





دانشگاه صنعتی شاهرود

دانشکده ریاضی
گروه ریاضی کاربردی

حل مسائل برنامه ریزی ریاضی با استفاده از شبکه های عصبی

دانشجو: سمانه آذربیک

اساتید راهنما:

دکتر جعفر فتحعلی

دکتر داود شاهسونی

استاد مشاور:

دکتر علیرضا ناظمی

پایان نامه جهت اخذ درجه کارشناسی ارشد ریاضی کاربردی

تابستان ۱۳۸۹

تقدیم به

حضور سبز مادر گرانقدرم

و

خاطره ارزشمند پدر بزرگوارم

و

آنان که آموختن بی وجودشان میسر نبود.

تشکر و قدردانی

ستایش شایسته ذات بی همتای آن عزیز مهربان است و تقدیم سزاوار هر آنچه و هر آنکه لطف و محبت اوست. خدا را شاکرم که به من این توان را داد که این مرحله از زندگی را با سلامتی و سر بلندی پشت سر بگذارم و سر بر آستان پر برکت او می سایم که هر چه عزت و سرافرازی هست، از اوست. پس از خدای بلند مرتبه سر تعظیم در برابر مادرم فرود می آورم، بر پایش بوسه می زنم و او را می ستایم که تجلی مهر و لطف خداوندی برای من می باشد.

لازم می دانم که از زحمات فراوان و کارگشای اساتید فرزانه و بزرگوارم آقایان دکتر **جعفر فتحعلی**، دکتر **داود شاهسونی** و دکتر **علیرضا ناظمی** که با راهنمای ها و نظرات ارزنده و گهربارشان و صبر و حوصله فراوان، نقش مهمی در به ثمر رسیدن این پایان نامه داشته اند، صمیمانه تقدیر و تشکر نمایم. بی تردید انجام این پایان نامه بدون همکاری ایشان هرگز ممکن نبود. برای این بزرگواران از درگاه خداوند مهربان آرزوی سعادت و عاقبت به خیری دارم.

چکیده

با پیشرفت فن آوری اطلاعات و ارتباطات و توسعه ارتباطات درون سازمانی و بین سازمانی نیاز به استفاده از مدل های بهینه سازی را برای استفاده منطقی از داده ها و اطلاعات فراهم شده گسترش داده است. این مطلب متضمن بزرگ شدن اندازه مسائل بهینه سازی که در عمل وجود دارند خواهد بود. در این شرایط لزوم به کارگیری روش های کارآمدی که بتوانند با سرعت بالا مسائل بسیار بزرگ را با کیفیت قابل قبول حل کنند بیش از پیش احساس می شود.

اخیراً روش های بهینه سازی که بر پایه رویکرد هوش مصنوعی توسعه یافته اند، موفقیت های چشم گیری در حل مؤثر و کارای مسائل بهینه سازی به دست آورده اند. روش هایی چون الگوریتم ژنتیک^۱، جستجوی ممنوع^۲، گرم و سرد کردن شبیه سازی شده^۳ و شبکه عصبی^۴، قابلیت های خود را در حل مسائل بزرگ عملی به خوبی نشان داده اند. امتیازات ویژه ی موجود در شبکه های عصبی امکان کاربرد آنها را در حوزه وسیعی از تحقیقات فراهم ساخته است. از جمله آن امتیازات می توان به امکان یادگیری و بهبود عملکرد بر اساس داده های ورودی اشاره کرد. همچنین امکان انجام محاسبات به صورت موازی در شبکه های عصبی امتیاز دیگری است که با توجه به گسترش سخت افزارهای موازی، امکان حل مسائل بسیار بزرگ را توسط این رویکرد ممکن می سازد.

در این پایان نامه چند مدل مختلف شبکه عصبی بازگشتی^۵ برای حل مسائل برنامه ریزی خطی و درجه دوم ارائه می شود. تحلیل وجود یکتایی، پایداری و همگرایی سراسری جواب ها مورد بررسی قرار می گیرند و عملکرد روش های ارائه شده با به کارگیری چند مثال نشان داده می شود.

¹ Genetic Algorithms

² Tabu Search

³ Simulated Annealing

⁴ Neural Network

⁵ Recurrent

لیست مقالات مستخرج از پایان نامه

[۱] سمانه آذربیک، علیرضا ناظمی، داود شاهسونی (۱۳۸۹). "یک کاربرد از شبکه های عصبی برای حل مسائل برنامه ریزی خطی و درجه دوم". سومین کنفرانس بین المللی تحقیق در عملیات دانشگاه صنعتی امیر کبیر.

[2] A. R. Nazemi, S. Azarbeyk "An application of non linear complementary problem for linear and quadratic programming problems", **submitted**.

فهرست مندرجات

فصل ۱: مقدمه ای بر شبکه های عصبی.....	۱
۱-۱ مقدمه.....	۱
۲-۱ ساختار شبکه های مصنوعی.....	۴
۳-۱ اساس بیولوژیکی سلول های عصبی.....	۶
۴-۱ مدل ریاضی یک سلول عصبی.....	۸
۵-۱ شباهت سلول عصبی مصنوعی و نرون بیولوژیک.....	۱۱
۶-۱ فرایند یادگیری.....	۱۲
۱-۶-۱ شبکه های عصبی به عنوان سیستم های دینامیکی آموزش پذیر.....	۱۳
۲-۶-۱ الگوریتم پس انتشار.....	۱۳
۳-۶-۱ انتخاب یادگیری.....	۱۵
۷-۱ تاریخچه کاربرد شبکه های عصبی مصنوعی در حل مسائل بهینه سازی	۱۶
فصل ۲: پیش نیازها.....	۱۹
۱-۲ مقدمه.....	۲۰
۲-۲ توابع محدب.....	۲۰
۳-۲ شرایط لازم و کافی بهینگی برای مسائل نامقید	۲۱
۴-۲ شرایط لازم و کافی بهینگی برای مسائل مقید.....	۲۲
۵-۲ توابع جریمه ای.....	۲۴
۶-۲ پایداری.....	۲۴
۷-۲ تابع انرژی.....	۲۸
۸-۲ عملگر تصویر.....	۲۹

فصل ۳: مدل های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه سازی.....۳۱

۱-۳ مقدمه.....۳۲

۲-۳ مدل های مختلف شبکه عصبی.....۳۳

۱-۲-۳ م.....۳۳

اول.....۳۳

۲-۲-۳ مدل دوم.....۳۴

۳-۲-۳ مدل سوم.....۳۴

۴-۲-۳ مدل چهارم.....۳۵

۵-۲-۳ مدل پنجم.....۳۵

۶-۲-۳ مدل ششم.....۳۶

۷-۲-۳ مدل هفتم.....۳۹

۸-۲-۳ مدل هشتم.....۴۰

۹-۲-۳ مدل نهم.....۴۲

فصل ۴: حل مسائل برنامه ریزی خطی و درجه دوم با استفاده از عملگر تصویر.....۴۵

۱-۴ مقدمه.....۴۶

۲-۴ بیان مسئله.....۴۶

۳-۴ شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم محدب.....۴۷

۱-۳-۴ تحلیل پایداری و همگرایی.....۴۸

۴-۴ نتایج عددی.....۵۰

فصل ۵: بررسی پایداری مجانبی یک مدل شبکه عصبی برای حل مسائل برنامه ریزی درجه

دوم.....۵۶

۱-۵ مقدمه.....۵۷

۵۸.....	۲-۵	مدل شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم.....
۵۹.....	۳-۵	تحلیل همگرایی و پایداری مدل شبکه عصبی.....
۶۲.....	۴-۵	نتایج عددی.....
۶۵.....		نتیجه گیری و پیشنهادات.....
۶۷.....		منابع و مراجع.....

فصل اول

مقدمه ای بر شبکه های عصبی

کار بر روی شبکه های عصبی مصنوعی، با الهام از عملکرد مغز انسان از آنجا شروع شد که دانشمندان دریافتند مغز بشر به گونه ای کاملاً متفاوت از کامپیوترهای دیجیتالی مرسوم محاسبات را انجام می دهد. مغز یک سیستم پردازش داده بسیار پیچیده است که از واحدهای ساختاری به نام سلول عصبی یا نرون^۶ تشکیل شده است.

طبق تعریف، شبکه عصبی مصنوعی ماشینی است برای ساخت یک مدل که یا می تواند با استفاده از قطعات الکترونیکی (سخت افزار) ساخته شود یا به وسیله نرم افزار شبیه سازی شود. این عملکرد شبکه عصبی شبیه به عملکرد مغز انسان انجام می پذیرد [۱]. شبکه عصبی سیستمی پویا و غیر خطی است که از تعداد زیادی واحد پردازش (نرون ها) و اتصالات بین این واحدهای پردازش تشکیل می شود [۲].

شبکه های عصبی برای حل مسائلی به کار می رود که فرمول حل آنها ناشناخته است و مدل علت و معلولی یا برای آنها وجود ندارد و یا ابهام قابل ملاحظه ای در آن دیده می شود. علت نبود روابط ریاضی لازم برای تشریح چنین مسائلی این است که حتی خود مسأله به طور کامل و بدون ابهام شناخته نشده است [۳].

یک شبکه عصبی بر خلاف کامپیوتر که نیازمند دستورهای کاملاً صریح و مشخص است، به مدل های ریاضی محض نیازی ندارد، بلکه مانند انسان تجربه کسب کرده و سپس نتیجه این تجربیات را تعمیم می دهد. امروزه این شبکه ها تبدیل به ابزاری قدرتمند و عمومی شده اند که برای حل مسائل از قبیل تخمین (تقریب)، تشخیص الگو^۷، رباتیک، کنترل، و به طور کلی در هر جا که نیاز به یادگیری نگاشت خطی و یا غیر خطی باشد، به کار می روند [۴ و ۲۷].

برای حل هر مسأله ای، شبکه های عصبی سه مرحله را طی می کنند: آموزش^۸، تعمیم^۹

^۶ Neuron

^۷ Pattern recognition

^۸ Training

^۹ generalization

اجرا^{۱۰}، آموزش، فرایندی است که طی آن شبکه می آموزد تا الگوی موجود در ورودی ها را (که به صورت مجموعه داده های آموزشی است) بشناسد. برای این منظور هر شبکه عصبی از مجموعه ای از قوانین یادگیری که نحوه یادگیری را تعریف می کنند استفاده می کند. تعمیم توانایی شبکه برای ارائه جواب قابل قبول در قبال ورودی هایی است که در مجموعه آموزشی نبوده اند. یعنی پس از آنکه مثال های اولیه به شبکه آموزش داده شد، شبکه می تواند در قبال یک ورودی آموزش داده نشده قرار گیرد و یک خروجی مناسب را ارائه نماید. این خروجی بر اساس مکانیسم تعمیم که همانا چیزی جز پروسه درونیابی^{۱۱} نیست به دست می آید [۲۷]. استفاده از شبکه برای انجام عملکردی که به آن منظور طراحی شده است را/جر/ می گویند [۵].

در اثر آموزش دیدن شبکه، وزن های داخلی^{۱۲} (که بر روی ورودی های هر سلول اعمال می شود) تغییر می کند و به وضعیت مناسب می رسند. یکی از نقاط ضعف شبکه های عصبی این است که نتایج آموزش، یعنی وزن های داخلی، هیچگونه تصویر روشنی از اعتبار جواب های مسأله به دست نمی دهد. این وزن ها کاملاً قابل درک نیستند. با این وجود جواب های تولید شده توسط شبکه، اغلب صحیح است و با شرایط کمی حاکم بر محیط سازگاری دارد. این صحت جوابها و صدق شرایط کمی حاکم بر محیط، گاهی مهم تر از توضیح پذیر بودن آن است [۴]. تا کنون اسامی متفاوتی برای برای شبکه های عصبی به کار رفته است. از جمله آنها می توان به مدل های جعبه سیاه^{۱۳}، مدل های تجربی^{۱۴}، تقریب زن های عام^{۱۵} و مدل های موازی^{۱۶} اشاره نمود [۶].

¹⁰ Operation

¹¹ Interpolation

¹² Internal weight

¹³ Black-box models

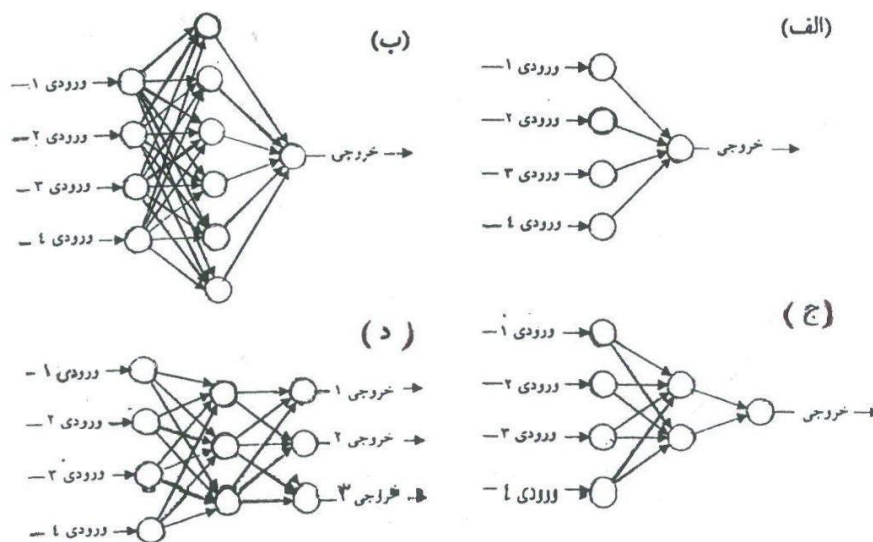
¹⁴ Empirical models

¹⁵ Universal approximators

¹⁶ Parallel models

۲-۱ ساختار شبکه های عصبی مصنوعی

شبکه های عصبی مصنوعی از یک سری واحد های ساختمانی اولیه تشکیل می شوند. هر واحد دارای چندین ورودی است که با هم ترکیب شده و پس از انجام عملیات پردازش یک خروجی را به دست می دهند. این واحد های اولیه مانند شکل (۱-۱) به هم متصل هستند به طوری که خروجی هر واحد به عنوان ورودی واحد های دیگر مورد استفاده قرار می گیرد. واحدهای ساختمانی را سلول عصبی، نرون یا گره^{۱۷} نیز می نامند (شکل (۱-۱)).



شکل (۱-۱): چهار مثال از شبکه های عصبی مصنوعی، الف-یک شبکه بسیار ساده شامل چهار ورودی و یک خروجی، نتیجه آموزش این شبکه تکنیک رگرسیون لجستیک^{۱۸} در آمار است. ب-شبکه دارای دو لایه میانی است که قدرت بیشتری در تشخیص الگو دارد. ج-افزایش تعداد لایه پنهان شبکه را قوی تر می کند ولی خطر بیش برازشی^{۱۹} در آن وجود دارد. د-شبکه عصبی با چندین خروجی.

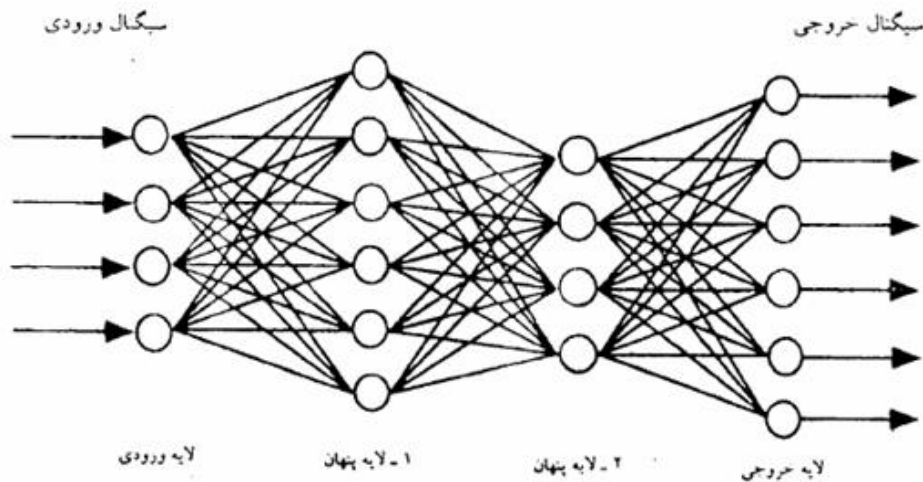
شبکه های عصبی ساختار لایه ای دارند شکل (۲-۱). در شکل (۲-۱) اولین لایه در سمت چپ، لایه ورودی است و لایه آخر در سمت راست نیز لایه خروجی است. داده ها در لایه ورودی جا می گیرند. به جز لایه ورودی تمام لایه های دیگر شبکه عصبی در کار پردازش شرکت می کنند که در نهایت،

¹⁷ Node

¹⁸ Logistic regression

¹⁹ Over fitting

خروجی به دست می آید. لایه های بین لایه های ورودی و خروجی را لایه های پنهان^{۲۰} می نامند [۷].



شکل (۱-۲): ساختمان یک شبکه عصبی مصنوعی

شبکه های عصبی اولیه تنها دو لایه (ورودی و خروجی) داشتند و به همین دلیل تنها در مسائل خطی مفید بودند. ساختارهای توسعه یافته امروزی شامل سه و یا تعداد بیشتری لایه می باشد. رایج ترین ساختار شبکه، سه لایه ای است (لایه ورودی، یک لایه پنهان و لایه خروجی) و عموماً قادر به حل اکثر مسائل پیچیده می باشد.

بر اساس نحوه اتصال گره ها به یکدیگر، شبکه های عصبی به دو گروه بزرگ تقسیم می شوند [۳]: ۱- شبکه هایی با تغذیه پیشرو^{۲۱} ۲- شبکه هایی با تغذیه برگشتی^{۲۲}.

شبکه های نشان داده شده در شکل (۱-۱) همگی از نوع شبکه ها با تغذیه پیشرو هستند بدین معنی که تنها یک جریان یک طرفه از لایه ورودی به سمت لایه خروجی موجود است و هیچ مسیر برگشتی وجود ندارد. این نوع شبکه ساده ترین و پرکاربردترین نوع شبکه عصبی است که برای مقاصد پیش بینی بسیار مناسب است [۴]. در شبکه هایی با تغذیه برگشتی حداقل یک حلقه برگشتی

²⁰ Hidden layers

²¹ Feed forward networks

²² Feed-backward (feed-back)

فعال است. شبکه ای که خروجی هر سلول آن به تمام سلول های دیگر آن لایه وارد می شود مثالی از این دسته است. وجود حلقه برگشتی اثر زیادی بر قابلیت یادگیری شبکه می گذارد [۱]. اگر دینامیکی برای نرون در نظر گرفته شده باشد آنگاه نرون دارای فیدبک می باشد، این بدان معناست که خروجی نرون نه تنها به ورودی در آن لحظه بلکه به مقدار خروجی خود نرون در لحظه گذشته بستگی دارد [۲۷].

