (x - x^*)^T B^T y^* + \frac{1}{\gamma} \|(y + Bx - d)^+\|^2 + \frac{1}{\gamma} \|y^*\|^2 \\ &\quad - y^T y^* + \frac{1}{\gamma} \|x - x^*\|^2 + \frac{1}{\gamma} \|y - y^*\|^2. \end{aligned} \tag{۱۰.۳}$$

در این صورت

$$V(x, y) \geq \frac{1}{\gamma} \|x - x^*\|^2 + \frac{1}{\gamma} \|y - y^*\|^2; \tag{۱۱.۳}$$

$$\begin{aligned} \frac{dV}{dt} &\leq -\|(I - P)[Wx + c + B^T(y + Bx - d)^+] + Q(Ax - b)\|^2 \\ &\quad - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2. \end{aligned} \tag{۱۲.۳}$$

برهان. برای اثبات (۱۱.۳) داریم، از آنجا که تابع  $\|(y + Bx - d)^+\|^2$  محدب و مشتق پذیر می‌باشد و

$$\nabla \|(y + Bx - d)^+\|^2 = 2 \begin{bmatrix} B^T(y + Bx - d)^+ \\ (y + Bx - d)^+ \end{bmatrix}$$

در نتیجه

$$\|(y + Bx - d)^+\|^2 - \|y^*\|^2 \geq \Upsilon(x - x^*)^T B^T y^* + \Upsilon(y - y^*)^T y^*$$

یعنی

$$\frac{1}{\Upsilon} \|(y + Bx - d)^+\|^2 + \frac{1}{\Upsilon} \|y^*\|^2 - (x - x^*)^T B^T y^* - y^T y^* \geq 0.$$

و با توجه به این که ماتریس  $W$  نیمه‌معین مثبت متقارن است، داریم

$$V(x, y) \geq \frac{1}{\Upsilon} \|x - x^*\|^2 + \frac{1}{\Upsilon} \|y - y^*\|^2$$

همچنین برای اثبات (۱۲.۳) داریم

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial x} \frac{dx}{dt} + \frac{\partial V}{\partial y} \frac{dy}{dt} \\ &= - [W(x - x^*) - B^T y^* + B^T (y + Bx - d)^+ + x - x^*]^T \\ &\quad \times [(I - P)(Wx + c + B^T (y + Bx - d)^+) + Q(Ax - b)] \\ &\quad - \frac{1}{\Upsilon} [y - (y + Bx - d)^+]^T \times [(y + Bx - d)^+ - \Upsilon y^* + y] \\ &= [-W(x - x^*) + x - x^* + B^T ((y + Bx - d)^+ - y^*)]^T \\ &\quad \times [(I - P)(W(x - x^*) + B^T ((y + Bx - d)^+ - y^*) + P(x - x^*)) \\ &\quad - \frac{1}{\Upsilon} [y - (y + Bx - d)^+]^T \times [y - (y + Bx - d)^+ + \Upsilon(y + Bx - d)^+ - \Upsilon y^*] \\ &= - (x - x^*)^T W(I - P)W(x - x^*) \\ &\quad - (x - x^*)^T \times W(I - P)B^T [(y + Bx - d)^+ - y^*] - (x - x^*)^T WP(x - x^*) \\ &\quad - (x - x^*)^T (I - P)W(x - x^*) - (x - x^*)^T (I - P)B^T [(y + Bx - d)^+ - y^*] \\ &\quad - (x - x^*)^T P(x - x^*) - [(y + Bx - d)^+ - y^*]^T \times B(I - P)W(x - x^*) \\ &\quad - [(y + Bx - d)^+ - y^*]^T BB^T [(y + Bx - d)^+ - y^*] - [(y + Bx - d)^+ - y^*]^T \\ &\quad \times BP(x - x^*) - \frac{1}{\Upsilon} \|y - (y + Bx - d)^+\|^2 \\ &\quad + [(y + Bx - d)^+ - y^*]^T [(y + Bx - d)^+ - y] \end{aligned}$$

توجه کنید که  $P(I - P) = 0$ ،  $P^2 = P$ ،  $(I - P)^2 = I - P$  و

داریم  $(y + Bx - d)^+ - y = Bx - d + (-y - Bx - d)^+$

$$\begin{aligned}
 \frac{dV}{dt} &= - (x - x^*)^T W (I - P)^\top W (x - x^*) \\
 &\quad - \top (x - x^*)^T W (I - P) B^T [(y + Bx - d)^+ - y^*] - (x - x^*)^T P^\top (x - x^*) \\
 &\quad - [(y + Bx - d)^+ - y^*]^T B B^T [(y + Bx - d)^+ - y^*] - (x - x^*)^T W (x - x^*) \\
 &\quad - (x - x^*)^T B^T [(y + Bx - d)^+ - y^*] - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2 \\
 &\quad + [(y + Bx - d)^+ - y^*]^T \times [Bx - d + (-y - Bx + d)^+] \\
 &= - \|(I - P)W(x - x^*) + (I - P)B^T[(y + Bx - d)^+ - y^*] + P(x - x^*)\|^2 \\
 &\quad - (x - x^*)^T W (x - x^*) - (x - x^*)^T B^T [(y + Bx - d)^+ - y^*] \\
 &\quad - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2 + [(y + Bx - d)^+ - y^*]^T \\
 &\quad \times [B(x - x^*) + (Bx^* - d) + (-y - Bx - d)^+] \\
 &= - \|(I - P)[Wx + c + B^T(y + Bx - d)^+] + Q(Ax - b)\|^2 \\
 &\quad - (x - x^*)^T W (x - x^*) - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2 \\
 &\quad + [(y + Bx - d)^+]^T (Bx^* - d) - (y^{*T})(Bx^* - d) \\
 &\quad + [(y + Bx - d)^+]^T (-y - Bx + d)^+ - (y^{*T})(-y - Bx + d)^+ \\
 &= - \|(I - P)[Wx + c + B^T(y + Bx - d)^+] + Q(Ax - b)\|^2 \\
 &\quad - (x - x^*)^T W (x - x^*) - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2 \\
 &\quad + [(y + Bx - d)^+]^T (Bx^* - d) - (y^{*T})(-y - Bx + d)^+ \\
 &\leq - \|(I - P)[Wx + c + B^T(y + Bx - d)^+] \\
 &\quad + Q(Ax - b)\|^2 - \frac{1}{\gamma} \|y - (y + Bx - d)^+\|^2
 \end{aligned}$$

□ که در معادله‌ی فوق با استفاده از (۵.۳) اثبات تمام است.

**ملاحظه ۴.۱.۳.**  $V(x, y)$  معرفی شده در (۱۰.۳) یک تابع لیاپانوف می‌باشد، بنابراین، شبکه‌ی عصبی (۳.۳) پایدار به مفهوم لیاپانوف می‌باشد.

**قضیه ۵.۱.۳.** [۷۹] برای هر نقطه‌ی اولیه‌ی  $(x(0), y(0)) \in \mathbb{R}^{n+p}$ ، یک جواب پیوسته منحصر به فرد  $(x(t), y(t)) \in \mathbb{R}^{n+p}$  از (۳.۳) برای  $t \geq 0$  وجود دارد.

برهان. فرض کنید

$$E(x, y) = -(I - P)(Wx + c - (y + BX - d)^+) - Q(Ax - b) - \frac{1}{\gamma} y + \frac{1}{\gamma} (y + Bx - d)^+.$$

برای هر  $(x, y), (x', y') \in \mathbb{R}^{n+p}$  داریم

$$\begin{aligned} \|E(x, y) - E(x', y')\| &\leq \|(I - P)[Wx + c - (y + BX - d)^+] + Q(Ax - b) \\ &\quad - (I - P)[Wx' + c - (y' + BX' - d)^+] - Q(Ax' - b)\| \\ &\quad + \|\frac{1}{\rho}y - \frac{1}{\rho}(y + Bx - d)^+ - \frac{1}{\rho}y' + \frac{1}{\rho}(y' + Bx' - d)^+\| \\ &\leq (\|(I - P)W\| + \|(I - P)B\| + \|P\| + \|B\|/\rho)\|x - x'\| \\ &\quad + (\|I - P\| + 1)\|y - y'\| \end{aligned}$$

از این رو،  $E(x, y)$  روی  $\mathbb{R}^{n+p}$  پیوسته‌ی لیپشیتز<sup>۱</sup> است. از قضیه‌ی وجودی موضعی معادله‌ی دیفرانسیل معمولی [۵۲] نتیجه می‌شود یک جواب منحصر به فرد (۳.۳) روی  $[0, T)$  وجود دارد. بنا به لم ۳.۱.۳ می‌دانیم که  $V$  نسبت به  $t$  یک تابع نزولی است، لذا

$$\frac{1}{\rho}\|x - x^*\|^2 \leq V(x(0), y(0)) \leq V(x(t), y(t)), \quad \forall t \geq 0 \quad (۱۳.۳)$$

نشان می‌دهد مسیر حالت شبکه‌ی عصبی (۳.۳) کراندار است. در نتیجه  $T = +\infty$ . این اثبات را کامل می‌کند.  $\square$

**قضیه ۶.۱.۳.** [۷۹] مسیر حالت شبکه‌ی عصبی (۳.۳) برای هر نقطه‌ی اولیه‌ی  $(x(0), y(0)) \in \mathbb{R}^{n+p}$  به یک نقطه‌ی تعادل همگرا است. به ویژه، شبکه‌ی عصبی (۳.۳) با هر نقطه‌ی اولیه‌ی  $(x(0), y(0)) \in \mathbb{R}^{n+p}$  هنگامی که  $\Omega^e$  نقطه‌ی تعادل منحصر به فرد دارد به طور سراسری پایدار مجانبی است.

برهان. تعریف کنید

$$D(x, y) = \|(I - P)[Wx + c + B^T(y + Bx - d)^+] + Q(Ax - b)\|^2 + \frac{1}{\rho}\|y - (y + Bx - d)^+\|^2.$$

در این صورت  $D(x, y) = 0$  اگر و تنها اگر  $(x, y)$  یک نقطه‌ی تعادل شبکه‌ی عصبی (۳.۳) باشد. از اثبات قضیه‌ی ۵.۱.۳ نتیجه می‌گیریم مسیر حالت  $(x(t), y(t))$  شبکه‌ی عصبی (۳.۳) کراندار است. بنابراین، دنباله‌ی صعودی  $\{t_n\}$  با  $\lim_{n \rightarrow \infty} t_n \rightarrow \infty$  و نقطه‌ی حدی  $(\hat{x}, \hat{y})^T$  وجود دارند به طوری که  $\lim_{n \rightarrow \infty} y(t_n) \rightarrow \hat{y}$  و  $\lim_{n \rightarrow \infty} x(t_n) \rightarrow \hat{x}$

حال، ثابت می‌کنیم  $(\hat{x}, \hat{y}) \in \Omega^e$  و مسیر حالت  $(x(t), y(t))$  به طور کلی به نقطه‌ی تعادل  $(\hat{x}, \hat{y})$  همگرا است.

