



دانشکده علوم ریاضی

گروه ریاضی کاربردی

پایان نامه

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

عنوان

یک کاربرد از مدل‌های شبکه عصبی برای حل مساله تخصیص

استاد راهنما

آقای دکتر علیرضا ناظمی

پژوهشگر

عذرا قزلسلی

آبان ۱۳۹۲

نام خانوادگی دانشجو: قزلسفلی

نام: عذرا

عنوان: یک کاربرد از مدل‌های شبکه عصبی برای حل مساله تخصیص

استاد راهنما: آقای دکتر علیرضا ناظمی

مقطع تحصیلی: کارشناسی ارشد رشته: ریاضی کاربردی گرایش: تحقیق در عملیات

دانشگاه: شاهرود

تعداد صفحات: ۷۶

تاریخ فارغ‌التحصیلی: آبان ۱۳۹۲

واژگان کلیدی: مساله تخصیص، کوتاهترین مسیر، دور، شبکه‌ی عصبی، پایداری

چکیده

در چند دهه اخیر روش‌های بهینه‌سازی که بر پایه رویکرد هوش مصنوعی توسعه یافته‌اند، موفقیت‌های چشم‌گیری در حل مؤثر و کارایی مسائل بهینه‌سازی بدست آورده‌اند. روش‌هایی چون الگوریتم ژنتیک، جستجوی ممنوع، شبیه‌سازی تبریدی، شبکه عصبی و ... قابلیت‌های خود را در حل مسائل بزرگ عملی به‌خوبی نشان داده‌اند. در این پایان‌نامه از یک مدل شبکه عصبی بازگشتی برای حل مساله تخصیص و کوتاهترین مسیر استفاده کرده‌ایم. ایده اصلی جانشینی مساله تخصیص با یک مساله برنامه‌ریزی خطی است. بر طبق قضیه بهینگی، قضیه تحلیل تحدب، قضیه پایداری لیاپانف و اصل تغییرناپذیری لازال، نقطه تعادل شبکه عصبی جواب بهینه مساله اصلی است و بر عکس. نشان داده می‌شود که شبکه عصبی ارائه شده در حالت لیاپانف پایدار می‌باشد و همگرای سراسری به جواب بهینه مساله تخصیص است.

تقدیم به همه آنهایی که

می خوانند. بیشتر بدانند.

خدایا...^۱

به من زیستنی عطا کن که در لحظه مرگ، بر بی‌ثمری لحظه‌ای که برای زیستن گذشته است، حسرت نخورم و مُردنی عطا کن که بر بیهودگیش، سوگوار نباشم. بگذار تا آن را، خود انتخاب کنم، اما آنچنان که تو دوست می‌داری. تو می‌دانی و همه می‌دانند که شکنجه دیدن بخاطر تو، زندانی کشیدن بخاطر تو و رنج بردن به پای تو تنها لذت بزرگ زندگی من است، از شادی توست که من در دل می‌خندم، از امید رهایی توست که برق امید در چشمان خسته‌ام می‌درخشد و از خوشبختی توست که هوای پاک سعادت را در ریه‌هایم احساس می‌کنم. نمی‌توانم خوب حرف بزنم. نیروی شگفتی را که در زیر کلمات ساده و جمله‌های ضعیف و افتاده، پنهان کرده‌ام دریاب، دریاب. تو می‌دانی و همه می‌دانند که زندگی از تحمیل لبخندی بر لبان من، از آوردن برق امیدی در نگاه من، از برانگیختن موج شعفی در دل من، عاجز است.

تو، چگونه زیستن را به من بیاموز، چگونه مردن را خود خواهم آموخت. به من توفیق تلاش در شکست، صبر در نومیدی، رفتن بی‌همراه، جهاد بی‌سلاح، کار بی‌پاداش، فداکاری در سکوت، دین بی‌دنیا، مذهب بی‌عوام، عظمت بی‌نام، خدمت بی‌نان، ایمان بی‌ریا، خوبی بی‌نمود، گستاخی بی‌خامی، قناعت بی‌غرور، عشق بی‌هوس، تنهایی در انبوه جمعیت، و دوست داشتن بی‌آنکه دوست بداند، روزی کن.

اگر تنها ترین تنها شوم، باز خدا هست

او جان‌شین همه ندانستن هست...

^۱ مناجاتی از دکتر علی شریعتی.

پاس‌گزازی...

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست. در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر علی‌رضا ناظمی، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید. همچنین لازم می‌دانم از پدید آورندگان بسته زی‌پرشین، مخصوصاً جناب آقای وفا خلیقی، که این پایان‌نامه با استفاده از این بسته، آماده شده است، کمال قدردانی را داشته باشم. در پایان، بوسه می‌زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می‌کنم وجود مقدس‌شان را و تشکر می‌کنم از برادران عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که در این سردترین روزگاران، بهترین پشتیبان من بودند.

عزرا قز لعلی

آبان ۱۳۹۲

فهرست مطالب

خ	لیست تصاویر
۳	۱ پیشینه پژوهش و تعاریف مقدماتی
۳	۱.۱ مقدمه
۳	۲.۱ توابع محدب و نرم‌ها
۵	۱.۲.۱ شرایط لازم و کافی برای مسائل نامقید [۱۹]
۶	۲.۲.۱ شرایط لازم و کافی برای مسائل مقید [۱۹]
۸	۳.۱ پایداری
۱۱	۴.۱ تابع انرژی
۱۲	۵.۱ اصل تغییرناپذیری لازال [۱۶]
۱۴	۲ تاریخچه مختصر از شبکه عصبی و کاربرد آن در بهینه‌سازی
۱۴	۱.۲ مقدمه
۱۵	۱.۱.۲ تاریخچه مختصر شبکه‌های عصبی [۴]
۱۶	۲.۱.۲ از زیست‌شناختی به شبکه عصبی مصنوعی [۲]، [۲۷]
۱۷	۳.۱.۲ تابع تحرک هر نرون در شبکه عصبی مصنوعی
۱۸	۴.۱.۲ توابع تحریک مورد استفاده در شبکه عصبی
۲۰	۵.۱.۲ معرفی سه شبکه عصبی ساده
۲۱	۶.۱.۲ کاربردهای شبکه عصبی
۲۲	۷.۱.۲ چرا از شبکه عصبی استفاده می‌کنیم؟
۲۳	۸.۱.۲ معایب شبکه‌ی عصبی
۲۳	۲.۲ شبکه‌های عصبی در بهینه‌سازی [۳]
۲۴	۳.۲ مدل‌های مختلف شبکه عصبی

۳۷	۳ کاربرد شبکه عصبی در مساله تخصیص
۳۷	۱.۳ مقدمه
۳۷	۲.۳ مساله تخصیص و تاریخچه مختصر
۳۸	۱.۲.۳ یک روش حل مساله تخصیص
۳۹	۳.۳ مساله تخصیص و فرمول‌بندی
۴۱	۴.۳ تحلیل پایداری
۴۹	۵.۳ کوتاهترین مسیر
۵۱	۱.۵.۳ ارتباط بین مساله‌ی تخصیص و کوتاهترین مسیر [۳۳]
۶۲	۴ استفاده از یک مدل شبکه عصبی برای مسائلی که دچار دور افتادگی می‌شوند
۶۲	۱.۴ مقدمه
۶۲	۲.۴ مساله دور
۶۳	۱.۲.۴ حقایق در باره مسائل برنامه‌ریزی خطی با جنبه دوری [۲۸]
۶۸	۳.۴ نتایج
۶۸	۱.۳.۴ پیشنهادات
۶۹	مراجع
۷۳	واژه‌نامه فارسی به انگلیسی

لیست تصاویر

۸	پایداری لیپانوف	۱.۱
۹	پایداری مجانبی	۲.۱
۱۷	یک نرون طبیعی و یک شبکه عصبی ساده	۱.۲
۱۹	انواع توابع تحریک	۲.۲
۳۰	مسیر جواب در مثال ۲.۳.۲	۳.۲
۳۲	دیاگرام مدل شبکه عصبی ۲۵.۲	۴.۲
۳۲	مسیر جواب در مثال ۳.۳.۲	۵.۲
۳۳	مسیر جواب در مثال ۴.۳.۲	۶.۲
۳۵	مسیر جواب در مثال ۵.۳.۲	۷.۲
۳۶	مسیر جواب در مثال ۶.۳.۲	۸.۲
۳۶	مسیر جواب در مثال ۷.۳.۲	۹.۲
۴۶	دیاگرام شبکه‌ی عصبی ارائه شده در ۱۷.۳-۱۵.۳	۱.۳
۴۷	رفتار x_{ij} در مثال ۶.۴.۳	۲.۳
۴۷	رفتار z_{ij} در مثال ۶.۴.۳	۳.۳
۴۸	مسیر اولیه مساله تخصیص در مثال ۷.۴.۳	۴.۳
۴۹	رفتار z_{ij} در مثال ۷.۴.۳	۵.۳
۵۲	گراف مثال ۱.۵.۳	۶.۳
۵۳	رفتار z_{ij} در مثال ۱.۵.۳	۷.۳
۵۳	مسیر هزینه در مثال ۱.۵.۳	۸.۳
۵۴	گراف مثال ۲.۵.۳	۹.۳
۵۵	رفتار z_{ij} در مثال ۲.۵.۳	۱۰.۳

۵۵	مسیر هزینه در مثال ۲.۵.۳	۱۱.۳
۵۶	گراف مثال ۳.۵.۳	۱۲.۳
۵۷	رفتار z_{ij} در مثال ۳.۵.۳	۱۳.۳
۵۷	مسیر هزینه در مثال ۳.۵.۳	۱۴.۳
۵۸	گراف مثال ۴.۵.۳	۱۵.۳
۵۸	رفتار z_{ij} در مثال ۴.۵.۳	۱۶.۳
۵۹	مسیر هزینه در ۴.۵.۳	۱۷.۳
۶۰	مسیر اولیه در مثال ۵.۵.۳	۱۸.۳
۶۰	رفتار z_{ij} در مثال ۵.۵.۳	۱۹.۳
۶۱	مسیر هزینه در مثال ۵.۵.۳	۲۰.۳
۶۵	مسیر جواب در مثال ۲.۲.۴	۱.۴
۶۶	مسیر جواب در مثال ۳.۲.۴	۲.۴
۶۷	مسیر جواب در مثال ۴.۲.۴	۳.۴
۶۷	مسیر جواب در مثال ۵.۲.۴	۴.۴

پیشگفتار

با پیشرفت فن‌آوری اطلاعات و ارتباطات و توسعه ارتباطات درون سازمانی و بین سازمانی نیاز به استفاده از مدل‌های بهینه‌سازی را برای استفاده منطقی از داده‌ها و اطلاعات فراهم شده گسترش داده است. این مطلب متضمن بزرگ شدن اندازه مسائل بهینه‌سازی که در عمل وجود دارند خواهد بود. در این شرایط لزوم به کارگیری روش‌های کارآمدی که بتوانند با سرعت بالا مسائل بسیار بزرگ را با کیفیت قابل قبول حل کنند بیش از پیش احساس می‌شود. در چند دهه گذشته، استفاده از مدل‌های شبکه عصبی برای حل مسائل بهینه‌سازی با موفقیت‌های قابل توجهی همراه بوده است. در این پایان‌نامه سعی کرده‌ایم مروری بر شبکه‌های عصبی داشته و با استفاده از این نظریه به حل رده‌ای از مسائل بهینه‌سازی بپردازیم. به این منظور در فصل اول به مفاهیم، تعاریف و قضایای مورد نیاز برای فصل‌های آتی پرداخته‌ایم. در فصل دوم تاریخچه‌ای از چگونگی پیدایش شبکه عصبی و کاربرد آن در زمینه‌های مختلف علوم را ذکر کرده و در نهایت برخی از مدل‌های شبکه عصبی که تاکنون برای حل مسائل بهینه‌سازی به وجود آمده است را معرفی کرده‌ایم. در فصل سوم با مساله تخصیص درجه اول آشنا می‌شویم، برای حل این نوع مسائل، یک مدل شبکه عصبی پیشنهاد داده و پایداری مجانبی و همگرایی سراسری این مدل را اثبات می‌کنیم و در آخر برای نشان دادن کارایی این شبکه چندین مثال عددی را حل می‌کنیم. در فصل آخر به بررسی مسائلی که دچار دور افتادگی می‌شوند می‌پردازیم و با مدل شبکه عصبی پیشنهاد شده در فصل سوم چندین مثال عددی از این گونه مسائل را حل می‌نماییم.

فصل ۱

پیشینه پژوهش و تعاریف مقدماتی

۱.۱ مقدمه

در این فصل تعاریف مورد نیاز در فصل‌های بعدی به‌طور مختصر آورده شده است. ابتدا تعاریف مرتبط با تابع محدب را بیان نموده و در خصوص مسائل بهینه‌سازی مقید و نامقید به ذکر شرایط لازم و کافی بهینگی^۱ می‌پردازیم. سپس مفاهیم پایداری^۲ و تابع انرژی در سیستم‌های دینامیکی را ارائه می‌دهیم.

۲.۱ توابع محدب و نرم‌ها

در توسعه برنامه‌ریزی غیرخطی، تابع محدب^۳ همواره به عنوان یک مفهوم ریاضی اصلی مورد نیاز است.

تعریف ۱.۲.۱. مجموعه $E \subset \mathbb{R}^n$ را محدب می‌گوییم اگر:

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

تعریف ۲.۲.۱. تابع $f : \mathbb{R}^n \rightarrow \mathbb{R}$ بر روی یک مجموعه نقاط واقع در یک مجموعه محدب $E \subset \mathbb{R}^n$ یک تابع

محدب نامیده می‌شود 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 می‌گوییم هرگاه $-f$ محدب باشد.

^۱Optimality

^۲Stability

^۳Convex

تعریف ۳.۲.۱. تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ را یک تابع آفینی^۴ گوئیم هرگاه f همزمان تابع محدب و تابع مقعر باشد.

تعریف ۴.۲.۱. تابع $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$ را تابع نرم گوئیم هرگاه:

$$\begin{cases} \|x\| \geq 0, \quad \forall x \in \mathbb{R}^n, \\ \|x\| = 0 \Leftrightarrow x = 0, \\ \|\alpha x\| = |\alpha| \|x\|, \\ \|x + y\| \leq \|x\| + \|y\| \end{cases}$$

و مثال‌هایی از نرم را به صورت زیر داریم:

$$\|x\|_1 = |x_1| + \dots + |x_n|,$$

$$\|x\|_2 = \sqrt{x^T x} = \sqrt{x_1^2 + \dots + x_n^2},$$

$$\|x\|_r = \sqrt[r]{x_1^r + \dots + x_n^r}, \quad r \geq 1.$$

تعریف ۵.۲.۱. نگاشت مشتق‌پذیر $f = (f_1, f_2, \dots, f_m)^T$ که $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ و بردار $x = (x_1, \dots, x_n)$

را در نظر بگیرید، در اینصورت ماتریس ژاکوبین^۵ f در x را به صورت زیر تعریف می‌شود:

$$J_f(x_1, \dots, x_n) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \dots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

همچنین به $\nabla f = (\frac{\partial f_1}{\partial x_1}, \dots, \frac{\partial f_1}{\partial x_n})$ گرادیان^۶ f گفته می‌شود.

تعریف ۶.۲.۱. تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ و بردار $X = (x_1, x_2, \dots, x_n)$ را در نظر بگیرید. ماتریس هسین^۷ f در X

عبارتست از:

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix},$$

ماتریس هسین با ماتریس ژاکوبین به صورت $H(f)(X) = J(\nabla f)(X)$ در ارتباط است.

تعریف ۷.۲.۱. اگر $\bar{x} \in X$ و ε -همسایگی از $\bar{x}$ مثل $N_\varepsilon(\bar{x})$ موجود باشد که به ازای هر

$x \in X \cap N_\varepsilon(\bar{x})$ داشته باشیم $f(\bar{x}) \leq f(x)$ آنگاه $\bar{x}$ را یک کمینه موضعی f می‌نامیم.

^۴ Affine

^۵ Jacobian

^۶ Gradient

^۷ Hessian

تعریف ۸.۲.۱. یک ماتریس $M(x)$ در اندازه $n \times n$ که عناصر آن m_{ij} ($i, j = 1, \dots, n$) توابعی روی $E \subset \mathbb{R}^n$ هستند،

۱. نیمه معین مثبت^۸ روی E نامیده می‌شود، اگر:

$$\forall V \in \mathbb{R}^n, V \neq 0, x \in E : V^T M(x) V \geq 0.$$

۲. معین مثبت^۹ روی E نامیده می‌شود، اگر:

$$\forall V \in \mathbb{R}^n, V \neq 0, x \in E : V^T M(x) V > 0.$$

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

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

قضیه ۹.۲.۱. اگر $\gamma(x)$ کوچکترین مقدار ویژه ماتریس $M(x)_{n \times n}$ که در آن $M(x) = n \times n$ ماتریس متقارن است باشد، آنگاه:

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

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

۳. $M(X)_{n \times n}$ روی E معین مثبت قوی است اگر و فقط اگر به ازای هر $x \in E$ ، $\gamma(x) \geq \alpha > 0$.

۱.۲.۱ شرایط لازم و کافی برای مسائل نامقید [۱۹]

قضیه ۱۰.۲.۱. (شرط لازم مرتبه اول): فرض کنید X یک مجموعه باز ناتهی در $\mathbb{R}^n$ باشد. مسأله برنامه‌ریزی نامقید زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (1.1)$$

$$\text{subject to } x \in X, \quad (2.1)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$ در $\hat{x}$ مشتق‌پذیر است. اگر $\hat{x}$ یک کمینه موضعی باشد، آنگاه $\nabla f(\hat{x}) = 0$.

^۸Positive semi-definite

^۹Positive definite

قضیه ۱۱.۲.۱. (شرط لازم مرتبه دوم): فرض کنید مشتق دوم $f: \mathbb{R}^n \rightarrow \mathbb{R}$ در $\hat{x}$ موجود باشد. اگر $\hat{x}$ یک کمینه موضعی باشد، آنگاه $\nabla f(\hat{x}) = 0$ و ماتریس هسین f'' در $\hat{x}$ نیمه معین مثبت است.

قضیه ۱۲.۲.۱. (شرط کافی): فرض کنید مشتق دوم $f: \mathbb{R}^n \rightarrow \mathbb{R}$ در $\hat{x}$ موجود باشد. اگر $\nabla f(\hat{x}) = 0$ و ماتریس هسین f'' در $\hat{x}$ معین مثبت باشد، آنگاه $\hat{x}$ یک کمینه موضعی اکید است.

قضیه ۱۳.۲.۱. : اگر $f(x)$ بر مجموعه‌ی محدب E تابعی محدب باشد آنگاه $f(x)$ دارای یک کمینه موضعی^{۱۱} است. بنابراین $f(x)$ یک کمینه سراسری^{۱۲} است و بر روی مجموعه محدب به دست آورده می‌شود.

قضیه ۱۴.۲.۱. فرض کنید:

$$E = \{x | x \in \mathbb{R}^n, g_k(x) \leq 0 \ (k = 1, \dots, m), h_p(x) = 0 \ (p = 1, \dots, L)\}.$$

اگر $g_k(x), k = 1, \dots, m$ تابعی محدب و $h_p(x), p = 1, \dots, L$ تابعی آفینی باشد، آنگاه E یک مجموعه‌ی محدب است.

قضیه ۱۵.۲.۱. : فرض کنید E مجموعه نقاطی از $\mathbb{R}^n$ باشد که در قیود $g_k(x) \leq 0$ و $x \leq 0$ برای $k = (1, \dots, m)$ صدق کند. اگر $g_k(x)$ ها توابعی محدب باشند آنگاه E یک مجموعه محدب است.