در بخش های بعد نحوه کارکرد شبکه های عصبی مصنوعی تشریح خواهد شد. از آنجا که مدل سازی سلول های عصبی مصنوعی با الهام از سلول های عصبی بیولوژیکی صورت گرفته است، شباهت های زیادی در نحوه عملکرد آنها وجود دارد. از این رو دانستن مفاهیم اولیه در مورد سلول های عصبی بیولوژیکی به فهم چگونگی عملکرد شبکه های عصبی کمک می کند.

۱-۳ اساس بیولوژیکی سلول های عصبی

الگو و الهام بخش شبکه های عصبی مصنوعی، سیستم عصبی موجودات زنده است که شامل واحد های تشکیل دهنده ای به نام نرون می باشد. یک نرون از جسم سلولی^{۲۳}، دندریت^{۲۴} و آکسون^{۲۵} تشکیل شده است (شکل (۱-۲)). دندریت ها اطلاعات ورودی را از سلول های عصبی دیگر دریافت می کنند ولی آکسون کانال خروجی نرون است. اطلاعات ورودی به صورت مواد شیمیایی (ناقل عصبی^{۲۶}) می باشد.

نرون این اطلاعات را به صورت سیگنال الکتریکی از دندریت به انتهای آکسون انتقال می دهد، هنگامی که سیگنال الکتریکی در انتهای آکسون به قدر کافی بزرگ باشد، سلول عصبی در طول آکسون تخلیه الکتریکی انجام می دهد و سیگنال شیمیایی در سیناپس^{۲۷} (که محل اتصال انتهای

²³ Cell body

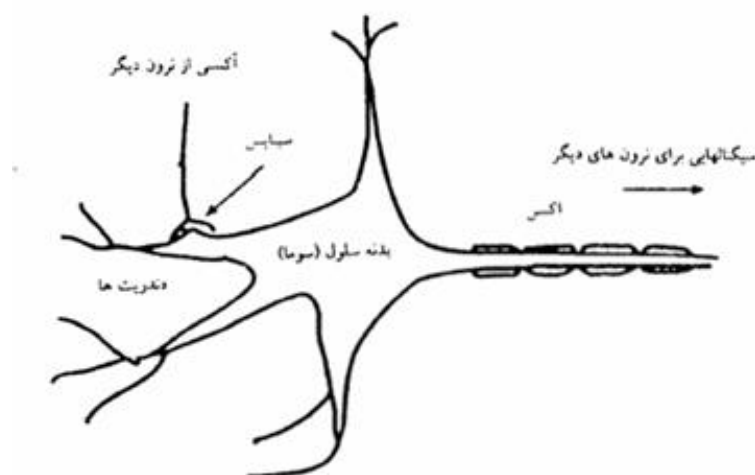
²⁴ dendrite

²⁵ axon

²⁶ Neurotransmitter

²⁷ Synapse

آکسون به دندریت های نرون بعدی است) ایجاد می شود [۶]. تفاوت اصلی یک نرون با سلول معمولی در قابلیت تحریک شدن آن است. غشای نرون در حالت عادی دارای پتانسیل الکتریکی ۷۰ میلی ولت است اما به دلیل ورودی دندریت ها، ممکن است نرون قادر به ابقای این پتانسیل عادی نباشد و در نتیجه به «پتانسیل عمل^{۲۸}» منجر شود. «پتانسیل عمل» یک پتانسیل الکتریکی است که به صورت یک طرفه به انتهای آکسون منتقل می شود. تحریک نرون ها از اصل همه یا هیچ تبعیت می کنند. به این معنی که تحریک شدن یک نرون بعد از رسیدن به یک سطح پتانسیل خاص (مثلاً ۵۰ میلی ولت) انجام می پذیرد. بعد از آزاد سازی پالس، نرون به وضعیت عادی خویش بر می گردد. پتانسیل عمل باعث رها شدن ناقل عصبی از انتهای آکسون می شود. این ناقل عصبی می تواند بر روی نرون بعدی اثر تحریک^{۲۹} یا جلوگیری^{۳۰} از تحریک داشته باشد [۳].



شکل (۱-۳): ساختمان یک سلول عصبی بیولوژیک یا نرون

شبکه عصبی مصنوعی، تقلیدی ساده سازی شده از شبکه عصبی مغز است. مغز انسان تقریباً از صد میلیارد (10^{11}) سلول عصبی تشکیل شده که بین آنها تقریباً صد تریلیون (10^{14}) اتصال وجود دارد. بعید است که بتوان حتی یک نرون مغز انسان را دقیقاً مدل سازی کرد. با تکنولوژی امروز، این

²⁸ Action potential

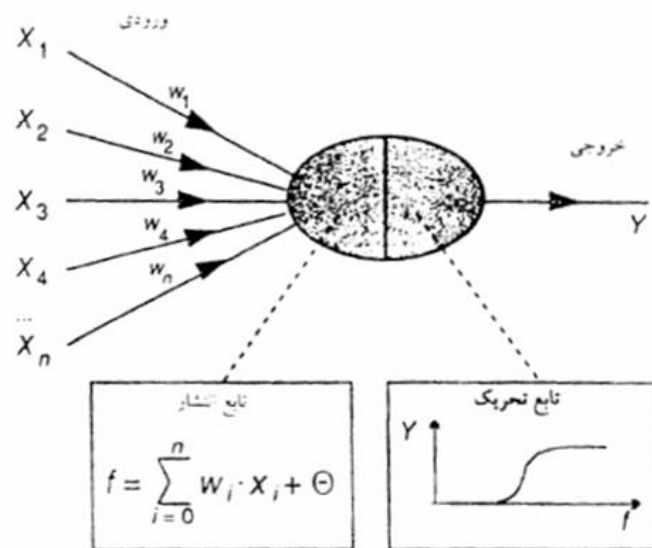
²⁹ Excitatory effect

³⁰ Inhibitory effect

کار در ماورای توانایی بشر قرار می گیرد. آنچه که امکان پذیر است الگو برداری مدل سلول های عصبی مصنوعی از نرون های طبیعی همراه با ساده سازی است که توأم با موفقیت نیز بوده است [۷].

۴-۱ مدل ریاضی یک سلول عصبی

همان طور که قبلا ذکر شد، شبکه های عصبی مصنوعی از واحدهای اولیه ای ساخته می شوند که مدل رفتاری آنها شبیه به نرون بیولوژیک است. شکل (۴-۱) یک مدل سلول عصبی را نشان می دهد. بدنه این سلول از دو بخش تشکیل شده است اولین بخش را تابع ترکیب^{۳۱} می گویند. وظیفه تابع ترکیب این است که تمام ورودی ها را ترکیب و یک عدد تولید کند. مطابق شکل (۴-۱) هر ورودی دارای وزن مختص به خود است. ورودی ها در اوزان مربوطه ضرب و سپس با هم جمع می شوند. مجموع حاصل را مجموع موزون^{۳۲} گویند که رایج ترین تابع ترکیب است.



شکل (۴-۱): مدل ساده یک سلول عصبی

³¹ Combination function

³² Weighted sum

بخش دوم سلول عصبی، تابع انتقال^{۳۳} نام دارد تابع ترکیب را به خروجی سلول تبدیل و سپس انتقال می دهد. تابع انتقال را تابع تحریک^{۳۴} نیز می نامند. رایج ترین توابع تحریک، بر پایه مدل های بیولوژیک استوار گردیده است.

اگر ورودی سلول i ام واقع در لایه ورودی را x_i در نظر بگیریم، این ورودی برای اتصال به سلول j ام لایه بعد در وزن w_{ji} ضرب می شود. حرف j در w_{ji} نشان دهنده شماره سلول در لایه بعد و حرف دوم اندیس معرف شماره سلول لایه قبلی است. شکل (۱-۵) مدل بلوکی یک سلول عصبی را نشان می دهد سلول مذکور دارای یک ورودی اضافی می باشد که به آن اریب^{۳۵} می گویند و با b_j نشان داده می شود. نقش اریب افزایش یا کاهش مجموع موزون است.

به بیان ریاضی، سلول j به وسیله دو رابطه زیر تعریف می شود:

$$u_j = \sum_{i=1}^m w_{ji} x_i \quad (1-1)$$

$$y_j = \varphi(u_j + b_j) \quad (2-1)$$

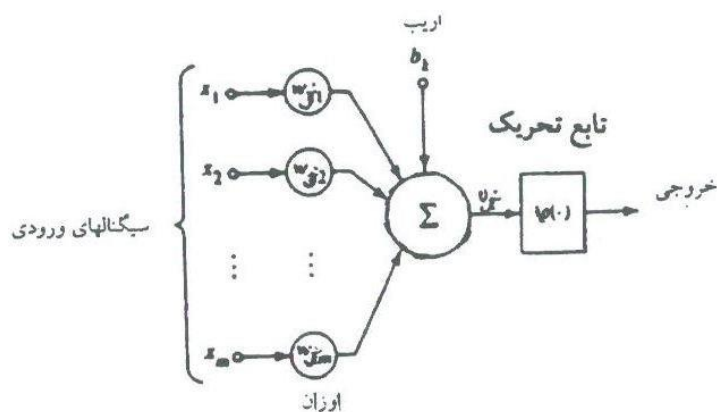
که $x_1, x_2, \dots, x_m$ داده های ورودی، $w_{j1}, w_{j2}, \dots$ اوزان اتصال ورودی های ۱، ۲ و ... به سلول j ، u_j ترکیب خطی ورودی ها، b_j اریب، φ تابع تحریک و y_j خروجی سلول است. می توان مجموع u_j و b_j را به صورت زیر نمایش داد:

$$v_j = (u_j + b_j) \quad (3-1)$$

³³ Transfer function

³⁴ Activation function

³⁵ Bias



شکل (۵-۱): مدل بلوکی سلول عصبی

تابع تحریک یا تابع انتقال که با ϕ نشان داده می شود بر روی مجموع موزون (شامل اریب) عمل می کند. این تابع می تواند خطی یا غیر خطی باشد و بر اساس نیاز خاص حل یک مسأله انتخاب می شود. انواع مختلفی از توابع تحریک وجود دارند که مهم ترین آنها به شرح زیر می باشند:

۱- تابع حد آستانه^{۳۶}: این تابع در شکل (۵-۱ الف) نشان داده شده و به صورت زیر بیان می شود:

$$\phi(v) = \begin{cases} 1 & v \geq 0, \\ 0 & v < 0. \end{cases} \quad (۴-۱)$$

۲- تابع قسمتی خطی^{۳۷}: تابع قسمتی خطی که در شکل (۵-۱ ب) نشان داده شده است به صورت زیر تعریف می شود:

$$\phi(v) = \begin{cases} 1 & v \geq 0.5, \\ v & -0.5 < v < 0.5, \\ 0 & v < -0.5. \end{cases} \quad (۵-۱)$$

۳- تابع سیگموئید^{۳۸}: نام این تابع به علت شکل S مانند آن است و پرکاربردترین تابع تحریک محسوب می شود. این تابع بین رفتار خطی و غیرخطی، به آرامی تغییر می کند. مثالی از تابع سیگموئید تابع لجستیک^{۳۹} است که به صورت زیر بیان می شود:

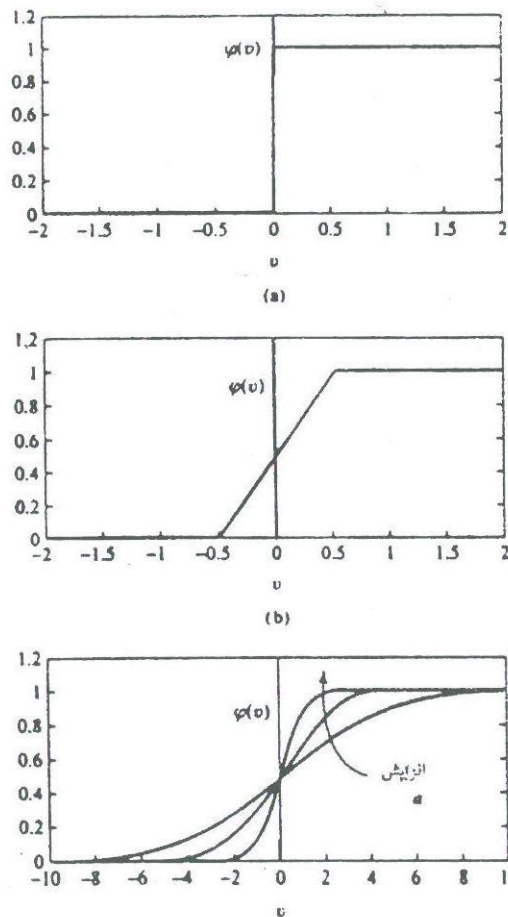
³⁶ Threshold function

³⁷ Piecewise-Linear function

³⁸ Sigmoid function

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (6-1)$$

که در آن a پارامتر شیب تابع سیگموئید است، با تغییر پارامتر a ، توابع مختلفی با شیب های متفاوت مانند آنچه که در شکل (۵-۱-ج) نشان داده شده به دست می آید.



شکل (۵-۱): انواع تابع تحریک، (a) تابع حد آستانه (b) تابع قسمتی خطی (c) تابع سیگموئید با شیب های متغیر

۵-۱ شباهت سلول عصبی مصنوعی و نرون بیولوژیک

از آنجا که ایده اولیه شبکه های عصبی به تقلید از ویژگی های نرون های بیولوژیک بوده است، لذا شباهت زیادی بین آنها وجود دارد. یک سلول عصبی مصنوعی یا گره می تواند ورودی هایش را از چندین گره دیگر دریافت کند. این ورودی های چند گانه را می توان شبیه دندریت

³⁹ Logistic function

ها در نرون بیولوژیک دانست. مشابه نرون ها هر گره یک خروجی تولید می کند که با آکسون قابل قیاس است. در محاسبه خروجی، به اطلاعات ورودی وزن مثبت یا منفی داده می شود. این وزن ها معادل اثر تحریک یا جلوگیری از اثر تحریک ناشی از ناقل عصبی در نرون ها است. در نرون ها فقط در صورتی که ولتاژ از آستانه خاصی فراتر رود خروجی تولید می شود. در گره ها این عمل در تابع انتقال نوع حد آستانه تقلید شده است. البته همه شبکه های عصبی از تابع انتقال نوع حد آستانه استفاده نمی کنند. با این وجود بقیه انواع انتقال نیز تقریباً شبیه به تابع حد آستانه رفتار می کنند [۳].

۱-۶ فرایند یادگیری

آموزش فرایندی است که منجر به دستیابی وزن های بهینه برای ورودی هر سلول شبکه عصبی می گردد. هدف آموزش آن است که وزن هایی تولید کند که خروجی شبکه را در نزدیک ترین حد ممکن به خروجی مطلوب قرار دهد [۴]. رفتار سیستم های یادگیر^{۴۰} توسط الگوریتم های بازگشتی بیان می شود. به همین دلیل به این الگوریتم ها قوانین یادگیری گویند و عموماً توسط معادلات دیفرانسیلی (تفاضلی) بیان می شود. این الگوریتم ها روی اطلاعات موجود آن گونه پردازش می کنند که شاخص اجرایی مشخص شده ای که عموماً تقریبی از هدف خاص که مقصود پروسه یادگیری است، بهینه گردد و این کار تنها راه جبران نمودن نقصان اطلاعات اولیه می باشد [۲۷]. یادگیری شبکه های عصبی به دو گروه عمده تقسیم می شود:

۱- یادگیری نظارتی^{۴۱} : در یادگیری نظارتی به شبکه آموخته می شود که بین داده های آموزشی و خروجی های مربوطه ارتباط برقرار کند. این کار از طریق حداقل کردن اختلاف بین نتایج حاصل از مجموعه های آموزشی و خروجی های مطلوب انجام می شود [۶]. بنابراین برای یادگیری نظارتی، وجود مجموعه آموزشی ضروری است. بعد از تنظیم اوزان، یک آزمون برای تعیین میزان دقت

⁴⁰ Learning systems

⁴¹ Supervised learning

جواب های شبکه انجام می شود تا تضمین کند که شبکه، الگوی داده های ورودی را شناخته است. در حالت ایده آل مجموعه داده های آزمون باید از داده های آموزش جدا باشد [۳].

۲- یادگیری غیر نظارتی^{۴۲}: در یادگیری غیر نظارتی از داده هایی با جواب های نامعلوم استفاده می شود. در این حالت، هدف یافتن الگویی بین داده هاست که بتواند بر اساس خصوصیات مشابه، داده ها را گروه بندی کند. روش اصلی انجام یادگیری غیر نظارتی، خوشه بندی است [۳].

۱-۶-۱ شبکه های عصبی به عنوان سیستم های دینامیکی آموزش پذیر

خاصیت یادگیری شبکه های عصبی از اهمیت ویژه ای برخوردار می باشند. شبکه های عصبی به عنوان سیستم های یادگیری دارای این توانایی هستند که از گذشته، تجربه و محیط بیاموزند و رفتار خود را حین یادگیری بهبود بخشند. بهبود در یادگیری در طول زمان باید براساس معیاری سنجیده شود. معیار بهبود، همان هدف سیستم یادگیر را مدل می کند.

در این جا قانون یادگیری توسط روابط بازگشتی، عموماً به صورت معادلات دینامیکی (تفاضلی) بیان می شود. قانون یادگیری در این جا روندی است که توسط آن ماتریس وزن ها و بردار های بایاس شبکه عصبی تنظیم می شود. هدف قانون یادگیری، آموزش شبکه عصبی جهت انجام کار مشخصی است. نوع یادگیری توسط روندی که طبق آن پارامترهای شبکه تنظیم می گردند، مشخص می شود [۲۷].