در مرحله اول، ثابت می‌کنیم که  $D(\hat{x}, \hat{y}) = 0$  یعنی  $(\hat{x}, \hat{y})$  یک نقطه‌ی تعادل شبکه‌ی عصبی (۳.۳) است. اگر این مورد برقرار نباشد، آنگاه  $D(\hat{x}, \hat{y}) > 0$ . چون  $D(x, y)$  به ترتیب نسبت به  $x$  و  $y$  پیوسته است،  $\varepsilon > 0$ ،  $q > 0$  و همسایگی  $\varepsilon$  از  $(\hat{x}, \hat{y})^T$ ،

$$B((\hat{x}, \hat{y}), \varepsilon) = \{(x, y)^T : \|x - \hat{x}\| + \|y - \hat{y}\| \leq \varepsilon\}$$

<sup>۱</sup>Lipschitz

وجود دارند به طوری که برای هر  $(x, y)^T \in B((\hat{x}, \hat{y}), \varepsilon)$ ،  $D(x, y) > q$ .

با توجه به اینکه  $\lim_{n \rightarrow \infty} x(t_n) \rightarrow \hat{x}$  و  $\lim_{n \rightarrow \infty} y(t_n) \rightarrow \hat{y}$ ، عدد صحیح مثبت  $N$  وجود دارد به طوری که برای هر  $n \geq N$ ،  $\|x(t_n) - \hat{x}\| \leq \frac{1}{4}\varepsilon$  و  $\|y(t_n) - \hat{y}\| \leq \frac{1}{4}\varepsilon$ .

از (۳.۳) و کراندار بودن  $(x(t), y(t))$  نتیجه می‌گیریم  $\dot{x}$  و  $\dot{y}$  نیز کراندار هستند که این کران با  $M$  نشان داده می‌شود. با در نظر گرفتن  $n \geq N$ ،  $t \in [t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M}]$  داریم

$$\begin{aligned} \|x(t) - \hat{x}\| + \|y(t) - \hat{y}\| &\leq \|x(t) - x(t_n)\| + \|y(t) - y(t_n)\| + \|x(t_n) - \hat{x}\| + \|y(t_n) - \hat{y}\| \\ &= \|\dot{x}(\xi_1)\| \times |t - t_n| + \|\dot{y}(\xi_2)\| \times |t - t_n| \\ &\quad + \|x(t_n) - \hat{x}\| + \|y(t_n) - \hat{y}\| \\ &\leq 2M|t - t_n| + \frac{\varepsilon}{4} \leq \varepsilon \end{aligned}$$

از این رو، برای هر  $n \geq N$ ،  $t \in [t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M}]$  داریم  $(x, y)^T \in B((\hat{x}, \hat{y}), \varepsilon)$ . بنابراین برای هر  $n \geq N$ ،  $t \in [t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M}]$  نتیجه می‌شود  $D(x(t), y(t)) > q$ . چون  $\lim_{n \rightarrow \infty} t_n = \infty$  و اندازه‌ی لبگ<sup>۲</sup> مجموعه‌ی  $t \in \cup_{n \geq N} [t_n - \frac{\varepsilon}{\lambda M}, t_n + \frac{\varepsilon}{\lambda M}]$  بی‌نهایت است، پس داریم

$$\int_0^{\infty} D(x(t), y(t)) dt = \infty. \quad (۱۴.۳)$$

با این حال از رابطه (۱۲.۳)

$$\begin{aligned} \int_0^{\infty} D(x(t), y(t)) dt &= \lim_{s \rightarrow \infty} \int_0^s D(x(t), y(t)) dt \\ &\leq - \lim_{s \rightarrow \infty} \int_0^s \dot{V}(x(t), y(t)) dt \\ &= \lim_{s \rightarrow \infty} [V(x(0), y(0)) - V(x(s), y(s))] \\ &\leq V(x(0), y(0)). \end{aligned} \quad (۱۵.۳)$$

که (۱۴.۳) را نقض می‌کند. لذا  $D(\hat{x}, \hat{y}) = 0$  که یعنی  $(\hat{x}, \hat{y})^T \in \Omega^e$ .

در مرحله دوم، ثابت می‌کنیم مسیر حالت  $(x(t), y(t))$  به طور سراسری به نقطه‌ی تعادل  $(\hat{x}, \hat{y})$  همگرا است.

<sup>۲</sup>Lebeque



تابع لیپانوف

$$\hat{V}(x, y) = \frac{1}{\alpha} W(x(t) - \hat{x}) - (x(t) - \hat{x})^T B^T \hat{y} + \frac{1}{\alpha} \|(y + Bx - d)^+\|^2 + \frac{1}{\alpha} \|\hat{y}\|^2 - y^T \hat{y} + \frac{1}{\alpha} \|x(t) - \hat{x}\|^2 + \frac{1}{\alpha} \|y(t) - \hat{y}\|^2$$

را تعریف کنید که  $\hat{V}(\hat{x}, \hat{y}) = 0$ .

چون

$$\lim_{n \rightarrow \infty} y(t_n) \rightarrow \hat{y} \text{ و } \lim_{n \rightarrow \infty} x(t_n) \rightarrow \hat{x}$$

به ازای هر  $\varepsilon > 0$  وجود دارد به طوری که  $\hat{V}(x(t_k), y(t_k)) < \varepsilon$ .

بنا به لم ۳.۱.۳ داریم  $\frac{1}{\alpha} \|x(t) - \hat{x}\|^2 + \frac{1}{\alpha} \|y(t) - \hat{y}\|^2 \leq \hat{V}(x(t), y(t)) \leq \hat{V}(x(t_k), y(t_k)) \leq \varepsilon$  و  $\hat{V}$  نزولی است.

بنابراین، برای هر  $t \geq t_k$  داریم

$$\frac{1}{\alpha} \|x(t) - \hat{x}\|^2 + \frac{1}{\alpha} \|y(t) - \hat{y}\|^2 \leq \hat{V}(x(t), y(t)) \leq \hat{V}(x(t_k), y(t_k)) \leq \varepsilon$$

که یعنی  $\lim_{n \rightarrow \infty} y(t) = \hat{y}$  و  $\lim_{n \rightarrow \infty} x(t) = \hat{x}$ . لذا مسیر حالت شبکه‌ی عصبی (۳.۳) همگرای سراسری به نقطه‌ی تعادل  $(\hat{x}, \hat{y})^T$  همگرا می‌شود.

به ویژه، اگر  $\Omega^e = \{(x^*, y^*)\}$ ، آنگاه بنا به تحلیل فوق مسیر حالت  $(x(t), y(t))$  با هر نقطه‌ی اولیه‌ی  $(x(0), y(0))$  به  $(x^*, y^*)$  میل خواهد نمود. پس این شبکه‌ی عصبی پایدار جانبی سراسری است.  $\square$

### ۲.۱.۳ مثال عددی

مثال ۷.۱.۳. [۷۹] مساله‌ی برنامه ریزی درجه دوم زیر را در نظر بگیرید.

$$\min \quad 3x_1^2 + 3x_2^2 + 4x_3^2 + 5x_4^2 + 3x_1x_2 + 5x_1x_3 + x_2x_4 - 11x_1 - 5x_4$$

s.t.

$$3x_1 - 3x_2 - 2x_3 + x_4 = 0$$

$$4x_1 + x_2 - x_3 - 2x_4 = 0$$

$$-x_1 + x_2 \leq -1$$

$$-2 \leq 3x_1 + x_3 \leq 4.$$

(۱۶.۳)

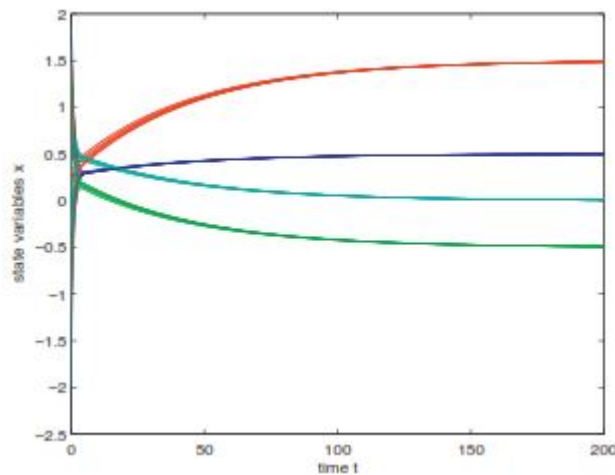
در این مساله ماتریس های متناظر عبارتند از

$$W = \begin{bmatrix} 6 & 3 & 5 & 0 \\ 3 & 6 & 0 & 1 \\ 5 & 0 & 8 & 0 \\ 0 & 1 & 0 & 10 \end{bmatrix}, \quad c = \begin{bmatrix} -11 \\ 0 \\ 0 \\ -5 \end{bmatrix},$$

$$B = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ -3 & 0 & -1 & 0 \end{bmatrix}, \quad d = \begin{bmatrix} -1 \\ 4 \\ 2 \end{bmatrix},$$

$$A = \begin{bmatrix} 3 & -3 & -2 & 1 \\ 4 & 1 & -1 & -2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

این مساله جواب بهینه‌ی  $x^* = (0.5, -0.5, 1.5, 0)^T$  دارد. شبکه‌ی عصبی را با استفاده از  $10$  نقطه‌ی اولیه‌ی تصادفی بررسی کرده‌ایم. تمامی نتایج شبیه سازی که شبکه‌ی عصبی (۳.۳) را نشان می‌دهند به این جواب بهینه همگرا می‌شوند. نتایج شبیه سازی در شکل ۱.۳ نشان داده شده است.



شکل ۱.۳: رفتار مدل شبکه‌ی عصبی (۳.۳) در مثال ۷.۱.۳

مثال ۸.۱.۳. [۷۹] مسئله‌ی برنامه ریزی درجه دوم زیر را در نظر بگیرید.

$$\min \quad x_1^2 + x_2^2 + x_3^2 - 2x_1 - 3x_2$$

s.t.

$$2x_1 + x_2 + x_3 + x_4 = 7 \quad (17.3)$$

$$x_1 + x_2 + 2x_3 + x_4 = 6$$

$$x_i \geq 0, \quad i = 1, 2, 3, 4.$$

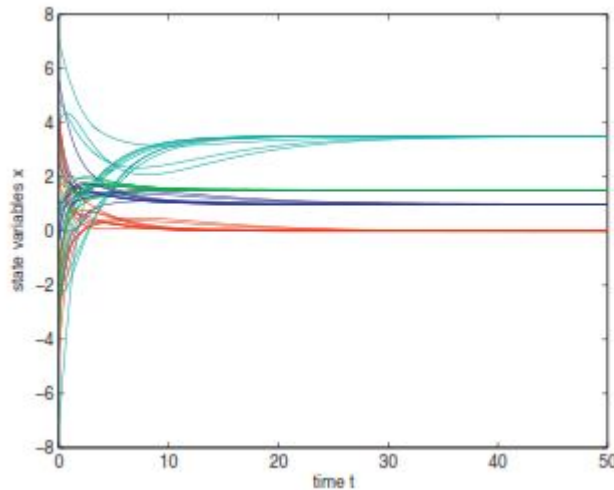
صورت های ماتریسی متناظر عبارتند از

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} -2 \\ -3 \\ 0 \\ 0 \end{bmatrix},$$

$$A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 6 \end{bmatrix},$$

$$B = -I, \quad d = 0.$$

این مساله جواب بهینه  $x^* = (1, 1/5, 0, 3/5)^T$  دارد. شبکه‌ی عصبی را با استفاده از  $1^\circ$  نقطه‌ی اولیه‌ی تصادفی بررسی کرده‌ایم. تمامی نتایج شبیه سازی که شبکه‌ی عصبی (۳.۳) را نشان می‌دهند به این جواب بهینه همگرا می‌شوند. نتایج شبیه سازی در شکل ۲.۳ نشان داده شده است.