۲.۲.۱ شرایط لازم و کافی برای مسائل مقید [۱۹]

قضیه ۱۶.۲.۱. (شرایط لازم کروش-کان-تاکر)^{۱۳} : فرض کنید X یک مجموعه‌ی باز ناتهی در $\mathbb{R}^n$ باشد. مسأله‌ی کمینه‌سازی مقید زیر را در نظر بگیرید:

$$\text{Minimize } f(x) \quad (3.1)$$

$$\text{subject to } \begin{cases} g_k(x) \leq 0, & k = 1, \dots, m, \\ h_j(x) = 0, & j = 1, \dots, l, \\ x = (x_1, x_2, \dots, x_n)^T \in X. \end{cases} \quad (4.1)$$

^{۱۰} Hessian

^{۱۱} Local minimum

^{۱۲} Global minimum

^{۱۳} Karush-Kuhn-Tucker

که در آن توابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ، $g_k: \mathbb{R}^n \rightarrow \mathbb{R}$ و $h_j: \mathbb{R}^n \rightarrow \mathbb{R}$ فرض کنید $\hat{x}$ یک جواب شدنی این مسأله باشد و $K = \{k: g_k(\hat{x}) = 0\}$ فرض کنید g_k برای هر k در $\hat{x}$ پیوسته باشد، همچنین فرض کنید f و g_k برای $k \in K$ در $\hat{x}$ مشتق‌پذیر و h_j دارای مشتق پیوسته باشند. به‌علاوه فرض کنید $\nabla g_k(\hat{x})$ برای $k \in K$ و $\nabla h_j(\hat{x})$ برای $j = 1, \dots, l$ مجموعه‌ی مستقل خطی باشند. اگر $\hat{x}$ کمینه موضعی برای (۳.۱) و (۴.۱) باشد، آنگاه اسکالرهایی یکتای $\hat{u}_k \geq 0$ برای $k \in K$ و $\hat{v}_j \geq 0$ موجود هستند به‌طوری‌که:

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

علاوه بر فرض‌های بالا اگر g_k برای $k \notin K$ در $\hat{x}$ مشتق‌پذیر باشد، آنگاه شرایط کروش-کان-تاکر می‌تواند به‌صورت زیر نوشته شود:

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

همچنین شرایط کروش-کان-تاکر می‌تواند به‌فرم ماتریسی زیر نوشته شود:

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

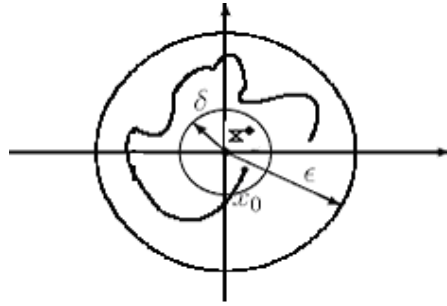
که در آن $\nabla g(\hat{x})$ ماتریس ژاکوبی $m \times n$ و $\nabla h(\hat{x})$ ماتریس ژاکوبی $l \times n$ است و $\hat{u}$ یک بردار m تایی و $\hat{v}$ یک بردار l تایی است. $(\hat{u}^T, \hat{v}^T)^T$ بردار ضرایب لاگرانژ نامیده شده است.

قضیه ۱۷.۲.۱. (شرایط کافی کروش-کان-تاکر): فرض کنید X یک مجموعه باز ناتهی در $\mathbb{R}^n$ باشد، فرض کنید $\hat{x}$ یک جواب شدنی برای مسأله‌ی (۴.۱) و (۳.۱) باشد و $K = \{k: g_k(\hat{x}) = 0\}$ ، اگر شرایط کروش-کان-تاکر در $\hat{x}$ برقرار باشد، یعنی اسکالرهایی $u_k \geq 0$ برای $k \in K$ و $v_j \geq 0$ برای $j = 1, \dots, l$ موجود باشند به‌طوری‌که:

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

آنگاه $\hat{x}$ یک کمینه موضعی برای (۴.۱) خواهد بود.

قضیه ۱۸.۲.۱. : اگر f در مسأله (۳.۱) و (۴.۱) g_k و h_j ($k = 1, \dots, m$) توابعی محدب و ($j = 1, \dots, l$) تابعی آفینی باشد آنگاه شرایط کروش-کان-تاکر لازم و کافی می‌باشند.



شکل ۱.۱: پایداری لیپانوف

۳.۱ پایداری

سیستم دینامیکی زیر را در نظر بگیرید:

$$\begin{cases} \dot{x} = f(x(t)), \\ x(t_0) = x_0 \in \mathbb{R}^n, \end{cases} \quad (5.1)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$. x^* یک نقطه تعادل^{۱۴} برای (۵.۱) است، اگر $f(x^*) = 0$.

تعریف ۱.۳.۱. فرض کنید که $x(t)$ یک جواب برای (۵.۱) باشد، نقطه تعادل x^* پایدار به مفهوم لیپانوف^{۱۵} است،

اگر برای هر $x(t_0) = x_0$ و هر $\epsilon > 0$ یک $\delta > 0$ وجود داشته باشد، به طوری که اگر $\|x(t) - x_0\| < \delta$ آنگاه:

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

تعریف ۲.۳.۱. سیستم دینامیکی (۵.۱) همگرای سراسری^{۱۶} به مجموعه‌ی

$$\Omega^* = \{x | x \text{ یک جواب سیستم (۵.۱) است}\},$$

است. اگر هر جواب دلخواه $x(t)$ از سیستم در رابطه‌ی زیر صدق کند:

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

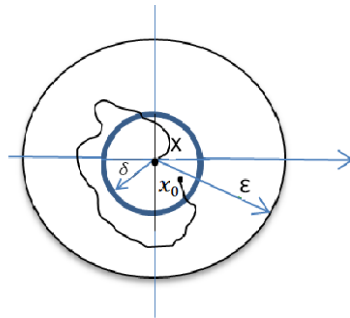
که در آن

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

^{۱۴}Equilibrium point

^{۱۵}Lyapunov

^{۱۶}Global convergent



شکل ۲.۱: پایداری مجانبی

به عبارت دیگر اگر Ω^* تنها شامل یک نقطه‌ی x^* باشد، آنگاه $\lim_{t \rightarrow \infty} x(t) = x^*$ و اگر سیستم در x^* دارای پایداری به مفهوم لیاپانف باشد، آنگاه سیستم در نقطه‌ی x^* پایدار مجانب سراسری^{۱۷} نامیده می‌شود.

تعریف ۳.۳.۱. [۱]. سیستم دینامیکی در نقطه‌ی تعادل یکتای x^* پایدار مجانبی سراسری نامیده می‌شود اگر به مفهوم لیاپانف پایدار باشد و در شرط زیر صدق کند.

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

تعریف ۴.۳.۱. [۱]. مجموعه‌ی $M \subseteq \mathbb{R}^n$ ، یک مجموعه‌ی پایدار نسبت به سیستم (۵.۱) نامیده می‌شود اگر $x(t_0) \in M$ و $t_0 \geq 0$ ، آنگاه به ازای هر $t \geq t_0$ ، $x(t) \in M$ باشد.

روش لیاپانف کلیترین روشی است که در مورد تحلیل پایداری سیستم می‌شناسیم. این روش برای سیستم‌های خطی و غیر خطی از هر مرتبه‌ای قابل استفاده است. برای سیستم‌های خطی هر شرط لازم و کافی را فراهم می‌سازد، ولی برای سیستم‌های غیر خطی فقط شرایط کافی برای پایداری مجانبی را می‌دهد.

تعریف ۵.۳.۱. یک تابع $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ را پیوسته لیپ شیتز^{۱۸} با ثابت L روی $\mathbb{R}^n$ برای هر جفت $x, y \in \mathbb{R}^n$ گوئیم اگر:

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

^{۱۷}Globally asymptotically stable

^{۱۸}Lipshitz

F را لیپ شیتز موضعی روی $\mathbb{R}^n$ گوئیم اگر یک همسایگی $D \subset \mathbb{R}^n$ باشد به طوری که نامساوی بالا برای هر جفت نقطه $x, y \in D$ برقرار باشد.

تعریف ۶.۳.۱. نگاشت $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ یکنوا^{۱۹} گفته می شود، اگر برای هر زوج از نقاط $x, y \in \mathbb{R}^n$ داشته باشیم:

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

همچنین روی $\mathbb{R}^n$ اکیداً یکنوا^{۲۰} است، اگر نامساوی فوق به صورت اکید برای هر $x \neq y$ برقرار باشد. و F روی $\mathbb{R}^n$ یکنوای قوی است اگر برای هر زوج از نقاط $x, y \in \mathbb{R}^n$ ثابت $\beta > 0$ وجود داشته باشد، به طوری که:

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

قضیه ۷.۳.۱. [۱۲]: فرض می کنیم که $F: K \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ ، 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) است.

قضیه ۸.۳.۱. [۹]. در سیستم دینامیکی (۵.۱) فرض می کنیم که $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ یک تابع پیوسته، آنگاه برای هر $t_0 > 0$ و $x_0 \in \mathbb{R}^n$ ، یک جواب موضعی^{۲۱} $x(t)$ 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 موضعی صدق کند آنگاه جواب یکتا خواهد بود و اگر f در $\mathbb{R}^n$ 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 تغییر متغیر، یک سیستم جدید با نقطه تعادل y^* تعریف کرد. برای این منظور تعریف می کنیم:

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

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

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

^{۱۹} Monotone

^{۲۰} Strictly monotonic

^{۲۱} Local solution

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.

۴.۱ تابع انرژی

قضیه ۱.۴.۱. [۳۰]: فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم زیر باشد.

$$\begin{cases} \dot{x} = f(x(t)), \\ x(t_0) = x_0 \in \mathbb{R}^n, \end{cases} \quad (۴.۱)$$

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

$$۱. \quad E(0) = 0.$$

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

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

آنگاه $x = 0$ نقطه‌ی پایداری سیستم خواهد بود و $E(x)$ را «تابع لیپانوف» یا «تابع انرژی» برای سیستم (۴.۱) نامیم.

قضیه ۲.۴.۱. [۳۰]: فرض کنید که $x = 0$ یک نقطه‌ی تعادل برای سیستم (۴.۱) باشد و $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$

یک تابع به طور پیوسته مشتق پذیر باشد اگر:

$$۱. \quad E(0) = 0.$$

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

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

آنگاه $x = 0$ پایدار مجانبی خواهد بود.

لم ۳.۴.۱. [۲۵]: اگر یک نگاشت F روی یک مجموعه باز محدب D شامل Ω به طور پیوسته مشتق پذیر باشد،

سپس F یکنواخت (اکیدا یکنواخت) روی Ω است اگر و فقط اگر ماتریس ژاکوبین آن نیمه معین مثبت (مثبت

معین) برای هر $x \in \Omega$ باشد.

لم ۴.۴.۱. [۲۶]. اگر $A \in \mathbb{R}_{m \times n}$ با رتبه کامل باشد، بنابراین $A^T A$ یک ماتریس متقارن معین مثبت می‌باشد.

۵.۱ اصل تغییرناپذیری لازال [۱۶]

تعریف ۱.۵.۱. فرض کنید $x(t)$ جوابی از سیستم دینامیکی $\dot{x} = f(x)$ باشد. نقطه $p \in \mathbb{R}^n$ حد مثبت^{۲۲}

$x(t)$ نامیده می‌شود، اگر دنباله $(t_n)_{n \geq 0}$ موجود باشد بطوریکه $\lim_{n \rightarrow \infty} t_n = \infty$ و $\lim_{n \rightarrow \infty} x(t_n) = p$.

مجموعه حدود مثبت^{۲۳} دنباله مسیرهای $(x(t))_{t \geq 0}$ را با L^+ نمایش می‌دهیم.

تعریف ۲.۵.۱. مجموعه M را مجموعه تغییرناپذیر مثبت^{۲۴} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر:

$$x(0) \in M \Rightarrow x(t) \in M, \quad \forall t \geq 0.$$

تعریف ۳.۵.۱. مجموعه M را مجموعه تغییرناپذیر منفی^{۲۵} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر:

$$x(0) \in M \Rightarrow x(t) \in M, \quad \forall t \leq 0.$$

تعریف ۴.۵.۱. مجموعه M را مجموعه تغییرناپذیر اساسی^{۲۶} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر

تغییرناپذیر مثبت و همچنین، تغییرناپذیر منفی باشد.

تعریف ۵.۵.۱. دنباله مسیرهای $(x(t))_{t \geq 0}$ پایدار مجانبی وابسته به M نامیده می‌شود، اگر $t \rightarrow \infty$ و برای هر

$\varepsilon > 0$ وجود داشته باشد $T > 0$ ، به‌طوری‌که:

$$d(x(t), M) < \varepsilon, \quad \forall t > T.$$

که در آن d یک متر است.

قبل از بیان اصل تغییرناپذیری لازال^{۲۷} به بیان یک لم می‌پردازیم.

لم ۶.۵.۱. فرض کنید تابع $f : D \rightarrow \mathbb{R}^n$ روی $D \subseteq \mathbb{R}^n$ پیوسته لیپ‌شیتز باشد که D مجموعه‌ای فشرده

و شامل مبدأ است. همچنین، فرض کنید تابع $V : D \rightarrow [0, \infty)$ بطور پیوسته مشتق‌پذیر و $x = x(t)$ جوابی

^{۲۲}Positive limit

^{۲۳}Positive limit set

^{۲۴}Positive invariant set

^{۲۵}Negative invariant set

^{۲۶}Functional invariant set

^{۲۷}LaSalle's invariance principle

برای سیستم دینامیکی $\dot{x} = f(x)$ باشد و این جواب‌ها برای هر $t \geq 0$ توسط مجموعه D کراندار باشند. در اینصورت مجموعه حدود مثبت (L^+) ، ناتهی، فشرده و تغییرناپذیر مثبت است.

قضیه ۷.۵.۱. (اصل تغییرناپذیری لازال) سیستم دینامیکی $\dot{x} = f(x)$ را که $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ در نظر گرفته و فرض کنید $\Omega \subset D$ مجموعه‌ای فشرده و تغییرناپذیر مثبت باشد. همچنین، فرض کنید تابع $V : D \rightarrow \mathbb{R}$ بطور پیوسته مشتق‌پذیر بوده و روی Ω داشته باشیم $\dot{V}(x) \leq 0$. فرض کنید E مجموعه نقاطی از D باشد که $\dot{V}(x) = 0$ است. اکنون اگر M بزرگترین مجموعه تغییرناپذیر مثبت در E باشد، آنگاه هر جواب این سیستم دینامیکی با نقطه شروعی در مجموعه Ω ، پایدار مجانبی وابسته به مجموعه M است.

حال نتیجه مهم زیر را که مستقیماً از اصل تغییرناپذیری لازال به‌دست می‌آید بیان می‌کنیم. این نتیجه قضیه پایداری لیاپانوف را کامل می‌کند.

نتیجه ۸.۵.۱. : فرض کنید مبدأ نقطه تعادل سیستم دینامیکی $\dot{x} = f(x)$ باشد و تابع $V : D \rightarrow \mathbb{R}$ روی $D \subset \mathbb{R}^n$ پیوسته مشتق‌پذیر و شامل مبدأ باشد بطوریکه روی مجموعه D داشته باشیم $\dot{V}(x) \leq 0$. همچنین مجموعه $E = \{x \in D \mid \dot{V}(x) = 0\}$ را در نظر می‌گیریم. اگر E فقط شامل نقطه $x(t) \equiv 0$ باشد، آنگاه مبدأ به عنوان یک نقطه تعادل سیستم دینامیکی بالا، پایدار مجانبی خواهد بود.

فصل ۲

تاریخچه مختصر از شبکه عصبی و کاربرد آن در بهینه‌سازی

۱.۲ مقدمه

دهه‌های آغازین سده بیستم میلادی و دوران پیشرفت شگرف صنعتی، همراه با تولید خودرو بود که انقلاب همه جانبه این در ترابری، افزایش شتاب جابجایی و صدها کار و پیشه جدید در رشته‌های بازرگانی بوجود آورده است. به نظر می‌رسد که سبیل دوران فراصنعتی و نماد فرآورده‌های بی‌همتای قرن آینده «هوش مصنوعی» است. امروزه موضوع هوش مصنوعی داغ‌ترین بحث میان کارشناسان دانش رایانه و اطلاعات و دیگر دانشمندان و تصمیم‌گیرندگان است. در سراسر تاریخ تا به امروز انسان از جنبه تن و روان، مرکز و محور بحث‌ها و پژوهش‌ها بوده است. ولی اکنون موجودی با رتبه‌ای پائین‌تر، بی‌جان و ساختگی می‌خواهد جانشین او شود، امری که بدون شک می‌توان ادعا نمود بیشتر انسان‌ها با آن مخالفند. هوش مصنوعی چنانچه به هدف‌های والای خود برسد، جهش بزرگی در راه دستیابی بشر به رفاه بیشتر و حتی ثروت افزون‌تر خواهد بود. هم اکنون نمونه‌های خوب و پذیرفتن از هوش مصنوعی در دنیای واقعی ما به کار افتاده است. چنین دستاوردهایی صرف منابع لازم در آینده را همچنان توجیه خواهد کرد. از سوی دیگر، منتقدین هوش مصنوعی چنین استدلال می‌کنند که صرف زمان و منابع ارزشمند دیگر در راه ساخت فرآورده‌ای که پر از نقص و کاستی و دست‌آوردهای مثبت اندکی است، مایه بدنام کردن و زیر پا گذاشتن توانمندی‌ها و هوشمندی‌های انسان می‌باشد. تلخ‌ترین انتقادهای بر این باور است که هوش مصنوعی، توهین آشکار به گوهر طبیعت و نقش انسان است. شبکه عصبی مصنوعی روشی عملی برای یادگیری توابع گوناگون نظیر توابع با مقادیر

حقیقی، توابع با مقادیر گسسته و توابع با مقادیر برداری می‌باشد. یادگیری شبکه‌ی عصبی در برابر خطاهای داده‌های آموزشی مصون بوده و شبکه با موفقیت به مسائلی نظیر شناسایی گفتار، شناسایی و تعبیر تصاویر، و یادگیری روبات اعمال شده‌اند. لذا در این فصل تاریخچه‌ای از شبکه‌ی عصبی و کاربرد آن در زمینه‌های مختلف علوم اشاره کرده و در انتها مدل‌هایی از شبکه‌های عصبی که برای حل مسائل گوناگون بهینه‌سازی (مسائل خطی یا غیر خطی) به دست آمده است را معرفی می‌کنیم.

۱.۱.۲ تاریخچه مختصر شبکه‌های عصبی [۴]

دیدگاه جدید شبکه‌های عصبی در دهه‌ی چهل قرن بیستم شروع شد، زمانیکه وارن مک کلوت^۱ و والتر پیتز^۲ نشان دادند که شبکه‌های عصبی در اصل می‌توانند هر تابع حسابی^۳ و منطقی^۴ را محاسبه نمایند. کار این افراد را می‌توان نقطه‌ی شروع حوزه‌ی علمی شبکه‌های عصبی مصنوعی نامید و این موضوع با دونالد هب^۵ ادامه یافت. نخستین کاربرد عملی شبکه‌های عصبی در اواخر دهه‌ی پنجاه قرن بیستم مطرح شد، زمانیکه فرانک روزنبلات^۶ در سال ۱۹۵۸ شبکه‌ی پرسپترون^۷ را معرفی نمود. روزنبلات و همکارانش شبکه‌ای ساختند که قادر بود الگوها را از هم شناسایی نماید. در همین زمان بود که برنارد ویدرو^۸ در سال ۱۹۶۰ شبکه‌ی عصبی تطبیقی خطی آدلاین^۹ را با قانون یادگیری جدید مطرح نمود که از لحاظ ساختار شبیه شبکه‌ی پرسپترون می‌باشد. پیشرفت شبکه‌های عصبی تا دهه هفتاد قرن بیستم ادامه یافت. در ۱۹۷۲ تئوکوهونن^{۱۰} و جیمز اندرسون^{۱۱} به طور مستقل و بدون اطلاع از هم شبکه‌های عصبی جدید را معرفی نمودند که قادر بودند به عنوان عناصر ذخیره‌ساز عمل نمایند. استفان گروسبرگ^{۱۲} در این دهه روی شبکه‌های خود سازمانده^{۱۳} فعالیت می‌کرد. علاقه به کار کردن روی شبکه‌های عصبی در دهه شصت قرن بیستم در قیاس با دهه هشتاد به علت عدم بروز ایده‌های جدید و نبود کامپیوترهای سریع جهت

^۱ Warren McCulloch

^۲ Walter Pitts

^۳ Logical function

^۴ Arithmetic

^۵ Donald Hebb

^۶ Frank Rosenblatt

^۷ Perceptron

^۸ Bernard Widrow

^۹ Adaptive linear element

^{۱۰} Teo Kohonen

^{۱۱} James Anderson

^{۱۲} Stefan Grossberg

^{۱۳} Self-organizing

پیاده‌سازی کمرنگ می‌نمود. لیکن در خلال دهه هشتاد، تحقیقات روی شبکه‌های عصبی دوباره افزایش یافت، در این زایش دوباره شبکه‌های عصبی دو نگرش جدید قابل تأمل می‌باشند. استفاده از مکانیسم تصادفی جهت توضیح عملکرد یک طبقه وسیع از شبکه‌های برگشتی^{۱۴} که می‌توان آن‌ها را جهت ذخیره‌سازی اطلاعات استفاده نمود. این ایده توسط جان هاپفیلد^{۱۵} در سال ۱۹۸۲ مطرح شد. دومین ایده‌ی مهم که کلید توسعه‌ی شبکه‌های عصبی در دهه هشتاد شد، الگوریتم پس انتشار خطا^{۱۶} می‌باشد که توسط دیوید راملهارت^{۱۷} و جیمز مک‌کلند^{۱۸} مطرح گردید. با بروز این دو ایده شبکه‌های عصبی متحول شدند و هزاران مقاله نوشته شد و شبکه‌های عصبی کاربردهای زیادی در رشته‌های مختلف علوم پیدا کردند. بیشتر پیشرفت‌ها در شبکه‌های عصبی به ساختارهای نوین و روش‌های یادگیری جدید مربوط می‌شوند.