۱-۶-۲ الگوریتم پس انتشار

رایج ترین تکنیک آموزش نظارتی، الگوریتم پس انتشار خطا است یادگیری این الگوریتم بر پایه قانون تصحیح خطا^{۴۳} بنا شده است که می توان آن را تعمیم الگوریتم موسوم به حداقل میانگین

⁴² Unsupervised learning

⁴³ Error correction learning rule

مربعات^{۴۴} دانست. یادگیری از طریق این روش (پس انتشار) دو مرحله دارد: مرحله پیشروی و مرحله بازگشت در مرحله پیشروی ورودی ها به صورت لایه به لایه در شبکه پیش می رود و در پایان یک سری خروجی به عنوان جواب حقیقی شبکه بدست می آید. در این مرحله اوزان اتصال ثابت است. در مرحله بازگشت، اوزان اتصال بر اساس قانون تصحیح خطا، تغییر می کند. تفاضل پاسخ حقیقی شبکه و پاسخ مورد انتظار که خطا نامیده می شود، در جهت مخالف اتصالات در شبکه منتشر می شود و اوزان به گونه ای تغییر می یابند که پاسخ حقیقی شبکه به پاسخ مطلوب نزدیکتر شود[1].

مراحل الگوریتم پس انتشار را می توان به صورت زیر بیان کرد:

- ۱- شبکه یک مثال آموزشی دریافت می کند و با استفاده از اوزان موجود در شبکه (که در ابتدا به صورت تصادفی مقدار دهی می شوند) خروجی ها را محاسبه می کند.
 - ۲- خطا یعنی اختلاف بین نتیجه محاسبه شده (خروجی) و مقدار مورد انتظار محاسبه می شود.
 - ۳- خطا درون شبکه منتشر می شود و اوزان برای حداقل نمودن خطا، از نو تنظیم می شوند.
- مهم ترین بخش این الگوریتم، تنظیم اوزان یک سلول است. این روش، اوزان را به گونه ای تغییر می دهد که خطا کاهش یابد (نه اینکه حذف شود). این تغییر بسیار کوچک است و در واقع هر تنظیم وزن برای هر مثال از مجموعه آموزشی، وزن را به آرامی به سمت مقدار بهینه اش سوق می دهد. پس از نشان دادن تعداد کافی مثال آموزشی، اوزان شبکه تغییر معنی داری نشان نمی دهند. در این صورت خطا کاهش بیشتری نشان نخواهد داد. درچنین حالتی شبکه ورودی را فرا گرفته و آموزش پایان یافته است. لازم به ذکر است که هدف از آموزش، شناخت الگوهای موجود در مثال هاست نه اینکه شبکه صرفاً مثال ها را بیاموزد و جواب آنها را به خاطر بسپارد[۴].

اکنون مراحل انجام الگوریتم را به صورت ریاضی بیان می کنیم[۱]، خطای خروجی سلول j

(در لایه خروجی) در تکرار n ام عبارت است از

$$e_j(n) = d_j(n) - y_j(n)$$

⁴⁴ Least mean square

که $d_j(n)$ پاسخ مطلوب و $y_j(n)$ خروجی سلول j است. اگر انرژی خطا برای سلول j را به صورت $\frac{1}{2}e_j^2(n)$ تعریف کنیم، انرژی خطای کل (برای تمام سلول های لایه خروجی) عبارت است از:

$$E(n) = \frac{1}{2} \sum_{j=1}^c e_j^2(n)$$

که در آن c تعداد سلول های لایه های خروجی است. فرض کنید تعداد مثال ها در مجموعه آموزشی، برابر N باشد. میانگین مربعات انرژی خطا به صورت زیر نشان داده می شود:

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n)$$

E_{av} تابعی از تمام پارامتر های آزاد شبکه (مانند اوزان اتصال و ارب ها) می باشد. برای یک مجموعه آموزشی، E_{av} معرف تابع هزینه (تابع خطا) است. هدف فرایند یادگیری حداقل نمودن E_{av} است.

الگوریتم پس انتشار، مقدار تصحیح وزن معادل $\Delta w_{ji}(n)$ را به وزن قبلی $w_{ji}(n)$ اضافه می کند. این مقدار تصحیح وزن متناسب است با مشتق جزئی $E(n)$ نسبت به $w_{ji}(n)$ ، یعنی $\frac{\partial E(n)}{\partial w_{ji}(n)}$. مشتق جزئی مذکور، جهت جستجو را در «فضای اوزان» برای یافتن w_{ji} مناسب تعیین

می کند. فرمول تعیین مقدار تصحیح وزن را می توان به صورت زیر نوشت:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}$$

η را ضریب یادگیری^{۴۵} گویند.

۱-۶-۳ انتخاب نرخ یادگیری

هر چه ضریب یادگیری η کوچکتر باشد، تغییر اوزان شبکه در هر تکرار، کوچکتر خواهد بود و این به معنای نرخ یادگیری کمتر است. از طرف دیگر، اگر سعی شود که با افزایش ضریب η سرعت

⁴⁵ Learning rate parametr

یادگیری زیاد شود، تغییرات بزرگی در اوزان اتصال به وجود می آید که ممکن است شبکه را ناپایدار کند [۱].

۱-۷ تاریخچه کاربرد شبکه های عصبی مصنوعی در حل مسائل بهینه سازی

مسائل بهینه سازی در طیف وسیعی از کاربرد های علوم و مهندسی شامل پردازش تصویر^{۴۶}، شناسایی سیستم ها^{۴۷}، حل معادلات، طراحی فیلتر^{۴۸}، تقریب توابع^{۴۹}، آنالیز رگرسیون^{۵۰} و... پدیدار می شود. در بسیاری از کاربردهای علوم و مهندسی جواب زمان واقعی^{۵۱} مسائل بهینه سازی بسیار مورد نیاز می باشد.

الگوریتم های قدیمی برای کامپیوترهای دیجیتال چندان موثر نیستند زیرا زمان مورد نیاز برای به دست آوردن جواب وابستگی زیادی به بُعد^{۵۲} و ساختار^{۵۳} مسأله دارد. یک رهیافت ممکن و بسیار امیدوارکننده کاربرد شبکه های عصبی مصنوعی می باشد.

به واسطه ذات شبکه های عصبی که می توانند حجم عظیمی از کارهای موازی را با هم انجام دهند رهیافت شبکه عصبی می تواند مسائل بهینه سازی در زمان واقعی را با سرعت بیشتری نسبت به سایر الگوریتم ها ی بهینه سازی موجود حل کند. هاپفیلد^{۵۴} و تنک^{۵۵} [۸] ابتدا شبکه عصبی را برای حل مسائل برنامه ریزی خطی پیشنهاد دادند، کار بنیادین آنها الهام بخش تحقیقات زیادی برای ارائه شبکه های عصبی دیگر برای حل مسائل برنامه ریزی خطی و غیر خطی شد.

⁴⁶ Signal processing

⁴⁷ System identification

⁴⁸ Filter design

⁴⁹ Function approximation

⁵⁰ Regression analysis

⁵¹ Real time

⁵² Dimension

⁵³ Structure

⁵⁴ Hopfield

⁵⁵ Tank

بدین ترتیب شبکه های عصبی بهینه سازی زیادی گسترش یافتند. به عنوان مثال کندی^{۵۶} و چاو^{۵۷} [۹] یک شبکه عصبی برای حل مسائل برنامه ریزی غیر خطی پیشنهاد کردند که از روش گرادیان و روش تابع جریمه استفاده می نمودند و نقاط تعادلشان متناظر با تقریبی از جواب های بهینه بود. با استفاده از روش گرادیان و تصویر، بازردوم^{۵۸} و پاتیسن^{۵۹} [۱۰] یک شبکه عصبی برای حل مسائل بهینه سازی درجه دوم تنها با متغیرهای کراندار معرفی کردند. این شبکه عصبی تحت شرایط خاصی به تنها جواب بهینه همگرا می باشد.

بر پایه روش های گرادیان و تصویر، ایکسیا^{۶۰} و همکارانش [۱۱]-[۱۶] چندین شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم و خطی با جواب های بهینه غیر یکتا معرفی کردند و همگرایی سراسری آنها را به جواب های دقیق ثابت نمودند.

بر پایه بعضی از ملاحظات در اجرای سخت افزاری، یک شبکه عصبی با عملکرد محاسباتی خوب باید سه ویژگی اساسی داشته باشد:

۱. همگرایی شبکه عصبی به ازای هر وضعیت دلخواه داده شده تضمین شود.
 ۲. در طراحی شبکه ترجیحاً هیچ پارامتر متغیری وجود نداشته باشد.
 ۳. نقاط تعادل شبکه متناظر با جواب های دقیق یا تقریبی باشد.
- از دیدگاه ریاضی این خصوصیات مربوط به تکنیک های بهینه سازی هستند که برای طراحی مدل های شبکه های عصبی بهینه سازی به کار می روند.

در مجموع برای فرمول بندی یک مسأله بهینه سازی برحسب یک شبکه عصبی دو نوع روش وجود دارد. یک رهیافت که عموماً در ایجاد یک شبکه عصبی بهینه سازی استفاده می شود این است که در ابتدا مسأله بهینه سازی را به یک مسأله بهینه سازی نامقید تبدیل می کنند و سپس یک شبکه عصبی طراحی می کنند که مسأله نامقید را با استفاده از روش گرادیان حل کند. روش گرادیان

⁵⁶ Kennedy

⁵⁷ chau

⁵⁸ Bouzardoum

⁵⁹ Pattison

⁶⁰ Xia

یک مزیت برای آن دسته از شبکه های عصبی است که مستقیماً از مشتق تابع انرژی استفاده می کنند اما عیب آنها این است که همگرایی در این موارد تضمین نمی شود به ویژه درحالتی که مجموعه های جواب کراندار نیستند.

رهیافت دیگر این است که یک مجموعه از معادلات دیفرانسیل ایجاد می کنند به طوری که نقاط تعادل آنها متناظر با جواب مطلوب باشد و بعد یک تابع لیاپانوف مناسب پیدا می کنند به طوری که همه مسیر های سیستم به نقاط تعادل همگرا باشند.

فصل دوم

پیش نیازها

۲-۱ مقدمه

در این فصل تعاریف مورد نیاز در فصول بعدی به طور مختصر آورده شده است. ابتدا تعاریف مرتبط با تابع محدب را بیان نموده و در خصوص مسائل بهینه سازی مقید و نامقید به ذکر شروط لازم و کافی بهینگی می پردازیم. سپس تابع جریمه را معرفی کرده و مفاهیم پایداری و تابع انرژی در سیستم های دینامیکی را ارائه می دهیم. در پایان نیز مفهوم عملگر تصویر را معرفی خواهیم کرد.

۲-۲ توابع محدب

در توسعه برنامه ریزی غیر خطی، تابع محدب همواره به عنوان یک مفهوم ریاضی اصلی مورد نیاز است.

تعریف ۲-۲-۱: مجموعه $E \subset R^n$ را محدب می گوئیم اگر:

$$\forall x, y \in E, \lambda \in (0,1), \lambda x + (1-\lambda)y \in E.$$

تعریف ۲-۲-۲: تابع $f(x) = f(x_1, x_2, \dots, x_n)$ بر روی یک مجموعه نقطه واقع در یک مجموعه

محدب $E \subset R^n$ یک تابع محدب نامیده می شود اگر:

$$\forall x, y \in E, \lambda \in (0,1), f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y).$$

تابع f را روی E مقعر گوئیم هر گاه f -محدب باشد.

قضیه ۲-۲-۳ ([۲۸]): اگر $f(x)$ بر مجموعه محدب E تابعی محدب باشد آنگاه $f(x)$ دارای یک

کمینه موضعی^{۶۱} است. اگر چنین کمینه ای یافت شود، یک کمینه سراسری^{۶۲} است و بر روی مجموعه محدب به دست آورده می شود.

⁶¹ Local minimum

⁶² Global minimum

قضیه ۴-۲-۲ ([۲۸]): فرض کنید E مجموعه نقاطی از R^n باشد که در قید های $g_k(x) \leq 0, x \geq 0, k = (1, \dots, m)$ صدق کند. اگر $g_k(x)$ ها توابعی محدب باشند آنگاه E یک مجموعه محدب است.

قضیه ۳-۲-۲ و ۴-۲-۲ بیان می کند که برای یک مسأله برنامه ریزی کلی کمینه سازی، اگر $f(x)$ و همه $g_k(x)$ ها توابعی محدب باشند آنگاه یک کمینه موضعی از $f(x)$ تحت این مجموعه از قیود، یک کمینه سراسری است.

تعریف ۵-۲-۲: یک ماتریس $M(X)_{n \times n}$ که عناصر آن $m_{ij}, i=1, \dots, n, j=1, \dots, n$ توابعی روی مجموعه $E \subset R^n$ هستند،

۱. نیمه معین مثبت روی E نامیده می شود، اگر:

$$\forall V \in R^n, V \neq 0, X \in E, V^T M(X) V \geq 0.$$

۲. معین مثبت روی E نامیده می شود، اگر:

$$\forall V \in R^n, V \neq 0, X \in E, V^T M(X) V > 0.$$

۳. معین مثبت قوی روی E نامیم اگر عددی مثبت مانند α وجود داشته باشد به طوری که:

$$\forall V \in R^n, X \in E, V^T M(X) V \geq \alpha \|V\|^2.$$

تعاریف فوق را می توان بر اساس مفهوم مقدار ویژه نیز ارائه داد.

تعریف ۶-۲-۲: اگر $\gamma(x)$ کوچکترین مقدار ویژه ماتریس $M(X)_{n \times n}$ باشد آنگاه:

۱. $M(X)_{n \times n}$ روی E نیمه معین مثبت است اگر فقط اگر به ازای هر $X \in E, \gamma(X) \geq 0$.

۲. $M(X)_{n \times n}$ روی E معین مثبت است اگر فقط اگر به ازای هر $X \in E, \gamma(X) > 0$.

۳. $M(X)_{n \times n}$ روی E معین مثبت قوی است اگر فقط اگر به ازای هر

$$\gamma(X) \geq \alpha \geq 0, X \in E$$

۳-۲ شرایط لازم و کافی بهینگی برای مسائل نا مقید

قضیه ۳-۲-۱ ([۱۷]): شرط لازم مرتبه اول: فرض کنید X یک مجموعه باز در R^n باشد. مسأله

برنامه ریزی نامقید زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t} \quad & \\ & x \in X, \end{aligned}$$

که در آن $f: R^n \rightarrow R^1$ در $\hat{x}$ مشتق پذیر می باشد. اگر $\hat{x}$ یک مینیمم موضعی f باشد، آنگاه $\nabla f(\hat{x}) = 0$.

قضیه ۳-۲-۲ ([۱۷]): شرط لازم مرتبه دوم: فرض کنید مشتق دوم $f: R^n \rightarrow R^1$ در $\hat{x}$ موجود باشد. اگر $\hat{x}$ یک مینیمم موضعی باشد، آنگاه $\nabla f(\hat{x}) = 0$ و ماتریس هسیان f'' در $\hat{x}$ نیمه معین مثبت است.

قضیه ۳-۳-۲ ([۱۷]): شرط کافی: فرض کنید مشتق دوم $f: R^n \rightarrow R^1$ در $\hat{x}$ موجود باشد. اگر $\nabla f(\hat{x}) = 0$ و ماتریس هسیان f در $\hat{x}$ معین مثبت باشد. آنگاه $\hat{x}$ یک مینیمم موضعی اکید است.

۴-۲ شرایط لازم و کافی بهینگی برای مسائل مقید

قضیه ۴-۲-۱ ([۱۷]): شرایط لازم (کاروش - کان - تاکر KKT)^{۶۴}: فرض کنید X یک مجموعه باز در

R^n است. مسأله مینیمم سازی مقید زیر را در نظر بگیرید:

$$\left\{ \begin{array}{ll} \min & f(x_1, x_2, \dots, x_n) \\ \text{s.t} & \\ & g_k(x_1, x_2, \dots, x_n) \leq 0, \quad k = 1, \dots, m \\ & h_j(x_1, x_2, \dots, x_n) = 0, \quad j = 1, \dots, l \\ & x = (x_1, x_2, \dots, x_n)^T \in X, \end{array} \right. \quad (۴-۲-۱)$$

⁶³ Hessian

⁶⁴ Karush- Kuhn- Tucker

که در آن توابع $f: R^n \rightarrow R^1$ و $g_k: R^n \rightarrow R^1$ ، $k = (1, \dots, m)$ و $h_j: R^n \rightarrow R^1$ ، $j = (1, \dots, l)$. فرض کنید $\hat{x}$ یک جواب شدنی این مسأله باشد، و $K = \{k: g_k(\hat{x}) = 0\}$. همچنین فرض کنید f و $g_k, k \in K$ در $\hat{x}$ مشتق پذیر باشند، $g_k, k \notin K$ در $\hat{x}$ پیوسته باشند و h_j برای $j = 1, \dots, l$ دارای مشتق پیوسته باشند. بعلاوه فرض کنید $\nabla g_k(\hat{x}) = 0$ برای $k \in K$ و $\nabla h_j(\hat{x})$ برای $j = 1, \dots, l$ مجموعاً مستقل خطی باشند. اگر $\hat{x}$ مینیمم موضعی برای (۲-۴-۱) باشد، آنگاه اسکالر های یکتای $u_k \geq 0$ برای $k \in K$ و $v_j \geq 0$ برای $j = 1, \dots, l$ موجود هستند به طوری که:

$$\begin{cases} \nabla f(\hat{x}) + \sum_{k \in K} \hat{u}_k \nabla g_k(\hat{x}) + \sum_{j=1}^l \hat{v}_j \nabla h_j(\hat{x}) = 0, \\ \hat{u}_k \geq 0, k \in K. \end{cases}$$

با فرض های بالا اگر g_k برای $k \notin K$ در $\hat{x}$ مشتق پذیر باشند، آنگاه شرایط KKT می تواند به فرم زیر نوشته شود:

$$\begin{cases} \nabla f(\hat{x}) + \sum_{k=1}^m \hat{u}_k \nabla g_k(\hat{x}) + \sum_{j=1}^l \hat{v}_j \nabla h_j(\hat{x}) = 0, \\ u_k g_k(\hat{x}) = 0, \quad k = 1, \dots, m, \\ \hat{u}_k \geq 0, \quad k = 1, \dots, m. \end{cases}$$

همچنین شرایط KKT می تواند به فرم ماتریسی زیر نوشته شود:

$$\begin{cases} \nabla f(\hat{x}) + \nabla g(\hat{x})^T u + \nabla h(\hat{x})^T v = 0, \\ u^T g(\hat{x}) = 0, \\ u \geq 0, \end{cases}$$

که در آن $\nabla g(\hat{x})$ ماتریس ژاکوبین $m \times n$ است که سطر i ام آن $\nabla g_k(\hat{x})$ ، ماتریس ژاکوبین $\nabla h(\hat{x})$ ماتریس ژاکوبین $l \times n$ است که سطر i ام آن $\nabla h_j(\hat{x})$ و $\hat{u}$ یک بردار m تایی و $\hat{v}$ یک بردار l تایی می باشد. $(\hat{u}^T, \hat{v}^T)^T$ بردار ضرایب لاگرانژ نامیده می شود.