شکل ۲.۳: رفتار مدل شبکه‌ی عصبی (۳.۳) در مثال (۸.۱.۳)

مثال ۹.۱.۳. [۷۹] مساله‌ی برنامه ریزی درجه دوم زیر را در نظر بگیرید.

$$\min \quad 11x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 + 6x_1x_3 - 4x_1$$

s.t.

$$2x_1 + 2x_2 + x_3 = 0$$

$$-x_1 + x_2 \leq 1$$

$$3x_1 + x_3 \leq 4$$

$$-6 \leq x_i \leq 6, \quad i = 1, 2, 3.$$

(۱۸.۳)

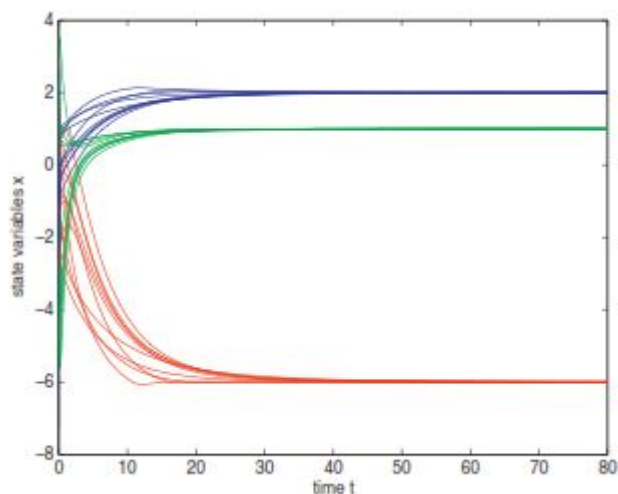
صورت های ماتریسی متناظر عبارتند از

$$W = \begin{bmatrix} 11 & -1 & 3 \\ -1 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} -4 \\ 0 \\ 0 \end{bmatrix},$$

$$A = \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}, \quad b = 0,$$

$$B = \begin{bmatrix} 3 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad d = \begin{bmatrix} 4 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \end{bmatrix}.$$

این مساله جواب بهینه‌ی  $x^* = (2, 1, -6)^T$  دارد. شبکه‌ی عصبی را با استفاده از  $10$  نقطه‌ی اولیه‌ی تصادفی بررسی کرده‌ایم. تمامی نتایج شبیه‌سازی که شبکه‌ی عصبی (۳.۳) را نشان می‌دهند به این جواب بهینه همگرا می‌شوند. نتایج شبیه‌سازی در شکل ۳.۳ نشان داده شده است.



شکل ۳.۳: رفتار مدل شبکه‌ی عصبی (۳.۳) در مثال (۹.۱.۳)



## فصل ۴

# حل یک مساله‌ی برنامه‌ریزی غیرخطی محدب نما با یک شبکه عصبی جدید کارا

### ۱.۴ معرفی

بهینه‌سازی غیرخطی اصطلاحی برای توصیف مساله‌ی بهینه‌سازی است هنگامی که تابع هدف یا قیدها خطی نباشند. بنابراین بسیاری از مسائل واقعی می‌توانند به‌طور مناسبی به شکل یک مساله‌ی غیرخطی نمایش داده شده یا تقریب زده شوند. در دهه‌های گذشته، تلاش‌ها برای حل چنین مسائل غیرخطی‌ای به صورت مطلوب، پیشرفت سریعی داشته است. فرآیندهای عددی مختلفی برای حل مسائل غیرخطی مطرح شده است.

در سال ۱۹۸۶، یک شبکه‌ی عصبی توسط تانک و هافیلد برای حل مسائل برنامه‌ریزی خطی پیشنهاد شد. از آن زمان به بعد بسیاری از دانشمندان انواع مختلفی از تکنیک‌ها را به منظور حل مساله‌ی برنامه‌ریزی غیرخطی در این زمینه مطرح کردند. به دلیل خصوصیت خوب شبکه‌های عصبی در همگرایی سریع و قابلیت محاسبه‌ی موازی، دانشمندان از همان ابتدای مطرح شدن این بحث حساب ویژه‌ای روی آن باز نمودند.

اخیراً، شبکه‌ی عصبی به‌طور گسترده‌ای برای بهینه‌سازی مطالعه شده و نتایج بسیار خوبی به دست آمده است. به‌عنوان مثال، ژیا و وانگ [۷۱]–[۷۵] شبکه‌های عصبی مختلفی را برای حل مسائل برنامه‌ریزی محدب خطی یا غیرخطی توسعه دادند. ژیا و همکاران [۷۶] و [۷۷] شبکه‌ی عصبی تصویری

را برای حل مسائل نامساوی وردشی یکنواخت مطرح کردند. یانگ و کاو [۸۲] یک شبکه‌ی عصبی تصویری برای حل مسائل برنامه‌ریزی غیرخطی مشتق ناپذیر ارائه کردند. لیو و وانگ [۴۲] یک شبکه‌ی عصبی تک‌لایه‌ای را به منظور حل مسائل برنامه‌ریزی غیرخطی محدب غیرهموار با قیود تساوی خطی مطرح کردند. برای برنامه‌ریزی ناهموار کلی‌تر، ژئو و بیان [۵]–[۶] یک شبکه‌ی عصبی بازگشتی بر پایه‌ی روش پارامتری جریمه‌ای و شبه‌گرادیان مطرح نمودند. چنگ و همکاران [۱۰] یک شبکه‌ی عصبی مصنوعی بازگشتی ناهموار به منظور حل مسائل برنامه‌ریزی محدب ناهموار با قیود نامساوی محدب و قیود تساوی خطی مطرح نمودند. بر پایه‌ی روش مرتب‌سازی، تیوخونوف، بیان و ژئو [۷۸] یک شبکه‌ی عصبی برای حل مسائل برنامه‌ریزی محدب ناهموار مقید مطرح نمودند. به هر صورت شبکه‌های عصبی فوق، تنها می‌توانستند برای حل مساله‌ی برنامه‌ریزی محدب استفاده شوند.

در حالی که دانشمندان مدت زیادی به برنامه‌ریزی محدب توجه داشتند، به تدریج کشف کردند که برنامه‌ریزی محدب نما نیز به اندازه‌ی برنامه‌ریزی محدب مهم است. کاربردهای بسیاری از برنامه‌ریزی محدب‌نما در حوزه‌های مختلفی مانند برنامه‌ریزی کسری، شیمی، کامپیوتر، برنامه‌ریزی تولید، برنامه‌ریزی مالی شرکت [۵] و [۶] را ببینید) ارائه شده است. هرچند این نوع بهینه‌سازی سخت‌تر از برنامه‌ریزی محدب است. همان‌طور که می‌دانیم، اکثر شبکه‌های عصبی که در بالا مطرح شده نمی‌توانند به‌طور مستقیم به منظور حل برنامه‌ریزی محدب‌نما به کار گرفته شوند. سپس چند شبکه‌ی عصبی جدید، به منظور حل برنامه‌ریزی محدب‌نما مطرح شدند. به‌عنوان مثال، هو و وانگ [۳۲]، یک شبکه‌ی عصبی برای حل نامساوی‌های وردشی و مسائل بهینه‌سازی محدب‌نمای متناظر ارائه کردند. گوئو و همکاران [۲۶] یک شبکه‌ی عصبی تک‌لایه‌ای برای حل مسائل بهینه‌سازی محدب نما با قیود تساوی خطی مطرح کردند. بر مبنای تابع جریمه و روش گرادیان، لیو و همکاران [۴۳] یک شبکه‌ی عصبی تک‌لایه‌ای بازگشتی برای حل مسائل بهینه‌سازی محدب‌نما با قیدهای کراندار و قیود تساوی خطی ارائه کردند، هرچند که تابع جریمه به‌طور عمده‌ای به پارامترهای دقیق جریمه وابسته است. با الهام از مطالب گذشته و به منظور کاهش پیچیدگی، در این فصل یک روش جدید و ساده شده‌ی شبکه‌ی عصبی را برای برنامه‌ریزی محدب‌نما با قیود خطی ارائه می‌دهیم [۵۷].

## ۲.۴ مدل شبکه‌ی عصبی

مساله‌ی برنامه‌ریزی محدب نما به صورت زیر مطرح می‌شود:

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{s.t} \quad & \\ & Ax = b \end{aligned} \tag{۱.۴}$$

که  $x \in \mathbb{R}^n$  و  $A \in \mathbb{R}^{m \times n}$ . تابع  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  دوبار مشتق‌پذیر محدب نما است. در اینجا روی روش شبکه‌ی عصبی با این فرض که مساله‌ی بهینه‌سازی حداقل یک جواب بهینه دارد، متمرکز می‌شویم [۵۷]. بدون از دست دادن کلیت، فرض می‌کنیم  $A$  یک ماتریس با رتبه‌ی سطری کامل است

(۱.۴) را ببینید. در این بخش فرمول‌بندی معادل مسالهی بهینه‌سازی (۱.۴) را بحث می‌کنیم و یک مدل شبکه‌ی عصبی می‌سازیم. از لم ۳.۲.۲ می‌فهمیم که  $x^*$  یک جواب بهینه‌ی مسالهی بهینه‌سازی (۱.۴) است اگر و فقط اگر  $x^*$  در رابطه‌ی زیر صدق کند

$$\nabla f(x^*)(x - x^*) \geq 0, \forall x \in \Omega = \{x \in \mathbb{R}^n | Ax = b\}.$$

از این پس نماد  $\Omega$  مجموعه‌ی  $\{x \in \mathbb{R}^n | Ax = b\}$  را جز در حالت خاص نشان می‌دهد، به ویژه اگر  $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$  وجود داشته باشند که در رابطه‌ی زیر صدق کنند

$$\begin{cases} \nabla f(x^*) - A^T y^* = 0, \\ Ax^* - b = 0, \end{cases} \quad (۲.۴)$$

آنگاه برای هر  $x \in \Omega = \{x \in \mathbb{R}^n | Ax = b\}$  داریم

$$\nabla f(x^*)(x - x^*) = y^*(Ax - Ax^*) = 0.$$

بنابراین، با استفاده از لم ۳.۲.۲،  $x^*$  یک جواب بهینه‌ی مسالهی بهینه‌سازی (۱.۴) است. از این رو، برای حل مسالهی بهینه‌سازی (۱.۴)، فقط نیاز به حل دستگاه (۲.۴) داریم. از فرمول اول معادله‌ی (۲.۴) داریم  $A \nabla f(x^*) - AA^T y^* = 0$ . چون ماتریس  $A$  رتبه‌ی سطری کامل است، واضح است که  $AA^T$  معکوس‌پذیر است. پس

$$y^* = (AA^T)^{-1} A \nabla f(x^*). \quad (۳.۴)$$

با جایگزین کردن فرمول بالا در معادله‌ی (۲.۴) داریم

$$\nabla f(x^*) - A^T (AA^T)^{-1} A \nabla f(x^*) = 0, \quad (۴.۴)$$

به این معنی که

$$(I - A^T (AA^T)^{-1} A) \nabla f(x^*) = 0. \quad (۵.۴)$$

فرض کنید

$$P = A^T (AA^T)^{-1} A, \quad (۶.۴)$$

$$(I - P) \nabla f(x^*) = 0.$$

بر مبنای نتایج بالا، مدل شبکه‌ی عصبی زیر معرفی می‌شود:

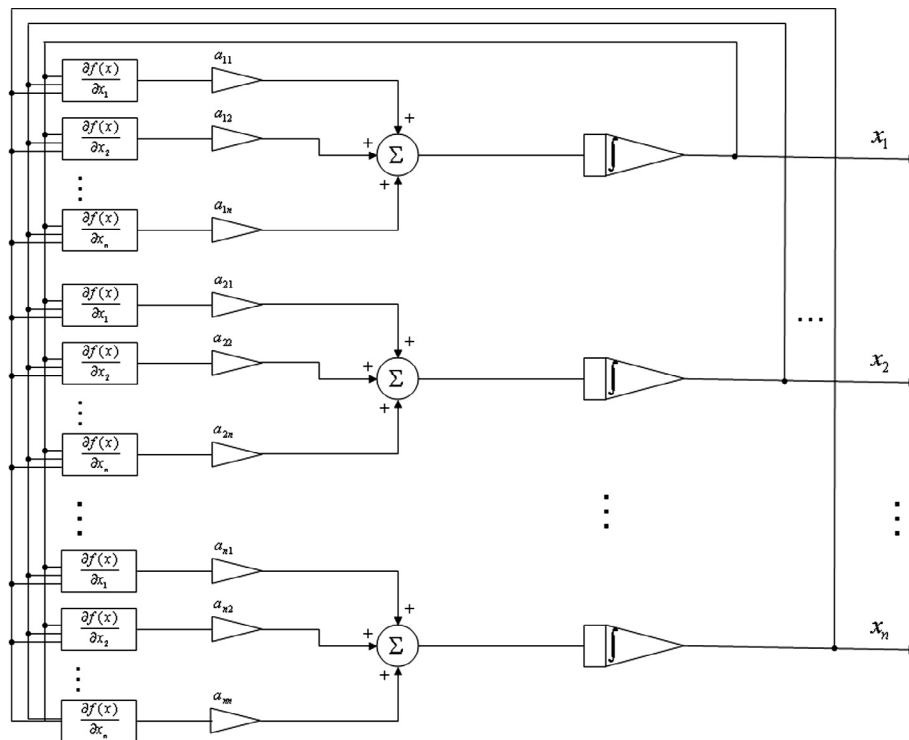
$$\dot{x}(t) = -(I - P) \nabla f(x(t)) \quad (۷.۴)$$

که  $\dot{x}(t) = (\dot{x}_1(t), \dots, \dot{x}_n(t))^T$  و  $\nabla f(x) = (\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n})^T$  و  $I \in \mathbb{R}^{n \times n}$  ماتریس همانی است.



از مباحث بالا می‌فهمیم که جواب بهینه‌ی مساله‌ی (۱.۴) با یافتن نقطه‌ی تعادل شبکه‌ی عصبی (۷.۴) به دست می‌آید. از طرف دیگر اگر نقطه‌ی تعادل  $\bar{x}$  شبکه‌ی عصبی (۷.۴) با  $A\bar{x} = b$  برابر باشد، پس  $\bar{x}$  جواب بهینه‌ی مساله‌ی (۱.۴) است. به عبارت دیگر، محدب نما نسبت به محدب، شرایط ضعیف تری دارد. اگر مساله‌ی بهینه‌سازی محدب نما را توسط شبکه‌ی عصبی (۷.۴) حل کنیم، پس مساله‌ی بهینه‌سازی محدب را می‌توانیم به‌عنوان یک مورد خاص، حل کنیم.

معادله‌ی (۷.۴) با مدار شکل ۱.۴ بررسی می‌شود.  $\{a_{i,j}\}_{n \times n}$  ها عناصر ماتریس  $(I - P)$  هستند. به دلیل پیش زمینه‌ی کاربردی، بیشتر روی مطالعه‌ی شبکه‌ی عصبی تمرکز می‌کنیم. بعلاوه شبکه‌ی عصبی تک‌لایه نسبت به سایر شبکه‌ها به دلیل پیچیدگی کمتر و تأثیر زمان واقعی خوب مزیت‌هایی دارد.



شکل ۱.۴: مداری از شبکه‌ی عصبی پیشنهادی

### ۳.۴ پایداری و تحلیل همگرایی

در این بخش در مورد همگرایی سراسری و پایداری لیاپانوف شبکه‌ی عصبی، بحث می‌کنیم. ابتدا قبل از رسیدن به هدف، یک لم را بیان می‌کنیم.

لم ۱.۳.۴. [۵۷] برای هر نقطه‌ی ابتدایی  $x_0 \in \mathbb{R}^n$ ، یک جواب موضعی یکتای  $x(t)$  برای شبکه‌ی عصبی (۷.۴) با  $x(0) = x_0$  وجود دارد. بعلاوه، اگر  $x_0 \in \Omega$  باشد، آنگاه برای  $t \geq 0$ ،  $x(t) \in \Omega$ .

برهان. چون  $\nabla f(x)$  به طور پیوسته مشتق پذیر روی مجموعه‌ی  $\Omega$  است، اثبات پیوستگی لیم شیتز موضعی  $-(I - P)\nabla f(x)$  واضح می‌باشد. بدیهی است که یک جواب یکتا برای مساله‌ی مقدار اولیه‌ی شبکه‌ی عصبی (۷.۴) وجود دارد. پس دومین بخش از لم اثبات می‌شود.

فرض کنید  $B(x(t)) = \frac{1}{2}\|Ax(t) - b\|^2$ . بر اساس قانون زنجیره ای داریم

$$\frac{dB(x(t))}{dt} = \dot{B}(x(t)) = \nabla B(x)\dot{x}(t) = -(Ax(t) - b)^T A(I - P)\nabla f(x(t)).$$

چون  $A(I - P) = 0$ ، بنابراین  $\dot{B}(x(t)) = 0$  است یعنی  $B(x(t))$  ثابت است. از طرف دیگر، برای  $x_0 \in \Omega$ ،  $B(x_0) = 0$ ، برای هر  $t \geq 0$  داریم  $B(x(t)) = 0$ . نتیجه می‌شود که برای هر  $t \geq 0$ ،  $x(t) \in \Omega$ ،  $x(0) = x_0 \in \Omega$  اگر  $\square$

**تعریف ۲.۳.۴ [۵۷]** تابع به طور شعاعی بی کران، یک تابع  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  است که برای آن

$$\|x\| \rightarrow \infty \implies f(x) \rightarrow \infty.$$

توجه داشته باشید که نرم استفاده شده در این تابع می‌تواند هر نرم تعریف شده روی  $\mathbb{R}^n$  باشد و رفتار تابع در راستای محورها لزوماً مشخص نمی‌کند که به طور شعاعی بی کران است یا خیر، یعنی برای به طور شعاعی بی کران بودن این شرط باید در راستای هر مسیری که به

$$\|x\| \rightarrow \infty$$

منجر می‌شود، بررسی گردد.

اکنون نتایج اصلی را بر مبنای همگرایی سراسری شبکه‌ی عصبی (۷.۴) معرفی می‌کنیم.

**قضیه ۳.۳.۴ [۵۷]** برای هر نقطه‌ی اولیه‌ی  $x_0 \in \Omega$ ، جواب شبکه‌ی عصبی (۷.۴) پایدار و سراسری لیاپانوف است و یک جواب برای مساله‌ی بهینه سازی (۱.۴) می‌باشد.

برهان. از لم ۱.۳.۴ برای هر نقطه‌ی اولیه‌ی  $x_0 \in \Omega$ ، همیشه جواب موضعی یکتای  $x(t)$  در  $\Omega$  وجود دارد. بنابراین برای هر  $t \geq 0$ ،  $Ax(t) = b$ . فرض کنید  $x^*$  یک جواب بهینه‌ی مساله‌ی بهینه‌سازی (۱.۴) است. پس بنا به لم ۶.۲.۲ داریم:

$$\nabla f(x(t))(x(t) - x^*) \geq 0. \quad (۸.۴)$$

یک تابع لیاپانوف را به صورت زیر تعریف می‌کنیم:

$$V(x) = f(x) - f(x^*) + \frac{1}{2}\|x - x^*\|^2$$

واضح است که برای هر  $x \in \Omega$ ،  $V(x^*) = 0$  و  $V(x) \geq 0$ . سپس مشتق  $V(x)$  را به صورت زیر محاسبه می‌کنیم:

$$\begin{aligned} \frac{dV(x(t))}{dt} &= \nabla V(x(t))\dot{x}(t) = -\nabla f(x(t))(I - P)\nabla f(x(t)) \\ &\quad - (x(t) - x^*)(I - P)\nabla f(x(t)). \end{aligned}$$

چون

$$(I - P)^\dagger = (I - A^T(AA^T)^{-1}A)(I - A^T(AA^T)^{-1}A) = I - P$$

و

$$(x(t) - x^*)P\nabla f(x(t)) = 0,$$

داریم:

$$\frac{dV(x(t))}{dt} = -\nabla f(x(t))(I - P)^\dagger \nabla f(x(t)) - (x(t) - x^*)\nabla f(x(t)).$$

براساس (۸.۴) داریم:

$$\frac{dV(x(t))}{dt} \leq -\nabla f(x(t))(I - P)^\dagger \nabla f(x(t)) = -\|(I - P)\nabla f(x(t))\|^2 \leq 0.$$

بنابراین، برای هر نقطه‌ی اولیه‌ی  $x_0 \in \Omega$ ، یک جواب سراسری یکتای  $x(t)$  وجود دارد که به مفهوم لیاپانوف ثابت است.

از ساختار  $V$ ، می‌فهمیم که  $V$  به‌طور شعاعی بی‌کران است چون  $V(x) \geq \frac{1}{\gamma}\|x - x^*\|$ . از طرف دیگر  $V(x(t)) \leq V(x_0)$  چون  $\frac{dV}{dt} \leq 0$ . بنابراین  $x(t)$  کراندار است. به این معنی که زیردنباله‌ی همگرای  $x(t_k)$  وجود دارد به‌طوری‌که  $\lim_{k \rightarrow \infty} x(t_k) = \bar{x}$ . به دلیل کرانداری  $x(t)$  و پیوستگی  $\nabla f$  واضح است که  $\dot{x}(t)$  کراندار است. اکنون تعریف می‌کنیم  $M = \sup_{t \geq 0} \|\dot{x}(t)\| < +\infty$  و

$$H(x) = \|(I - P)\nabla f(x)\|^2.$$

ثابت می‌کنیم  $H(\bar{x}) = 0$ .

فرض کنید  $H(\bar{x}) \neq 0$  و  $H(\bar{x}) > \varepsilon$  است. از پیوستگی  $H(x)$ ،  $\varepsilon > 0$  و  $\delta > 0$  وجود دارند به‌طوری‌که برای  $H(x) \geq \varepsilon, \forall x \in B(\bar{x}, \delta)$  که  $B(\bar{x}, \delta) = \{x \in \mathbb{R}^n \mid \|x - \bar{x}\| \leq \delta\}$  به عبارت دیگر، چون  $\lim_{k \rightarrow \infty} x(t_k) = \bar{x}$ ، لذا عدد مثبت  $N$  وجود دارد به‌طوری‌که برای  $k > N$ ،  $\|x(t_k) - \bar{x}\| \leq \frac{\delta}{4}$  وقتی که  $t \in \{t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}\}$  و  $k > N$  داریم:

$$\|x(t) - \bar{x}\| \leq \|x(t) - x(t_k)\| + \|x(t_k) - \bar{x}\| \leq M|t - t_k| + \frac{\delta}{4} \leq \delta$$

یعنی برای هر  $t \in [t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}]$  داریم  $H(x(t)) \geq \varepsilon$ .