۲.۱.۲ از زیست‌شناختی به شبکه عصبی مصنوعی [۲]، [۲۷]

کار بر روی شبکه‌های عصبی مصنوعی، با الهام از عملکرد مغز انسان از آنجا شروع شد که دانشمندان دریافتند مغز بشر به‌گونه‌ای کاملاً متفاوت از کامپیوترهای دیجیتالی مرسوم محاسبات را انجام می‌دهد. مغز یک سیستم پردازش داده بسیار پیچیده است که از واحدهای ساختاری به نام سلول عصبی یا نرون^{۱۹} تشکیل شده است. مغز انسان شامل ۱۰^{۱۱} عصب می‌باشد که به‌طور گسترده باهم در ارتباط اند. هر نرون با ۱۰^۴ نرون دیگر در ارتباط است که با هم تقریباً ۱۰^{۱۴} سیناپس (دستگاه عصبی)^{۲۰} را به وجود می‌آورند. شکل (۱.۲) یک نرون و یک شبکه عصبی ساده تک خروجی را نشان می‌دهد: قسمت الف یک مدل بیولوژیکی عصب را در مغز نشان می‌دهد. ترکیب اصلی یک عصب بیولوژیکی، دندریت‌ها هستند که کار آن‌ها انتقال فعالیت از عصب‌های دیگر به سلول‌های بدن می‌باشد و اکسون که اطلاعات را انتقال می‌دهد. قسمت ب یک مدل ساده ولی بنیادی را نشان می‌دهد که نرون‌ها به طور منفرد در خود پیچیده‌اند. یکی از موارد مهمی که در بسیاری از مدل‌های شبکه عصبی مصنوعی مورد غفلت است، مقدار زمانی است که یک سیگنال در طول اکسون عبور می‌کند، البته در بعضی ساختارها این مورد را به طور واضح در نظر می‌گیرند.

^{۱۴}Feedback(Recurrent)

^{۱۵}John Hopfield

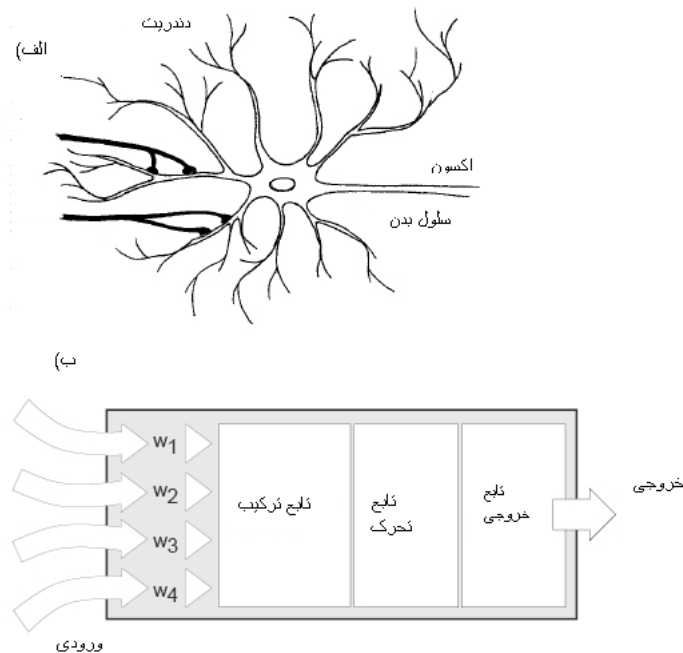
^{۱۶}Error back-propagation

^{۱۷}Daivid Rummelhurt

^{۱۸}James McLand

^{۱۹}Neuron

^{۲۰}Sinaps

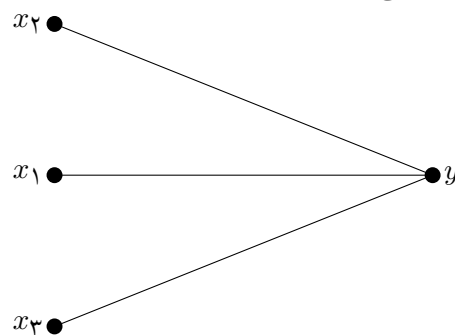


شکل ۱.۲: یک نرون طبیعی و یک شبکه عصبی ساده

۳.۱.۲ تابع تحرک هر نرون در شبکه عصبی مصنوعی

هر شبکه عصبی شامل مجموعه ای از اجزاء کوچک داده پرداز به نام نرون، واحد یا گره می باشد. هر نرون به نرون دیگر از طریق یک رابط جهت دار که دارای وزن است مرتبط می گردد، وزن ها نمایش دهنده ی اطلاعات مورد نیاز شبکه می باشند. برای حل یک مساله در یک شبکه عصبی، هر یک از نرون ها برای محاسبه ی خروجی خود، یک تابع تحرک را که معمولا غیرخطی است به ورودی هایش اعمال می نماید. به عنوان مثال در شکل ساده ی زیر که

یک نرون مصنوعی بسیار ساده را نشان می دهد:



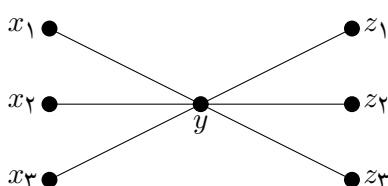
نرون y داده‌هایی را از نرون‌های x_1, x_2 و x_3 دریافت می‌نماید که پاسخ تحرک یا جواب خروجی از این نرون‌ها به ترتیب x_1, x_2, x_3 هستند. وزن رابط بین نرون‌های y و x_1, x_2 و x_3 به ترتیب w_1, w_2, w_3 می‌باشد. ورودی خالص y_{in} به نرون y برابر مجموع داده‌های وزن‌دار شده از نرون‌های x_1, x_2, x_3 می‌باشد بدین گونه:

$$y_{in} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

پاسخ تحریک Y مربوط به نرون y به صورت تابعی از ورودی خالص این نرون بیان می‌گردد، $y = f(y_{in})$ به‌طوری‌که تابع تحریک سیگموئید مورد استفاده در تعیین این پاسخ تحرک y به‌صورت زیر می‌باشد:

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

با توجه به شکل زیر در حالت گسترده فرض می‌کنیم که Y به نرون‌های z_1, z_2 و z_3 با وزن‌های v_1, v_2 و v_3 مرتبط گردد.

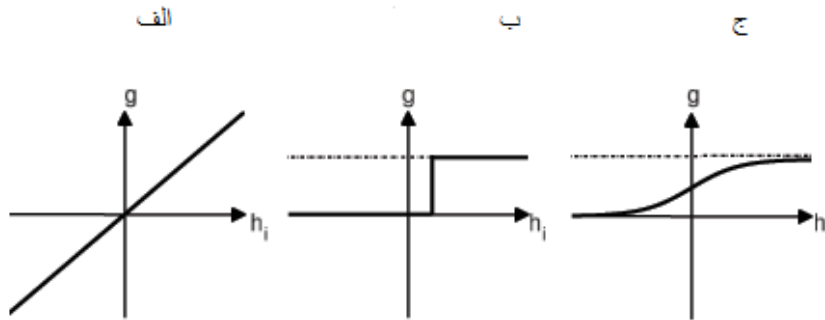


اگرچه شکل بالا ساده می‌باشد، ولی وجود یک نرون پنهان توام با تابع تحریک غیر خطی موجب افزایش توانایی شبکه می‌شود، به‌طوری‌که بسیاری از مسائل قابل حل با این شبکه، با شبکه‌هایی که تنها شامل نرون‌های ورودی و خروجی هستند قابل حل نمی‌باشد. در مقابل آموزش شبکه حاوی لایه میانی و تعیین مقادیر بهینه وزن‌ها مشکل‌تر می‌گردد.

۴.۱.۲ توابع تحریک مورد استفاده در شبکه عصبی

نمودار (۲.۲) بیشترین توابع تحریک مورد استفاده را نشان می‌دهد: g تابع تحریک و h_i مجموع ورودی را نشان می‌دهد. نمودار قسمت الف مربوط به یک تابع تحریک خطی است. نمودار ب یک تابع پله‌ای که ناخطی است را نشان می‌دهد و نمودار قسمت ج تابع تحریک سیگموئید^{۲۱} است. تابع سیگموئید در واقع یک مدل هموار تابع پله‌ای است که به صورت زیر است:

^{۲۱} Sigmoid



شکل ۲.۲: انواع توابع تحریک

$$g(h_i) = \frac{1}{1 + e^{-2\beta h_i}}$$

با $\beta = 1/k_B T$ که T دمای مطلق را نشان می‌دهد. تابع سیگموئید بین صفر و یک تغییر می‌کند، ولی گاهی اوقات نیز تابع تحریک بین -1 و 1 تغییر می‌کند، که در این صورت تابع تحریک به فرم زیر تعریف می‌شود:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

و $\tanh(\beta h) = 2g(h) - 1$ داده شده است.

بایاس^{۲۲} آستانه تحریک

بایاس هر نرون بمانند وزنی با مقدار مشخص که از یک نرون مفروض با پاسخ ثابت 1 بر آن تحمیل می‌شود عمل می‌کند. افزایش بایاس موجب افزایش ورودی نرون می‌گردد.

ملاحظه ۱.۱.۲. شبکه‌های عصبی به صورت شبکه‌های عصبی تک‌لایه و چند لایه می‌باشند، شبکه‌های عصبی چند لایه دارای توانایی بیشتری نسبت به شبکه‌های عصبی تک‌لایه‌اند. شبکه‌های عصبی پیش‌خور دولایه با تابع سیگموئید در لایه‌ی اول قادرند هر تابعی را با دقت دلخواه تقریب بزنند، در حالی که شبکه‌های عصبی تک‌لایه از چنین توانایی برخوردار نیستند.

ملاحظه ۲.۱.۲. به نظر می‌رسد که تعداد درجات آزادی (مثلاً تعداد ورودی‌ها، خروجی‌ها و نرون‌های هر لایه) برای

^{۲۲} Bias

طراحی یک شبکه چند لایه پیش‌خور زیاد باشد. ولی باید توجه داشت که تعداد ورودی‌های شبکه و تعداد خروجی‌ها، بر اساس مسأله‌ی خاص که شبکه قرار است حل کند، به دست می‌آیند، می‌توان گفت که ساختار شبکه‌های عصبی تک لایه پیش‌خور را کاملاً از روی ویژگی‌های مسأله‌ی مورد بررسی، می‌توان مشخص نمود. در مورد شبکه‌های عصبی چند لایه این مطلب صادق نیست. به عبارتی دیگر از روی ویژگی‌های مسأله‌ی مورد بررسی، نمی‌توان اطلاعاتی در مورد تعداد نرون‌های لایه‌ی میانی کسب نمود. اما شبکه‌های عصبی پس‌خور یا بازگشتی از توانایی بیشتری نسبت به شبکه‌های پیش‌خور برخوردارند و بهتر می‌توانند رفتار مربوط به ویژگی‌های زمانی سیستم‌ها را نشان دهند. تفاوت شبکه‌های پس‌خور با شبکه‌های پیش‌خور در این است که در شبکه‌های پس‌خور، حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون‌های همان لایه و یا لایه‌ی قبل وجود دارد.

ملاحظه ۳.۱.۲. وجود نرون‌های لایه‌ی میانی زمانی مفید است که توابع محرک آن‌ها غیر خطی باشند، هر چه تعداد لایه‌های میانی افزایش یابد قابلیت‌های بیشتری می‌توانند استخراج گردند، بنابراین با افزایش لایه‌های میانی شبکه‌های عصبی قادر خواهند بود تصویر بهتری از فضای پارامترهای سیستم تحت بررسی به دست دهند.

۵.۱.۲ معرفی سه شبکه عصبی ساده

شبکه‌ی عصبی پرسپترون، شبکه‌ی عصبی همینگ، شبکه‌ی عصبی هاپفیلد^{۲۳}، به ترتیب نمونه‌های ساده‌ای از شبکه‌های عصبی پیش‌خور، شبکه‌های عصبی رقابتی و شبکه‌های حافظه انجمنی برگشتی می‌باشند.

پرسپترون: این شبکه عصبی بر مبنای یک واحد محاسباتی به نام پرسپترون ساخته می‌شود. یک پرسپترون برداری از ورودی‌ها با مقادیر حقیقی را گرفته و یک ترکیب خطی از ورودی‌ها را محاسبه می‌کند. اگر حاصل از یک مقدار آستانه بیشتر بود خروجی پرسپترون برابر با ۱ و در غیر اینصورت معادل ۰ خواهد بود. شبکه‌های عصبی پرسپترون به ویژه پرسپترون چندلایه در زمره کاربردی‌ترین شبکه‌های عصبی می‌باشند، این شبکه‌ها قادرند با انتخاب مناسب تعداد لایه‌ها و سلول‌های عصبی، که اغلب هم زیاد نیستند، یک نگاشت غیر خطی را با دقت دلخواه انجام دهند.

همینگ: نخستین بار توسط اشتاین بوخ^{۲۴} در سال ۱۹۶۱ مطرح شد و در سال‌های اخیر توسط کسانی چون

^{۲۳}Hopfield

^{۲۴}Stein buch

لیپمن^{۲۵} بازنگری شده است. این شبکه اساساً جهت حل مسأله‌ی شناسایی الگوهای باینری (الگوهای برداری که عناصرشان فقط دو مقدار مثلاً ۱ و ۰ را قبول می‌کند) طراحی شده است. این شبکه هم در چارچوب شبکه‌های عصبی قرار می‌گیرد، چون که از یک سری نرون‌ها به مثابه گره‌ها و یک سری وزنه‌های ارتباطی بین گره‌ها تشکیل یافته است. هر گره یک سطح فعال دارد که خروجی نرون را می‌سازد. هدف اصلی در شبکه‌ی همینگ این است که تشخیص دهد کدام الگوی مرجع^{۲۶} بیشترین نزدیکی را به الگوی ورودی دارد و سپس آن را در خروجی شبکه ظاهر کند. شبکه‌ی همینگ از سه لایه‌ی پیش‌خور اول، لایه‌ی برگشتی و لایه‌ی سوم تشکیل شده است.

هاپفیلد: شبکه هاپفیلد دارای معماری خاصی است که آن را از سایر شبکه‌ها جدا کرده است. اصولاً این نوع شبکه‌ها دارای یک لایه از نرون‌های ورودی به نرون‌های ورودی است و به نوعی می‌توان گفت نرون‌های ورودی همان نرون‌های خروجی هستند. برخلاف سایر شبکه‌ها این شبکه در الگوریتم تربیت وزن‌های خود را با تکرار معین نمی‌کند بلکه با فرمول خاصی این کار را انجام می‌دهد و در الگوریتم شناسایی، ورودی با تکرار، تغییر و به یک شکل معین می‌رسد. در این شبکه‌ها در هر زمان فقط یک نرون فعال و سایر نرون‌ها غیر فعال است به عبارتی چون یک نرون از سایر نرون‌ها ورودی می‌گیرد به همین دلیل آن نرون در حال تغییر و دیگر نرون‌ها ثابت هستند. از این نوع شبکه‌ها معمولاً در برطرف کردن نویز از تصویر و یا هر الگوی دیگر استفاده می‌شود.

۶.۱.۲ کاربردهای شبکه عصبی

شبکه‌های عصبی مصنوعی دارای دامنه‌ی کاربرد وسیعی می‌باشند، از جمله سامانه‌های آنالیز ریسک، کنترل هواپیما بدون خلبان، آنالیز کیفیت جوشکاری، آنالیز کیفیت کامپیوتر، آزمایش اتاق اورژانس، اکتشاف نفت و گاز، سامانه‌های تشخیص ترمز کامیون، تخمین ریسک وام، شناسایی طیفی، تشخیص دارو، فرآیندهای کنترل صنعتی، مدیریت خطا، تشخیص صدا، تشخیص هیپاتیت، بازیابی اطلاعات راه دور، شناسایی مین‌های زیردریایی، تشخیص اشیاء سه بعدی و دست نوشته‌ها و چهره و.... در کل می‌توان کاربردهای شبکه‌های عصبی را به صورت زیر دسته بندی کرد: تناظر (شبکه الگوهای مغشوش و به هم ریخته را بازشناسی می‌کند)، خوشه‌یابی، دسته‌بندی، شناسایی، بازسازی الگو، تعمیم‌دهی (به دست آوردن یک پاسخ صحیح برای محرک ورودی که قبلاً به شبکه آموزش داده نشده)، بهینه سازی. امروزه شبکه‌های عصبی در کاربردهای مختلفی نظیر مسائل تشخیص الگو که خود شامل مسائلی مانند

^{۲۵}Lipman

^{۲۶}Prototype

تشخیص خط، شناسایی گفتار، پردازش تصویر و مسائلی از این دست می‌شود و نیز مسائل دسته بندی مانند دسته بندی متون یا تصاویر، به کار می‌روند. در کنترل یا مدل‌سازی سامانه‌هایی که ساختار داخلی ناشناخته یا بسیار پیچیده‌ای دارند نیز به صورت روز افزون از شبکه‌های عصبی مصنوعی استفاده می‌شود. به عنوان مثال می‌توان در کنترل ورودی یک موتور از یک شبکه عصبی استفاده نمود که در این صورت شبکه عصبی خود تابع کنترل را یاد خواهد گرفت.

۷.۱.۲ چرا از شبکه عصبی استفاده می‌کنیم؟

شبکه‌های عصبی با توانایی قابل توجه خود در استنتاج نتایج از داده‌های پیچیده می‌توانند در استخراج الگوها و شناسایی گرایش‌های مختلفی که برای انسان‌ها و کامپیوتر شناسایی آنها بسیار دشوار است استفاده شوند. از مزایای شبکه‌های عصبی می‌توان موارد زیر را نام برد:

۱. یادگیری تطبیقی: توانایی یادگیری اینکه چگونه وظایف خود را بر اساس اطلاعات داده شده به آن و یا تجارب اولیه انجام دهد در واقع اصلاح شبکه را گویند.

۲. خود سازماندهی: یک شبکه عصبی مصنوعی به صورت خودکار سازماندهی و ارائه داده‌هایی که در طول آموزش دریافت کرده را انجام دهد. نرون‌ها با قاعده‌ی یادگیری سازگار شده و پاسخ به ورودی تغییر می‌یابد.

۳. عملگرهای بی‌درنگ: محاسبات در شبکه‌ی عصبی مصنوعی می‌تواند به صورت موازی و به وسیله سخت‌افزارهای مخصوصی که طراحی و ساخت آن برای دریافت نتایج بهینه قابلیت‌های شبکه عصبی مصنوعی است انجام شود.

۴. تحمل خطا: با ایجاد خرابی در شبکه مقداری از کارایی کاهش می‌یابد ولی برخی امکانات آن با وجود مشکلات بزرگ همچنان حفظ می‌شود.

۵. دسته بندی: شبکه‌های عصبی قادر به دسته بندی ورودی‌ها بر ای دریافت خروجی مناسب می‌باشند.

۶. تعمیم‌دهی: این خاصیت شبکه را قادر می‌سازد تا تنها با برخورد با تعداد محدودی نمونه، یک قانون کلی از آن را به دست آورده، نتایج آموخته‌ها را به موارد مشابه از قبل تعمیم دهد. توانایی که در صورت نبود آن

سامانه باید بی نهایت واقعیت‌ها و روابط را به خاطر سپارد.

۷. پایداری-انعطاف پذیری: یک شبکه عصبی هم به حد کافی پایدار است تا اطلاعات فراگرفته خود را حفظ کند و هم قابلیت انعطاف و تطبیق را دارد و بدون از دست دادن اطلاعات قبلی می‌تواند موارد جدید را بپذیرد.

۸.۱.۲ معایب شبکه‌ی عصبی

با وجود برتری‌هایی که شبکه‌های عصبی نسبت به سامانه‌های مرسوم دارند، معایبی نیز دارند که پژوهشگران این رشته تلاش دارند که آنها را به حداقل برسانند، از جمله:

۱. قواعد یا دستورات مشخصی برای طراحی شبکه جهت یک کاربرد اختیاری وجود ندارد.

۲. در مورد مسائل مدل سازی، صرفاً نمی‌توان با استفاده از شبکه عصبی به فیزیک مساله پی برد. به عبارت دیگر مرتبط ساختن پارامترها یا ساختار شبکه به پارامترهای فرآیند معمولاً غیر ممکن است.

۳. دقت نتایج بستگی زیادی به اندازه مجموعه آموزش دارد.

۴. آموزش شبکه ممکن است مشکل و یا حتی غیر ممکن باشد.

۵. پیش بینی عملکرد آینده شبکه (عمومیت یافتن) آن به سادگی امکان پذیر نیست.

۲.۲ شبکه‌های عصبی در بهینه‌سازی [۳]

همانطور که گفته شد یکی از کاربردهای هوش مصنوعی در مسائل بهینه سازی می‌باشد، بهینه نمودن تابع هدف (ماکزیمم یا مینیمم کردن آن) با استفاده از روش‌های تحلیلی سر راست بسیار پیچیده می‌باشد. با وجود این امکان رسیدن به جواب با بکارگیری روش‌های عددی رایانه‌ای میسر می‌گردد. در این روش‌ها با تغییر متغیرهای طراحی در دوره‌های متوالی و با در نظر گرفتن قیود به جواب بهینه نزدیک می‌شویم. استفاده از روش‌های بهینه‌سازی متداول و تحلیل مستقیم سازه‌ها در دفعات مکرر موجب افزایش قابل ملاحظه‌ای زمان محاسبه می‌گردد. 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 بر می‌گیرد. ایده 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 مرتبه اول است. انتظار می‌رود که برای یک نقطه‌ی آغازین نقطه‌ی پایداری دستگاه معادلات دیفرانسیل غیرخطی به دست آمده، جواب بهینه‌ی مسأله بهینه‌سازی اصلی باشد. یک اصل اساسی در استفاده از تابع انرژی این است که برای همگرایی دستگاه معادلات دیفرانسیل به نقطه‌ی پایداری آن، تابع انرژی متناظر با آن حتماً باید نامنفی باشد. البته می‌توان مدل شبکه‌های عصبی نظیر مسائل بهینه‌سازی را حتی بدون در نظر گرفتن تابع انرژی نیز بیان کرد. بنابراین برای یک مدل متناظر با مسائل بهینه‌سازی، اصل اساسی استفاده از شبکه‌های عصبی در اینگونه مسائل به صورت زیر بیان می‌شود:

”برای یک نقطه‌ی آغازین دلخواه، نقطه‌ی پایداری دستگاه معادلات دیفرانسیل غیرخطی بدست آمده جواب بهینه مسأله اصلی است و برعکس.”