قضیه ۲-۴-۲: ([۱۷]) شرایط کافی (کاروش - کان - تاکر): فرض کنید X یک مجموعه باز در R^n است. و فرض کنید $f: R^n \rightarrow R^1$ و $g_k: R^n \rightarrow R^1$ ، $k = (1, \dots, m)$ و $h_j: R^n \rightarrow R^1$ ، $j = (1, \dots, l)$.

مسأله (۱-۴-۲) را در نظر بگیرید. فرض کنید $\hat{x}$ یک جواب شدنی باشد، و $K = \{k : g_k(\hat{x}) = 0\}$. اگر شرایط KKT در $\hat{x}$ برقرار باشد، یعنی اسکالر $u_k \geq 0$ برای $k \in K$ و $v_j \geq 0$ برای $j = 1, \dots, l$ موجود باشند به طوریکه:

$$\nabla f(\hat{x}) + \sum_{k \in K} \hat{u}_k \nabla g_k(\hat{x}) + \sum_{j=1}^l \hat{v}_j \nabla h_j(\hat{x}) = 0,$$

آنگاه $\hat{x}$ یک مینیمم موضعی برای (۱-۴-۲) خواهد بود.

۲-۵ توابع جریمه ای

مسأله مینیمم سازی زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, \\ & g(x) \leq 0. \end{aligned} \tag{۱-۵-۲}$$

روش تابع جریمه ای مبتنی بر این اندیشه است که به جای حل مسأله (۱-۵-۲) یک مسأله نامقید به صورت زیر حل شود:

$$\begin{aligned} \min \quad & f(x) + \mu \alpha(x) \\ & x \in R^n \end{aligned}$$

که در آن $\alpha(x) = \sum_{i=1}^m |h_i(x)|^p + \sum_{j=1}^l [\max(0, g_j(x))]^p$ ، p یک عدد صحیح مثبت و $\mu > 0$ یک عدد خیلی بزرگ می باشد.

۲-۶ پایداری

سیستم دینامیکی زیر را در نظر بگیرید:

$$\dot{x} = f(x(t)), \quad x(t_0) = x_0 \in R^n, \tag{۱-۶-۲}$$

که در آن f یک تابع از R^n به R^n می باشد. x^* یک نقطه تعادل^{۶۵} برای (۱-۶-۲) نامیده می شود اگر $f(x^*) = 0$.

تعریف ۱-۶-۲: فرض می کنیم که $x(t)$ یک جواب (۱-۶-۲) باشد، نقطه تعادل تنها x^* ، پایدار به مفهوم لیاپانوف^{۶۶} است، با توجه به شکل (۱-۶-۲) اگر برای هر $x_0 = x(t_0)$ و هر $\varepsilon > 0$ ، یک $\delta > 0$ وجود داشته باشد به طوریکه اگر $\|x(t_0) - x^*\| \leq \delta$ آنگاه:

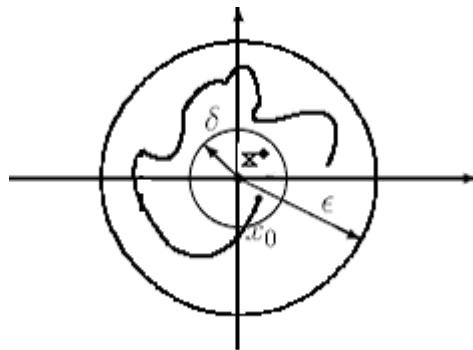
$$\forall t \geq t_0, \quad \|x(t) - x^*\| < \varepsilon.$$

تعریف ۲-۶-۲: سیستم دینامیکی (۱-۶-۲) همگرای سراسری^{۶۷} به مجموعه $\Omega^* = \{x \mid \text{جواب (۱-۶-۲) می باشد} \}$ گفته می شود اگر هر جواب دلخواه $x(t)$ از سیستم در رابطه زیر صدق کند:

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

که در آن

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



شکل (۱-۶-۲) پایداری لیاپانوف

⁶⁵ Equilibrium point

⁶⁶ Lyapunov

⁶⁷ Global convergence

تعریف ۳-۶-۲: سیستم دینامیکی (۱-۶-۲) در نقطه تعادل یکتا x^* پایدار مجانبی سراسری^{۶۸}

نامیده می شود اگر x^* به مفهوم لیاپانوف پایدار باشد و در شرط زیر صدق کند:

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

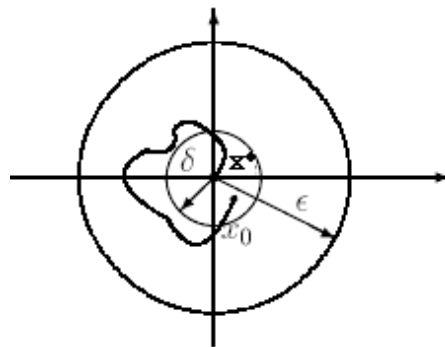
تعریف ۴-۶-۲: سیستم دینامیکی (۱-۶-۲) در نقطه x^* پایدار نمایی سراسری^{۶۹} با نرخ η نامیده

می شود اگر برای هر نقطه آغازین دلخواه، مسیر $x(t)$ در رابطه زیر صدق کند:

$$\forall t \geq t_0, \quad \|x(t) - x^*\| \leq \mu_1 \|x(t_0) - x^*\| \exp(-\eta(t - t_0)),$$

که در آن η و μ_1 ثابت های مثبت و مستقل از نقاط آغازین هستند. واضح است که پایداری نمایی

سراسری پایداری مجانبی سراسری را نتیجه می دهد (شکل (۲-۶-۲)).



شکل (۲-۶-۲) پایداری نمایی

تعریف ۵-۶-۲: مجموعه $M \subseteq R^n$ ، یک مجموعه پایدار^{۷۰} نسبت به سیستم (۱-۶-۲) گفته می شود

اگر $x(t_0) \in M$ و $t_0 \geq 0$ ، آنگاه به ازای هر $t \geq t_0$ ، $x(t) \in M$ باشد.

تعریف ۶-۶-۲: نگاشت F در شرط لپ شیتز^{۷۱} صدق می کند اگر عدد ثابت L وجود داشته باشد

به طوریکه برای هر دو نقطه $x, y \in R^n$ داشته باشیم:

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

⁶⁸ Globally asymptotically stable

⁶⁹ Globally exponentially stable

⁷⁰ Invariant set

⁷¹ Lipschitz

F را لیپ شیتز محلی روی R^n نامیم اگر به ازای هر نقطه از R^n ، یک همسایگی مانند $D_0 \subset R^n$ وجود داشته باشد به طوریکه نامساوی بالا برای هر دو نقطه $x, y \in D_0$ برقرار باشد.

تعریف ۲-۶-۷: نگاشت $F: R^n \rightarrow R^n$ یکنوا^{۷۲} گفته می شود، اگر برای هر زوج از نقاط $x, y \in R^n$ داشته باشیم:

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

همچنین روی R^n اکیداً یکنوا^{۷۳} گفته می شود، اگر نامساوی فوق به صورت اکید برای هر $x \neq y$ برقرار باشد.

F روی R^n یکنوای قوی است اگر برای هر زوج از نقاط $x, y \in R^n$ ثابت $\beta > 0$ وجود داشته باشد، به طوریکه:

$$(x - y)^T (F(x) - F(y)) \geq \beta \|x - y\|^2.$$

قضیه ۲-۶-۸ ([۱۸]): فرض می کنیم که $F: K \subset R^n \rightarrow R^n$ ، یک تابع به طور پیوسته مشتق پذیر روی K باشد و ماتریس ژاکوبین F که لزوماً متقارن نیست، نیمه معین مثبت (معین مثبت) باشد، آنگاه $F(x)$ یکنوا (یکنوای قوی) می باشد.

قضیه ۲-۶-۹ ([۱۹]): در سیستم دینامیکی (۲-۶-۱) فرض می کنیم که f یک تابع پیوسته از R^n به R^n باشد، آنگاه برای هر $x_0 \in R^n$ و $t_0 > 0$ ، یک جواب محلی^{۷۴} $x(t)$ به ازای $t \in [t_0, \tau)$ که $\tau > t_0$ وجود دارد. علاوه بر این اگر f در x_0 در شرط لیپ شیتز محلی صدق کند آنگاه جواب یکتا خواهد بود و اگر f در R^n در شرط لیپ شیتز صدق کند آنگاه τ می تواند تا $+\infty$ توسعه داده شود.

تذکر: همیشه می توان نقطه تعادل را به صورت $x^* = 0$ در نظر گرفت. برای هر نقطه تعادل دیگر می توان با استفاده از تغییرمتغیر، یک سیستم جدید با نقطه تعادل y^* تعریف کرد. برای این منظور تعریف می کنیم:

⁷² Monotone

⁷³ Strictly monotonic

⁷⁴ local solution

$$\begin{aligned}
y &= x - x^* \\
\Rightarrow \dot{y} &= \dot{x} = f(x) \\
\Rightarrow f(x) &= f(y + x^*) = g(y).
\end{aligned}$$

در نتیجه نقطه تعادل سیستم جدید $\dot{y} = g(y), y^* = 0$ می باشد زیرا

$$g(0) = f(0 + x^*) = f(x^*) = 0$$

در نتیجه x^* نقطه تعادل برای سیستم $\dot{x} = f(x)$ است اگر و فقط اگر $y = 0$ نقطه تعادل سیستم $\dot{y} = g(y)$ باشد.

۷-۲ تابع انرژی

قضیه ۷-۲-۱ ([۲۹]): فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم (۱-۶-۲) و تابع

$$E: \Omega \subseteq R^n \rightarrow R \text{ به طور پیوسته مشتق پذیر است اگر}$$

$$1. \quad E(0) = 0$$

$$2. \quad \text{به ازای هر } x \in \Omega \subseteq R^n \setminus \{0\}, E(x) > 0$$

$$3. \quad \text{به ازای هر } x \in \Omega \subseteq R^n \setminus \{0\}, \dot{E}(x) \leq 0$$

آنگاه $x = 0$ نقطه پایداری سیستم خواهد بود، و $E(x)$ را «تابع لیپانوف» یا «تابع انرژی» برای سیستم (۱-۶-۲) نامیم.

قضیه ۷-۲-۲ ([۲۹]): فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم (۱-۶-۲) باشد، و

$$E: \Omega \subseteq R^n \rightarrow R \text{ یک تابع به طور پیوسته مشتق پذیر باشد اگر}$$

$$1. \quad E(0) = 0$$

$$2. \quad \text{به ازای هر } x \in \Omega \subseteq R^n \setminus \{0\}, E(x) > 0$$

$$3. \quad \text{به ازای هر } x \in \Omega \subseteq R^n \setminus \{0\}, \dot{E}(x) < 0$$

آنگاه $x = 0$ پایدار مجانبی خواهد بود.

قضیه ۳-۷-۲ ([۲۹]): فرض کنید که $x=0$ یک نقطه تعادل برای سیستم (۱-۶-۲) و تابع

$E: \Omega \subseteq R^n \rightarrow R$ به طور پیوسته مشتق پذیر باشد اگر

$$1. \quad E(0) = 0$$

$$2. \quad \text{به ازای هر } x \in R^n \setminus \{0\}, E(x) > 0$$

$$3. \quad \text{به ازای هر } x \in R^n \setminus \{0\}, \dot{E}(x) \leq 0$$

آنگاه نقطه $x=0$ پایدار مجانبی سراسری است.

قضیه ۴-۷-۲ ([۲۹]): فرض کنید که $x=0$ یک نقطه تعادل برای سیستم (۱-۶-۲) و تابع

$E: R^n \rightarrow R$ به طور پیوسته مشتق پذیر است، و ثابت های مثبت K_3, K_2, K_1 و P وجود داشته

باشند اگر:

$$K_1 \|x\|^P \leq E(x) \leq K_2 \|x\|^P,$$

$$\dot{E}(x) \leq -K_3 \|x\|^P,$$

آنگاه $x=0$ پایدار نمایی است.

۸-۲ عملگر تصویر

فرض می کنیم که $\Omega = \{x \in R^l \mid d_i \leq x_i \leq h_i, \forall i \in I \subset L\}$ ، $L = \{1, \dots, l\}$. عملگر $P_\Omega: R^l \rightarrow \Omega$

یک عملگر تصویر^{۷۵} نامیده شده و به صورت زیر تعریف می شود:

$$P_\Omega(x) = \arg \min_{v \in \Omega} \|x - v\|, \quad (1-8-2)$$

که در آن $\|\cdot\|$ نرم l_2 را نشان می دهد.

⁷⁵ Projection operator

تعریف ۱-۸-۲: معادله $P_{\Omega}(x - (Mx + q)) = x$ را معادله تصویر خطی می نامند که در آن $q \in R^l$ و $M \in R^{l \times l}$ یک ماتریس نیمه معین مثبت می باشد که لزوماً متقارن نیست و P_{Ω} در (۱-۸-۲) تعریف شده است.

عملگر تصویر $P_{\Omega}(\cdot)$ در (۱-۸-۲) را می توان به صورت $P_{\Omega}(x) = [P_{\Omega}(x_1), \dots, P_{\Omega}(x_m)]^T$ بیان نمود، که در آن $P_{\Omega}(x_i)$ یک تابع تکه ای^{۷۶} می باشد. برای هر $i \in L - I$ ، $P_{\Omega}(x_i) = x_i$ و برای $i \in I$ داریم:

$$P_{\Omega}(x_i) = \begin{cases} d_i & x_i < d_i \\ x_i & d_i \leq x_i \leq h_i \\ h_i & x_i > h_i. \end{cases}$$

اگر $d_i = 0$ ، $h_i = +\infty$ ، $i = (1, \dots, l)$ آنگاه معادله تصویر خطی به صورت زیر تبدیل می شود:

$$P_{\Omega}(x - (Mx + q)) = (x - (Mx + q))^+ = x,$$

که در آن $(x)^+ = [(x_1)^+, \dots, (x_l)^+]^T$ و $(x_i)^+ = \max(x_i, 0)$.

معادله تصویر خطی در بسیاری از کاربردها ظاهر می شود و نقشی اساسی در مسائل بهینه سازی مقید ایفا می کند، زیرا بسیاری از مسائل بهینه سازی مقید در برنامه ریزی می تواند به معادله تصویر خطی معادل تبدیل شوند. معادلات تصویر خطی کاربردهای مهمی در علوم و مهندسی، شامل آنالیز مدارهای مقاومت خطی تکه ای، نامعادلات وردشی، مسائل نقطه ثابت، مدل سازی شبکه ترافیک و ... دارد.

دو نوع سیستم دینامیکی بر پایه عملگر تصویر پیشنهاد شده است، نوع اول توسط فریز [۲۰] به عنوان یک سیستم دینامیکی تصویر عمومی و نوع دوم توسط ناگورنی و داپیوس [۲۱] به عنوان یک سیستم دینامیکی تصویر موضعی طراحی شده است.

⁷⁶ Piecewise function

فصل سوم

مدل های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه سازی

۳-۱ مقدمه

بحث و بررسی در مورد کاربرد شبکه های عصبی (یا شبکه های عصبی مصنوعی) در بهینه سازی از اوایل سال ۱۹۸۰ میلادی آغاز شده است. نتایج پژوهش های انجام شده از آن زمان تا کنون، مواردی چون برنامه ریزی خطی، برنامه ریزی درجه دوم، مسائل مکمل خطی و برنامه ریزی غیر خطی را در بر می گیرد. ایده اصلی در استفاده از شبکه های عصبی مصنوعی برای مسائل بهینه سازی استفاده از یک تابع انرژی (نامنفی) و یک سیستم دینامیکی است که این دو بیان کننده مدل های شبکه های عصبی مصنوعی متناظر مسائل بهینه سازی می باشند. سیستم دینامیکی بیان شده معمولاً یک دستگاه معادلات دیفرانسیل غیر خطی مرتبه اول است. انتظار می رود که برای یک نقطه آغازین نقطه پایداری دستگاه معادلات دیفرانسیل غیر خطی به دست آمده، جواب بهینه مسأله بهینه سازی اصلی باشد. یک اصل اساسی در استفاده از تابع انرژی این است که برای همگرایی دستگاه معادلات دیفرانسیل به نقطه پایداری آن، تابع انرژی متناظر با آن حتماً باید نامنفی باشد. البته می توان مدل شبکه های عصبی نظیر مسائل بهینه سازی را حتی بدون در نظر گرفتن تابع انرژی نیز بیان کرد. بنابراین برای یک مدل متناظر با یک مسائل بهینه سازی، اصل اساسی استفاده از شبکه های عصبی در اینگونه مسائل به صورت زیر بیان می شود:

برای یک نقطه آغازین دلخواه، نقطه پایداری دستگاه معادلات دیفرانسیل غیر خطی

به دست آمده جواب بهینه مسأله اصلی می باشد و برعکس.

مدل های مطرح شده متناظر مسائل مختلف بهینه سازی را می توان به دو قسمت مدل های دو گانی و مدل های جریمه ای تقسیم نمود. نظریه دوگانی و توابع جریمه ای دو نظریه بسیار مهم در مسائل بهینه سازی هستند که اکثر این مسائل بر مبنای این دو روش کلاسیک قابل حل می باشند. در نظریه توابع جریمه ای معمولاً از مدل های گرادیانی برای معرفی مدل مورد نظر استفاده می شود، ولی در نظریه دوگانی از مدل های اولیه - دو گان برای حل این مسائل استفاده می شود. اگر بتوان متناظر با هر مسأله بهینه سازی، روش مشخصی را ارائه نمود که آن روش برای حل آن

مسأله شرایط لازم و کافی را برآورده سازد، آنگاه می توان متناظر با آن روش، یک مدل شبکه عصبی برای مسأله مورد نظر بسازیم. در زیر به بیان چند مدل که تا کنون برای مسائل برنامه ریزی خطی و درجه دوم و غیر خطی ارائه شده اند می پردازیم.