مجموعه‌ی اندازه پذیر لبگ  $\cup_{k \geq N} [t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}]$  را جلوتر بررسی می‌کنیم. چون اندازه‌ی لبگ این مجموعه بی‌نهایت است و  $H(x(t)) \geq \varepsilon$  داریم  $\int_0^\infty H(x(t))dt = \infty$ . به عبارت دیگر، چون  $\frac{dV(x(t))}{dt} \leq -H(x(t)) \cdot V(x(t))$ ، صعودی و کراندار است. بنابراین ثابت  $V_0$  وجود دارد به‌طوری‌که  $\lim_{t \rightarrow \infty} V(x(t)) = V_0$ ، پس داریم:

$$\begin{aligned} \int_0^\infty H(x(t))dt &= \lim_{s \rightarrow \infty} \int_0^s H(x(t))dt \leq - \lim_{s \rightarrow \infty} \int_0^s \dot{V}(x(t))dt \\ &= - \lim_{s \rightarrow \infty} [V(x(s)) - V(x(0))] = -V_0 + V(x(0)) < \infty. \end{aligned}$$

این تناقض است. بنابراین  $H(\bar{x}) = 0$  یعنی  $\bar{x}$  یک نقطه‌ی تعادل شبکه‌ی عصبی (۷.۴) است. در ضمن از لم ۱.۳.۴ واضح است که  $\bar{x} \in \Omega$ . بنابراین  $\bar{x}$  جواب بهینه‌ی مسأله‌ی بهینه‌سازی (۱.۴) است. اکنون ثابت می‌کنیم هنگامی که  $x(t) \rightarrow \bar{x}, t \rightarrow +\infty$

تابع لیپانوف دیگر  $\bar{V}$  را به صورت زیر تعریف می‌کنیم:

$$\bar{V}(x) = f(x) - f(\bar{x}) + \frac{1}{\eta} \|x - \bar{x}\|^2.$$

ثابت می‌کنیم که  $\bar{V}(\bar{x}) = 0$  و  $\bar{V}(x) \geq 0$ .  $\bar{V}$  به طور پیوسته مشتق پذیر است و  $\frac{d\bar{V}}{dt} \leq 0$ . با توجه به اینکه  $\lim_{k \rightarrow \infty} x(t_k) = \bar{x}$ ، لذا  $\lim_{t \rightarrow \infty} \bar{V}(x(t)) = \bar{V}(\bar{x}) = 0$ . بنابراین  $\forall \varepsilon > 0, q \geq 0$  وجود دارد به طوری که برای هر  $t > t_q$  داریم  $\bar{V}(x) < \varepsilon$ .  
به علاوه

$$\bar{V}(x) \geq \frac{1}{\eta} \|x - \bar{x}\|^2,$$

که برای  $t > t_q$  به صورت زیر است:

$$\frac{1}{\eta} \|x - \bar{x}\|^2 \leq \bar{V}(x) < \varepsilon.$$

به این معنی که  $\lim_{k \rightarrow \infty} x(t) = \bar{x}$ . بنابراین مسیرهای شبکه‌ی عصبی (۷.۴) به جواب بهینه‌ی مسأله‌ی بهینه‌سازی (۱.۴) همگرا است. این اثبات را کامل می‌کند.  $\square$

**ملاحظه ۴.۳.۴.** بر اساس قضیه‌ی ۳.۳.۴، برای هر نقطه‌ی اولیه‌ی  $x_0 \in \Omega$ ، مسیر شبکه‌ی عصبی (۷.۴) به جواب بهینه‌ی مسأله‌ی بهینه‌سازی (۱.۴) همگرا است. اگرچه اگر نقطه‌ی اولیه‌ی  $x_0 \notin \Omega$ ، نمی‌توانیم همگرایی شبکه‌ی عصبی (۷.۴) را به جواب بهینه ثابت کنیم. به عبارت دیگر، نیاز به انتخاب نقطه‌ی اولیه‌ای که در مجموعه‌ی  $\Omega$  باشد، داریم.

در حالت خاص زمانی که  $\nabla f(x)$  اکیداً شبه‌یکنواخت باشد، نتیجه‌ی زیر را داریم که مستقیماً از قضیه‌ی ۳.۳.۴ به دست می‌آید. اثبات را حذف می‌کنیم؛ زیرا شبیه اثبات قضیه‌ی ۳ در [۶۲] است.

**نتیجه ۵.۳.۴.** فرض کنید  $\nabla f$  در  $\Omega$  اکیداً شبه‌یکنواخت باشد، در این صورت شبکه‌ی عصبی (۷.۴) به صورت نمایی به جواب بهینه‌سازی مسأله‌ی (۱.۴) همگرا است.

در ادامه فصل یک مثال عددی ساده و دو کاربرد برای رسیدگی و تحلیل نتایج نظری فصول قبل جهت نشان دادن عملکرد و اثر شبکه‌ی عصبی نشان داده شده است. نتیجه‌ی پایان نامه در بخش آخر این فصل مشخص می‌شود.

## ۴.۴ یک مثال عددی ساده

این بخش، مثالی که اثرات شبکه‌ی عصبی پیشنهادی (۷.۴) در حل مسأله‌ی بهینه‌سازی (۱.۴) را

توضیح می‌دهد، آورده شده است.

مثال ۱.۴.۴. [۵۷] یک مثال از مساله برنامه‌ریزی غیرخطی به صورت زیر در نظر بگیرید:

$$\begin{aligned} \text{Min } & f(x) \\ \text{s.t } & \\ & Ax = b \end{aligned} \quad (۹.۴)$$

که در آن

$$f(x) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2,$$

$$A = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 0 & 1 & 2 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix}.$$

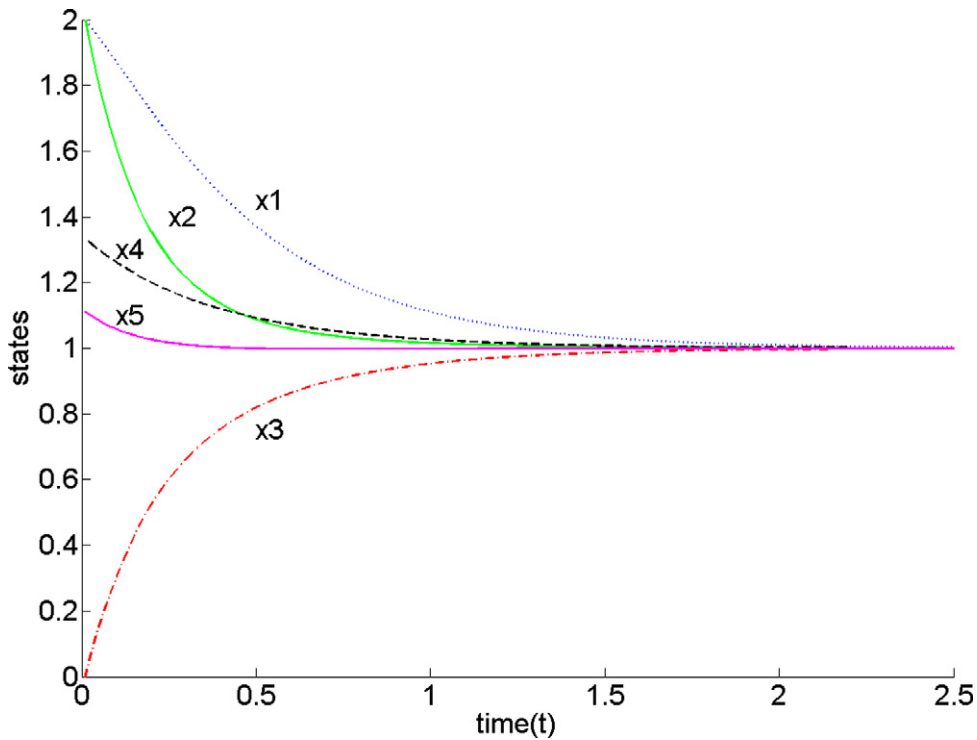
در این مساله، بررسی اینکه تابع هدف  $f(x)$  محدب‌نما است، خیلی آسان است. این مساله جواب بهینه‌ی  $x^* = (1, 1, 1, 1, 1)^T$  دارد. پس می‌توانیم از شبکه‌ی عصبی (۷.۴) برای حل این مساله‌ی بهینه‌سازی استفاده کنیم. نتایج نشان می‌دهد که مسیر شبکه با هر نقطه‌ی اولیه در  $\Omega = \{x | Ax = b\}$  برای جواب بهینه‌ی  $x^*$  همگرا خواهد بود. مثلاً شکل ۲.۴ مسیر شبکه‌ی عصبی (۷.۴) را با نقطه‌ی اولیه‌ی  $x_0 = (2, 2, 0, \frac{4}{3}, \frac{1}{9})$  نشان می‌دهد.

## ۵.۴ کاربرد در تجارت

در این بخش یک کاربرد در تجارت را معرفی می‌کنیم [۵۷]. قبل از کاربرد در تصمیمات تجاری، ابتدا برخی از مفاهیم اولیه‌ی آن را ارائه می‌دهیم.

بهینه‌سازی موضوعی مهم در تصمیم منطقی تجارت است. هنگامی که تصمیم گیرنده بیشتر از یک هدف دارد، یا وقتی معنی نامعلومی در سیستم دارد، بهینه‌سازی یک چارچوب منطقی برای تصمیمات موثر است. مدل انتخاب مجموعه‌ی کلاسیک، مثال خوبی از مساله‌ی بهینه‌سازی در این زمینه است.

احتمالاً هر سرمایه‌گذاری با چنین مساله‌ای مواجه می‌شود، با توجه به دسترسی اوراق بهادار، چگونه بهترین راه برای ترکیب اوراق بهادار به سبد سهام تعیین می‌شود؟ برای پاسخ به این سوال، در مورد مجموعه‌هایی که مقدار مورد نظر سود را در شرایط پایین‌ترین سطح ریسک نشان می‌دهند، بررسی شدند [۷۸]. مجموعه‌ای که از ۳ سرمایه‌ی موجود  $A$ ،  $B$  و  $C$  تشکیل می‌شود را در نظر می‌گیریم. نرخ انتظار بازگشت و کوواریانس دو موضوع مهم آماری هستند. نرخ انتظار بازگشت در مورد آنچه که از سهام به عنوان نسبت سود در ماه آینده انتظار داشتیم می‌گوید زمانی که یک سهام بالاتر از مقدار مورد



شکل ۲.۴: مسیری از شبکه‌ی عصبی (۷.۴) با نقطه‌ی اولیه‌ی  $x_0 = (2, 2, 0, \frac{4}{3}, \frac{10}{9})$  در مثال ۱.۴.۴.