مدل‌های مطرح شده متناظر مسائل مختلف بهینه‌سازی را می‌توان به دو قسمت مدل‌های دوگانی و مدل‌های جریمه‌ای تقسیم نمود. نظریه دوگانی و توابع جریمه‌ای دو نظریه 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 گرادینانی برای معرفی مدل مورد نظر استفاده می‌شود، ولی در نظریه دوگانی از مدل‌های اولیه -دوگان برای حل این مسائل استفاده می‌شود. اگر بتوان متناظر با هر مسأله بهینه‌سازی، روش مشخصی را ارائه نمود که آن روش برای حل آن مسأله شرایط لازم و کافی را برآورده سازد، آنگاه می‌توان متناظر با آن روش، یک مدل شبکه عصبی برای مسأله مورد نظر بسازیم. در زیر به بیان چند مدل که تاکنون برای مسائل برنامه‌ریزی خطی و درجه دوم و غیرخطی ارائه شده‌اند می‌پردازیم.

۳.۲ مدل‌های مختلف شبکه عصبی

مدل اول

مسأله برنامه‌ریزی خطی زیر را در نظر بگیرید:

$$\text{minimize } f(x) = a^T x \quad (۱.۲)$$

$$\text{subject to } \begin{cases} g(x) = Dx - b = \circ, \\ x \geq \circ, \end{cases} \quad (۲.۲)$$

که در آن $D \in \mathbb{R}^{m \times n}$, $\text{rank}(D) = m$, $x, a \in \mathbb{R}^n$ و $b \in \mathbb{R}^m$ است. دوگان (۱.۲) و (۲.۲) به صورت زیر بیان می شود:

$$\text{maximize } \bar{f}(y) = b^T y \quad (۳.۲)$$

$$\text{subject to } \begin{cases} \bar{g}(y) = \bar{D}^T y - a \leq \circ, \\ y \in \mathbb{R}^m. \end{cases} \quad (۴.۲)$$

در [۱۱] مدل متناظر برای حل (۱.۲) و (۳.۲) به صورت 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 $\beta = \| (x + D^T y - a)^+ - x \|_p^2$ و $(x, y)^T \in \{(x, y)^T \mid x \geq \circ\}$.
 $(Dx - D^T y - a)^+ = \max\{(Dx - D^T y - a), \circ\}$.

مدل دوم

مسأله برنامه ریزی درجه دوم زیر را در نظر بگیرید:

$$\text{minimize } f(x) = \frac{1}{p} x^T A x + a^T x \quad (۵.۲)$$

$$\text{subject to } \begin{cases} Dx - b = \circ, \\ x \geq \circ, \end{cases} \quad (۶.۲)$$

که در آن A یک ماتریس متقارن معین مثبت است. دوگان (۵.۲) و (۶.۲) به صورت زیر است:

$$\text{maximize } \bar{f}(y) = b^T y - \frac{1}{p} x^T A x \quad (۷.۲)$$

$$\text{subject to } \bar{D} y - A x - a \leq \circ. \quad (۸.۲)$$

در [۱۱] مدل متناظر برای حل (۵.۲) و (۷.۲) به صورت زیر نمایش داده شده است:

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

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

مدل سوم

مسئله برنامه‌ریزی غیرخطی زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (9.2)$$

$$\text{subject to } g(x) = [g_1(x), g_2(x), \dots, g_m(x)]^T \leq 0, \quad (10.2)$$

که در آن $f(x), g(x) \in C^2$ و $f(x), g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}^1$. برای حل (9.2) هاپفیلد و تنک^{۲۷} در [۱۴]، یک مدل شبکه عصبی به صورت زیر ارائه کردند:

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

که در آن C یک ماتریس قطری $n \times n$ بیان کننده ظرفیت هر نرون و Q یک ماتریس قطری $n \times n$ است که اعضای آن بصورت زیر تعریف می‌شوند:

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

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

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

مدل چهارم

برای حل (9.2)، کندی و چوآ^{۲۸} در [۱۸] مدلی دیگر به صورت زیر ارائه داده‌اند:

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

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

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

^{۲۷}Hopfield and Tank

^{۲۸}Kennedy and Chua

مدل پنجم

مجدداً برای حل (۹.۲) توسط رودریگز-واسکوئز^{۲۹} در [۵] مدلی 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

$$u_x = \begin{cases} 1 & g(x) \leq 0, \\ 0 & \text{در غیر این صورت} \end{cases}$$

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

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

همان طور که ملاحظه می کنید، مدل های بیان شده به دلیل مشکلاتی که هر کدام داشته اند به طور متوالی اصلاح شده اند.

مدل ششم

برنامه ریزی محدب درجه دوم^{۳۰} زیر را در نظر بگیرید:

$$\text{minimize} \quad f(x) = \frac{1}{\Psi} x^T Q x + D^T x \quad (11.2)$$

$$\text{subject to} \begin{cases} g(x) = Ax - b \leq 0, \\ h(x) = Ex - f = 0, \end{cases} \quad (12.2)$$

که در آن $Q \in \mathbb{R}^{n \times n}$ یک ماتریس متقارن معین مثبت، $A \in \mathbb{R}^{m \times n}$ ، $b \in \mathbb{R}^m$ ، $E \in \mathbb{R}^{l \times n}$ ، $f \in \mathbb{R}^l$ و $x \in \mathbb{R}^n$ است. بر طبق شرایط کروش-کان-تاکر، $x^* \in \mathbb{R}^n$ جواب بهینه (۱۱.۲) و (۱۲.۲) است اگر و تنها اگر

بردارهای $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ وجود داشته باشد که در رابطه زیر صدق کنند:

$$\begin{cases} u^* \geq 0, Ax^* - b \leq 0, u^{*T}(Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (13.2)$$

برای حل مسأله (۱۱.۲) تابع زیر را که معروف به تابع فیشر-برمیستر^{۳۱} است، معرفی می کنیم:

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

^{۲۹}Rodríguez Vazquez

^{۳۰}Convex quadratic programming

^{۳۱}Ficher-Burmeister

قضیه ۱.۳.۲. [۲۰]:

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

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

۳. $\phi(a, b)$ همه جا به جز در مبدأ دو بار به‌طور پیوسته مشتق پذیر است، ولی در مبدأ قویاً^{۳۲} هموار است.

تابع ϕ که در شرط ۱ قضیه‌ی (۱.۳.۲) صدق کند، مساله مکمل غیر خطی^{۳۳} نامیده می‌شود.

متناظر با دستگاه (۱۲.۲) و (۱۱.۲) دستگاه زیر را در نظر می‌گیریم:

$$\phi(x, u, v) = \begin{bmatrix} \phi(\alpha, Qx + D + A^T u + E^T v) \\ \phi(u, Ax - b) \\ \phi(\beta, Ex - f) \end{bmatrix}, \quad (14.2)$$

که در آن $\alpha = (\alpha_1, \dots, \alpha_m)^T > 0$ ، $u = (u_1, \dots, u_m)$ و $\beta = (\beta_1, \dots, \beta_l)^T > 0$ است. مدل شبکه

عصبی متناظر با مساله‌ی (۱۴.۲) به‌صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = -\tau \nabla E(y(t)), & \tau > 0 \\ y(t_0) = y_0 \in \mathbb{R}^{n+m+L}. \end{cases} \quad (15.2)$$

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

انتخاب شده برای حل دستگاه دینامیکی (۱۴.۲) است. تابع انرژی متناظر این مدل به‌صورت زیر انتخاب شده است:

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

مدل هفتم

برای طراحی مدل هفتم، مجدداً شرایط کروش-کان-تاکر را برای حل مسائل (۱۲.۲) و (۱۱.۲) به‌صورت زیر در نظر

می‌گیریم:

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

^{۳۲}Strongly smooth

^{۳۳}Nonlinear complementarity problem

آنگاه برای حل (۱۱.۲) دستگاه زیر تعریف می شود:

$$U(x, u, v) = \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}. \quad (۱۶.۲)$$

مدل شبکه عصبی برای حل (۱۱.۲) در [۹] به صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = \kappa U(y), \\ y(t_0) = y_0, \end{cases} \quad (۱۷.۲)$$

که در آن κ نرخ همگرایی (نرخ همگرایی پارامتری است که سرعت همگرایی مدل شبکه عصبی با آن قابل تنظیم است) می باشد.

مثال ۲.۳.۲.

$$\begin{aligned} & \text{minimize } x_1^2 + x_2^2 + x_3^2 \\ & \text{subject to } \begin{cases} 2x_1 + x_2 - 5 \leq 0, \\ x_1 + x_3 - 2 \leq 0, \\ -x_1 + 1 \leq 0, \\ -x_2 + 2 \leq 0, \\ -x_3 \leq 0, \end{cases} \end{aligned}$$

جواب بهینه ۱ $\kappa = 1$, $x = (1, 2, 0)^T$ است. با استفاده از شبکه عصبی (۱۶.۲) نقطه تعادل بدست آمده جواب بهینه مساله می باشد.

مدل هشتم

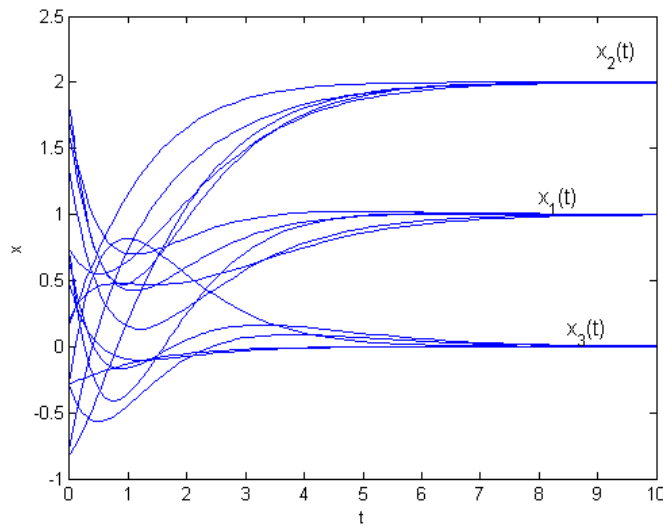
مسأله ی برنامه ریزی درجه دوم (۱۱.۲) را در نظر بگیرید که در آن $Q \in \mathbb{R}^{n \times n}$ یک ماتریس نیمه معین مثبت

است. برای حل مساله (۱۱.۲) در [۵] یک مدل شبکه عصبی به صورت زیر ارائه می شود:

$$\frac{dy}{dt} = \kappa(I + M^T) \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}, \quad (۱۸.۲)$$

که در آن κ نرخ همگرایی است و

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



شکل ۳.۲: مسیر جواب در مثال ۲.۳.۲

مدل نهم

مسأله برنامه‌ریزی درجه دوم زیر را در نظر بگیرید:

$$\text{minimize } f(x) = \frac{1}{2}x^T A x + a^T x \quad (۱۹.۲)$$

$$\text{subject to } \begin{cases} D x \leq b, \\ x \geq 0. \end{cases} \quad (۲۰.۲)$$

دوگان آن عبارت است از:

$$\text{maximize } \bar{f}(w) = b^T w - \frac{1}{2}x^T A x \quad (۲۱.۲)$$

$$\text{subject to } \begin{cases} \bar{D}^T w - A x \leq a, \\ w \geq 0. \end{cases} \quad (۲۲.۲)$$

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

به شرایط شرایط کروش-کان-تاکر برای مسائل برنامه‌ریزی محدب، $(x^{*T}, w^{*T})^T$ به ترتیب یک جواب بهینه برای

مسائل (۱۹.۲) و (۲۱.۲) است اگر و فقط اگر $(x^{*T}, w^{*T})^T$ در شرایط کروش-کان-تاکر زیر صدق کند:

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

مدل متناظر با (۱۹.۲) و (۲۱.۲) در [۸] به صورت زیر بیان شده است:

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

که در آن

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

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

مدل دهم [۲۴]

فرم کلی یک مساله ی برنامه ریزی غیر خطی محدب^{۲۴} را به صورت زیر در نظر بگیرید:

$$\text{minimize } f(x) \quad (23.2)$$

$$\text{subject to } \begin{cases} g(x) \leq \circ, \\ h(x) = \circ, \end{cases} \quad (24.2)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$ یک تابع برداری m -بعدی است. توابع $f, g_1, g_2, \dots, g_m$ محدب و دو بار مشتق پذیر فرض شده اند. $A \in \mathbb{R}^{l \times n}$, $h(x) = Ax - b$. در اینجا فرض می کنیم که مساله (۲۳.۲)-(۲۴.۲) یک جواب بهینه دارد. $b \in \mathbb{R}^l$ و $\text{rank}(A) = l$, $\circ \leq l < n$.

متناظر با مساله (۲۳.۲)-(۲۴.۲) شبکه عصبی زیر را معرفی می کنیم:

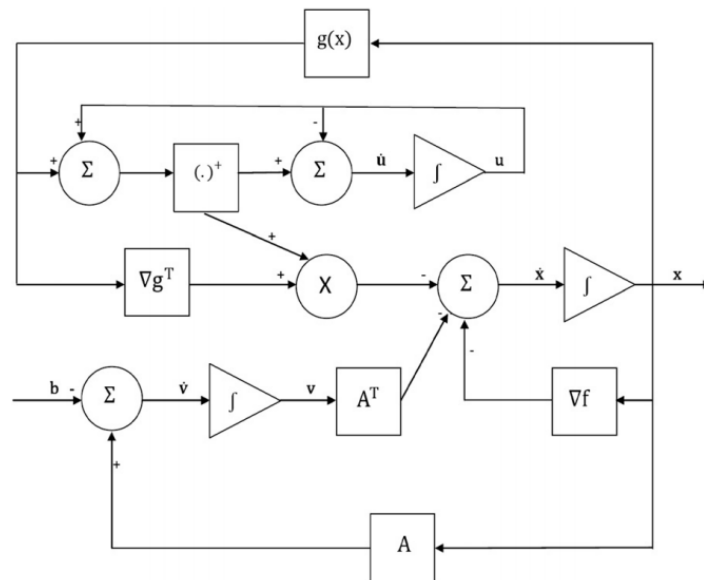
$$\begin{cases} \frac{dx}{dt} = -(\nabla f(x) + \nabla g(x)^T(u + g(x))^+ + \nabla h(x)^T v), \\ \frac{du}{dt} = (u + g(x))^+ - u, \\ \frac{dv}{dt} = h(x), \end{cases} \quad (25.2)$$

با نقطه ی آغازی $y_\circ = (x_\circ^T, u_\circ^T, v_\circ^T)^T$.

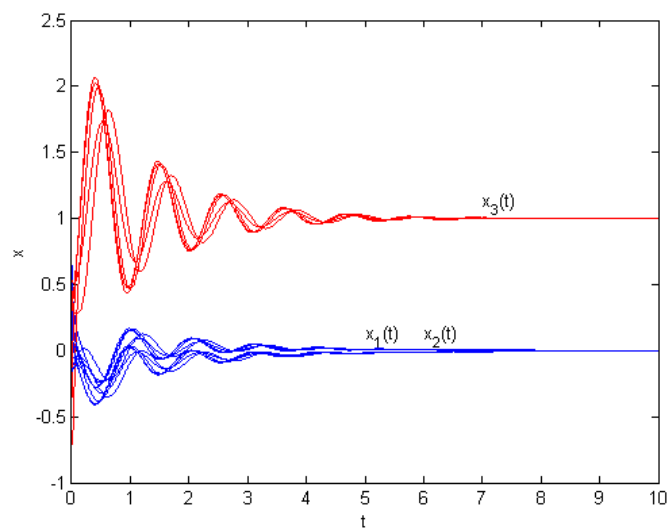
مثال ۳.۳.۲

$$\text{minimize } (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2$$

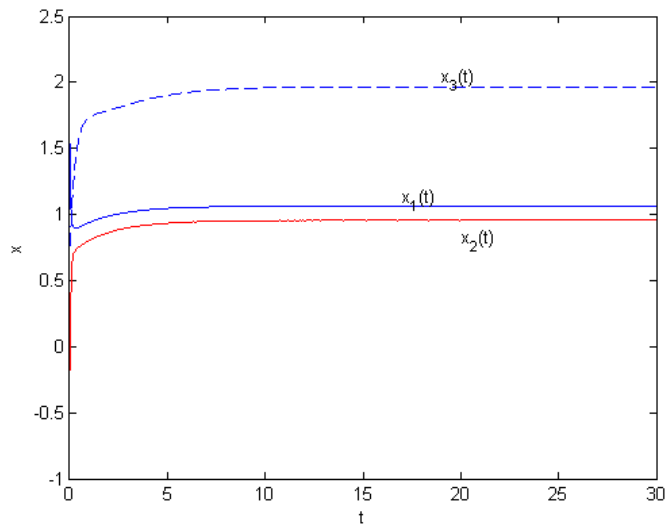
$$\text{subject to } \begin{cases} x_1^3 - 6x_2 - 4x_3 \leq \circ, \\ x_1 + x_2 + x_3 - 1 = \circ, \\ x_1, x_2, x_3 \geq \circ, \end{cases}$$



شکل ۴.۲: دیاگرام مدل شبکه عصبی ۲۵.۲



شکل ۵.۲: مسیر جواب در مثال ۳.۳.۲



شکل ۳.۲: مسیر جواب در مثال ۳.۲.۲

جواب بهینه $x^* = (0, 0, 1)^T$ است. مسیرهای جواب در نمودار (۵.۲) نشان داده شده‌اند.

مثال ۴.۳.۲

$$\begin{aligned} & \text{minimize} \quad \frac{1}{4}((x_1 - x_2)^4 + (x_2 + x_3)^2 + (x_1 + x_3)^2) \\ & \text{subject to} \quad \begin{cases} x_1^2 + x_2^4 - x_3 \leq 0, \\ (2 - x_1)^2 + (2 - x_2)^2 - x_3 \leq 0, \\ 2e^{-x_1 + x_2} - x_3 \leq 0, \\ x_1^2 + x_2^4 - 2x_1 + x_2 - 4 \leq 0, \\ |x_1| \leq 2, |x_2| \leq 2, x_3 \geq 0, \end{cases} \end{aligned}$$

جواب بهینه $x^* = (1/109, 0/922, 1/954)^T$ است. مسیرهای جواب در نمودار (۶.۲) نشان داده شده‌اند.

مدل یازدهم [۲۲]

مساله برنامه‌ریزی غیر خطی محدب^{۳۵} زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (26.2)$$

^{۳۴}General convex nonlinear programming

^{۳۵}Convex nonlinear programming

$$\text{subject to } \begin{cases} g(x) \leq 0, \\ h(x) = 0. \end{cases} \quad (27.2)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$ یک تابع برداری m -بعدی است. توابع $f, g_1, g_2, \dots, g_m$ محدب و دو بار مشتق‌پذیر فرض شده‌اند. $A \in \mathbb{R}^{l \times n}$, $h(x) = Ax - b$.

$b \in \mathbb{R}^l$ و $\text{rank}(A) = l$, $0 \leq l < n$ ابتدا تابع لاگرانژ زیر را در نظر می‌گیریم:

$$L(x, u, v) = f(x) + \frac{1}{2} \sum_{k=1}^m u_k g_k(x) + \sum_{p=1}^L v_p h_p(x)$$

که در آن $u \in \mathbb{R}^m$ و $v \in \mathbb{R}^l$ را ضرایب لاگرانژ می‌نامند. $x^* \in \mathbb{R}^n$ یک جواب بهینه (۲۶.۲) و (۲۷.۲) اگر و فقط

اگر $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ وجود داشته باشد به‌طوری‌که (x^{*T}, u^{*T}, v^{*T}) در شرایط فروش-کان-تاکر زیر صدق

کند:

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

برای حل مسالهی (۲۶.۲) و (۲۷.۲) از سیستم دینامیکی زیر استفاده می‌کنیم:

$$\begin{cases} \frac{dx}{dt} = -\nabla_x L(x, u, v) = -(\frac{\partial f}{\partial x_i} + \frac{1}{2} \sum_{k=1}^m u_k \frac{\partial g_k}{\partial x_i} + \sum_{p=1}^L v_p \frac{\partial h_p}{\partial x_i}), \\ \frac{du}{dt} = \nabla_u L(x, u, v) = \text{diag}(u_1, \dots, u_m) g(x), \\ \frac{dv}{dt} = \nabla_v L(x, u, v) = h_p(x). \end{cases} \quad (28.2)$$

با یک نقطه آغازین $(x_0^T, u_0^T, v_0^T)^T$ و $u_k(t_0) \neq 0$ ($k = 1, 2, \dots, m$) توجه شود که در شبکه عصبی (۲۸.۲)

ضرایب متناظر با قیود نامساوی به صورت $u_k^*(k = 1, \dots, m)$ تعریف می‌شوند. برای سادگی بحث، فرض کنید

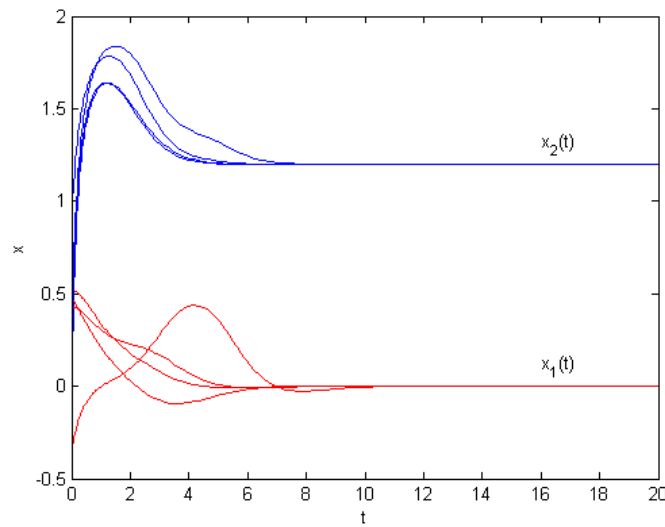
$y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ و D^* جواب بهینه‌ی مسالهی (۲۶.۲)-(۲۷.۲) می‌باشد.