۲-۳ مدل های مختلف شبکه های عصبی

مدل اول [۲۲]

مسأله برنامه ریزی خطی زیر را در نظر بگیرید:

$$\begin{cases} \min & f(x) = a^T x \\ \text{s.t} & \\ & g(x) = Dx - b = 0, \\ & x \geq 0. \end{cases} \quad (۱-۳)$$

که در آن D یک ماتریس $n \times m$ ، $rank(D) = m$ ، $b \in R^n$ و $x, a \in R^m$ می باشند. دوگان (۱-۳) به فرم زیر بیان می شود:

$$\begin{cases} \max & \bar{f}(y) = b^T y \\ \text{s.t} & \\ & \bar{g}(y) = D^T y - a \leq 0, \\ & y \in R^n. \end{cases} \quad (۲-۳)$$

در مرجع [۲۲] مدل متناظر با (۱-۳) و (۲-۳) به فرم زیر نمایش داده شده است:

$$\begin{cases} \frac{dx}{dt} = -[D^T(Dx - b) - \beta(D^T y - a)], \\ \frac{dy}{dt} = -\beta[(Dx + D^T y - a)^+ - b], \end{cases}$$

که در آن $\beta = \|(x + D^T y - a)^+ - x\|_2^2$ و $(x, y) \in \{(x, y) | x \geq 0\}$.

مدل دوم [۲۲]

مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{cases} \min & f(x) = \frac{1}{2} x^T A x + a^T x \\ \text{s.t} & \\ & D x - b = 0, \\ & x \geq 0. \end{cases} \quad (۳-۳)$$

که در آن A یک ماتریس متقارن معین مثبت می باشد. دوگان (۳-۳) به فرم زیر می باشد:

$$\begin{cases} \max & \bar{f}(y) = b^T y - \frac{1}{2} x^T A x \\ \text{s.t} & \\ & D^T y - A x - a \leq 0. \end{cases} \quad (۴-۳)$$

در مرجع [۲۲] مدل متناظر با (۳-۳) و (۴-۳) به فرم زیر بیان شده است:

$$\begin{cases} \frac{dx}{dt} = -\{D^T(x-b) + \gamma(-D^T y + A x + a) + \gamma A[x - (x + D^T y - A x - a)^+]\} \\ \frac{dy}{dt} = -\gamma\{D x - b + D[(x + D^T y - A x - a)^+ - x]\} \end{cases}$$

که در آن $\gamma = \|(x + D^T y - A x - a)^+ - x\|_2^2$ و $(x, y) \in \{(x, y) \mid x \geq 0\}$.

مدل سوم [۸]

مسأله برنامه ریزی غیر خطی زیر را در نظر بگیرید:

$$\begin{cases} \min & f(x) \\ \text{s.t} & \\ & g(x) = [g_1(x), g_2(x), \dots, g_m(x)]^T \leq 0, \end{cases} \quad (۵-۳)$$

که در آن $f(x), g(x) : R^n \rightarrow R^1$ و $f(x), g(x) \in C^2$. برای حل (۵-۳) هاپفیلد و تنک در [۸] یک

مدل شبکه عصبی به صورت زیر ارائه داده اند:

$$\dot{x} = C^{-1} \{-\nabla f(x) - \nabla g(x) g^+(x) - \frac{1}{s} Q^{-1} x\} s$$

که در آن C یک ماتریس قطری $n \times n$ مبین ظرفیت هر نرون، $\nabla f(x)$ و $\nabla g(x)$ به ترتیب گرادیان های توابع f و g ، و Q یک ماتریس قطری $n \times n$ است که اعضای آن به صورت زیر تعریف می شوند:

$$q_{ii} = \frac{1}{\frac{1}{\rho_i} + \sum_{j=1}^m - (d_{ji})},$$

که $\frac{1}{\rho_i}$ ضریب هدایت گرمایی هر نرون، d_{ji} وزن های تخصیص داده شده به نرونهای داخلی و s پارامتر جریمه است. تابع انرژی متناظر به صورت زیر انتخاب شده است:

$$E_1(x) = f(x) + \sum_{j=1}^m [g_j^+(x)]^2 + \sum_{i=1}^n \frac{x_i^2}{2sr_{ii}}.$$

مدل چهارم [۹]

برای حل (۵-۳) کندی و چوآ [۹] مدلی دیگر به صورت زیر ارائه داده اند:

$$\dot{x} = C^{-1}[-\nabla f(x) - \nabla g(x)g^+(x)],$$

که در آن C و s شرایط مدل سوم را دارند. تابع انرژی متناظر این مدل به صورت زیر انتخاب شده است:

$$E_2 = f(x) + \frac{s}{2} \sum_{j=1}^m [g_j^+(x)]^2.$$

مدل پنجم [۲۳]

مجدداً برای حل (۵-۳) توسط رودریگز-واسکوئز [۲۳] به صورت زیر ارائه شده است:

$$\dot{x} = -u_x \nabla f(x) - s \nabla g(x)g^+(x),$$

که در آن u_x متناظر اندیس های شدنی x است چنان که:

$$u_x = \begin{cases} 1 & g(x) \leq 0, \\ 0 & otherwise. \end{cases}$$

تابع انرژی متناظر این مدل نیز به صورت زیر انتخاب شده است:

$$E_3 = -u_x f(x) + \frac{s}{2} \sum_{j=1}^m [g_j^+(x)]^2.$$

همان طور که ملاحظه می کنید، مدل های بیان شده به طور متوالی اصلاح شده اند و این به دلیل مشکلاتی بوده است که هرکدام از این مدل ها داشته اند [۲۲].

مدل ششم [۱۹]

برنامه ریزی محدب درجه دوم (CQP)^{۷۷} زیر را در نظر بگیرید:

$$\begin{cases} \min & f(x) = \frac{1}{2} x^T Q x + D^T x \\ \text{s.t} & \\ & g(x) = Ax - b \leq 0, \\ & h(x) = Ex - f = 0, \end{cases} \quad (۶-۳)$$

که در آن $Q \in R^{n \times n}$ یک ماتریس نیمه معین مثبت، $A \in R^{m \times n}$ ، $b \in R^m$ ، $E \in R^{l \times n}$ ، $f \in R^l$ و $x \in R^n$ می باشند.

دوگان برنامه ریزی محدب درجه دوم (DCQP)^{۷۸} از CQP (۶-۳) به فرم زیر نوشته می شود:

$$\begin{cases} \max & \bar{f}(x, u, v) = f(x) + u^T (Ax - b) + v^T (Ex - f) \\ \text{S.t} & \\ & \bar{g}(x, u, v) = Qx + D + A^T u + E^T v = 0, \\ & u \geq 0. \end{cases} \quad (۷-۳)$$

بر طبق شرایط کاروش - کان - تاکر (KKT) برای (۶-۳) و (۷-۳)، جواب بهینه (۶-۳)

می باشد اگر و تنها اگر بردارهای $u^* \in R^m$ و $v^* \in R^l$ وجود داشته باشد به طوری

که $(x^{*T}, u^{*T}, v^{*T})^T$ در رابطه زیر صدق نماید:

$$\begin{cases} u^* \geq 0, Ax^* - b \leq 0, u^{*T} (Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (۸-۳)$$

⁷⁷ Convex quadratic programming

⁷⁸ Dual Convex quadratic programming

برای حل مسأله (۶-۳) تابع زیر را که معروف به تابع فیشر برمیستر^{۷۹} می باشد را معرفی می کنیم:

$$\begin{cases} \phi: R^2 \rightarrow R^1, \\ \phi(A, B) = \sqrt{a^2 + b^2} - a - b. \end{cases}$$

قضیه ۱-۳ ([۲۴]):

۱. $\phi(a, b) = 0$ اگر و فقط اگر $ab = 0, b \geq 0, a \geq 0$.

۲. مربع $\phi(a, b)$ به طور پیوسته مشتق پذیر باشد.

۳. $\phi(a, b)$ همه جا به جز در مبدأ دو بار به طور پیوسته مشتق پذیر است، ولی در مبدأ قویاً^{۸۰}

هموار است.

متناظر با دستگاه (۶-۳) و (۷-۳) دستگاه زیر را در نظر می گیریم:

$$\phi(x, u, v) = \begin{bmatrix} \phi(\alpha, Qx + D + A^T u + E^T v) \\ \phi(u, Ax - b) \\ \phi(\beta, Ex - f) \end{bmatrix} = 0, \quad (۹-۳)$$

که در آن $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T > 0$ ، $u = (u_1, u_2, \dots, u_m)$ و $\beta = (\beta_1, \beta_2, \dots, \beta_l)^T > 0$ می باشند.

مدل شبکه عصبی متناظر با (۹-۳) به صورت زیر بیان شده است [۱۹]:

$$\begin{cases} \dot{y} = -\tau \nabla E(y(t)), & \tau > 0, \\ y(t_0) = y_0 \in R^{n+m+l}, \end{cases} \quad (۱۰-۳)$$

که در آن $y = (x^T, u^T, v^T)^T$ و τ ضریبی برای افزایش سرعت همگرایی و کاهش تعداد تکرار ها در

روش عددی انتخاب شده برای حل دستگاه (۱۰-۳) می باشد. تابع انرژی متناظر این مدل به صورت

زیر انتخاب شده است:

$$E(y) = \frac{1}{2} \|\phi(y)\|^2.$$

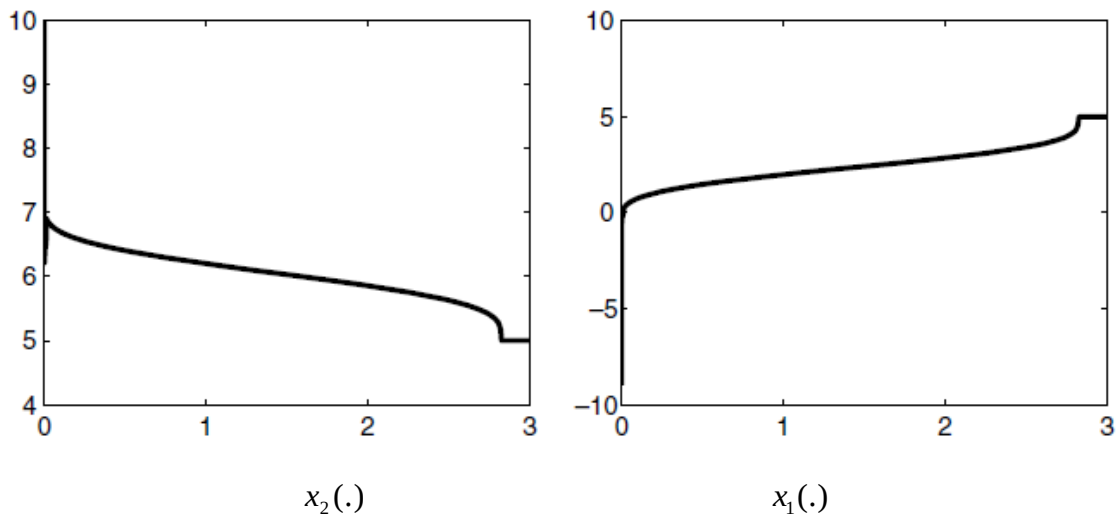
⁷⁹ Fisher Bermister

⁸⁰ Stringly smooth

مثال ۱-۳ [۱۹] مسأله برنامه ریزی خطی زیر را در نظر بگیرید:

$$\left\{ \begin{array}{ll} \min & f(x) = -x_1 - x_2 \\ \text{s.t} & \\ & \frac{5}{12}x_1 - x_2 \leq \frac{35}{12}, \\ & \frac{5}{2}x_1 + x_2 \leq \frac{35}{2}, \\ & -x_1 \leq 5, \quad x_2 \leq 5. \end{array} \right.$$

با توجه به شبکه عصبی معرفی شده در (۱۰-۳) جواب بهینه این مسأله $x^* = (5, 5)^T$ است. همگرایی شبکه عصبی به x^* در شکل (۱-۳) نشان داده شده است.

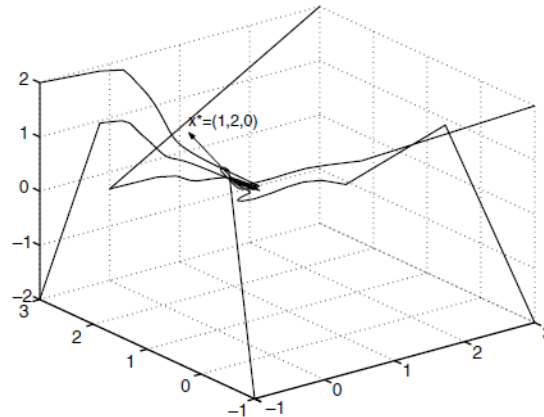


شکل (۱-۳): رفتار شبکه عصبی (۱۰-۳) در مثال (۱-۳) با نقطه آغازین $x_0 = (10, -9)$.

مثال ۲-۳ [۱۹] مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\left\{ \begin{array}{ll} \min & x_1^2 + x_2^2 + x_3^2 \\ \text{s.t} & \\ & 2x_1 + x_2 - 5 \leq 0, \\ & x_1 + x_3 - 2 \leq 0, \\ & -x_1 + 1 \leq 0, \\ & -x_2 + 2 \leq 0, \\ & -x_3 \leq 0. \end{array} \right.$$

با توجه به شبکه عصبی معرفی شده در (۳-۱۰) جواب بهینه $x^* = (1, 2, 0)^T$ است که همگرایی شبکه عصبی به این نقاط در شکل (۳-۲) نشان داده شده است. ملاحظه می شود که با شروع از شش نقطه آغازین دلخواه، همه مسیرها به x^* همگرا شده اند و بنابراین مدل همگرایی سراسری است.



شکل (۳-۲): همگرایی سراسری شبکه عصبی (۳-۱۰) در مثال (۳-۲) با شش نقطه آغازین دلخواه.

مدل هفتم [۱۹]:

برای طراحی مدل هفتم، مجدداً شرایط KKT را برای حل مسائل (۳-۶) و (۳-۷) به صورت

زیر در نظر می گیریم:

$$\begin{cases} (u + Ax - b)^+ = u, \\ Qx + D + A^T u + E^T v = 0, \\ Ex - f = 0. \end{cases}$$

آنگاه متناظر (۳-۶) و (۳-۷) دستگاه (۳-۱۱) تعریف می شود:

$$U(x, u, v) = \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}. \quad (۳-۱۱)$$

مدل شبکه عصبی متناظر برای حل (۳-۶) و (۳-۷) به صورت زیر بیان شده است [۱۹]:

$$\begin{cases} \frac{dy}{dt} = \kappa U(y), \\ y(t_0) = y_0. \end{cases} \quad (۱۲-۳)$$

مدل هشتم [۳۰]:

مسأله برنامه ریزی درجه دوم (۶-۳) را در نظر بگیرید. دوگان و شرایط بهینگی برای آن به ترتیب در فرم (۷-۳) و (۸-۳) بیان شده اند که در آن $Q \in R^{n \times n}$ یک ماتریس نیمه معین مثبت است. برای مسائل (۶-۳) - (۸-۳) یک مدل شبکه عصبی به صورت زیر ارائه می شود [۳۰]:

$$\frac{dy}{dt} = (I + M^T) \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}, \quad (۱۳-۳)$$

که در آن κ نرخ همگرایی است و

$$M = \begin{bmatrix} -Q & A^T & E^T \\ A & 0_{m \times m} & 0_{m \times l} \\ E & 0_{l \times m} & 0_{l \times l} \end{bmatrix}.$$

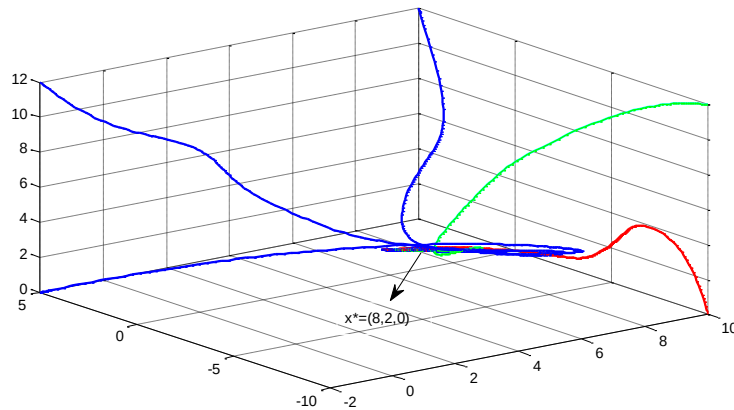
مثال ۳-۳: مسأله برنامه ریزی خطی زیر مفروض است:

$$\begin{aligned} \min \quad & -4x_1 - 5x_2 + 3x_3 \\ \text{s.t.} \quad & -x_1 + x_2 \leq -1, \\ & x_1 + 3x_2 + x_3 \leq 14, \\ & -x_1 \leq 0, \quad -x_2 \leq 0, \quad -x_3 \leq 0, \\ & x_1 + x_2 + x_3 = 10. \end{aligned}$$

جواب دقیق این مسأله $y^* = (8, 2, 0)^T$ می باشد. که در آن

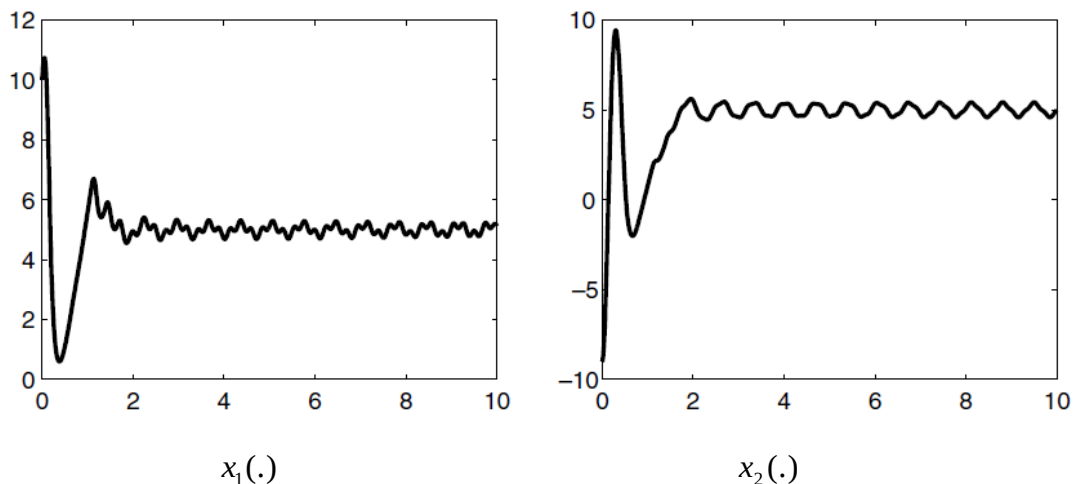
$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -4 \\ -5 \\ 3 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 3 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ 14 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}.$$

رفتار شبکه عصبی (۳-۱۳) در شکل (۳-۳) نشان دهنده آن است که با شروع از پنج نقطه آغازین دلخواه و با نرخ همگرایی $\kappa = 1$ ، تمام مسیرها به y^* همگرا شده اند.