نظر سوددهی دارد، کوواریانس میان دو سهام، در مورد سایر سهامها و گرایش یا عدم گرایش آنها، اطلاعاتی ارائه می‌کند. فرض کنید نسبت مورد انتظار سود این سهامها به  $E(r_A)$ ،  $E(r_B)$  و  $E(r_C)$  اختصاص دارد. ماتریس کوواریانس سهامها به صورت زیر نشان داده شده است:

$$\begin{bmatrix} Cov(r_A, r_A) & Cov(r_B, r_A) & Cov(r_C, r_A) \\ Cov(r_A, r_B) & Cov(r_B, r_B) & Cov(r_C, r_B) \\ Cov(r_A, r_C) & Cov(r_B, r_C) & Cov(r_C, r_C) \end{bmatrix}.$$

واضح است که مجموع وزن برای تمام سهامها در مجموعه باید برابر با ۱ باشد. از نماد  $x_A$ ،  $x_B$  و  $x_C$  به ترتیب برای سهامهای  $A$ ،  $B$  و  $C$  استفاده می‌کنیم. در زمینه‌ی سرمایه‌گذاری، واریانس کوچکتر به معنای ریسک کمتر است. بنابراین مساله‌ی بهینه‌سازی اصلی در مورد وزن مجموعه و کاهش واریانس نسبت به سود مورد نظر است. واریانس بازگشت به سبد سهام داده شده است ([۷۸] را ببینید):

$$\sigma^2(r_p) = x_A^2 \sigma^2(r_A) + x_B^2 \sigma^2(r_B) + x_C^2 \sigma^2(r_C) + 2x_A x_B Cov(r_A, r_B) + 2x_A x_C Cov(r_A, r_C) + 2x_B x_C Cov(r_B, r_C) \quad (10.4)$$

نرخ سود یک مجموعه به صورت زیر به دست می‌آید [۷۸]:

$$E(r_p) = x_A E(r_A) + x_B E(r_B) + x_C E(r_C) \quad (11.4)$$

**مثال ۱.۵.۴.** در این مثال، سود مورد انتظار  $E(r_p) = ۱۰\%$  و ارقام برای واریانس، کواریانس و سود مورد نظر سهام  $A$  و  $B$  و  $C$  در جدول زیر نشان داده شده است.

	$E(r_A)$	$E(r_B)$	$E(r_C)$
مقدار	۵٪	۱۰٪	۱۵٪
کواریانس	$A$	$B$	$C$
$A$	۰٫۲۵	۰٫۱۵	۰٫۱۷
$B$	۰٫۱۵	۰٫۲۱	۰٫۰۹
$C$	۰٫۱۷	۰٫۰۹	۰٫۲۸

جدول ۱.۴: سودهای مورد انتظار

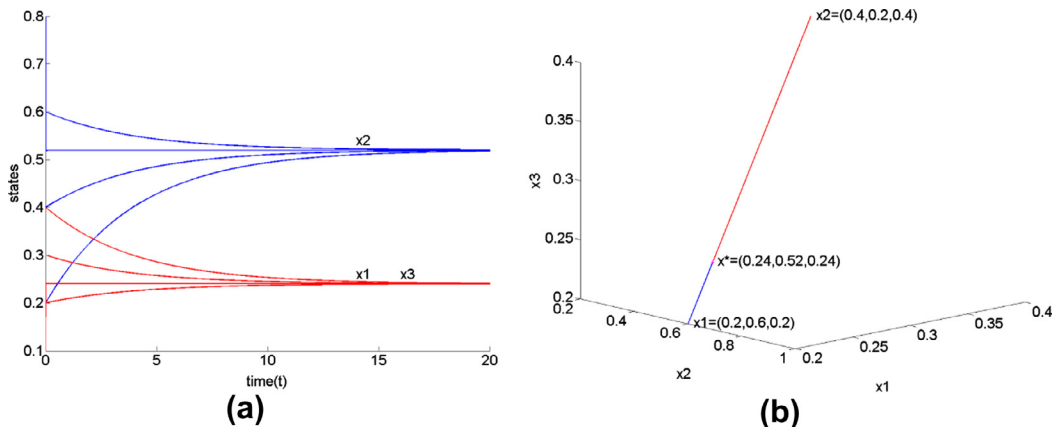
مساله‌ی بهینه‌سازی با قیدهای خطی زیر بررسی می‌شوند:

$$\begin{aligned} \text{Min } \sigma^2(r_p) &= 0.25x_A^2 + 0.21x_B^2 + 0.28x_C^2 + 0.3x_{AB} \\ &+ 0.34x_{AC} + 0.18x_{BC} \\ \text{s.t} \\ &0.05x_A + 0.1x_B + 0.15x_C = 0.1, \\ &x_A + x_B + x_C = 1. \end{aligned}$$

واضح است که تابع هدف  $\sigma^2(r_p)$  محدب است. چون یک تابع محدب باید محدب‌نما باشد، برای حل این مساله می‌توانیم از شبکه‌ی عصبی (۷.۴) استفاده کنیم. با استفاده از شبکه‌ی عصبی (۷.۴) برای حل این مساله، نتایج شبیه‌سازی نشان می‌دهد که مسیر به نقطه‌ی تعادل  $x^* = (0.24, 0.52, 0.24)^T$  همگرا می‌شود. حال، حداقل واریانس مجموعه برابر ۰٫۱۶۶۸ است.

شکل (a) ۳.۴ رفتار گذرای شبکه‌ی مورد نظر با نقاط اولیه‌ی  $(0.1, 0.8, 0.1)^T$ ،  $(0.2, 0.6, 0.2)^T$ ، و  $(0.3, 0.4, 0.3)^T$  را نشان می‌دهد.

شکل (b) ۳.۴ مسیر شبکه‌ی عصبی در سیستم سه بعدی با نقاط اولیه‌ی هم‌هنگ  $(0.2, 0.6, 0.2)^T$  و  $(0.4, 0.2, 0.4)^T$  را نشان می‌دهد. از شکل ۳.۴ می‌توانیم ببینیم که اگر به ترتیب از وزن‌های  $(0.24, 0.52, 0.24)^T$  برای خرید سهام  $A$ ،  $B$  و  $C$  در شرایط مشابه بازه مورد انتظار ۱۱٪ استفاده کنیم، پایین‌ترین ریسک را پذیرفته‌ایم.



شکل ۳.۴: رفتار گذرای شبکه‌ی عصبی (۷.۴) با نقطه‌ی ابتدایی متفاوت در مثال ۱.۵.۴.

## ۶.۴ کاربرد در شیمی

کاربرد دیگر در زمینه‌ی شیمی است [۷۷]. وقتی یک واکنش شیمیایی پیچیده را بررسی می‌کنیم، بیشتر برای توسعه‌ی ساده و روش‌های عمومی محاسبه‌ی تعادل شیمیایی مهم است. روشی که مطرح خواهیم کرد برای تمام واکنش‌های متعادل کاربرد دارد. در این روش تعادل شیمیایی با حداقل انرژی آزاد سیستم به دست می‌آید [۸۲].

انرژی آزاد یک ترکیب حاوی  $n$  نمونه‌ی شیمیایی محتوی  $x_i$  مول ( $1 \leq i \leq n$ ) از  $i$ مین نوع به صورت زیر بیان می‌شود:

$$F(x) = \sum_{i=1}^n f_i, \quad (12.4)$$

که در آن  $x = (x_1, x_2, \dots, x_n)^T$  مجموعه تعداد مول،  $f_i = x_i [c_i + \ln(\frac{x_i}{x_i^0})]$ ،  $c_i = (f^\circ / RT)_i +$ ،  $\bar{x} = \sum_{i=1}^n x_i$ ،  $\ln P$ ، فشار کل در اتمسفر،  $F^\circ / RT$  تابع انرژی آزاد مولال استاندارد است و در بیشتر موارد از جدول به دست می‌آید. سپس تعیین نقطه‌ی تعادل برای یافتن مینیمم (۱۲.۴) تحت شرایط توازن جرم به صورت زیر است:

$$\sum_{i=1}^n a_{ij} x_i = b_j, \quad j = 1, 2, \dots, m,$$

یعنی

$$Ax = b,$$

که در آن  $A \in \mathbb{R}^{m \times n}$ ،  $b \in \mathbb{R}^m$ ،  $a_{ij}$  فرمول تعداد اتم‌های عنصر  $j$  در نمونه‌ی  $i$  و  $b_j$  تعداد کل وزن اتمی عنصر  $j$  است. با توجه به اینکه تابع (۱۲.۴) محدب است بنابراین، می‌توانیم از شبکه‌ی عصبی پیشنهادی برای حل مسأله‌ی بهینه‌سازی استفاده کنیم.



مثال ۱.۶.۴. [۵۷] به‌عنوان یک مثال، در مورد اجزای تعادل گاز و احتراق ترکیب هیدروژن و اکسیژن در دمای  $3500K$  و فشار  $750\text{psi}$  ( $\ln P = 3.932$ ) بحث خواهیم کرد. مقادیر تابع انرژی آزاد در جدول ۲.۴ داده شده‌اند.

$i$	۱	۲	۳	۴	۵	۶	۷
انواع	$H$	$H_2$	$H_2O$	$N$	$N_2$	$NH$	$N$
$F^\circ/RT$	$-10,021$	$-21,096$	$-37,986$	$-9,846$	$-28,653$	$-18,918$	$-28,032$
	۸	۹	۱۰				
	$OH$	$O_2$	$O$				
	$-26,111$	$-30,594$	$-14,640$				

جدول ۲.۴: مقادیر تابع انرژی آزاد

با جایگزین کردن مقادیر داده شده در معادله‌ی (۱۲.۴) داریم:

$$\text{Min } F(x) = \sum_{i=1}^n c_i x_i + \sum_{i=1}^n x_i \ln\left(\frac{x_i}{\bar{x}}\right) \quad (13.4)$$

s.t

$$Ax = b,$$

که در آن

$$A = \begin{bmatrix} 1 & 2 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 1 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix},$$

$$c = (c_1, c_2, \dots, c_n)^T = (-6,089, -17,164, -34,054, -5,914, -24,721, -14,986, -24,100, -10,708, -26,662, -22,179)^T.$$

از شبکه‌ی عصبی (۷.۴) برای حل مساله استفاده می‌کنیم. شبیه‌سازی نشان می‌دهد که مسیرهای با نقاط اولیه‌ی

$$(0.8, 0.4, 0.1, 0.4, 0.2, 0.1, 0.1, 0.1, 0.3, 0.1)^T$$

و

$$(0.4, 0.6, 0.1, 0.2, 0.3, 0.1, 0.1, 0.1, 0.2, 0.1)^T$$

به نقطه‌ی تعادل

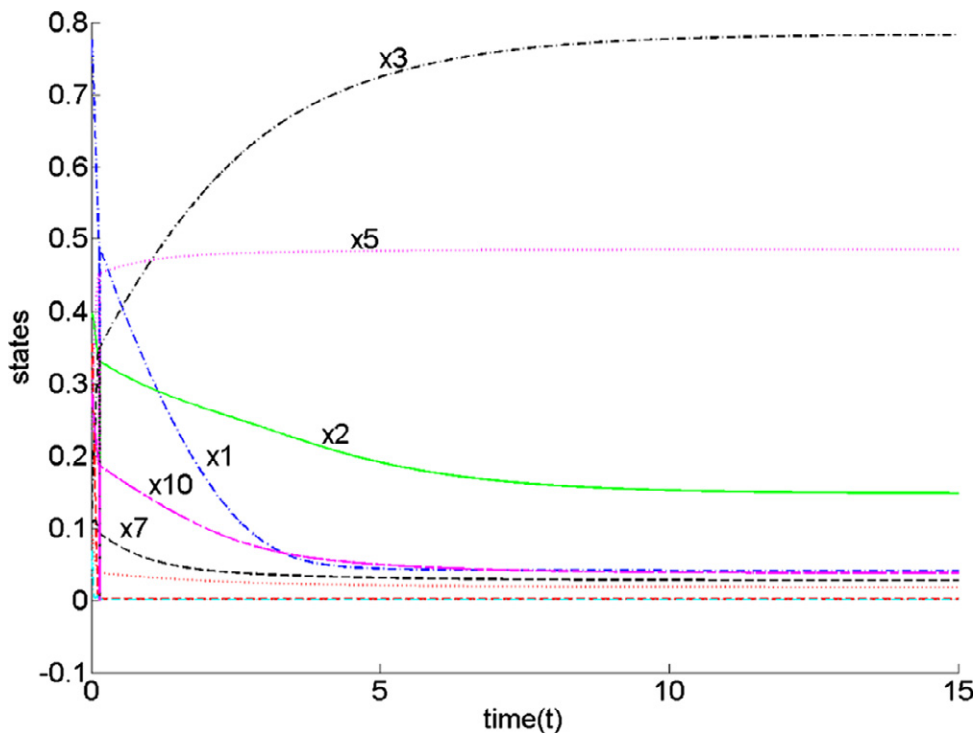
$$x^* \approx (0.14, 0.148, 0.783, 0.001, 0.485, 0.001, 0.027, 0.018, 0.037, 0.097)^T$$

همگرا هستند.