مثال ۵.۳.۲

$$\text{minimize } x_1^2 + x_2^2 + 2x_1x_2 + (x_1 - 1)^4 + (x_2 - 3)^4$$

$$\text{subject to } \begin{cases} x_1^2 + x_2^2 - 64 \leq 0, \\ (x_1 + 3)^2 + (x_2 + 4)^2 - 36 \leq 0, \\ (x_1 - 3)^2 + (x_2 + 4)^2 - 36 \leq 0, \\ x_1 \geq 0, x_2 \geq 0, \end{cases}$$

جواب بهینه $x^* = (0, 1/96)^T$ است و مسیرهای خروجی $x(t)$ در شکل (۲.۷) نشان داده شده است.



شکل ۷.۲: مسیر جواب در مثال ۵.۳.۲

مثال ۶.۳.۲

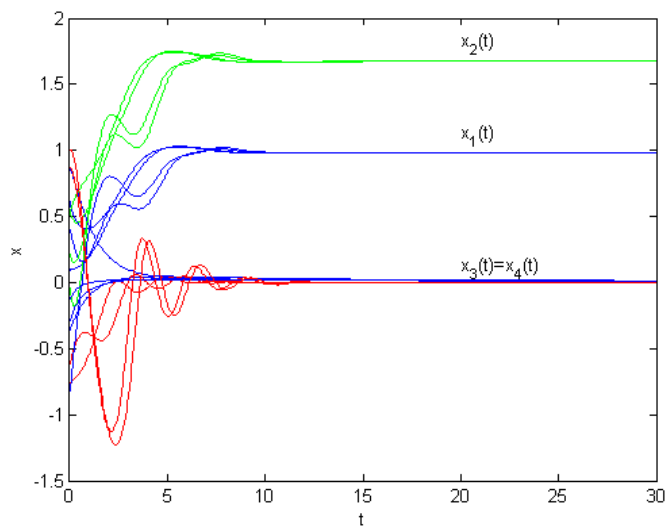
$$\begin{aligned} &\text{minimize } 0.4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{1}{3}x_1^2 \\ &\text{subject to } \begin{cases} -x_1 + x_2 - x_3 \leq 2, \\ 3x_1 + x_2 - x_3 - x_4 \leq 18, \\ \frac{1}{6}x_1 + x_2 - x_4 = 2, \\ x \geq 0, \end{cases} \end{aligned}$$

جواب بهینه $x^* = (0.982, 1.672, 0, 0)^T$ است که همگرایی مسیرها در شکل (۲.۸) نشان داده شده است.

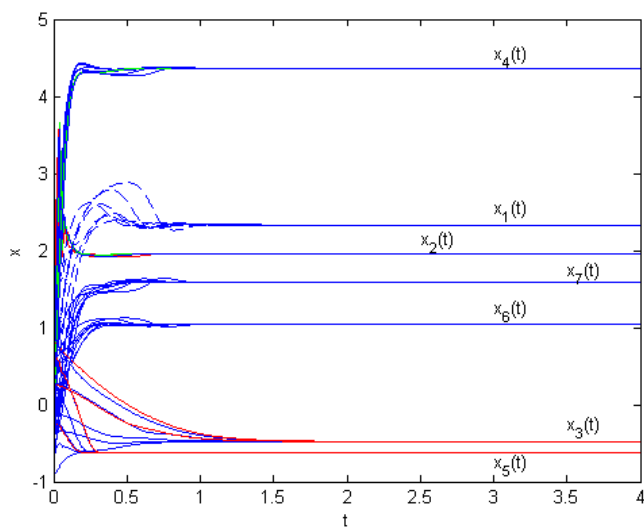
مثال ۷.۳.۲

$$\begin{aligned} &\text{minimize } (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\ &\text{subject to } \begin{cases} 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0, \\ 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0, \\ 23x_1 + x_2^2 + 6x_3^2 - 8x_7 - 196 \leq 0, \\ 4x_1^2 + x_2^2 + 2x_3^2 - 3x_1x_2 + 5x_6 - 11x_7 \leq 0, \end{cases} \end{aligned}$$

نقطه بهینه به صورت $x^* = (2.330, 1.951, -0.4775, 4.366, -0.625, 1.038, 1.594)^T$ می باشد.



شکل ۸.۲: مسیر جواب در مثال ۶.۳.۲



شکل ۹.۲: مسیر جواب در مثال ۷.۳.۲

فصل ۳

کاربرد شبکه عصبی در مساله تخصیص

۱.۳ مقدمه

مساله‌ی تخصیص یک حالت خاص مساله‌ی حمل و نقل است که هدف آن مینیمم سازی همه‌ی هزینه یا ماکزیمم کردن سود است. مساله‌ی تخصیص یک ترکیب بنیادی مساله‌ی بهینه‌سازی در شاخه تحقیق در عملیات ریاضی است. مساله‌ی تخصیص اغلب به مساله‌ی تخصیص خطی و مساله تطابق مشهور است که به کاربردهای آن در طبقه‌بندی الگو، تخصیص کار، طرح‌بندی مکان، زمان‌بندی تولید و طراحی مدار می‌توان اشاره کرد. در این فصل به حل مساله‌ی تخصیص^۱ و مساله‌ی کوتاهترین مسیر^۲ با استفاده از یک مدل شبکه‌ی عصبی می‌پردازیم و در ادامه‌ی فصل نشان داده می‌شود که مدل ارائه شده پایدار مجانبی بوده و نقطه‌ی تعادل شبکه عصبی با جواب بهینه یکی می‌باشد و برای نشان دادن عملکرد سیستم دینامیکی مثال‌هایی از آن را حل می‌کنیم.

۲.۳ مساله تخصیص و تاریخچه مختصر

تا کنون روش‌های زیادی برای حل مساله‌ی تخصیص وجود داشته است که می‌توان چهار روش را نام برد مثل: روش فهرست‌بندی، روش سیمپلکس، روش حمل‌ونقل و روش مجارستانی^۳ که در بین همه‌ی این روش‌ها روش مجارستانی بهترین روش موجود برای حل مساله تخصیص است. ریاضی‌دان مجارستانی کونینگ^۴ روش مجاری تخصیص را که یک روش کافی برای یافتن جواب بهینه بدون مقایسه مستقیم جواب است را گسترش داد.

^۱ Assignment problem

^۲ Shortest path

^۳ Hungarian

^۴ Konig

البته در چهار روش ذکر شده مشکلاتی وجود دارد، از جمله هنگامی که تعداد متغیرها افزایش می‌یابد حل مساله با روش‌های ذکر شده دشوار می‌باشد. با استفاده از مدل شبکه عصبی ارائه شده در این فصل این مشکل از بین می‌رود و در واقع این مدل برای مساله تخصیص با متغیرای زیاد نیز جوابگو است.

۱.۲.۳ یک روش حل مساله تخصیص

همان طور که قبلا توضیح داده شد برای حل مساله تخصیص چهار روش وجود دارد، از بین روش‌های نام برده فقط روش مجارستانی را به اختصار توضیح می‌دهیم.

روش مجارستانی: در این روش ابتدا ماتریس هزینه را در جدولی به صورت زیر قرار می‌دهیم و مراحل زیر را انجام می‌دهیم:

c_{11}	c_{12}	c_{13}	$\dots$	$\dots$	c_{1n}
c_{21}	c_{22}	c_{23}	$\dots$	$\dots$	c_{2n}
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
c_{n1}	c_{n2}	c_{n3}	$\dots$	$\dots$	c_{nn}

۱. مینیمم هزینه هر سطر را یافته و از عناصر آن سطر کم می‌کنیم.

۲. در جدول حاصل مینیمم عناصر هر ستون را یافته و از عناصر آن ستون کم می‌نماییم، به ماتریس حاصل ماتریس تقلیل یافته می‌گوییم.

۳. اگر در هر سطر و ستون دقیقا یک صفر وجود داشته باشد، به جواب بهینه رسیده‌ایم.

۴. مینیمم تعداد خطوط پوشش را پیدا می‌کنیم. (منظور از خطوط پوشش، خطوط افقی یا عمودی (نه مورب) هستند که تمام صفرهای جدول را می‌پوشانند).

۵. در حالت بهینه مینیمم خطوط پوشش با بعد مساله برابر است، در غیر این صورت:

۶. از بین عناصر خط نخورده عنصر مینیمم را یافته این عنصر را از عناصر خط نخورده کم و به عناصر دو بار خط خورده اضافه می‌کنیم و به مرحله (۴) می‌رویم.

۳.۳ مساله تخصیص و فرمول بندی

مساله‌ی تخصیص می‌تواند به صورت برنامه‌ریزی خطی صفر و یک زیر فرمول بندی شود:

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (۱.۳)$$

subject to

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n. \quad (۲.۳)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n. \quad (۳.۳)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, 2, \dots, n. \quad (۴.۳)$$

که در آن c_{ij} و x_{ij} به ترتیب، مقدار هزینه و متغیر تصمیم را نشان می‌دهد. به طور کلی هزینه می‌تواند مثبت یا منفی باشد. هدف مینیمم سازی هزینه کل می‌باشد. قید اول نشان می‌دهد که هر شغل j -ام باید تنها به یک فرد اختصاص یابد. قید دوم نشان دهنده‌ی این است که تنها یک شغل باید به فرد i -ام اختصاص یابد. قید آخر نشان می‌دهد که

$$x_{ij} = \begin{cases} 1 & \text{اگر شغل } j\text{-ام به فرد } i\text{-ام تخصیص داده شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

اگر مساله‌ی تخصیص دارای جواب بهینه‌ی یکتا باشد، آنگاه مساله‌ی تخصیص با یک مساله برنامه‌ریزی خطی به صورت زیر معادل می‌باشد.

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (۵.۳)$$

subject to

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n. \quad (۶.۳)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n. \quad (۷.۳)$$

$$x_{ij} \geq 0 \quad i, j = 1, 2, \dots, n. \quad (۸.۳)$$

در [۱۷] و [۱۵] دوگان مساله تخصیص به فرم زیر ذکر شده است.

$$\text{maximize } \sum_{i=1}^n (u_i + v_j) \quad (9.3)$$

$$\text{subject to } u_i + v_j \leq c_{ij} \quad i, j = 1, 2, \dots, n. \quad (10.3)$$

u_i و v_i متغیرهای تصمیم می‌باشند. تعداد متغیرهای تصمیم و قیدهای نامساوی در مساله‌ی دوگان تخصیص به ترتیب $2n$ و n^2 می‌باشد. با توجه به قضایای اولیه-دوگان برنامه‌ریزی خطی مقدار تابع هدف در مساله‌ی دوگان تخصیص با مقدار تابع هدف در مساله‌ی اولیه آن برابر می‌باشد. چون تعداد قیود در مساله‌ی (۸.۳)-(۵.۳) از تعداد قیود مساله‌ی (۹.۳) و (۱۰.۳) بیشتر است برای یافتن جواب بهینه مساله‌ی اولیه از مساله‌ی دوگان استفاده می‌کنیم

[۳۵]. x_{ij} برای مساله‌ی (۹.۳)-(۱۰.۳) بهینه است اگر و فقط اگر رابطه‌ی زیر برقرار باشد

$$\begin{cases} x_{ij} = 1 & \text{اگر و فقط اگر } z_{ij} = 0 \\ x_{ij} = 0 & \text{اگر و فقط اگر } z_{ij} < 0 \end{cases} \quad (11.3)$$

قرار می‌دهیم: $z_{ij} = u_i + v_j - c_{ij}$ از آنجا که مساله تخصیص به صورت مساله برنامه‌ریزی خطی نوشته شد، بنابراین برای حل آن از یک مدل شبکه عصبی استفاده می‌شود. سپس با استفاده از شرایط مکمل زاید در (۱۱.۳) جواب مساله‌ی تخصیص (۸.۳)-(۵.۳) را پیدا می‌کنیم. ابتدا یک مساله‌ی برنامه‌ریزی خطی را به‌صورت کلی در زیر در نظر می‌گیریم

$$\text{minimize } D^T x \quad (12.3)$$

$$\text{subject to } \begin{cases} Ax - b \leq 0, \\ Ex - f = 0. \end{cases} \quad (13.3)$$

که در آن $\text{rank}(A, E) = m + l$ ($0 \leq m, l < n$) و $x \in \mathbb{R}^n, f \in \mathbb{R}^l, E \in \mathbb{R}^{l \times n}, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$

قضیه ۱.۳.۳. [۱۵] $x^* \in \mathbb{R}^n$ یک جواب بهینه (۱۲.۳)-(۱۳.۳) باشد اگر و فقط اگر $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ موجود

$$\begin{cases} u^* \geq 0, & Ax^* - b \leq 0, & u^{*T}(Ax^* - b) = 0, \\ D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0, \end{cases} \quad (14.3)$$

x^* یک نقطه شرایط کروش-کان-تاکر است و $(u^{*T}, v^{*T})^T$ بردار ضرایب لاگرانژ متناظر x^* نامیده می‌شود.

قضیه ۲.۳.۳. x^* یک جواب بهینه (۱۲.۳)-(۱۳.۳) است اگر و فقط اگر x^* یک نقطه شرایط فروش-کان-تاکر برای (۱۲.۳)-(۱۳.۳) باشد.

فرض کنید $x(\cdot), u(\cdot), v(\cdot)$ متغیرهای وابسته به زمان باشند، هدف ساختن یک سیستم دینامیکی با زمان پیوسته است که در شرایط فروش-کان-تاکر مساله (۱۲.۳)-(۱۳.۳) و دوگان آن صدق می‌کند. برای حل (۱۲.۳)-(۱۳.۳)، از معادله دینامیکی به صورت زیر استفاده می‌کنیم:

$$\frac{dx}{dt} = -(D + A^T(u + Ax - b)^+ + E^T v), \quad (۱۵.۳)$$

$$\frac{du}{dt} = (u + Ax - b)^+ - u, \quad (۱۶.۳)$$

$$\frac{dv}{dt} = Ex - f. \quad (۱۷.۳)$$

با نقطه‌ی آغازین (x_0^T, u_0^T, v_0^T) که

$$(u + Ax - b)^+ = [(u + Ax - b)_1]^+, [(u + Ax - b)_2]^+, \dots, [(u + Ax - b)_m]^+,$$

$y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ را تعریف می‌کنیم. D^* را به عنوان مجموعه‌ی نقطه‌ی بهینه‌ی (۱۲.۳)-(۱۳.۳)

تعریف می‌کنیم و

$$\eta(y) = \begin{pmatrix} -(D + A^T(u + Ax - b)^+ + E^T v) \\ (u + Ax - b)^+ - u \\ Ex - f \end{pmatrix}, \quad (۱۸.۳)$$

بنابراین شبکه عصبی (۱۵.۳)-(۱۷.۳) به صورت زیر نوشته می‌شود:

$$\frac{dy}{dt} = \eta(y), \quad (۱۹.۳)$$

$$y(t_0) = y_0. \quad (۲۰.۳)$$

۴.۳ تحلیل پایداری

قضیه ۱.۴.۳. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})$ نقطه‌ی تعادل شبکه عصبی (۱۹.۳)-(۲۰.۳) باشد، در این

صورت $x^* \in \mathbb{R}^n$ اگر برعکس، است. برای مساله‌ی (۱۲.۳)-(۱۳.۳)

جواب بهینه‌ی (۱۲.۳)-(۱۳.۳) باشد، u^* و v^* وجود دارد به طوری که $y^* = (x^{*T}, u^{*T}, v^{*T})$ یک نقطه‌ی تعادل شبکه‌ی عصبی در (۱۹.۳)-(۲۰.۳) باشد.

برهان. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})$ نقطه‌ی تعادل شبکه‌ی عصبی (۱۹.۳)-(۲۰.۳) باشد. سپس $\frac{dx^*}{dt} = \frac{dv^*}{dt} = 0$ ، یعنی:

$$D + A^T(u^* + Ax^* - b)^+ + E^T v^* = 0, \quad (21.3)$$

$$(u^* + Ax^* - b)^+ - u^* = 0, \quad (22.3)$$

$$Ex^* - f = 0. \quad (23.3)$$

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

$$u^* \geq 0, \quad Ax^* - b \leq 0, \quad u^{*T}(Ax^* - b) = 0. \quad (24.3)$$

با جانشینی (۲۲.۳) در (۲۱.۳) داریم:

$$D + A^T u^* + E^T v^* = 0, \quad (25.3)$$

از (۲۳.۳)-(۲۵.۳) واضح است که $y^* = (x^{*T}, u^{*T}, v^{*T})$ در شرایط فروش-کان-تاکر صدق می‌کند، عکس آن برقرار است. $\square$

لم ۲.۴.۳. برای هر نقطه‌ی اولیه دلخواه $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ در سیستم (۱۹.۳)-(۲۰.۳) یک جواب پیوسته یکتای $y(t) = (x(t)^T, u(t)^T, v(t)^T)^T$ وجود دارد.

برهان. $D \subseteq \mathbb{R}^{n+m+l}$ روی بازه باز و محدب $Ex - f$ و $D + A^T(u + Ax - b)^+ + E^T v, (u + Ax - b)^+ - u$ پیوسته‌ی لیپشیتز موضعی هستند (با توجه به قضیه آنالیز عددی: تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ لیپشیتز است اگر و فقط اگر مشتق آن کراندار باشد). بر طبق یکتایی جواب در معادلات دیفرانسیل معمولی [۲۲]، شبکه‌ی عصبی (۱۹.۳)-(۲۰.۳) یک جواب پیوسته یکتای $y(t), t \in [t_0, \tau)$ برای $\tau > t_0$ که $\tau \rightarrow \infty$ دارد. $\square$

لم ۳.۴.۳. فرض $A \in \mathbb{R}^{m \times n}$ رتبه‌ی کامل باشد. سپس ماتریس ژاکوبین $\nabla \eta(y)$ نگاشت η تعریف شده در (۱۸.۳) یک ماتریس نیمه‌معین منفی است.

برهان. بدون از دست دادن کلیت، فرض می‌کنیم که یک $0 < p < m$ وجود داشته باشد به‌طوری‌که:

$$(u + Ax - b)^+ = ((u + Ax - b)_1, (u + Ax - b)_2, \dots, (u + Ax - b)_p, \underbrace{0, 0, \dots, 0}_{m-p}),$$

با یک محاسبه‌ی ساده نشان داده می‌شود که:

$$\nabla \eta(y) = \begin{pmatrix} -(A^p)^T A^p & -(A^p)^T & -E^T \\ A^p & S_{m \times m} & O_{m \times l} \\ E & O_{l \times m} & O_{l \times l} \end{pmatrix},$$

که O ماتریس صفر را نشان می‌دهد،

$$A^p = \begin{pmatrix} U_{p \times n} \\ O_{(m-p) \times n} \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ \vdots \\ A_p \\ O_{l \times n} \\ O_{l \times n} \\ \vdots \\ \vdots \\ O_{l \times n} \end{pmatrix},$$

9

$$S_{m \times m} = \begin{pmatrix} O_{p \times p} & O_{p \times (m-p)} \\ O_{(m-p) \times p} & -I_{(m-p) \times (m-p)} \end{pmatrix},$$

چون A رتبه‌ی کامل است، p سطر اول ماتریس A^p مستقل خطی هستند، به عبارت دیگر ماتریس U رتبه‌ی کامل است. با استفاده از لم (۴.۴.۱) و این که $(A^p)^T A^p = U^T U$ می‌بینیم که $(A^p)^T A^p$ یک ماتریس معین مثبت است. علاوه بر این واضح است که ماتریس $S_{m \times m}$ ماتریس نیمه معین منفی است. از موارد گفته شده می‌توان

نتیجه گرفت که ماتریس $\nabla \eta(y)$ یک ماتریس نیمه معین منفی است. اگر $p = m$ آنگاه

$$(u + Ax - b)^+ = ((u + Ax - b)_1, \dots, (u + Ax - b)_m).$$

سپس

$$\nabla \eta(y) = \begin{pmatrix} -A^T A & -A^T & -E^T \\ A & O_{m \times m} & O_{m \times l} \\ E & O_{l \times m} & O_{l \times l} \end{pmatrix},$$

مجدد با استفاده از لم (۴.۴.۱) و این که A رتبه‌ی کامل است، به آسانی اثبات می‌شود که $\nabla \eta(y)$ ماتریس نیمه

معین منفی است. سرانجام، اگر $p = 0$ ، آنگاه

$$(u + Ax - b)^+ = \underbrace{(\circ, \circ, \dots, \circ)}_m,$$

بنابراین

$$\nabla \eta(y) = \begin{pmatrix} O_{n \times n} & O_{n \times m} & -E^T \\ O_{m \times n} & -I_{m \times m} & O_{m \times l} \\ E & O_{l \times m} & O_{l \times l} \end{pmatrix},$$

□

واضح است که $\nabla \eta(y)$ ماتریس نیمه معین منفی است و این اثبات را کامل می‌کند.