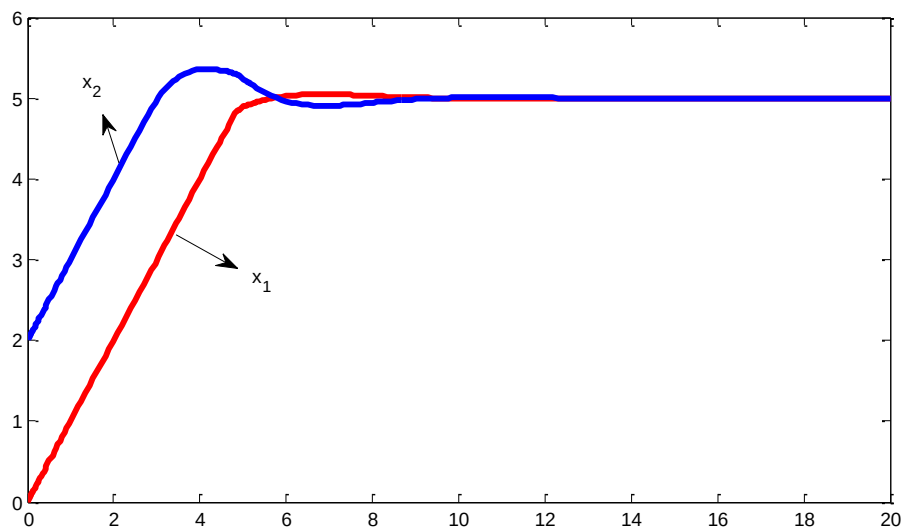


شکل (۳-۳): همگرایی سراسری شبکه عصبی (۳-۱۳) در مثال (۳-۳) با پنج نقطه آغازین دلخواه .

مثال ۳-۴: در مسأله برنامه ریزی خطی مثال (۱-۳) جواب بهینه این مسأله $x^* = (5, 5)^T$ می باشد. اگر این مسأله را با مدل معرفی شده در (۳-۱۲) حل نماییم، مشاهده می کنیم که مدل (۳-۱۲) برای حل این مسأله پایدار است اما پایدار مجانبی به x^* نیست (شکل (۳-۴) را ببینید). بنابراین برای حل این مسأله از مدل (۳-۱۳) استفاده می کنیم. قابل ذکر است که $(I + M^T)$ نقش پایدار کننده سیستم (۳-۱۲) را دارد (شکل (۳-۵)).



شکل (۴-۳): رفتار پایدار شبکه عصبی (۱۲-۳) برای مثال (۴-۳).



شکل (۵-۳): شبکه عصبی (۱۳-۳) برای مثال (۴-۳) همگرای سراسری و پایدار مجانبی به $x^* = (5, 5)^T$ می باشد.

مدل نهم [۲۵]:

مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{cases} \min & f(x) = \frac{1}{2} x^T A x + a^T x \\ \text{s.t} & \\ & D x \leq b, \\ & x \geq 0, \end{cases} \quad (۱۴-۳)$$

دوگان آن عبارت است از:

$$\begin{cases} \max & \bar{f}(w) = b^T w - \frac{1}{2} x^T A x \\ \text{s.t} & \\ & D^T w - A x \leq a, \\ & w \geq 0, \end{cases} \quad (۱۵-۳)$$

که در آن $A \in R^{n \times n}$ متقارن و نیمه معین مثبت می باشد، $D \in R^{m \times n}$ ، $b \in R^m$ و $a \in R^n$ است. با توجه

به شرایط KKT برای مسائل برنامه ریزی محدب، $(x^{*T}, w^{*T})^T$ به ترتیب یک جواب بهینه برای مسائل

(۱۴-۳) و (۱۵-۳) می باشند اگر و فقط اگر $(x^{*T}, w^{*T})^T$ در شرایط کاروش-کان-تاکر KKT زیر صدق کند:

$$\begin{cases} w^* \geq 0, Dx^* - b \leq 0, w^{*T} (Dx^* - b) = 0, \\ Ax^* + a + D^T w^* \geq 0, x^* \geq 0, \\ x^{*T} (Ax^* + a + D^T w^*) = 0. \end{cases}$$

مدل متناظر با (۱۴-۳) و (۱۵-۳) به صورت زیر بیان شده است ([۲۵]):

$$\dot{u} = B(I + M^T) \{ (u - (Mu + q))^+ - u \}, \quad (۱۶-۳)$$

که در آن

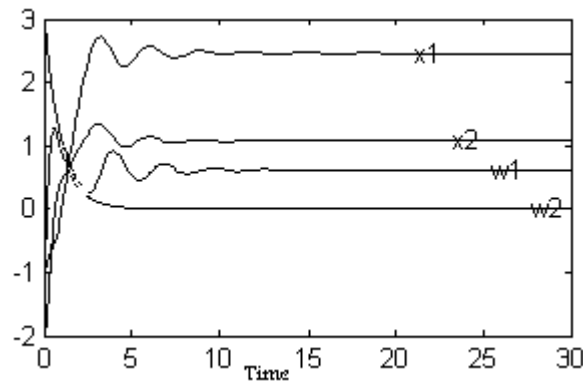
$$u = \begin{pmatrix} x \\ w \end{pmatrix}, q = \begin{pmatrix} a \\ b \end{pmatrix}, M = \begin{pmatrix} A & D^T \\ -D & 0 \end{pmatrix}.$$

M یک ماتریس نیمه معین مثبت می باشد زیرا $u^T M u = x^T A x \geq 0$ همچنین $B = \lambda I$ می باشد که $\lambda > 0$ و با افزایش λ سرعت همگرایی افزایش می یابد.

مثال ۳-۵: مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & -4x_1 + x_2^2 - 2x_1x_2 + 2x_2^2 \\ \text{s.t} \quad & 2x_1 + x_2 \leq 6, \\ & x_1 - 4x_2 \leq 0, \\ & x_1, x_2 \geq 0. \end{aligned}$$

با استفاده از شبکه عصبی (۱۶-۳) جواب بهینه اولیه $x_1 = 2.4615$ و $x_2 = 1.0796$ و جواب بهینه دوگان $w_1 = 0.6154$ و $w_2 = 0$ به دست می آید. همگرایی شبکه عصبی به نقطه بهینه با نقطه آغازین $u_0 = (-1, -2, 3, -1)^T$ در شکل (۶-۳) نشان داده شده است.

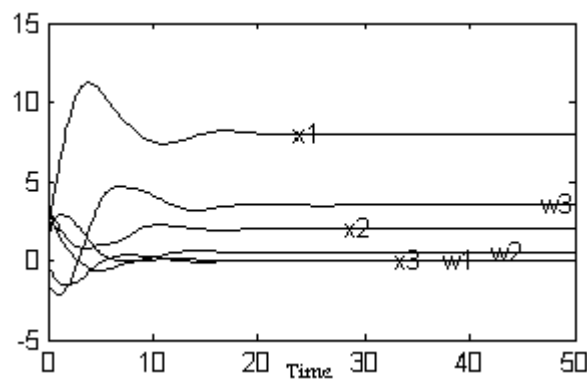


شکل (۳-۶) [۲۵]: پایدار مجانبی شبکه عصبی (۳-۱۶) در مثال (۳-۵) با نقطه آغازین $u_0 = (-1, -2, 3, -1)^T$.

مثال ۳-۶ مسأله ریزی خطی زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & -4x_1 - 5x_2 + 3x_3 \\ \text{s.t} \quad & x_1 + x_2 + x_3 = 10, \\ & -x_1 + x_2 \leq -1, \\ & x_1 + 3x_2 + x_3 \leq 14, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

با توجه به شبکه عصبی معرفی شده در (۳-۱۶)، $x^* = (8, 2, 0)^T$ و $w^* = (3.5, 0, 0.5)^T$ جواب های بهینه وبهینه دوگان هستند. همگرایی شبکه عصبی به نقطه بهینه با نقطه آغازین $u_0 = (1, 2, 3, -1, -2, -3)$ نیز در شکل (۳-۷) نشان داده شده است.



شکل (۳-۷) [۲۵]: پایداری مجانبی شبکه عصبی (۳-۱۶) برای مثال (۳-۶) با نقطه آغازین

$$u_0 = (1, 2, 3, -1, -2, -3)$$

فصل چهارم

حل مسائل برنامه ریزی خطی و درجه دوم با استفاده از عملگر تصویر

۴-۱ مقدمه

در این فصل با استفاده از تکنیک تابع تصویر، یک شبکه عصبی بازگشتی برای حل مسائل برنامه ریزی درجه دوم محدب معرفی می شود. سپس شرایط یکتایی وجود جواب و پایداری مجانبی مدل شبکه عصبی بررسی شده و در نهایت کارایی شبکه عصبی بازگشتی همراه با چند مثال عددی نشان داده می شود.

۴-۲ بیان مسأله

برنامه ریزی محدب درجه دوم (CQP) زیر را در نظر بگیرید:

$$\begin{cases} \min & \frac{1}{2} x^T Q x + D^T x \\ \text{s.t} & \\ & Ax - b \leq 0, \\ & Ex - f = 0, \end{cases} \quad (۱-۴)$$

که در آن $x \in R^n$ و $D \in R^n$ ، $f \in R^l$ ، $E \in R^{l \times n}$ ، $b \in R^m$ ، $A \in R^{m \times n}$ ، $Q \in R^{n \times n}$ می باشند.

دوگان برنامه ریزی محدب درجه دوم (DCQP) از (CQP) (۱-۴) به فرم زیر نوشته می شود:

$$\begin{cases} \max & \bar{f}(x, u, v) = \frac{1}{2} x^T Q x + D^T x + u^T (Ax - b) + v^T (Ex - f) \\ \text{s.t} & \\ & \bar{g}(x, u, v) = Qx + D + A^T u + E^T v = 0, \\ & u \geq 0. \end{cases} \quad (۲-۴)$$

بر طبق شرایط کاروش - کان - تاکر (KKT)، $x^* \in R^n$ جواب بهینه برای (۱-۴) می باشد اگر و تنها اگر

بردارهای $u^* \in R^m$ و $v^* \in R^l$ وجود داشته باشد به طوری که $(x^{*T}, u^{*T}, v^{*T})^T$ در رابطه زیر صدق

نماید:

$$\begin{cases} u^* \geq 0, Ax^* - b \leq 0, u^{*T} (Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (۳-۴)$$

۳-۴ شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم محدب

هدف ما در این قسمت طراحی یک مدل شبکه عصبی است که به یک نقطه تعادل برسد و این نقطه جواب بهینه مسأله CQP (۱-۴) نیز باشد. با استفاده از تکنیک تابع تصویر رابطه بین جواب های (۱-۴)، (۲-۴) و یک دستگاه از معادلات تصویر را اثبات می کنیم. برای این منظور به بیان یک رابطه تعادلی بین شرایط بهینگی (۳-۴) و ساختار یک شبکه عصبی می پردازیم.

گزاره ۱-۴: $(x^{*T}, u^{*T}, v^{*T})^T$ جواب یکتا برای CQP (۱-۴) و DCQP (۲-۴) می باشد، اگر

و تنها اگر $(x^{*T}, u^{*T}, v^{*T})^T$ در معادلات تصویر زیر صدق نماید:

$$\begin{cases} (\alpha - (Qx^* + D + A^T u^* + E^T v^*))^+ - \alpha = 0, \\ (u^* + Ax^* - b)^+ - u^* = 0, \\ (\beta + Ex^* - f)^+ - \beta = 0, \end{cases} \quad (۴-۴)$$

که در آن بردارهای $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T > 0$ و $\beta = (\beta_1, \beta_2, \dots, \beta_l)^T > 0$ معلوم می باشند.

اثبات: واضح است که $(u^* + Ax^* - b)^+ - u^* = 0$ اگر و تنها اگر

$$u^* \geq 0, \quad Ax^* - b \leq 0, \quad u^{*T}(Ax^* - b) = 0.$$

از طرف دیگر $(\alpha - (Qx^* + D + A^T u^* + E^T v^*))^+ - \alpha = 0$ اگر و تنها اگر

$$(Qx^* + D + A^T u^* + E^T v^*) = 0.$$

به طور مشابه $(\beta + Ex^* - f)^+ - \beta = 0$ اگر و تنها اگر $Ex^* - f = 0$ باشد. بنابراین

$(x^{*T}, u^{*T}, v^{*T})^T$ یک نقطه KKT است و اثبات کامل می شود.

برای سادگی تعریف می کنیم:

$$\begin{aligned} y &= (x^T, u^T, v^T)^T, \quad z = (\alpha^T, u^T, \beta^T)^T, \\ U(y) &= \begin{bmatrix} (Qx + D + A^T u + E^T v) \\ -(Ax - b) \\ -(Ex - f) \end{bmatrix}. \end{aligned} \quad (۵-۴)$$

بنابراین حل CQP (۱-۴) و DCQP (۲-۴) معادل است با پیدا کردن یک جواب از معادلات تصویر

زیر:

$$\phi(y) = (z - (U(y))^+ - z = 0, \quad (۴-۶)$$

که در آن U در (۴-۵) تعریف شده است.

اکنون تعریف می کنیم:

$$E(y) = \frac{1}{2} \|\phi(y)\|^2. \quad (۴-۷)$$

واضح است که $E(y) \geq 0$. بنابراین حل دستگاه (۴-۶) معادل است با پیدا کردن یک مینیمم

سراسری از مسأله بهینه سازی زیر:

$$\min E(y)$$

$$s.t$$

$$y \in R^{m+n+l}.$$

فرض کنید $x(\cdot)$ ، $u(\cdot)$ و $v(\cdot)$ متغیرهای وابسته به زمان باشند. مدل شبکه عصبی متناظر با دستگاه

(۴-۶) توسط سیستم دینامیکی غیر خطی زیر بیان می شود:

$$\frac{dy}{dt} = \frac{d}{dt} \begin{bmatrix} x \\ u \\ v \end{bmatrix} = -\kappa \nabla E(y(t)), \quad (۴-۸)$$

که در آن κ پارامتر اسکالر می باشد و نرخ همگرایی شبکه عصبی (۴-۸) با نقطه آغازین

$$y_0 = (x_0^T, u_0^T, v_0^T)^T$$

را نشان می دهد.

۴-۳-۱ تحلیل پایداری و همگرایی

در این بخش پایداری نقطه تعادل و همگرایی جواب بهینه مدل (۴-۸) بررسی می شود.

لم (۴-۲): برای هر نقطه آغازین $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ ، سیستم (۴-۸) دارای

جواب یکتای پیوسته $y(t) = (x(t)^T, u(t)^T, v(t)^T)^T$ می باشد.

برهان: واضح است که توابع $(\alpha - (Qx^* + D + A^T u^* + E^T v^*))^+ - \alpha = 0$ و $(u^* + Ax^* - b)^+ - u^* = 0$ و $(\beta + Ex^* - f)^+ - \beta = 0$ پیوسته و لیب شیتز محلی هستند. با توجه به قضیه وجود جواب موضعی برای معادلات دیفرانسیل معمولی شبکه عصبی (۸-۴) دارای جواب یکتای $y(t)$ به ازای $t \in [t_0, \tau]$ ، $\tau > t_0$ است وقتی که $\tau \rightarrow \infty$.

گزاره (۳-۴): اگر $y^* = (x^{*T}, u^{*T}, v^{*T}) \in R^{n+m+l}$ یک جواب بهینه برای CQP (۱-۴) و DCQP (۲-۴) باشد آنگاه y^* نقطه تعادل برای مدل شبکه عصبی (۸-۴) است.

برهان: فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ یک جواب بهینه برای CQP (۱-۴) و DCQP (۲-۴) باشد، بنابراین $\phi(y^*) = 0$. از طرفی $\nabla E(y(t)) = J_\phi \phi(y)$ که در آن J_ϕ ماتریس ژاکوبین ϕ است، پس $\nabla E(y^*) = 0$. بنابراین $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل (۸-۴) است.

قضیه (۴-۴): فرض کنید y^* یک نقطه تعادل از شبکه عصبی (۸-۴) باشد. آنگاه شبکه عصبی (۸-۴) پایدار مجانبی به y^* می باشد.

برهان: با در نظر گرفتن تابع $E(y)$ در (۷-۴) به عنوان تابع انرژی داریم:

$$\frac{dE}{dt} = \nabla E \cdot \frac{dy}{dt} = -\kappa \nabla E(y) \cdot \nabla E(y) = -\kappa \|\nabla E(y)\|^2 \leq 0.$$

بنابراین سیستم (۸-۴) پایدار مجانبی به y^* است و اثبات کامل می شود.

۴-۴ نتایج عددی

در این بخش برای نشان دادن کارایی مدل (۴-۸) در حل مسائل برنامه ریزی درجه دوم محدب به ذکر پنج مثال می پردازیم. در شبیه سازی همه مثال ها از دستور ode45s در Matlab استفاده شده است.