در این لحظه، واکنش به تعادل رسیده و انرژی آزاد ترکیب برابر  $۴۷/۷۶۱۱ -$  شد. شکل ۴.۴، رفتار گذرای شبکه‌ی عصبی پیشنهادی را با نقطه‌ی اولیه‌ی

$$(۰/۸, ۰/۴, ۰/۱, ۰/۴, ۰/۲, ۰/۱, ۰/۱, ۰/۱, ۰/۳, ۰/۱)^T$$

نشان می‌دهد. فقط بخشی از اجزای  $x(t)$  مشخص می‌شوند، زیرا بخش دیگر به صفر نزدیک است.



شکل ۴.۴: رفتار مسیر شبکه‌ی عصبی پیشنهادی در مثال ۱.۶.۴.

## ۷.۴ نتیجه گیری

در این پایان نامه، دو مدل شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم و محدب نامورد بررسی قرار گرفت. همچنین ثابت کردیم نقطه‌ی تعادل مدل‌های شبکه‌ی عصبی با جواب مسائل بهینه‌سازی یکسان است و مدل‌های ارائه شده پایدار لیاپانف و همگرایی سراسری است. مساله‌ی برنامه ریزی درجه دوم در حالت کلی و مساله‌ی بهینه‌سازی تابع محدب نما تحت قیود تساوی به عنوان دو مساله‌ی مهم با این دو مدل حل شده‌اند. دو مساله کاربردی در تجارت و شیمی نیز با مدل بیان شده در فصل آخر حل شده است. مثال‌های عددی حل شده کارایی مدل‌های ارائه شده را بیان می‌کنند.



## مراجع

- [۱] برقز، د. و گراهام، آ.، (۱۳۷۲). مقدمه‌ای بر نظریه‌ی کنترل و کنترل بهینه، ترجمه علی وحیدیان و ابوالقاسم بزرگ نیا، انتشارات دانشگاه فردوسی مشهد.
- [۲] پتکین، فیلیپ، (۱۳۷۸). شبکه‌های عصبی، ترجمه‌ی میرمجتبی میرصالحی و حسین تقی‌زاده کاخکی، انتشارات دانشگاه فردوسی مشهد.
- [۳] منهاج، م. ب.، (۱۳۷۷). هوش مصنوعی، جلد ۱، مرکز نشر پرفسور حسابی، تهران.
- [۴] مهرجو، ا.، (۱۳۸۶). شبکه‌های عصبی مصنوعی، دانشگاه آزاد اسلامی واحد اسلامشهر، اسلامشهر.
- [5] Bian, W., Xue, X. (2009). Subgradient-based neural networks for nonsmooth non-convex optimization problems. **IEEE Transactions on Neural Networks**, 20(6), 1024-1038.
- [6] Bian, W., Xue, X. (2013). Neural network for solving constrained convex optimization problems with global attractivity. **IEEE Transactions on Circuits and Systems I: Regular Papers**, 60(3), 710-723.
- [7] Bouzerdoum, A., Pattison, T. R. (1993). Neural network for quadratic optimization with bound constraints. **IEEE Transactions on Neural Networks**, 4(2), 293-304..
- [8] Chen, Y. H., Fang, S. C. (1998). Solving convex programming problems with equality constraints by neural networks. **Computers Mathematics With Applications**, 36(7), 41-68.
- [9] Chen, Y., Jiao, Y., Gao, X., Leung, K., (2001). "A new gradient-based neural network for solving linear and quadratic programming problems", **IEEE Transactions on Neural Networks**, No. 12(5), 1074–1083

- 
- [10] Cheng, L., Guang, Z., (2011). "Recurrent neural network for non-Smooth convex optimization problems with application to the identification of genetic regulatory networks", **IEEE Transactions Neural Networks**, 22, 714–26.
- [11] Cheng, L., Hou, Z., Tan, M., (2009). "A delayed projection neural network for solving linear variational inequalities", **IEEE Transactions , Neural Networks**, No. 20(6), 915–925.
- [12] Cichocki, A., Unbehauen, R., Weizierl, K., Holzl, R., (1996). "A new neural network for solving linear programming problems", **European Journal of Operational Research**, 93, 244-245.
- [13] Effati, S., Baymani, M., (2005). "A new nonlinear neural network for solving convex nonlinear programming problems", **Applied Mathematics and Computation**, 168, 1370 -1379
- [14] Effati, S., Baymani, M., (2005). "A new nonlinear neural network for solving quadratic programming problems", **Applied Mathematics and Computation**, 165, 719-729.
- [15] Effati, S., Nazemi, A. R. (2006). Neural network models and its application for solving linear and quadratic programming problems. **Applied mathematics and Computation**, 172(1), 305-331.
- [16] Effati, S., Ranjbar, M. (2008). Neural network models for solving the maximum flow problem. **Applications and Applied Mathematics**, 3(3), 149-162.
- [17] Effati, S., Ranjbar, M., (2011). "A novel neural network for solving quadratic programming problems", **Applied Mathematics and Computation**, Model, No. 35(4), 1688–1695.
- [18] Effati, S., Ghomashi, A., Nazemi, A.R., (2007). "Application of projection neural network in solving convex programming problems", **Applied Mathematics and Computation**, 188, 1103-1114.
- [19] Farley, B. G., Clark, W., (1954). "Simulation of Self-Organizing Systems by Digital Computer", **IEEE Transactions of the IRE Professional Group on Information Theory** 4, No. 4, pp. 76-84.

- [20] Fukushima, M. (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. **Mathematical programming**, 53(1-3), 99-110.
- [21] Gao, X. B. (2004). A novel neural network for nonlinear convex programming. **IEEE Transactions on Neural Networks**, 15(3), 613-621.
- [22] Gao, X., Liao, L. Z. (2010). A new one-layer neural network for linear and quadratic programming. **IEEE Transactions on Neural Networks**, 21(6), 918-929.
- [23] Ghasabi-Oskoei, H., Mahdavi-Amiri, N. (2006). An efficient simplified neural network for solving linear and quadratic programming problems. **Applied Mathematics and Computation**, 175(1), 452-464.
- [24] Ghasabi-Oskoei, H., Malek, A., Ahmadi, A. (2007). Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems. **Computers Mathematics with Applications**, 53(9), 1439-1454.
- [25] Goldfarb, D., Lapidus, L. (1968). Conjugate gradient method for nonlinear programming problems with linear constraints. **Industrial Engineering Chemistry Fundamentals**, 7(1), 142-151.
- [26] Guo, Z., Liu, Q., Wang, J., (2011). "A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints", **IEEE Transactions Neural Networks**, 892–900.
- [27] Haugen, R., (1997). "Modern investment theory, Upper Saddle River", **New Jersey, Prentice Hall**, 92–130.
- [28] Hebb, DO, (1949). "The Organization of Behavior: A Neuropsychological Theory", Wiley.
- [29] Hopfield, J.J., (1982). "Neural network and physical systems with collective computational abilities", **Proceedings of the National Academy of Science, USA**.
- [30] Hopfield, J.J., Tank, D.W., (1985). "Neural computation of decisions in optimization problems", **Biological Cybernetics**, 533-541.
- [31] Hu, X.L., (2009). "Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints", **Neuro-computing**, No. 72, 1131–1137.

- 
- [32] Hu, X., Wang, J. (2006). Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. **IEEE Transactions on Neural Networks**, 17(6), 1487-1499.
- [33] Hu, X., Wang, J. (2008). An improved dual neural network for solving a class of quadratic programming problems and its  $k$ -winners-take-all application. **IEEE Transactions on Neural Networks**, 19(12), 2022-2031.
- [34] Karamardian, S., Schaible, S. (1990). Seven kinds of monotone maps. **Journal of Optimization Theory and Applications**, 66(1), 37-46.
- [35] Kennedy, M. P., Chua, L. O. (1988). neural networks for nonlinear programming. **IEEE Transactions on Circuits and Systems**, 35(5), 554-562.
- [36] Kinderlehrer, D., Stampacchia, G., (1980). "An introduction to variational inequalities and their applications", **New York, Academic**.
- [37] Konnov, I. V. (1997). A class of combined iterative methods for solving variational inequalities. **Journal of Optimization Theory and Applications**, 94(3), 677-693.
- [38] Li, X. S., Fang, S. C. (1997). On the entropic regularization method for solving min-max problems with applications. **Mathematical Methods of Operations Research**, 46(1), 119-130.
- [39] Lillo, W. E., Loh, M. H., Hui, S., Zak, S. H. (1993). On solving constrained optimization problems with neural networks: A penalty method approach. **IEEE Transactions on Neural Networks**, 4(6), 931-940.
- [40] Liu, S., Wang, J. (2006). A simplified dual neural network for quadratic programming with its KWTA application. **IEEE Transactions on Neural Networks**, 17(6), 1500-1510.
- [41] Liu, Q., Wang, J. (2008). A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. **IEEE Transactions on Neural Networks**, 19(4), 558-570.
- [42] Liu, Q., Wang, J. (2008, November). A one-layer recurrent neural network for non-smooth convex optimization subject to linear equality constraints. **In International Conference on Neural Information Processing** (1003-1010). Springer Berlin Heidelberg.

- [43] Liu, Q., Guo, Z., Wang, J., (2012). "A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization", **Neural Networks**, 99–109.
- [44] Luenberger, D.G., (1973). "Introduction to Linear and Nonlinear Programming", **Addison-Wesley, Reading, MA**.
- [45] Maa, C. Y., Schanblatt, M. A. (1992). A two-phase optimization neural network. **IEEE Transactions on Neural Networks**, 3(6), 1003-1009.
- [46] Maa, C.Y., Shanblatt, M.A., (1992). "Linear and quadratic neural network Analysis", **IEEE Transactions on Neural Networks**, 3(4), 580-594.
- [47] Malek, A., Alipour, M. (2007). Numerical solution for linear and quadratic programming problems using a recurrent neural network. **Applied Mathematics and Computation**, 192(1), 27-39.
- [48] Malek, A., Oskoei, H.G., (2005). "Numerical solutions for constrained quadratic problems using high-performance neural networks", **Applied Mathematics and Computation**, 169, 451-471.
- [49] Malek, A., Oskoei, H.G., Ahmadi, A., (2007). "Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems", **Applied Mathematics and Computation**, 53, 1439-1454.
- [50] McCulloch, W. S. Pitts, W., "A logical calculus of the ideas immanent in nervous activity", **Bulletin of Mathematical Biophysics**, 5, 115-133.
- [51] Mícheál Searcóid, Ó., (2006). "Metric spaces", **Springer undergraduate mathematics series, Berlin**, New York, Springer-Verlag.
- [52] Miller, R. K., Michel, A. N., (1981). "Ordinary Differential Equations", **Academic Press, New York**.
- [53] Nazemi, A. R., (2014). "A neural network model for solving convex quadratic programming problems with some applications", **Engineering Applications of Artificial Intelligence**, 32, 54-62.
- [54] Nazemi, A. R., Nazemi, M., (2014). "A gradient-based neural network method for solving strictly convex quadratic programming problems", **Cognitive Computation**, 6, 484–495.