قضیه ۴.۴.۳. فرض کنیم (۳.۴.۳) برقرار باشد. سپس شبکه‌ی عصبی ارائه داده شده در (۱۹.۳)–(۲۰.۳) پایدار

مجانبی است و همگرای به $y^* = (x^{*T}, u^{*T}, v^{*T})$ است. که جواب بهینه‌ی (۱۲.۳)–(۱۳.۳) است.

برهان. تابع لیاپانف $V : \mathbb{R}^{m+n+l} \rightarrow \mathbb{R}$ را به‌صورت زیر در نظر بگیرید:

$$V(y) = \|\eta(y)\|^2 + \frac{1}{\gamma} \|y - y^*\|^2, \quad (26.3)$$

از (۱۸.۳)، داریم

$$\frac{d\eta}{dt} = \frac{\partial \eta}{\partial y} \frac{dy}{dt} = \nabla \eta(y) \eta(y),$$

با محاسبه‌ی مشتق $V(t)$ داریم:

$$\frac{dV(y(t))}{dt} = \left(\frac{d\eta}{dt}\right)^T \eta + \eta^T \left(\frac{d\eta}{dt}\right) + (y - y^*)^T \frac{dy(t)}{dt} = \eta^T (\nabla \eta(y))^T + \nabla \eta(y) \eta + (y - y^*)^T \eta(y), \quad (27.3)$$

با بکارگیری لم (۳.۴.۳) به دست می‌آوریم:

$$\eta^T(y) (\nabla \eta(y))^T + \nabla \eta(y) \eta(y) \leq 0, \quad \forall y \neq y^*, \quad (28.3)$$

سپس:

$$\frac{dV(y(t))}{dt} \leq 0, \quad (29.3)$$

این به معنی این است که شبکه‌ی عصبی (۱۹.۳)–(۲۰.۳) پایدار مجانبی است، چون:

$$V(y) \geq \frac{1}{\gamma} \|y - y^*\|^2, \quad (30.3)$$

یک زیر دنباله‌ی همگرا به‌صورت زیر وجود دارد

$$\{(x(t_k)^T, u(t_k)^T, v(t_k)^T)^T | t_0 < t_1 < \dots < t_k < t_{k+1}\}, \quad t_k \rightarrow \infty, k \rightarrow \infty$$

به طوری که $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T = \lim_{k \rightarrow \infty} (x(t_k)^T, u(t_k)^T, v(t_k)^T)^T$ که در آن $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$

$$\frac{dV(y(t))}{dt} = 0,$$

نشان می‌دهد که $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ یک ω -حد نقطه‌ای $\{(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T | t \geq t_0\}$ می‌باشد. با استفاده از قضیه‌ی

ناوردای لازال [۲۱] که یک $\{(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T \rightarrow M\}$ دارد که $t \rightarrow \infty$ که M بزرگترین مجموعه‌ی تغییرناپذیر

در $\{(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T | \frac{dV(y(t))}{dt} = 0\}$ است. از (۱۹.۳)-(۲۰.۳) و (۲۹.۳) نشان داده می‌شود که:

$$\frac{dx}{dt} = 0, \quad \frac{du}{dt} = 0, \quad \frac{dv}{dt} = 0 \iff \frac{dV(y(t))}{dt} = 0,$$

سپس: $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T \in D^*$ با $M \subseteq K \subseteq D^*$ با جایگزین کردن $x^* = \bar{x}, u^* = \bar{u}, v^* = \bar{v}$ در (۲۶.۳) تابع

لیاپانف دیگری را تعریف می‌کنیم:

$$\bar{V}(y) = \|\eta(y)\|^2 + \frac{1}{\gamma} \|y - \bar{y}\|^2, \quad (۳۱.۳)$$

سپس $\bar{V}(y)$ به طور پیوسته مشتق‌پذیر و $\bar{V}(\bar{y}) = 0$ توجه شود که:

$$\lim_{k \rightarrow \infty} (x(t_k)^T, u(t_k)^T, v(t_k)^T)^T = \bar{V}(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T,$$

بنابراین داریم: $\lim_{k \rightarrow \infty} \bar{V}(x(t_k)^T, u(t_k)^T, v(t_k)^T)^T = \bar{V}(\bar{x}, \bar{u}, \bar{v})$ از این‌رو:

$$\forall \epsilon > 0, \exists q > 0, \text{ s.t. } \forall t \geq t_q \quad \bar{V}(y(t)) < \epsilon,$$

به طور مشابه می‌توان به دست آورد $\epsilon < \frac{d\bar{V}(y(t))}{dt}$ ، آن نشان می‌دهد که برای $t \geq t_q$

$$\frac{1}{\gamma} \|y(t) - \bar{y}\|^2 \leq \bar{V}(y(t)) \leq \epsilon,$$

استنباط می‌شود که: $\lim_{t \rightarrow \infty} \|y(t) - \bar{y}\| = 0$ و $\lim_{t \rightarrow \infty} y(t) = \bar{y}$. بنابراین شبکه‌ی عصبی ارائه شده

در (۱۹.۳)-(۲۰.۳) همگرای سراسری به نقطه تعادل $\bar{y} = (\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ که $\bar{x}$ جواب بهینه‌ی (۱۲.۳)-(۱۳.۳)

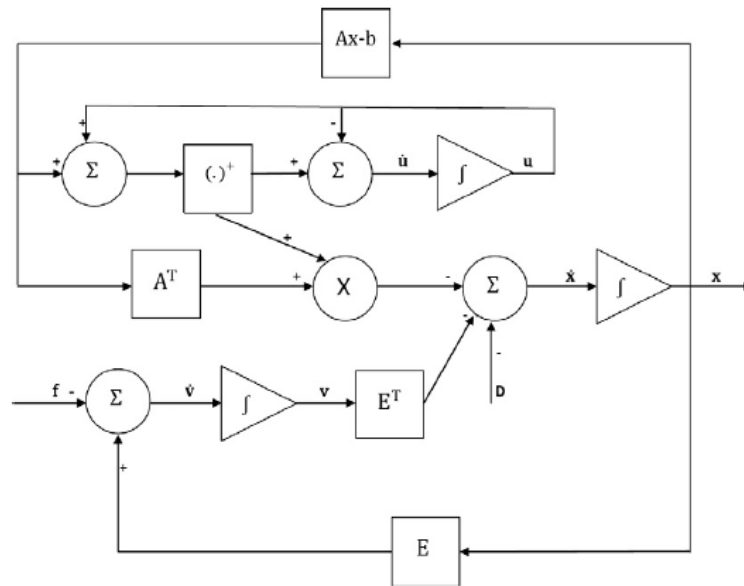
□

است.

نتیجه ۵.۴.۳. فرض کنید که فرض (۳.۴.۳) برقرار باشد. اگر $D^* = \{(x^{*T}, u^{*T}, v^{*T})\}$ ، سپس شبکه‌ی

عصبی (۱۹.۳)-(۲۰.۳) برای حل (۱۲.۳)-(۱۳.۳) به طور مجانبی پایدار به نقطه‌ی تعادل $y^* = (x^{*T}, u^{*T}, v^{*T})$

می‌باشد.

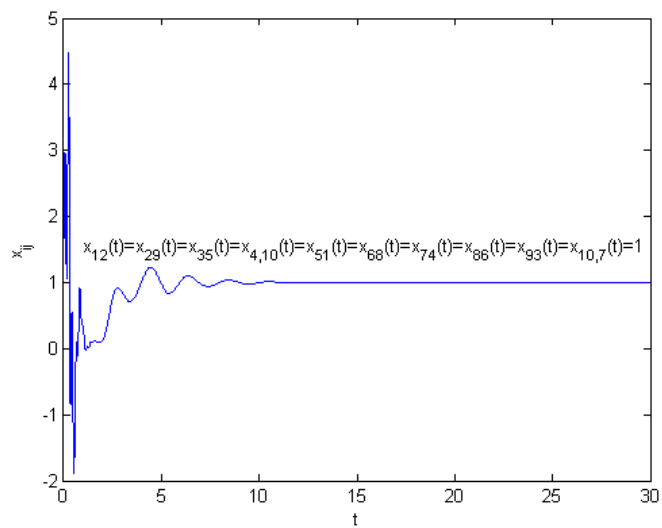


شکل ۱.۳: دیاگرام شبکه‌ی عصبی ارائه شده در ۱۵.۳-۱۷.۳

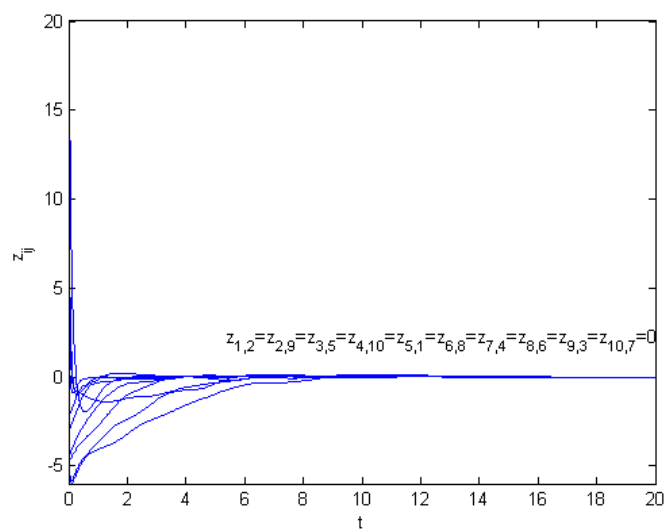
مثال ۶.۴.۳. [۳۲] در مساله تخصیص زیر ماتریس هزینه به صورت زیر می‌باشد:

$$[c_{ij}] = \begin{pmatrix} -13/0 & -11/7 & -10/4 & -9/1 & -7/8 & -6/5 & -5/2 & -3/9 & -2/6 & -1/3 \\ 17/0 & 15/3 & 13/6 & 11/9 & 10/2 & 8/5 & 6/8 & 5/1 & 3/4 & 1/7 \\ 5/0 & 4/5 & 3/0 & 2/5 & 1/0 & 0/5 & 0/0 & 1/5 & 0/0 & 0/5 \\ 22/0 & 19/8 & 17/6 & 15/4 & 13/2 & 11/0 & 8/8 & 6/6 & 4/4 & 2/2 \\ -26/0 & -23/4 & -20/8 & -18/2 & -15/6 & -13/0 & -10/4 & -7/8 & -5/2 & -2/6 \\ 15/0 & 13/5 & 12/0 & 10/5 & 9/0 & 7/5 & 6/0 & 4/5 & 3/0 & 1/5 \\ -6/0 & -5/4 & -4/8 & -4/2 & -3/6 & -3/0 & -2/4 & -1/8 & -1/2 & -0/6 \\ 9/0 & 8/1 & 7/2 & 6/3 & 5/4 & 4/5 & 3/6 & 2/7 & 1/8 & 0/9 \\ -12/0 & -10/8 & -9/6 & -8/4 & -7/2 & -6/0 & -4/8 & -3/6 & -2/4 & -1/2 \\ 11/0 & 9/9 & 8/8 & 7/7 & 6/6 & 5/5 & 4/4 & 3/3 & 2/2 & 1/1 \end{pmatrix}$$

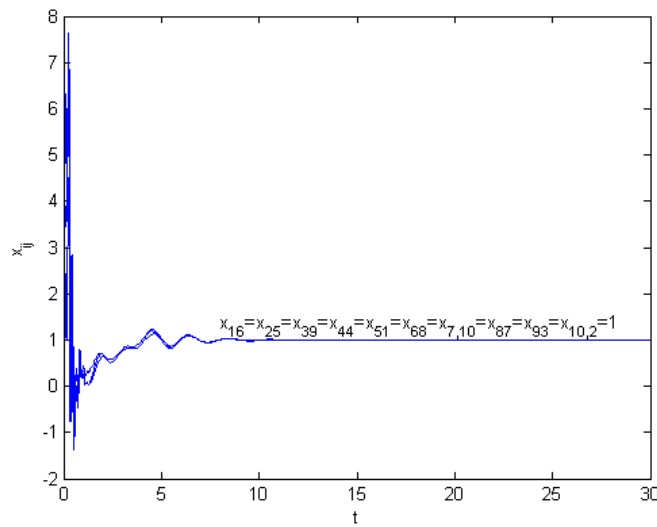
$$[x_{ij}^*] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$



شکل ۲.۳: رفتار x_{ij} در مثال ۶.۴.۳



شکل ۳.۳: رفتار z_{ij} در مثال ۶.۴.۳

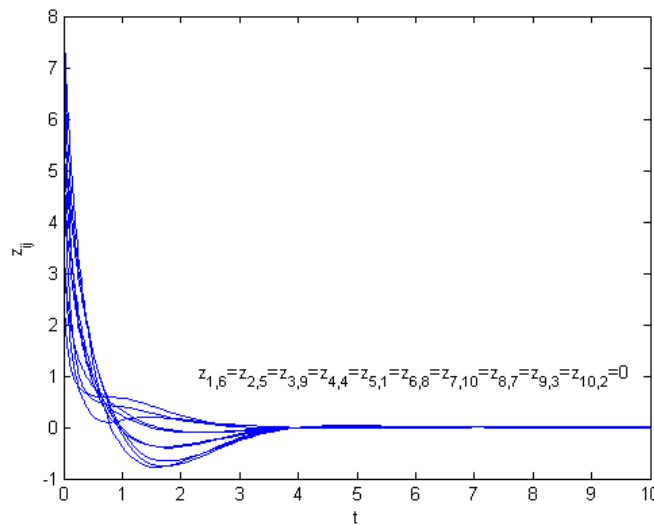


شکل ۴.۳: مسیر اولیه مساله تخصیص در مثال ۷.۴.۳

مثال ۷.۴.۳. [۲۴] ماتریس هزینه مساله تخصیص را به صورت زیر در نظر می گیریم:

$$[c_{ij}] = \begin{pmatrix} 1/6 & 6/7 & -3/6 & 2/8 & 5/6 & -4/1 & 10/5 & 4/2 & -2/8 & 10/9 \\ 1/6 & 9/3 & -5/8 & 5/1 & -4/6 & 10/2 & -1/2 & -2/0 & -2/2 & 8/3 \\ 8/0 & 11/8 & -5/2 & -1/7 & -2/4 & 10/4 & 3/9 & 9/1 & -7/8 & 10/6 \\ 1/4 & 2/1 & 1/2 & -8/4 & 3/2 & -7/7 & 4/3 & -5/7 & 3/5 & -1/8 \\ -4/2 & 4/6 & 7/5 & 0/5 & 1/7 & 7/4 & -3/9 & -2/2 & -5/0 & -2/6 \\ -3/6 & -6/2 & -2/4 & -2/1 & -5/0 & -4/9 & -3/6 & -6/0 & -2/1 & -2/7 \\ 5/3 & 1/0 & -3/5 & 5/5 & 3/4 & 8/3 & 9/1 & 0/5 & 4/6 & -2/7 \\ 5/5 & 2/5 & 10/2 & 10/8 & 1/6 & 8/5 & -5/8 & -4/3 & -2/9 & -3/8 \\ 10/9 & -4/6 & -7/8 & -1/2 & 2/5 & -4/0 & 0/8 & 8/9 & -5/1 & -3/6 \\ 7/4 & -6/6 & 3/7 & 3/3 & 7/8 & 5/5 & 6/0 & 8/7 & 5/1 & -1/2 \end{pmatrix}$$

$$[x_{ij}^*] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



شکل ۵.۳: رفتار z_{ij} در مثال ۷.۴.۳

۵.۳ کوتاهترین مسیر

مسالهای کوتاهترین مسیر عبارتست از یافتن کوتاهترین مسیر از یک ناحیه به ناحیه دیگر در حالی که هزینه کل مینیمم شود. یک گراف $G = (V, E)$ که در آن V مجموعه‌ی رئوس و E یال‌های متناظر با آن است را در نظر بگیرید. یک هزینه‌ی ثابت متناظر با یال‌ها از راس i به راس j در گراف G می‌باشد. در یک سیستم رباتیک یا حمل و نقل هزینه می‌تواند فاصله بین رئوس باشد، برای یک سفر هزینه می‌تواند زمان یا انرژی بین رئوس باشد، در یک سیستم ارتباطی هزینه می‌تواند از یک راس به راس دیگر بر طبق ارسال زمان و ظرفیت ارتباط تعیین شود. هزینه‌ی از راس i به راس j ممکن است با هزینه‌ی از راس j به i برابر نباشد، بنابراین ممکن است بین بعضی از راس‌ها یال موجود نباشد. اگر از راسی به راس دیگر یالی موجود نباشد مقدار هزینه را نامتناهی در نظر می‌گیریم. در کل هزینه می‌تواند منفی یا مثبت باشد. مقدار هزینه‌ی مثبت نشانگر یک ضرر است، حال آنکه برعکس هزینه منفی سود را نشان می‌دهد. تعیین کوتاهترین مسیر با ضرایب مثبت و منفی مشکل است، لذا فرض می‌کنیم که در شبکه هیچ دور و حلقه‌ی منفی وجود ندارد. از این رو در یک شبکه راس‌ها به طور قراردادی برچسب‌گذاری می‌شوند. در این قسمت فاصله را از راس ۱ به n در نظر می‌گیریم. از الگوریتم کوتاهترین مسیر در طیف گسترده‌ای از مسائل دنیا استفاده می‌شود، برای مثال در یک شبکه جاده، رئوس می‌توانند چهارراه باشند و خیابان‌ها معرف یال باشند

و فاصله بین راس‌ها مقدار هزینه باشد. برای راحتی کوتاهترین مسیر از راس ۱ به راس n را مسیر مورد نظر فرض می‌کنیم. از کاربردهای آن می‌توان در نقشه‌ی دریانوردی، برنامه‌ی سفر پرواز، مسیر سیم‌کشی ساختمان، کابل‌کشی و مسیریابی شبکه یاد کرد. به طور ریاضی کوتاهترین مسیر را می‌توان به صورت زیر فرمول‌بندی کرد:

$$\text{minimize } \sum_{i=1}^n \sum_{j \neq i=1}^n c_{ij} x_{ij} \quad (32.3)$$

subject to

$$\sum_{k=1, k \neq i}^n x_{ik} - \sum_{l=1, l \neq i}^n x_{li} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i = 2, 3, \dots, n-1 \\ -1 & \text{if } i = n \end{cases} \quad (33.3)$$

$$x_{ij} \in \{0, 1\} \quad i \neq j, \quad i, j = 1, 2, \dots, n \quad (34.3)$$

(۳۲.۳) بیانگر مجموع مینیمم هزینه است، سمت راست (۳۳.۳) که ۱، ۰ یا -۱ است، نشان می‌دهد که مقدار سمت

چپ (۳۳.۳) برای راس اول آن ۱ است برای رئوس میانی صفر و برای راس انتهایی -۱ می‌باشد، x_{ij} متغیرهای

تصمیم متناظر با یال‌ها از راس i به راس j را به صورت زیر تعریف می‌کنیم:

$$x_{ij} = \begin{cases} 1 & \text{اگر از راس } i \text{ به راس } j \text{ یالی موجود باشد.} \\ 0 & \text{در غیر اینصورت} \end{cases}$$

اگر مساله‌ی کوتاهترین مسیر جواب یکتا داشته باشد (۳۴.۳) را می‌توان با متغیرهای تصمیم نامنفی $x_{ij} \geq 0$

جایگزین کرد [۳۳]، به عبارت دیگر مساله‌ی (۳۲.۳)-(۳۴.۳) معادل برنامه‌ریزی خطی زیر است.

$$\text{minimize } \sum_{i=1}^n \sum_{j=2, j \neq i}^n c_{ij} x_{ij}$$

subject to

$$\sum_{k=1, k \neq i}^{n-1} x_{ik} - \sum_{l=2, l \neq i}^n x_{li} = \delta_{i1} - \delta_{in}, \quad i = 1, 2, \dots, n$$

$$x_{ij} \geq 0, \quad i \neq j \quad i = 1, 2, \dots, n-1, \quad j = 2, 3, \dots, n$$

δ_{ij} تابع دلتای کرونکر است که به صورت $\delta_{ij} = 1 (i = j)$ و $\delta_{ij} = 0 (i \neq j)$ روی مسیر یال‌ها تعریف می‌شود.