مثال ۴-۱: مسأله برنامه ریزی درجه دوم محدب زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_3^2 - 2x_1 - 3x_2 \\ \text{s.t} \quad & 2x_1 + x_2 + x_3 + x_4 = 7, \\ & x_1 + x_2 + 2x_3 + x_4 = 6, \\ & x_i \geq 0, \quad i = 1, \dots, 4. \end{aligned}$$

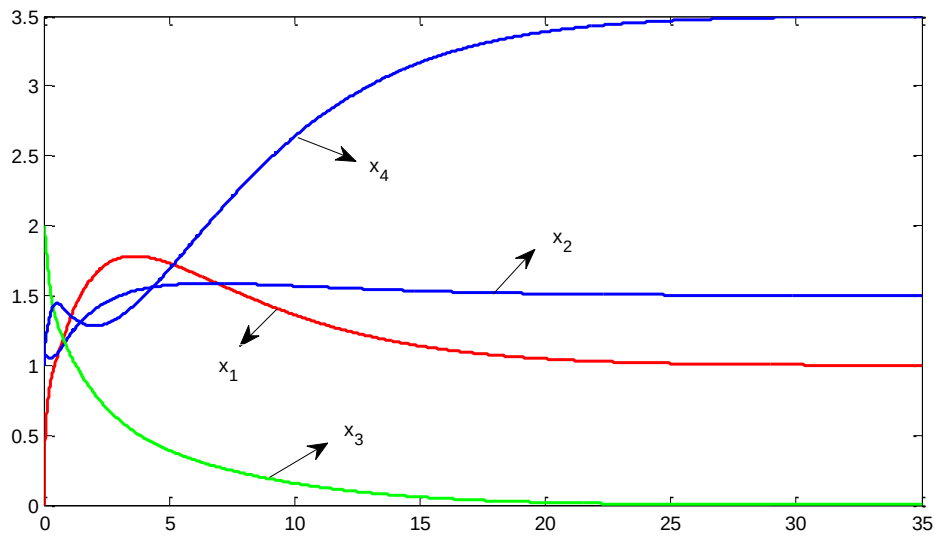
که در آن

$$\phi(y) = \begin{bmatrix} (3 - 2x_1 + u_1 - 2v_1 - v_2 + 2)^+ - 3 \\ (3 - 2x_2 + u_2 - v_1 - v_2 + 3)^+ - 3 \\ (3 - 2x_3 + u_3 - v_1 - 2v_2)^+ - 3 \\ (3 + u_4 - v_1 - v_2)^+ - 3 \\ (u_1 - x_1)^+ - u_1 \\ (u_2 - x_2)^+ - u_2 \\ (u_3 - x_3)^+ - u_3 \\ (u_4 - x_4)^+ - u_4 \\ (3 + 2x_1 + x_2 + x_3 + x_4 - 7)^+ - 3 \\ (3 + x_1 + x_2 + 2x_3 + x_4 - 6)^+ - 3 \end{bmatrix}$$

جواب بهینه این مسأله و دوگان آن با استفاده از شبکه عصبی (۴-۸) عبارت است از:

$y^* = (1, 1.5, 0, 3.5, 0, 0, 0, 0)^T$. نتایج شبیه سازی شبکه عصبی (۴-۸) با $\alpha = (3, 3, 3, 3)^T$ و

$\beta = (3, 3)^T$ و با نرخ همگرایی $\kappa = 1$ در شکل (۴-۱) نشان داده شده است.



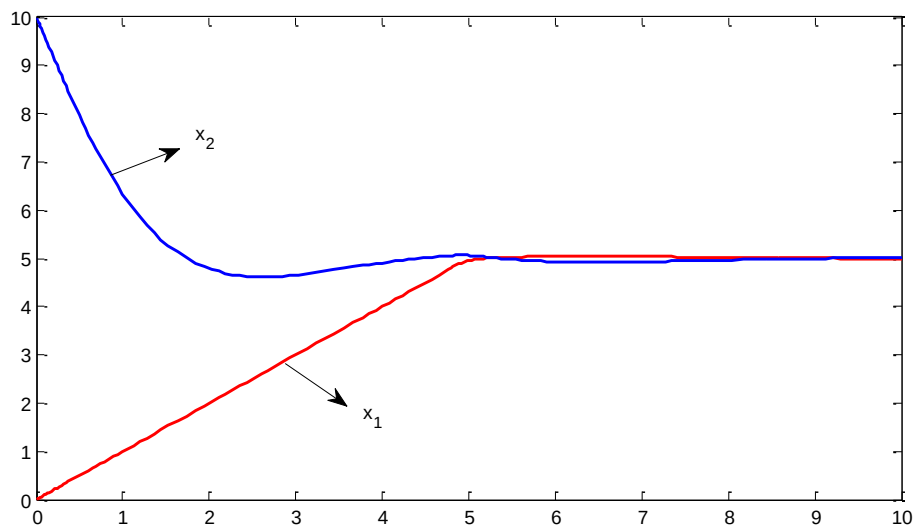
شکل (۴-۱): رفتار گذرا شبکه عصبی (۴-۸) با شش نقطه آغازین دلخواه.

مثال ۴-۲: مسأله برنامه ریزی خطی زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.t} \quad & \frac{5}{12}x_1 - x_2 - 1 \leq \frac{35}{12}, \\ & \frac{5}{2}x_1 + x_2 \leq \frac{35}{2}, \\ & -x_1 \leq 5, \\ & x_2 \leq 5. \end{aligned}$$

جواب بهینه مسأله که با استفاده از مدل (۴-۸) محاسبه شده است و پایدار مجانبی سراسری به

$x^* = (5, 5)^T$ می باشد. رفتار این مدل در شکل (۴-۲) با $(\alpha_1, \alpha_2)^T = (4, 5)^T$ نشان داده شده اند.



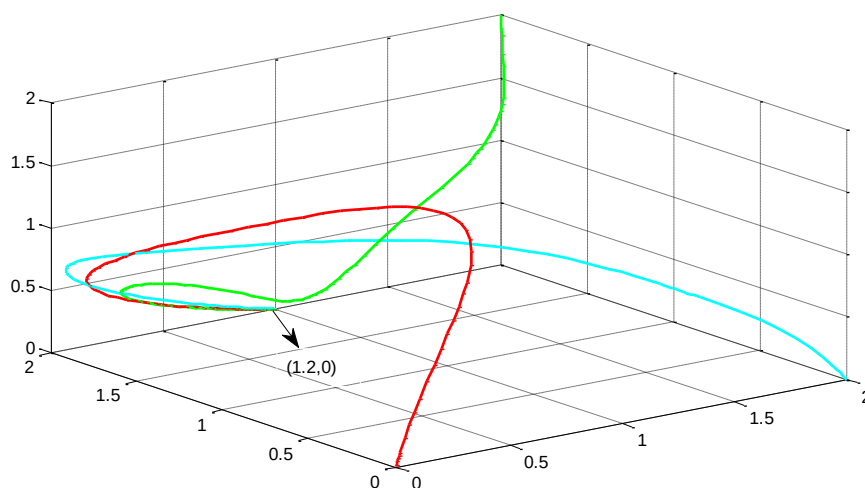
شکل (۲-۴): رفتار همگرا از شبکه عصبی (۸-۴) برای مثال (۲-۴).

مثال ۳-۴: مسأله بهینه سازی محدب زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_3^2 \\ \text{s.t} \quad & 2x_1 + x_2 - 5 \leq 0, \\ & x_1 + x_3 - 2 \leq 0, \\ & -x_1 + 1 \leq 0, \\ & -x_2 + 2 \leq 0, \\ & -x_3 \leq 0. \end{aligned}$$

جواب بهینه این مسأله با استفاده از مدل (۸-۴) به $x^* = (1, 2, 0)^T$ همگرا می باشد. شبیه سازی

عددی با $\kappa = 4$ و $\alpha = (3, 4, 5)^T$ در شکل (۳-۴) با سه نقطه آغازین متفاوت نشان داده شده است.



شکل (۳-۴): همگرایی سراسری مثال (۳-۴) به $x^* = (1, 2, 0)^T$ توسط شبکه عصبی (۸-۴).

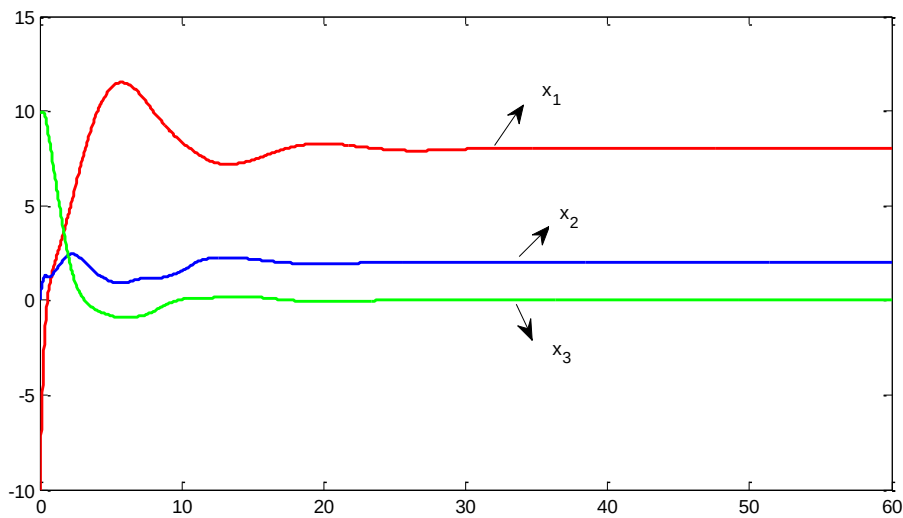
مثال ۴-۴: مسأله برنامه ریزی ریاضی زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & -4x_1 - 5x_2 + 3x_3 \\ \text{s.t} \quad & -x_1 + x_2 \leq -1, \\ & x_1 + 3x_2 + x_3 \leq 14, \\ & x_1 + x_2 + x_3 = 10, \\ & x_i \geq 0 \quad i=1, 2, 3. \end{aligned}$$

این یک مسأله برنامه ریزی خطی است. شبیه سازی عددی این مسأله با مدل (۸-۴) حاکی از آن

است که مسیر جواب $(x_1(t), x_2(t), x_3(t))^T$ به جواب یکتای $x^* = (8, 2, 0)^T$ همگرا می باشد. در این

مدل $\alpha = (3, 3, 3, 3, 3)^T$, $\kappa = 1$ و $\beta = 3$ انتخاب شده اند (شکل (۴-۴)).



شکل (۴-۴): رفتار پایدار مجانبی مثال (۴-۴) به $x^* = (8, 2, 0)^T$ توسط شبکه عصبی (۸-۴).

مثال ۴-۵: مسأله ریزی درجه دوم زیر را در نظر بگیرید:

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

s.t

$$\frac{5}{12}x_1 - x_2 < \frac{35}{12},$$

$$\frac{5}{2}x_1 + x_2 < \frac{35}{2},$$

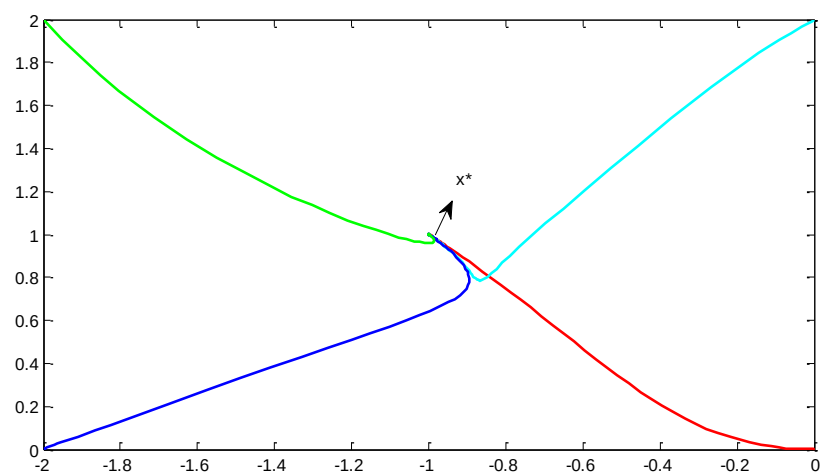
$$-x_1 < 5,$$

$$x_2 < 5.$$

شبیه سازی عددی این مسأله با مدل (۸-۴) نشان می دهد که تمام مسیر ها همگرا به جواب

یکتای $x^* = (-1, 1)^T$ که در آن $\kappa = 1$ و $\alpha = (3, 3)^T$ انتخاب شده اند. شکل (۵-۴) همگرایی مسیر

های مدل شبکه عصبی (۸-۴) را نشان می دهد.



شکل (۵-۴): همگرایی سراسری مثال (۵-۴)

فصل پنجم

بررسی پایداری مجانبی یک مدل شبکه عصبی برای حل مسائل برنامه ریزی

درجه دوم

۵-۱ مقدمه

یک روش کلی برای حل مسائل برنامه ریزی مقید، تبدیل آن به یک یا چند مسأله برنامه ریزی نامقید می باشد، که توسط تکنیک های عددی حل می شود. روش های جریمه و لاگرانژ روش های معمولی هستند که غالباً برای این تبدیل، مورد استفاده قرار می گیرند. قبلاً تبدیل مسائل مقید به مسائل نامقید با اضافه کردن یک جمله به تابع هدف انجام می گرفت که هزینه بالایی برای کنترل خطای محدودیت ها ایجاد می کرد. البته به شرط اینکه اطلاعات اولیه مناسب راجع به پارامتر پنالتی و نقطه شروع در دسترس باشد، نقطه مینیمم تابع پنالتی با جواب بهینه مسأله اولیه یکسان خواهد بود. در غیاب اطلاعات اولیه مناسب تقریبی از جواب بهینه به دست می آید، برای پیدا کردن جواب تقریبی نیاز به استفاده از پارامتر پنالتی و نقطه شروع مناسب می باشد، که مجبور به استفاده از قانون سعی و خطا می شویم. بعلاوه همیشه خطایی از افزودن نادرست قیدها به تابع هدف وجود دارد که برای تابع پنالتی دردسر ساز می باشد. به عنوان مثال می توان میل کردن پارامتر پنالتی به بی نهایت را ذکر کرد. از طرف دیگر اگر بعد فضای جواب افزایش یابد، هر چند ضرایبی که با محدودیت های نامساوی در تابع لاگرانژ مربوط می شوند، غیر منفی اند، حل مستقیم مسأله بهینه سازی آسان نمی باشد. با اصلاح ضرایب لاگرانژ مربوط به محدودیت های نامساوی، محدودیت های غیر منفی می توانند حذف شوند. از این رو محدودیت های نامساوی به طور مستقیم جا به جا می شوند، و برای تبدیل دیگر نیازی به متغیر های کمکی نداریم. بنابراین محاسبات مشکل و پیچیده می توانند کاهش یابند.

هدف استفاده از شبکه های عصبی برای حل مسائل بهینه سازی رفع این مشکلات است. تمام ساختارهای بیان شده شبکه عصبی برای حل مسائل بهینه سازی را می توان به دو گروه تقسیم کرد: ۱- توابعی که شامل یک نگاشت از محدودیت های نامساوی به مساوی می باشند و نیز توابع جریمه ای که مبنایی برای جریمه کردن خطای محدودیت های نامساوی می باشند. ۲- روش

الحاقی^{۸۱}، که این روش تبدیل محدودیت های نامساوی به مساوی را با اضافه کردن متغیر های کمکی یا مازاد انجام می دهد. هر دو روش به طور غیر مستقیم بر روی محدودیت های نامساوی بحث می کنند. مدل های شبکه های عصبی بر اساس یکی از موارد فوق فرمول بندی می شوند. در این فصل به بیان یک مدل شبکه عصبی برای بررسی مستقیم محدودیت های نامساوی در مسائل برنامه ریزی درجه دوم پرداخته و شرایط پایداری برای این مدل را بررسی می کنیم.

۵-۲ مدل شبکه عصبی برای حل مسائل برنامه ریزی درجه دوم

برنامه ریزی محدب درجه دوم (CQP) زیر را در نظر بگیرید:

$$\begin{cases} \min & \frac{1}{2} x^T Q x + D^T x \\ \text{s.t} & \\ & Ax - b \leq 0, \\ & Ex - f = 0, \end{cases} \quad (۱-۵)$$

که در آن $Q \in R^{n \times n}$ و $x \in R^n, A \in R^{m \times n}, b \in R^m, E \in R^{l \times n}, f \in R^{l \times 1}$ یک ماتریس معین مثبت می باشد. دو گان (۱-۵) به صورت زیر بیان می شود:

$$\begin{cases} \max & \bar{f}(x, u, v) = \frac{1}{2} x^T Q x + D^T x + u^T (Ax - b) + v^T (Ex - f) \\ \text{s.t} & \\ & \bar{g}(x, u, v) = Qx + D + A^T u + E^T v = 0, \\ & u \geq 0, \quad u = (u_1, \dots, u_m), \quad v = (v_1, \dots, v_l). \end{cases} \quad (۲-۵)$$

تعریف ۵-۱: $(x^{*T}, u^{*T}, v^{*T})^T$ نقطه KKT مسأله بهینه سازی (۱-۵) و (۲-۵) است اگر و

فقط اگر $(x^{*T}, u^{*T}, v^{*T})^T$ در شرایط زیر صدق کند:

$$\begin{cases} u^* \geq 0, \quad Ax^* - b \leq 0, \quad u^{*T} (Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (۳-۵)$$

⁸¹ State augmentation

هم اکنون فرض کنید که $x(\cdot)$ ، $u(\cdot)$ و $v(\cdot)$ متغیرهای وابسته به زمان باشند. هدف بیان یک ساختار دینامیکی زمان پیوسته است که به نقطه KKT مسائل (۱-۵) و (۲-۵) هدایت شود. مدل شبکه عصبی متناظر با (۱-۵) و (۲-۵) را توسط سیستم دینامیکی غیرخطی با نقطه آغازین $(x_0^T, u_0^T, v_0^T)^T$ و $u_k(0) \neq 0$ به صورت زیر بیان می کنیم:

$$\begin{cases} \frac{dx}{dt} = -(Qx + D + \frac{1}{2} A^T u^2 + E^T v), \\ \frac{du_k}{dt} = u_k(a_k x - b_k), \quad k=1, \dots, m, \\ \frac{dv}{dt} = Ex - f. \end{cases} \quad (۴-۵)$$

برای سادگی تعریف می کنیم:

$$y = (x^T, u^T, v^T) \in R^{n+m+l},$$

$$U(x, u, v) = \begin{bmatrix} -(Qx + D + \frac{1}{2} A^T u^2 + E^T v) \\ u_1(a_1 x - b_1) \\ \vdots \\ u_m(a_m x - b_m) \\ (Ex - f) \end{bmatrix}. \quad (۵-۵)$$

آنگاه شبکه عصبی (۴-۵) را می توان به صورت زیر نوشت:

$$\begin{cases} \frac{dy}{dt} = U(y), \\ y(t_0) = y_0, \quad u_0 \neq 0. \end{cases} \quad (۶-۵)$$

۳-۵ تحلیل همگرایی و پایداری مدل شبکه عصبی

در این قسمت به بررسی پایداری شبکه عصبی (۶-۵) خواهیم پرداخت.