- [55] Nazemi, A. R., Omid, F., (2012). "A capable neural network model for solving the maximum flow problem", **Computational and Applied Mathematics**, 236, 3498-3513.
- [56] Nazemi, A. R., Tahmasbi, N., (2013). "A high performance neural network model for solving chance constrained optimization problems", **Neurocomputing**, 121, 540-550.
- [57] Qin, S., Fan, D., Su, P., Liu, Q. (2014). "A simplified recurrent neural network for pseudoconvex optimization subject to linear equality constraints", **Communications in Nonlinear Science and Numerical Simulation**, 19(4), 789-798.
- [58] Qin, S.T., Xue, X.P., (2009). "Global exponential stability and global convergence in finite time of neural networks with discontinuous activations", **IEEE Transactions on Neural Networks**, 189–204.
- [59] Qin, S.T., Xue, X.P., (2010). "Dynamical analysis of neural networks of subgradient system", **IEEE Transactions on Neural Networks**, 47–52.
- [60] Qin, S.T., Xue, X.P., (2010). "Dynamical behavior of a class of nonsmooth gradient-like systems", **Neurocomputing**, 32–41.
- [61] Rodriguez-Va'z., nguez-Castro, A. Dom, Rueda, R., Huertas J.L.A., and nchez-Sinencio, Sa, (1990). "Nonlinear switched-capacitor neural networks for optimization problems", **IEEE Transactions on Circuits and Systems**, 37(3), 384-397.
- [62] Schittkowski, K., Hock, W., (1981). "Test examples for nonlinear programming codes", Berlin, Springer.
- [63] Tank, D., Hopfield, J. J. (1986). Simple'neural'optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. **IEEE Transactions on Circuits and Systems**, 33(5), 533-541.
- [64] Tao, Q., Cao, J., Xue, M., Qiao, H. (2001). A high performance neural network for solving nonlinear programming problems with hybrid constraints. **Physics Letters A**, 288(2), 88-94.
- [65] Vardi, A. (1992). New minimax algorithm. **Journal of Optimization Theory and Applications**, 75(3), 613-634.

- [66] White, W.B., Johnson, S.M., Dantzig, G.B., (1958). "Chemical equilibrium in complex mixtures", **The Journal of Chemical Physics**, 28(5),751-755.
- [67] Wu, X., Xia, Y., Li, J., Chen, W., (1996). "A high-performance neural network for solving linear and quadratic programming problems", **IEEE Transactions on Neural Networks**, No. 7(3), 643–651.
- [68] Xia, Y. (1996). A new neural network for solving linear and quadratic programming problems.**IEEE Transactions on Neural Networks**, 7(6), 1544-1548.
- [69] Xia, Y. (2004). An extended projection neural network for constrained optimization.**Neural Computation**, 16(4), 863-883.
- [70] Xia, Y., Feng, G., Wang, J. (2004). A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations. **Neural Networks**, 17(7), 1003-1015.
- [71] Xia, Y., Wang, J. (1998). A general methodology for designing globally convergent optimization neural networks. **IEEE Transactions on Neural Networks**, 9(6), 1331-1343.
- [72] Xia, Y.S., Wang, J., (2000). "A recurrent neural network for solving linear projection equations",**Neural Networks**, 37–50.
- [73] Xia, Y., Wang, J. (2001). A dual neural network for kinematic control of redundant robot manipulators. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, 31(1), 147-154.
- [74] Xia, Y., Wang, J. (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. **IEEE Transactions on Circuits and Systems I: Regular Papers**, 51(7), 1385-1394.
- [75] Xia, Y., Wang, J., (2005). "Recurrent neural networks for solving nonlinear convex programs with linear constraints", **IEEE Transactions on Neural Networks**, 16, No. 2, 379-386.
- [76] Xia, Y.S., Leung, H., Wang, J., (1998). "A projection neural network and its application to constrained optimization problems", **IEEE Transactions Circuits System**,47–58.

- 
- [77] Xia, Y., Leung, H., Wang, J. (2002). A projection neural network and its application to constrained optimization problems. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, 49(4), 447-458.
- [78] Xue, X., Bian, W. (2008). Subgradient-based neural networks for nonsmooth convex optimization problems. **IEEE Transactions on Circuits and Systems I: Regular Papers**, 55(8), 2378-2391.
- [79] Yang, Y., Cao, J., Xu, X., Hu, M., Gao, Y. (2014). A new neural network for solving quadratic programming problems with equality and inequality constraints. **Mathematics and Computers in Simulation** 101, 103-112.
- [80] Yang, Y., Cao, J. (2006). Solving quadratic programming problems by delayed projection neural network. **IEEE Transactions on Neural Networks**, 17(6), 1630-1634.
- [81] Yang, Y., Cao, J. (2008). A feedback neural network for solving convex constraint optimization problems. **Applied Mathematics and Computation**, 201(1), 340-350.
- [82] Yang, Y., Cao, J. (2010). The optimization technique for solving a class of non-differentiable programming based on neural network method. **Nonlinear Analysis: Real World Applications**, 11(2), 1108-1114.
- [83] Zhang, Y., Li, Z. (2009). Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints. **Physics Letters A**, 373(18), 1639-1643.
- [84] Zhang, S., Constantinides, A. G. (1992). Lagrange programming neural networks. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, 39(7), 441-452.
- [85] Zhang, Y., Wang, J. (2002). A dual neural network for convex quadratic programming subject to linear equality and inequality constraints. **Physics Letters A**, 298(4), 271-278.
- [86] <http://cse.stanford.edu/class/sophomore-college/projects-00/neural-networks/History/> [ ] J.J. Hopfield, Neural.
- [87] <http://www.ieee.org>.

- [88] Bazaraa, M.S., Sherali, H.D., Shetty, C.M., (1993). **"Nonlinear Programming Theory and Algorithm"**, 2nd ed., Wiley, New York.



# واژه‌نامه فارسی به انگلیسی

Initial	ابتدایی
Measurable	اندازه‌پذیر
ADELIN	آدلاین
Axon	آکسون
Bias	بایاس
Motion Vector	بردار حرکت
State Vector	بردار وضعیت (حالت)
Optimization	بهینه‌سازی
Stability	پایداری
Stability Asymptotic	پایداری مجانبی
Perceptron	پرسپترون
Continuous	پیوسته
Supporting Hyperplans Function	تابع اکیداً شبه‌محدب
Activation Function	تابع تحریک
Sigmoid Function	تابع سیگموئید
Strongly Quasi Convex Functions	تابع شبه‌محدب قوی
Logistic	تابع لجستیک
Liapunov Function	تابع لیاپانف
Convex Function	تابع محدب
Lower Semicontinuous Function	تابع نیم پیوسته‌ی پائینی
S-shap Function	تابع S-شکل
Mixture	ترکیب
Onstant	ثابت
Long Term Memory	حافظه‌ی دراز مدت
Short Term Memory	حافظه‌ی کوتاه مدت
Dendrit	دندریت
Penalty Function Method	روش تابع جریمه (پنالتی)

Projection Method	روش تصویر
Subsequence	زیر دنباله
Portfolio	سبد سهام
Combustion	سوختن
Soma	سوما
Systems Dynamical	سیستم‌های دینامیکی
Network	شبکه
Artificial Neural Networks	شبکه‌های عصبی مصنوعی
Pseudoconvex	شبه محدب
Pseudomonotone	شبه یکنواخت
Neural	عصبی
Adaptive Linear Elements	عنصر خطی تطبیقی
Nonlinear	غیر خطی
Synaptic Gap	فاصله‌ی سیناپسی
Bounded	کراندار
MADALINE	مادلاین
Convex Set	مجموعه‌ی محدب
Hard Limit	محدود کننده‌ی سخت
Differentiable	مشتق پذیر
Locally	موضعی
Regular Point	نقطه‌ی منظم
Reaction	واکنش
Synaptic Weight	وزن سیناپسی
Convenience	همگرایی

# واژه‌نامه انگلیسی به فارسی

Activation Function	تابع تحریک
Adaptive Linnear Elements	عنصر خطی تطبیقی
Adelin	آدلاین
Artificial Nearal Network	شبکه‌های عصبی مصنوعی
Axon	آکسون
Bias	بایاس
Bounded	کراندار
Combustion	سوختن
Continuous	پیوسته
Convenience	همگرایی
Convex Function	تابع محدب
Convex Set	مجموعه‌ی محدب
Dendrit	دندریت
Differentiable	مشتق پذیر
Hard Limit	محدودکننده‌ی سخت
Initial	ابتدایی
Liapunov Function	تابع لیاپانف
Locally	موضعی
Logistic	تابع لجستیک
Long Term Memory	حافظه‌ی دراز مدت
Lower Semicontinuous Function	تابع نیم پیوسته‌ی پائینی
MADALINE	مادلاین
Measurable	اندازه‌پذیر
Mixture	ترکیب
Motion Vector	برداری حرکت
Network	شبکه
Neural	عصبی



Nonlinear	غیرخطی
Onstant	ثابت
Optimization	بهینه‌سازی
Penalty Function Method	روش تابع جریمه (پنالتی)
Perceptron	پرسپترون
Portfolio	سبد سهام
Projection Method	روش تصویر
Pseudoconvex	شبه‌محدب
Pseudomonotone	شبه‌یکنواخت
Reaction	واکنش
Regular Point	نقطه‌ی منظم
Short Term Memory	حافظه‌ی کوتاه مدت
Sigmoid Function	تابع سیگموئید
Soma	سوما
S-shap Function	تابع S-شکل
Stability Asymptotic	پایداری مجانبی
Stability	پایداری
State Vector	بردار وضعیت (حالت)
Strongly Quasi Convex Functions	تابع شبه‌محدب قوی
Subsequence	زیر دنباله
Supporting Hyperplans Function	تابع اکیداً شبه‌محدب
Synaptic Gap	فاصله‌ی سیناپسی
Synaptic Weight	وزن سیناپسی
Systems Dynamical	سیستم‌های دینامیکی
Trajectory	مسیر

## **Aabstract**

A new neural network is proposed in this thesis for solving quadratic programming problems with equality and inequality constraints. Comparing with the existing neural networks for solving such problems, the proposed neural network has fewer neurons and an one-layer architecture.

the optimization techniques for solving pseudoconvex optimization problems are investigated. A simplified recurrent neural network is proposed according to the optimization problem. We prove that the optimal solution of the optimization problem is just the equilibrium point of the neural network, and vice versa if the equilibrium point satisfies the linear constraints. The proposed neural network is proven to be globally stable in the sense of Lyapunov and convergent to an exact optimal solution of the optimization problem. A numerical simulation is given to illustrate the global convergence of the neural network. Applications in business and chemistry are given to demonstrate the effectiveness of the neural network.

**Keywords:** Recurrent neural network, Pseudoconvex programming, Global convergence, Neural network, Convergence, Stability, Quadratic programming, Positive semidefinite



**Shahrood University of Technology**

**Faculty Of Mathematical Sciences**

**MSc Thesis in: Partial Differential equations**

**A New Capable Neural Network Model For  
Solving A Class Of Nonlinear Optimization  
Problems**

**By: Maryam Shafaii Node**

**Supervisor**

**Dr.Alireza.Nazemi**

**Feb 2017**