دوگان مساله کوتاهترین مساله به صورت زیر است:

$$\text{maximize } y_n - y_1$$

subject to

$$y_j - y_i \leq c_{ij} \quad i \neq j, \quad i, j = 1, 2, \dots, n$$

y_i متغیرهای تصمیم دوگان را نشان می‌دهد. $y_i - y_1$ ، کوتاهترین فاصله از راس ۱ به راس i می‌باشد. x_{ij} جواب بهینه است اگر و فقط اگر:

$$\begin{cases} x_{ij} = 1 & \text{اگر و فقط اگر } z_{ij} = 0 \\ x_{ij} = 0 & \text{اگر و فقط اگر } z_{ij} < 0 \end{cases} \quad i, j = 1, 2, \dots, n. \quad (35.3)$$

که در آن $z_{ij} = y_j - y_i - c_{ij}$ قرار می‌دهیم. از آنجا که مساله کوتاهترین مسیر را می‌توان بصورت مساله برنامه‌ریزی خطی نوشت لذا برای حل آن می‌توان از مدل شبکه‌ای عصبی (۱۵.۳)–(۱۷.۳) استفاده کرد.

۱.۵.۳ ارتباط بین مساله‌ی تخصیص و کوتاهترین مسیر [۲۳]

در مساله کوتاهترین مسیر $n + 1$ راس را می‌توان بدون در نظر گرفتن دور منفی به عنوان n -مؤلفه‌ی مساله‌ی تخصیص با n^2 متغیر تصمیم در نظر گرفت. یک دور منفی شامل مجموع هزینه منفی در یک مدار بسته در گراف G می‌باشد. یعنی:

$$\exists \quad i, j, k, \dots, l \quad \text{subject to} \quad d_{ij} + d_{jk} + \dots + d_{li} < 0,$$

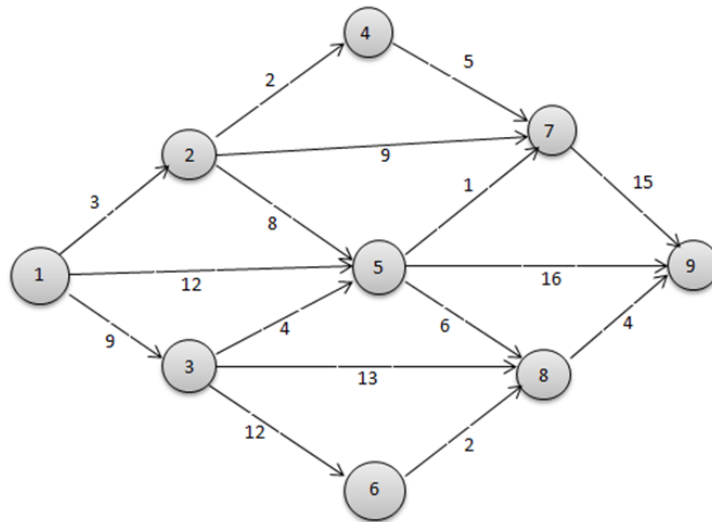
ماتریس هزینه مساله‌ی تخصیص برای یافتن کوتاهترین مسیر در الگوریتم کوتاهترین مسیر به صورت زیر تعریف می‌شود:

$$c_{ij} = d_{ii+1}, \quad c_{i+1,i} \equiv 0 \quad \text{for } i, j = 1, 2, \dots, n.$$

شرط لازم برای اینکه مساله جواب داشته باشد این است که در گراف G دور منفی وجود نداشته باشد.

مثال ۱.۵.۳. [۷]

$$\begin{aligned} \text{minimize } & 3x_{12} + 9x_{13} + 12x_{15} + 2x_{24} + 9x_{27} + 8x_{25} + 4x_{35} \\ & + 13x_{38} + 12x_{36} + 5x_{47} + x_{57} + 16x_{59} + 6x_{58} + 15x_{79} + 4x_{89} \end{aligned}$$



شکل ۶.۳: گراف مثال ۱.۵.۳

$$\text{subject to } \begin{cases} x_{12} + x_{13} + x_{15} = 1, \\ x_{24} + x_{27} + x_{25} - x_{12} = 0, \\ x_{35} + x_{38} + x_{36} - x_{13} = 0, \\ x_{47} - x_{24} = 0, \\ x_{57} + x_{59} + x_{58} - x_{15} - x_{25} - x_{35} = 0, \\ x_{68} - x_{36} = 0, \\ x_{79} - x_{57} - x_{27} - x_{47} = 0, \\ x_{89} - x_{38} - x_{58} - x_{68} = 0, \\ -x_{89} - x_{59} - x_{79} = -1, x_{ij} \geq 0, \end{cases}$$

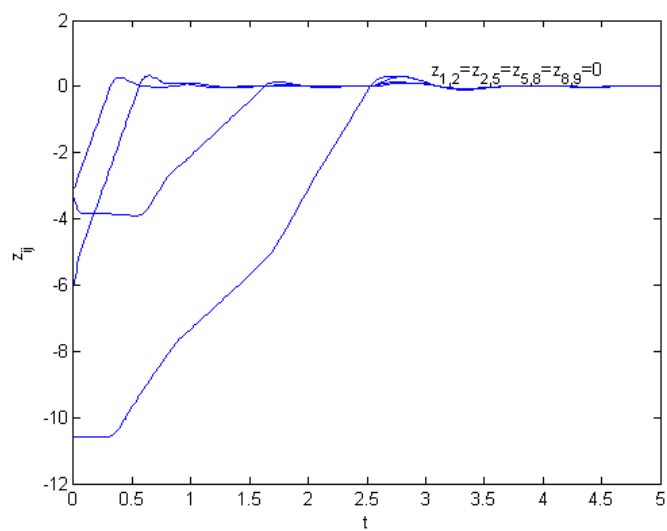
جواب بهینه ۱ $x_{12} = x_{25} = x_{58} = x_{89} = 1$ می‌باشد. مسیر z_{ij} و هزینه به ترتیب در شکل (۷-۳) و (۸-۳)

نشان داده شده است.

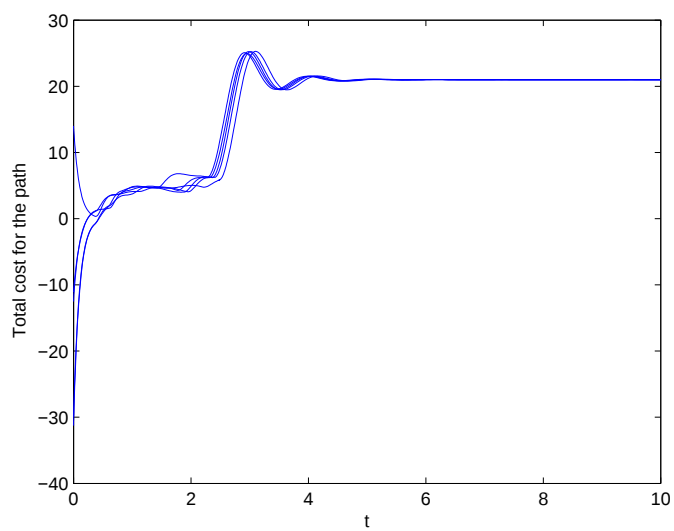
مثال ۲.۵.۳. [۷]

$$\text{minimize } 2x_{12} - x_{13} - 4x_{23} + 3x_{24} - 6x_{34}$$

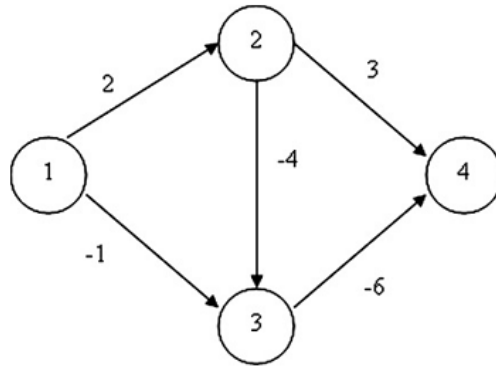
$$\text{subject to } \begin{cases} x_{12} + x_{13} = 1, \\ x_{23} + x_{24} - x_{12} = 0, \\ x_{34} - x_{13} - x_{23} = 0, \\ -x_{24} - x_{34} = -1, \\ x_{ij} \geq 0, \end{cases}$$



شکل ۷.۳: رفتار z_{ij} در مثال ۱.۵.۳



شکل ۸.۳: مسیر هزینه در مثال ۱.۵.۳



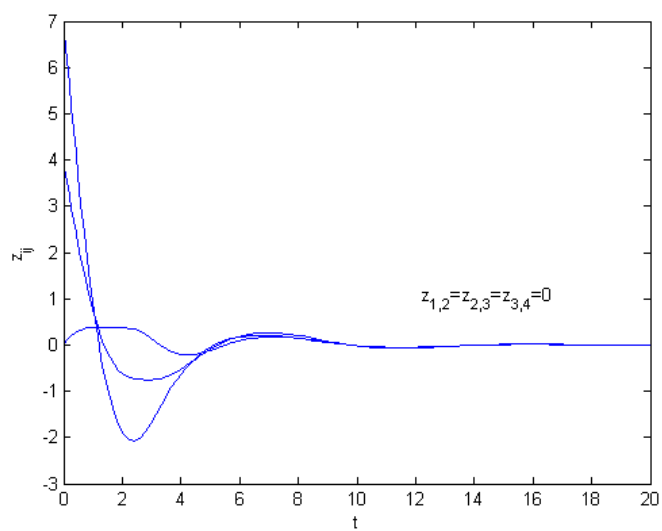
شکل ۹.۳: گراف مثال ۲.۵.۳

جواب بهینه $x_{12} = x_{23} = x_{34} = 1$ می‌باشد. مسیر z_{ij} و هزینه به ترتیب در شکل (۳-۱۰) و (۳-۱۱) نشان داده شده است.

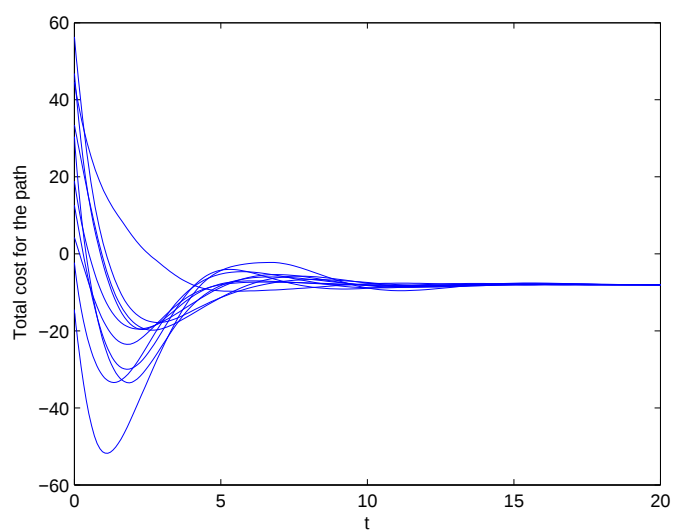
مثال ۳.۵.۳.

$$\text{minimize } 2x_{12} + x_{14} + 3x_{24} + 10x_{25} + 4x_{31} + 5x_{36}$$

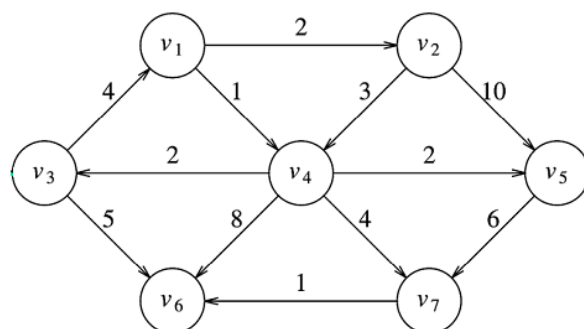
$$\begin{aligned} & 2x_{43} + 8x_{46} + 4x_{47} + 2x_{45} + 6x_{57} + x_{76} \\ \text{subject to } & \begin{cases} x_{12} + x_{14} - x_{31} = 1, \\ x_{24} + x_{25} - x_{12} = 0, \\ x_{31} + x_{36} - x_{43} = 0, \\ x_{43} + x_{46} + x_{47} + x_{45} - x_{14} - x_{24} = 0, \\ x_{57} - x_{25} - x_{45} = 0, \\ -x_{36} - x_{46} - x_{76} = 0, \\ x_{76} - x_{47} - x_{57} = -1, \\ x_{ij} \geq 0, \end{cases} \end{aligned}$$



شکل ۱۰.۳: رفتار z_{ij} در مثال ۲.۵.۳



شکل ۱۱.۳: مسیر هزینه در مثال ۲.۵.۳



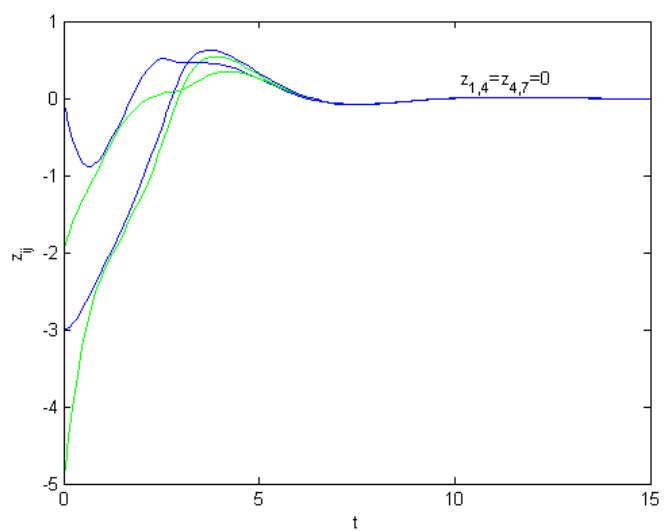
شکل ۱۲.۳: گراف مثال ۳.۵.۳

جواب بهینه $x_{14} = x_{47} = 1$ می‌باشد. مسیر z_{ij} و هزینه به ترتیب در شکل (۱۳-۳) و (۱۴-۳) نشان داده شده است.

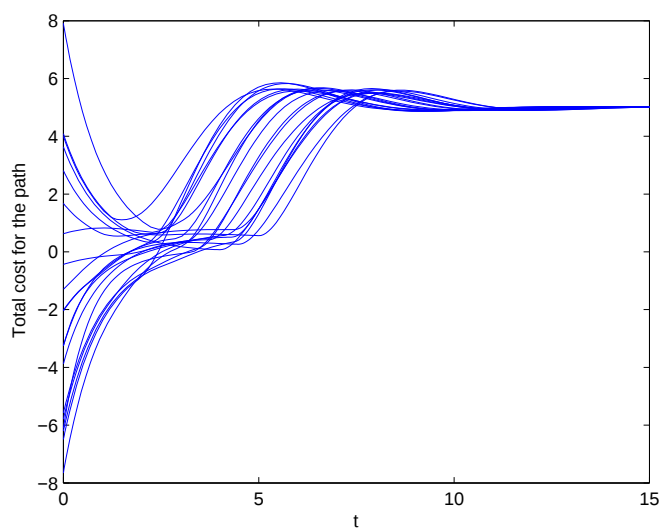
مثال ۴.۵.۳

$$\begin{aligned} &\text{minimize } 9x_{12} + 14x_{16} + 15x_{17} + 24x_{23} + 6x_{34} + 2x_{35} + 19x_{38} \\ &\quad + 6x_{48} + 11x_{54} + 16x_{58} + 18x_{63} + 30x_{65} + 5x_{67} + 20x_{75} + 44x_{78} \\ &\text{subject to } \begin{cases} x_{12} + x_{16} + x_{17} = 1, \\ x_{23} - x_{12} = 0, \\ x_{38} + x_{34} + x_{35} - x_{23} - x_{63} = 0, \\ x_{48} - x_{43} - x_{54} = 0, \\ x_{54} + x_{58} - x_{35} - x_{65} - x_{75} = 0, \\ x_{63} + x_{65} + x_{67} - x_{16} = 0, \\ x_{75} + x_{78} - x_{17} - x_{67} = 0, \\ -x_{38} - x_{48} - x_{58} - x_{78} = -1, x_{ij} \geq 0, \end{cases} \end{aligned}$$

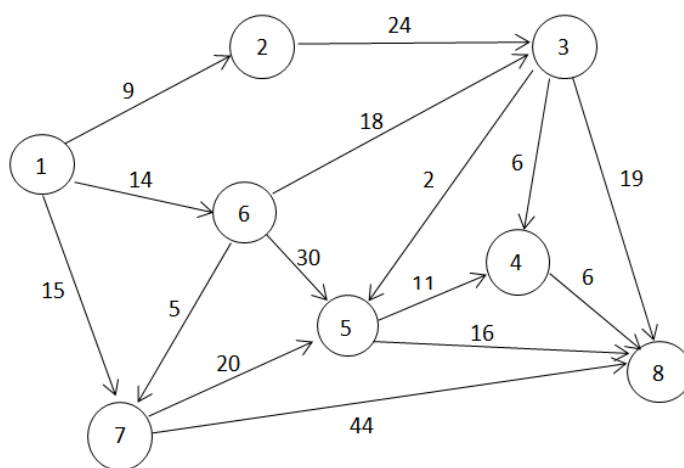
کوتاهترین مسیر $x_{16} = x_{63} = x_{35} = x_{58} = 1$ می‌باشد. مسیر z_{ij} و هزینه به ترتیب در شکل (۱۶-۳) و (۱۷-۳) نشان داده شده است.



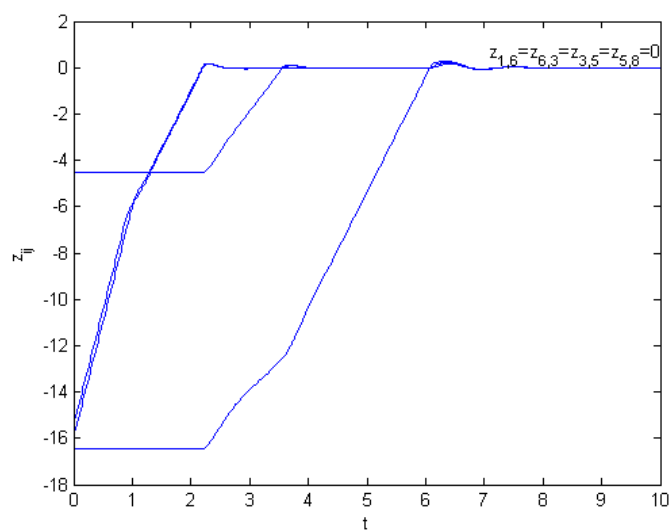
شکل ۱۳.۳: رفتار z_{ij} در مثال ۳.۵.۳



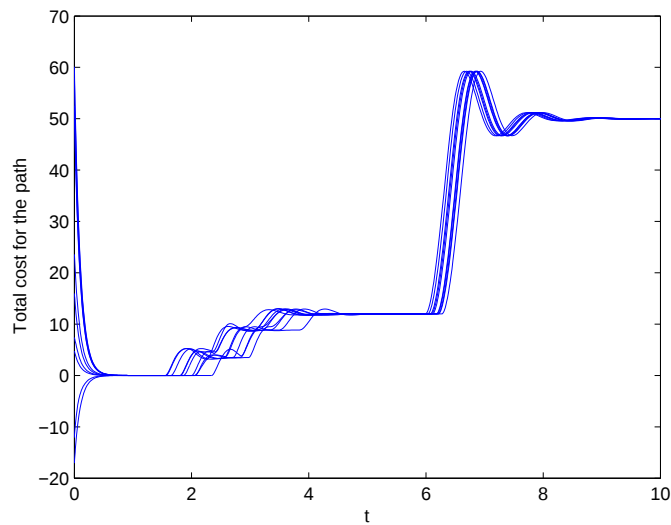
شکل ۱۴.۳: مسیر هزینه در مثال ۳.۵.۳



شکل ۱۵.۳: گراف مثال ۴.۵.۳



شکل ۱۶.۳: رفتار z_{ij} در مثال ۴.۵.۳

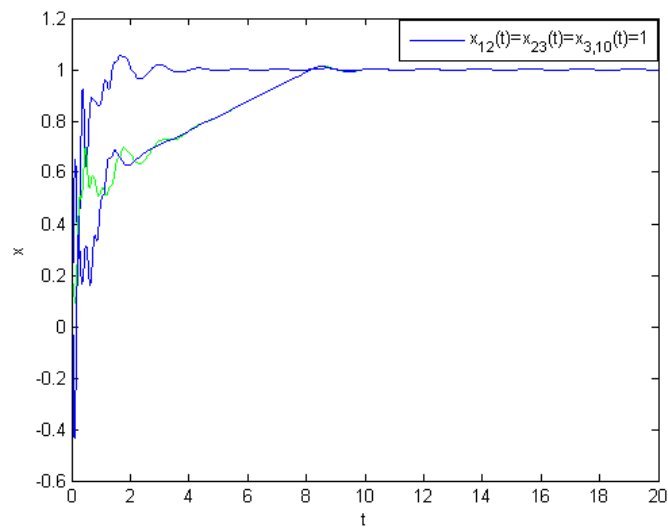


شکل ۱۷.۳: مسیر هزینه در ۴.۵.۳

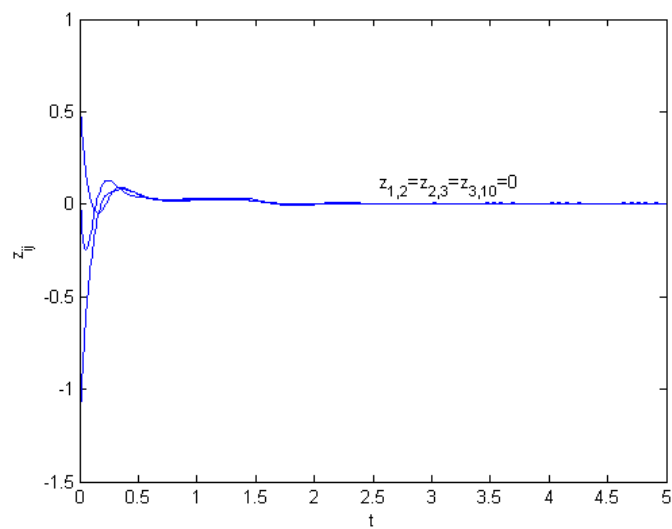
مثال ۵.۵.۳. [۱۳]. در این مثال کوتاهترین مسیر با ده راس داده شده است، یال‌هایی بین رئوس $a_{۱۲}, a_{۱۶}, a_{۱۷}$ ،
 $a_{۲۳}, a_{۲۴}, a_{۲۶}, a_{۲۷}, a_{۲۸}, a_{۳۴}, a_{۳۶}, a_{۳۸}, a_{۳,۱۰}, a_{۴۶}, a_{۴۸}, a_{۴,۱۰}, a_{۵۷}, a_{۵۹}, a_{۵,۱۰}, a_{۶۷}, a_{۶۸}, a_{۶۹}, a_{۷۹}, a_{۷,۱۰}$ ،
 $a_{۸,۱۰}, a_{۹,۱۰}$ وجود دارد. مختصات هر راس در جدول (۳۶.۳) داده شده است و فاصله‌ی اقلیدسی به عنوان ضرایب
 هزینه استفاده شده است. کوتاهترین مسیر این مساله $a_{۱۲}, a_{۲۳}, a_{۳,۱۰}$ می‌باشد که در شکل (۳-۱۸) نشان داده
 شده است. مسیر z_{ij} و هزینه به ترتیب در شکل (۳-۱۹) و (۳-۲۰) نشان داده شده است.