قضیه ۲-۵: فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل برای شبکه عصبی (۶-۵) باشد.

آنگاه y^* نقطه KKT مسأله (۱-۵) می باشد.

برهان: واضح است که در شبکه عصبی (۵-۶) از مربع ضرایب u استفاده شده است که منجر به حذف محدودیت های غیر منفی بیان شده ضرایب u در شرایط KKT (۵-۳) می شود. با مقایسه سیستم KKT (۵-۳) با معادله دینامیکی عصبی (۵-۶) تنها اثبات $(Ax-b) \leq 0$ و $u(Ax-b)=0$ باقی می ماند.

واضح است که در دستگاه دینامیکی (۵-۴) u را می توانیم به صورت زیر بنویسیم:

$$u_k(t) = u_k(0)e^{\int_{t_0}^t (a_k x - b_k) dt}, \quad k = 1, \dots, m,$$

که در آن $u_k(0) \neq 0$ مقدار آغازین غیر صفر از $u(t)$ می باشد. این عبارت نشان می دهد که اگر حداقل یک k موجود باشد به طوری که $a_k x - b_k > 0$ ، در این صورت $u(t)$ به صورت نمایی با زمان افزایش می یابد، به طوری که بعد از زمان متناهی مسیر $x(t)$ به سمت ناحیه شدنی هدایت می شود [۲۶]. این فرایند پیوسته تکرار می شود مگر این که تمام محدودیت ها برقرار باشند، در این حالت $(x^T, u^T, v^T)^T$ سر انجام به نقطه KKT $(x^{*T}, u^{*T}, v^{*T})^T$ مسأله میل خواهد کرد، به طوری که ضرایب $u_k(t)$ ، $k=1, \dots, m$ از محدودیت های غیر فعال (یعنی $a_k x - b_k < 0$) به صفر میل می کنند و ضرایب باقیمانده متناظر با قیود فعال به $u^* = u_0$ میل می کنند. بنابراین $(Ax^* - b) \leq 0$ و $u^*(Ax^* - b) = 0$ این اثبات را کامل می کند.

لم ۵-۳: برای هر نقطه آغازین $y_0 = (x_0^T, u_0^T, v_0^T)^T$ یک جواب پیوسته

$$y(t_0) = (x(t)^T, u(t)^T, v(t)^T)^T \text{ برای سیستم (۵-۶) وجود دارد.}$$

برهان: چون $Qx + D$ مشتق پذیر پیوسته روی هر مجموعه محدب باز $P \subseteq R^n$ می باشد، آنگاه

$$Qx + D + \frac{1}{2} A^T u^2 + E^T v \text{ و } (Ex - f) \text{ لیپ شیتز محلی می باشند. با توجه به قضیه}$$

وجود جواب موضعی برای معادلات دیفرانسیل معمولی، شبکه عصبی (۵-۶) دارای جواب پیوسته

یکتای $y(t)$ ، برای $t \in [t_0, \tau]$ و $\tau > t_0$ وقتی $\tau \rightarrow \infty$ می باشد.

قضیه ۴-۵: اگر ماتریس Q معین مثبت باشد آنگاه شبکه عصبی بیان شده در (۵-۶) پایدار

لیاپانوف می باشد.

برهان: ابتدا تابع لیاپانوف زیر را معرفی می کنیم:

$$E(y) = \|U(y)\|^2 + \frac{1}{2} \|y - y^*\|^2. \quad (۷-۵)$$

واضح است که $E(y)$ تابع غیر منفی روی R^{n+m+l} و $E(y^*) = 0$ می باشد، آنگاه:

$$\frac{dU}{dt} = \frac{\partial U}{\partial y} \frac{dy}{dt} = \nabla U(y) U(y)$$

و ماتریس ژاکوبین نگاشت U به صورت زیر به دست می آید:

$$\nabla U(y) = \begin{bmatrix} -Q & -A^T S & -E^T \\ S^T A & H & 0 \\ E & 0 & 0 \end{bmatrix}, \quad (۸-۵)$$

و ماتریس های S و H به صورت زیر تعریف می شوند:

$$H = \begin{bmatrix} a_1 x - b_1 & 0 & \dots & 0 \\ 0 & a_2 x - b_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & a_m x - b_m \end{bmatrix} \text{ و } S = \begin{bmatrix} u_1 & 0 & \dots & 0 \\ 0 & u_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & u_m \end{bmatrix}$$

با محاسبه مشتق $E(y(t))$ در طول مسیر سیستم (۵-۶) و بر طبق فرض قضیه داریم:

$$\begin{aligned} \frac{dE(y(t))}{dt} &= \left(\frac{dU}{dt}\right)^T U + U^T \left(\frac{dU}{dt}\right) + (y - y^*)^T \frac{dy(t)}{dt} \\ &= (\nabla U(y) U)^T U + U^T (\nabla U(y) U) + (y - y^*)^T \frac{dy(t)}{dt} \\ &= (U^T \nabla U^T(y)) U + U^T (\nabla U(y) U) + (y - y^*)^T U(y) \\ &= U^T (\nabla^T U(y) + \nabla U(y)) U + (y - y^*)^T U(y) \\ &= -2(Qx + D + \frac{1}{2} A^T u^2 + E^T v)^T Q (Qx + D + \frac{1}{2} A^T u^2 + E^T v) \\ &\quad + 2 \left(\sum_{k=1}^m (u_k (a_k x - b_k))^2 (a_k x - b_k) \right) + (y - y^*)^T U(y) \\ &\leq 2 \left(\sum_{k=1}^m (u_k (a_k x - b_k))^2 (a_k x - b_k) \right) + (y - y^*)^T U(y). \end{aligned}$$

اگر $a_k x - b_k \leq 0$, $k=1, \dots, m$ ، آنگاه $\sum_{k=1}^m (u_k (a_k x - b_k))^2 (a_k x - b_k) \leq 0$ بعلاوه در این حالت

$\nabla U(y)$ معین منفی است و بنا بر قضیه [۲۰-۲] $-U(y)$ یکنوا می باشد، یعنی:

$$(y - y^*)^T (U(y) - U(y^*)) = (y - y^*)^T U(y) \leq 0,$$

بنابراین $\frac{dE(y(t))}{dt} \leq 0$. از طرف دیگر اگر $a_k x - b_k > 0$ برای $k \in \{1, \dots, m\}$ ، بنا بر قضیه (۲-۵)

بعد از زمان متناهی $x(t)$ به ناحیه شدنی میل می کند (یعنی $T_1 < \infty$ وجود دارد به طوری که

$a_k x(t) - b_k \leq 0$ و $t \geq T_1$) آن چنان که ضرایب u_k نظیر تا زمانی که $u_k \geq 0$ شود، افزایش خواهند

یافت. بنابراین بعد از یک زمان متناهی و مشابه حالت قبلی $\frac{dE(y(t))}{dt} \leq 0$. بنابراین مشابه حالت

قبلی سیستم (۶-۵) پایدار مجانبی به y^* است.

۴-۵ نتایج عددی

در این بخش دو مثال برای نشان دادن کارایی مدل (۶-۵) برای حل مسائل درجه دوم محدب اکید

آورده شده است. در شبیه سازی مثال ها از دستور ode45s در Matlab استفاده شده است.

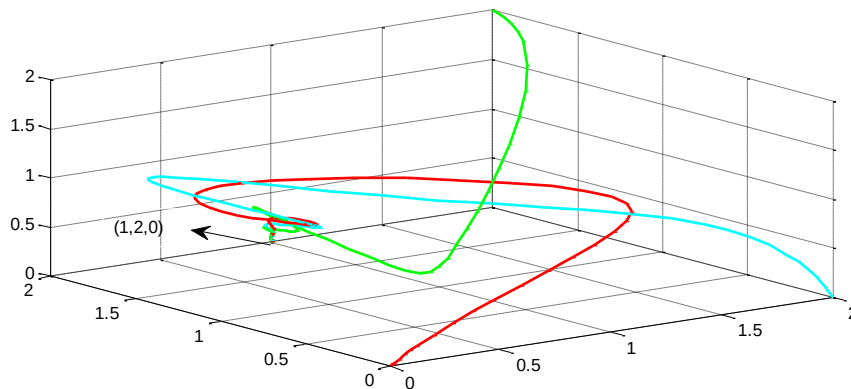
مثال ۵-۱: مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_3^3 \\ \text{s.t} \quad & 2x_1 + x_2 \leq 5, \\ & x_1 + x_3 \leq 2, \\ & x_1 \geq 1, x_2 \geq 2, x_3 \geq 0. \end{aligned}$$

جواب بهینه این مسأله با استفاده از شبکه عصبی (۶-۵) عبارت است از $y^* = (1, 2, 0)^T$ که در آن

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 2 \\ -1 \\ -2 \\ 0 \end{bmatrix}.$$

نتایج شبیه سازی شبکه عصبی (۶-۵) در شکل (۱-۴) نشان داده شده است. ملاحظه می کنیم که با شروع از سه نقطه آغازین متفاوت همه مسیرها به نقطه $y^* = (1, 2, 0)^T$ همگرا می باشند. این نشان می دهد که مدل شبکه عصبی استفاده شده برای این مثال همگرایی سراسری به y^* می باشد.



شکل (۱-۵): همگرایی سراسری مثال (۱-۵) به $y^* = (1, 2, 0)^T$ توسط مدل (۶-۵)

مثال ۲-۵: مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_1 x_2 + x_1 - x_2 \\ \text{s.t} \quad & \end{aligned}$$

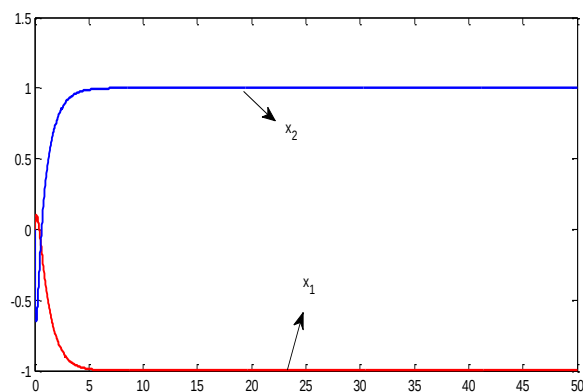
$$\begin{aligned} & \frac{5}{12}x_1 - x_2 < \frac{35}{12}, \\ & \frac{5}{2}x_1 + x_2 < \frac{35}{2}, \\ & -x_1 < 5, \\ & x_2 < 5. \end{aligned}$$

با توجه به مسأله داریم:

$$Q = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad A = \begin{bmatrix} \frac{5}{12} & -1 \\ \frac{5}{2} & 1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} \frac{35}{12} \\ \frac{35}{2} \\ 5 \\ 5 \end{bmatrix}.$$

شبیه سازی عددی این مسأله با مدل (۵-۶) نشان می دهد که تمام مسیر ها همگرا به جواب یکتای

$(x^*) = (-1, 1)$ می باشد (شکل (۵-۲)).



شکل (۵-۲): رفتار پایدار مجانبی مثال (۵-۲) توسط شبکه عصبی (۵-۶)

نتیجه گیری و پیشنهادات

در این پایان نامه در ابتدا به معرفی شبکه های عصبی و کاربرد آنها در علوم مختلف خصوصاً در بهینه سازی پرداخته شد. در فصل دوم تعاریف و قضایای مهم که در حل مسائل بهینه سازی در فصول بعد مورد استفاده قرار می گیرد بیان شد. در فصل سوم به بیان برخی مدل های ارائه شده پیشین برای حل مسائل برنامه ریزی خطی و غیر خطی پرداختیم. در دو فصل انتهایی دو مدل شبکه عصبی برای حل مسائل برنامه ریزی خطی و درجه دوم پیشنهاد کرده، پایداری و همگرایی سراسری آنها مورد بررسی قرار گرفت. با استفاده از چندین مثال عددی نشان داده شد که مدل های معرفی شده از کارایی بالایی در حل مسائل بهینه سازی خطی و درجه دوم برخوردار هستند.

از جمله تحقیقات دیگری که در ادامه می توان انجام داد عبارتند از:

۱. مدل های مختلف دیگری می توان برای حل مسائل برنامه ریزی درجه دوم و خطی ساخت.
۲. در تحلیل پایداری می توان از توابع انرژی دیگر نیز استفاده نمود.
۳. سرعت همگرایی مدل های بیان شده را می توان با تغییر پارامتر بیان کننده نرخ همگرایی بررسی کرد.
۴. می توان رفتار تابع انرژی و شکاف دوگانی را برحسب زمان برای همه مثال های یاد شده رسم کرد.
۵. می توان رفتار گذرا برای همه مدلهای بیان شده را برای حالت های مختلف با نقطه شروع از داخل و خارج ناحیه شدنی بررسی نمود.

منابع و مراجع

- [1] Haykin S. (1999) "*Neural Network: A Comprehensive Foundation*" prentice hall, pp 842.
- [2] Wu X., Zhou Y.(1993) "Reserve estimation using neural network technique", *Computer & Geosciences*, 19, 4, 567-575
- [3] Hudson D.L., Cohen M.E. (1999) "*Neural Network and Artificial Intelligence for Biomedical Engineering*" IEEE Press Series on Biomedical Engineering.
- [4] Berry M.J.A., Linoff G. (1997) "*Data Mining Techniques*" John Wiley & sons, pp 454.
- [5] Dowd P.A, Sarac C. (1994) "A neural network approach to geostatistical simulation" *Mathematical Geology*, 26, 4, pp 491-503.
- [6] Shelly B., Stephenson S. (2000), "The use of artificial neural network in completion simulation and design" *Computer & Geosciences*, 26, pp 941-951
- [7] Altrick C.V. (1995) "*Fuzzy Logic & Neurofuzzy Applications Explained*" prentice Hall, pp. 384.
- [8] Hopfield J.J, Tank D.W. (1986) "Simple neural optimization network: An A/ D converter, signal decision circuit, and a linear programming pircuit" *IEEE Trans on Circuits and Systems* 33, pp 533-541.
- [9] Kennedy M. P. and Chua L. O. (1988) "Neural networks for nonlinear programming" *IEEE Trans. on Neural Networks*, 35, pp 554-562.
- [10] Bouzerdoum A. and pattison T. R. (1993) "Neural network for quadratic optimization with bound constraints" *IEEE Trans. On Neural Networks*, 4, pp 293-304.
- [11] Xia Y. (1996) "A neural network for solving linear and quadratic programming problems" *IEEE Trans. On Neural Networks*, 7, pp 1544-1547.

- [12] Xia Y. and Feng G. (2005) "An improved network for convex quadratic optimization with application to real-time beam forming" *Neurocomputing*, 64, pp 359-374.
- [13] Xia Y. Feng G. and Wang J. (2004) "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equation" *Neural Networks*, 17, pp 1003-1015.
- [14] Xia Y. and Wang J. (2004) "A general projection neural network for solving monotone variational inequality and related optimization problems" *IEEE Trans. On Neural Networks*", 15, pp 318-328.
- [15] Xia Y., Leung H. and Wang J. (2002) "A projection neural network and its application to constrained optimization problems" *IEEE Transactions On Circuits And Systems*, 44, 4, pp 447-458.
- [16] Xia Y. and Wang J. (2000) "A recurrent neural network for solving linear projection equation" *Neural Networks*, 13, pp 337-350.
- [17] Bazarra M., Sherali H. D. and Shetty C. M. (1993) "*Non-linear Programing Theory and Algorithms*" John Wiley & Sons, N. Y.
- [18] Xing-Bao G., (2003) "Exponential Stability of Globally Projected Dynamic Systems" *IEEE Trans. On Neural Networks*, 14, 2, PP 426-431
- [19] Effati S., Nazemi A. R. (2006) "Neural network models and its application for solving linear and quadratic programming problems" *Appl. Math. Comput*, 172, pp 305-331
- [20] Friesz T. L., Bernstein D. H., Mehta N. J., Tobin R. L. and Ganjlizadeh S. (1994) "Day-to-day dynamic network disequilibria and idealized traveler information systems" *Oper . Res* 42, 6, pp 1120-1136.
- [21] Naguroney A. (2002) "*Variational Inequality*" Isenberg School of Management University of Massachusetts.

- [22] Wu X., Xia Y., Li J. and Chen W.(1996) "A High Performance Neural Network for solving Linear and Quadratic Programming Problems" *IEEE Trans. On Neural Networks*, 7, 3, 643-651.
- [23] Rodriguez-Vazquez A., Dommnguer-Castro R., Rueda A., Huertas J. L. and Sahnchez-Sinencio E. (1990) "Nonlinear switched-capacitor neural network for optimization problems " *IEEE Trans on Circuits and Sistems*", pp 384-390.
- [24] Qi L., Sun J. (1993) "A nonsmooth version of Newton method," *Math Program*" 58, pp 353-368.
- [25] Effati S., Ghomashi A. Nazemi A. R. (2007) "Application of projection neural network in solving convex programming problems" *Appl. Math. Comput*, 188, pp 1103-1114.
- [26] Huang Y. (2002) "A novel method to handle inequality constraint for convex programming neural network" *Neural Processing Letters*, 16, pp 17-27.

[۲۷] منهاج م، "مبانی شبکه های (هوش محاسباتی)" (۱۳۷۷)، جلد اول، انتشارات حسابی.

[۲۸] توتونیان ف، "برنامه ریزی خطی (روش ها و کاربردها) " (۱۳۷۸)، نوشته سل آی گس،

انتشارات دانشگاه فردوسی مشهد.

[۲۹] بابایی ع و میامئی ا، (۱۳۶۵)، "معادلات دیفرانسیل و کاربردهای آن " نوشته جرج سیمونز،

انتشارات مرکز دانشگاهی.

/۳۰/ آذریک س، ناظمی ع، شاهسونی د، (۱۳۸۹)، "یک کاربرد از شبکه های عصبی برای حل

مسائل برنامه ریزی خطی و درجه دوم"، سومین کنفرانس بین المللی انجمن ایرانی تحقیق در

عملیات، دانشگاه صنعتی امیرکبیر.