i	۱	۲	۳	۴	۵
x_i	۰/۰۱۰۵۵۹	۰/۳۳۵۱۵۴	۰/۳۵۵۷۲۴	۰/۵۳۶۹۷۳	۰/۷۰۰۳۳۹
y_i	۰/۰۰۳۹۶۷	۰/۰۳۳۲۶۵	۰/۲۱۷۲۰۰	۰/۱۹۵۷۷۵	۰/۹۴۹۹۱۹
i	۶	۷	۸	۹	۱۰
x_i	۰/۲۷۴۷۸۹	۰/۱۰۸۹۸۲	۰/۵۶۴۳۴۸	۰/۱۶۵۱۶۶	۰/۶۸۵۵۳۷
y_i	۰/۴۴۴۲۸۸	۰/۶۹۸۲۳۳	۰/۰۴۱۵۰۵	۰/۸۱۵۴۸۵	۰/۷۶۴۳۹۷

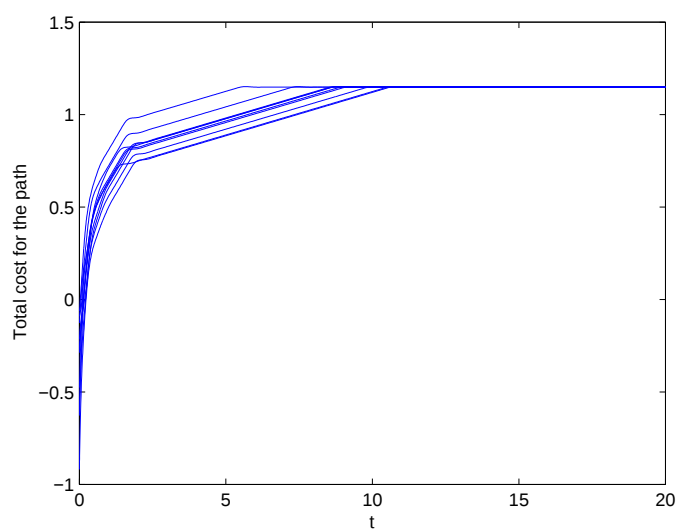
(۳۶.۳)



شکل ۱۸.۳: مسیر اولیه در مثال ۵.۵.۳



شکل ۱۹.۳: رفتار z_{ij} در مثال ۵.۵.۳



شکل ۲۰.۳: مسیر هزینه در مثال ۵.۵.۳

فصل ۴

استفاده از یک مدل شبکه عصبی برای مسائلی که دچار دور افتادگی می‌شوند

۱.۴ مقدمه

در این فصل با استفاده از یک مدل شبکه عصبی که در فصل سوم ارائه شد به حل مسائلی می‌پردازیم که با روش سیمپلکس دچار دور افتادگی می‌شوند و سپس مثال‌هایی را برای کارایی مدل ارائه شده حل می‌نماییم.

۲.۴ مساله دور

تعریف ۱.۲.۴. سیستم $Ax = b$ که $x \geq 0$ را در نظر بگیرید که در آن A یک ماتریس $m \times n$ و b یک m بردار است. فرض $\text{rank}(A, b) = \text{rank} A = m$. بعد از تغییر ترتیب ستون‌های A در صورت لزوم، فرض کنید $A = [B, N]$ که در آن B یک ماتریس معکوس‌پذیر $m \times m$ و N یک ماتریس $m \times (n - m)$ است. جواب $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ معادلات $Ax = b$ را که در آن :

$$\begin{cases} x_B = b^{-1}B, \\ x_N = 0, \end{cases}$$

جواب پایه سیستم گویند. اگر $x_B \geq 0$ ، آن وقت x جواب شدنی پایه سیستم می‌گویند. B را ماتریس پایه نامند و N را ماتریس غیر پایه می‌نامند. مولفه‌های x_B متغیرهای پایه یا متغیرهای وابسته می‌گویند و مولفه‌های x_N را متغیرهای غیر پایه یا متغیرهای مستقل می‌نامند. اگر $x_B > 0$ آن وقت x جواب شدنی پایه ناتبه‌ایده گفته می‌شود. اگر حداقل یکی از مولفه‌های x_B صفر باشد، آن وقت x جواب شدنی پایه‌ی تباه‌ایده نامیده می‌شود.

برای توضیح اینکه چگونه یک مساله‌ی برنامه‌ریزی خطی در روش سیمپلکس به دور می‌افتد می‌توان به [۱۵] رجوع کرد. با وجودی که دور افتادگی با احتمال کمی واقع می‌شود و در واقع فرمول‌بندی مساله‌ای که دوری شود کار آسانی نیست، با این حال برای اینکه در یک مساله برنامه‌ریزی خطی دور به وجود نیاید روش‌هایی به وجود آمده است. دو قاعده الفبایی^۱ یا آشفته‌گی و قاعده بلاند^۲ از قواعدی هستند که مانع دورافتادگی می‌شوند. با اینکه معلوم شده بیشتر مسائل جهان واقعی تباهیده هستند و این دو قاعده تضمین می‌کند که دور اتفاق نیفتد، ولی این قواعد به طور چشم‌گیری در بیشتر کدهای اقتصادی در حل مسائل برنامه‌ریزی خطی از طریق سیمپلکس نادیده گرفته می‌شوند. دو دلیل عمده وجود دارد، اول اینکه این قواعد از نظر محاسباتی در اجرا گران هستند، نظیر قاعده الفبایی، یا از نظر محاسباتی نسبت به طول مسیر سیمپلکس تولید شده ناکارا هستند، نظیر قاعده بلاند. دوم، به علت خطای گرد کردن کامپیوتر معمولاً استدلال می‌شود که طرف راست به هنگام در هر حالت با مقادیر واقعی‌شان اختلاف دارند و به ندرت سمت راست دقیقاً صفر می‌شود. نکته دیگری که نیاز به توضیح دارد، اگرچه دو قاعده نام برده شده از دورافتادگی جلوگیری می‌کند، اما به طور حتم این امکان وجود دارد که الگوریتم سیمپلکس تعداد زیادی (ولو متناهی) محورگیری تباهیده انجام دهد.

۱.۲.۴ حقایقی در باره مسائل برنامه‌ریزی خطی با جنبه دوری [۲۸]

فرم مساله‌ی برنامه‌ریزی خطی زیر را در نظر بگیرید.

$$\text{minimize } cy$$

subject to

$$Ix + Ay = b, \quad (1.4)$$

$$(x, y) \geq 0, \quad (2.4)$$

که در آن I ماتریس همانی $m \times m$ ، A یک ماتریس $(m \times (n - m))$ که $(m < n)$ ، x و y بردارهای ستونی هستند، b یک بردار ستونی و c یک بردار سطری است. با استفاده از سیمپلکس اولیه برای حل مساله (۲.۴)–(۱.۴)، برای اینکه دور اتفاق بیفتد باید $m \geq 2$ ، $m \geq m + 3$ و $n \geq 6$ برای اینکه دور در نقطه‌ای که بهینه نیست

^۱Alphabetic

^۲Bland

اتفاق بیفتد باید $m \geq 3$, $n \geq m + 3$ و $n \geq 7$ باشد [۲۱]. یک مساله برنامه‌ریزی خطی فقط با دو متغیر غیر پایه نمی‌تواند دوری باشد. یک مثال دوری باید در حداقل دو معادله و حداقل سه متغیر غیر پایه حداقل شش متغیر داشته باشد. یک جواب شدنی پایه با یک تباهیده نمی‌تواند دور ایجاد کند. مینیمم طول یک دور شش می‌باشد [۲۹، ۳۶].

مساله هافمن [۶]

در سال [۱۹۵۱]، اولین مساله‌ی برنامه‌ریزی خطی که تحت قواعد سیمپلکس به دور افتاد توسط هافمن^۳ نشان داده شد. بدین گونه نشان می‌دهد که روش نادروری برای اینکه روش سیمپلکس به عنوان یک الگوریتم که همیشه یک جواب برای مساله می‌دهد لازم می‌باشد و بعد از یک تعداد گام متناهی با یک جواب شدنی یا بهینه‌ی کران‌دار یا جواب بهینه‌ی بیکران متوقف می‌شود. مساله‌ی هافمن به صورت زیر می‌باشد:

$$\begin{aligned} & \text{minimize } [(\cos \varphi - 1)/\cos \varphi]x_4 + wx_5 + 2wx_7 + 4\sin^2 \varphi x_8 + w(2 - 4\cos^2 \varphi)x_9 + 4\sin^2 \varphi x_{10} \\ & \quad + w(1 - 2\cos \varphi)x_{11} \\ & \text{subject to } \begin{cases} x_1 = 1 \\ x_2 + \cos \varphi x_4 - w \cos \varphi x_5 + \cos 2\varphi x_6 - 2w \cos^2 \varphi x_7 + \\ \cos 2\varphi x_8 + 2w \cos^2 \varphi x_9 + \cos \varphi x_{10} + w \cos \varphi x_{11} = 0 \\ x_3 + [(\tan \varphi \sin \varphi)/w]x_4 + \cos \varphi x_5 + [(\tan \varphi \sin 2\varphi)/w]x_6 \\ + \cos 2\varphi x_7 - [2\sin^2 \varphi/w]x_8 + \cos 2\varphi x_9 - [(\tan \varphi \sin \varphi)/w]x_{10} \\ + \cos \varphi x_{11} = 0 \\ x_j \geq 0 \text{ for } j = 1, \dots, 11 \end{cases} \end{aligned}$$

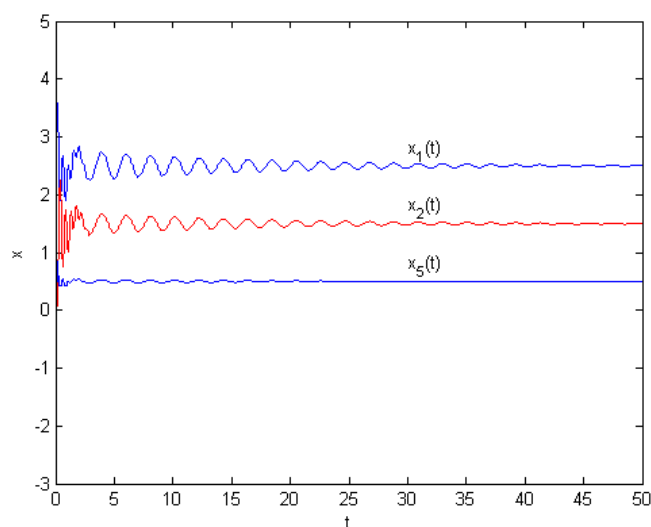
که $\psi = 2/5\pi$ قرار می‌دهیم و $w > (1 - \cos \psi)/(1 - 2\cos \psi)$ است. در این قسمت به حل مسائلی می‌پردازیم که با استفاده از روش سیمپلکس دچار دور افتادگی می‌شوند اما با استفاده از مدل شبکه عصبی ارائه شده در فصل سوم برای حل مساله تخصیص این مشکل از بین می‌رود.

مثال ۲.۲.۴. (یودین و گلدشتین)^۴ [۳۶]

$$\text{maximize } x_3 - x_4 + x_5 - x_6$$

^۳Hoffman

^۴Yudin and Gol'shtein



شکل ۱.۴: مسیر جواب در مثال ۲.۲.۴

$$\text{subject to } \begin{cases} x_1 + x_3 - 2x_4 - 3x_5 + 4x_6 = 0, \\ x_2 + 4x_3 - 3x_4 - 2x_5 + x_6 = 0, \\ x_3 + x_4 + x_5 + x_6 + x_7 = 1, \\ x \geq 0, \end{cases}$$

جواب بهینه‌ی آن $(x_1 = 2/5, x_2 = 1/5, x_5 = 0/5)$ می‌باشد.

مثال ۳.۲.۴. (مارشال و سوربال)^۵[۲۱]

$$\begin{aligned} & \text{minimize } 2x_1 + 4x_4 + 4x_6 \\ & \text{subject to } \begin{cases} x_1 - 3x_2 - x_3 - x_4 - x_5 + 6x_6 = 0, \\ 2x_2 + x_3 - 3x_4 - x_5 + 2x_6 = 0, \\ x \geq 0, \end{cases} \end{aligned}$$

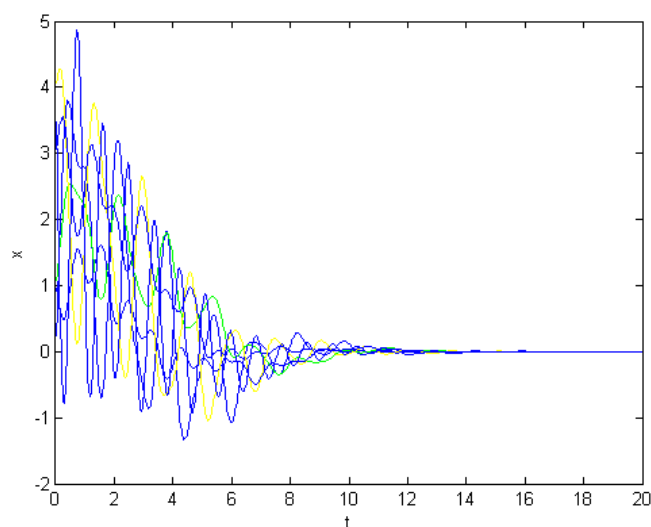
جواب بهینه $(x_j = 0, (j = 1, \dots, 6))$ می‌باشد.

مثال ۴.۲.۴. ^۶[۲۱]

$$\text{minimize } -0/4x_5 - 0/4x_6 + 1/8x_7$$

^۵Marshall and Suurballe

^۶Marshall and Suurballe



شکل ۲.۴: مسیر جواب در مثال ۳.۲.۴

$$\text{subject to } \begin{cases} x_1 + 0.6x_5 - 0.4x_6 + 4.8x_7 = 0, \\ x_2 + 0.2x_5 - 1.8x_6 + 0.6x_7 = 0, \\ x_3 + 0.4x_5 - 1.6x_6 + 0.2x_7 = 0, \\ x_4 + x_6 = 1, \\ x \geq 0, \end{cases}$$

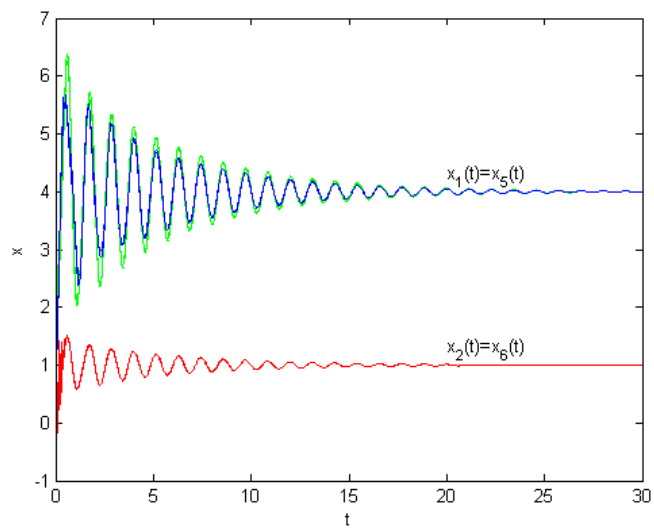
جواب بهینه $(x_1 = 4, x_2 = 1, x_5 = 4, x_6 = 1)$ می‌باشد.

مثال ۵.۲.۴. [۱۰]^۷

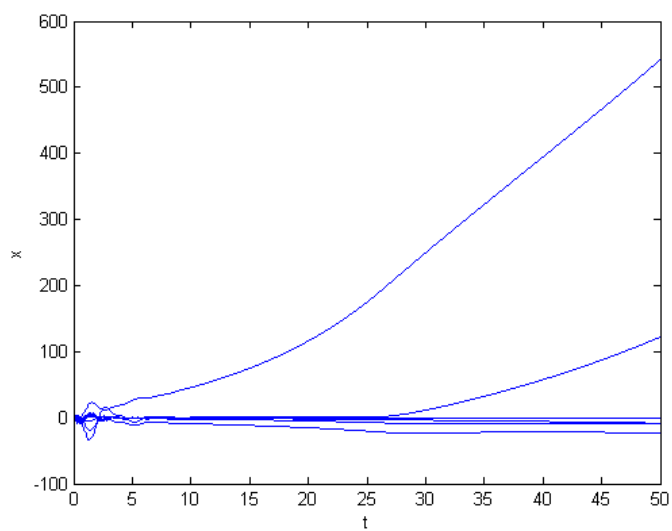
$$\begin{aligned} &\text{maximize } 3x_1 - 8x_2 + 2x_3 - 24x_4 \\ &\text{subject to } \begin{cases} x_1 - 32x_2 - 4x_3 + 36x_4 + x_5 = 0, \\ x_1 - 24x_2 - x_3 + 6x_4 + x_6 = 0, \\ x \geq 0, \end{cases} \end{aligned}$$

جواب بهینه نامتناهی است.

^۷Sierksma



شکل ۳.۴: مسیر جواب در مثال ۴.۲.۴



شکل ۴.۴: مسیر جواب در مثال ۵.۲.۴

۳.۴ نتایج

در این پایان نامه سعی کرده‌ایم مروری بر شبکه‌های عصبی داشته و با استفاده از این نظریه به حل رده‌ای از مسائل بهینه‌سازی همچون مسائل تخصیص و کوتاهترین مسیر بپردازیم، برای مسائل انتخاب شده به ارائه یک مدل از شبکه‌های عصبی پرداخته‌ایم. همچنین برای مسائل برنامه‌ریزی خطی که در دور می‌افتد نیز از مدل پیشنهادی شبکه‌ی عصبی استفاده کرده و پایداری و همگرایی سراسری مدل مورد بررسی قرار گرفت و کارایی مدل پیشنهاد شده را با حل چندین مثال عددی نشان دادیم.

از جمله مزیت‌های مدل ارائه شده می‌توان به موارد زیر اشاره کرد:

۱. همگرایی سراسری مدل و وابسته نبودن به نقطه اولیه لزوماً شدنی برخلاف بیشتر روشهای بهینه سازی از جمله روشهای نقطه درونی و سیمپلکس.

۲. استفاده از مدل برای مسائل برنامه‌ریزی خطی که ممکن است مساله در دور بیفتد.

۳. نداشتن پارامتر جریمه بر خلاف مدل‌های گرادیانی و سادگی مدل پیشنهاد شده.

۴. بعضی از مدل‌های شبکه‌ی عصبی نمی‌توانند مسائل برنامه‌ریزی محدب عمومی را وقتی که تابع هدف و تمام قیدها محدب‌اند، مانند برنامه‌ریزی خطی یا برنامه‌ریزی محدب درجه‌ی دوم تباهیده حل کنند، اما مدل شبکه‌ی عصبی ارائه شده این مسائل را حل می‌کند.

۱.۳.۴ پیشنهادات

از مدل ارائه شده در فصل سوم این پایان‌نامه برای حل مسائل حمل و نقل و ارتباطات، حل مسائل برنامه‌ریزی تخصیص درجه دوم استفاده شود. می‌توان با طراحی یک شبکه عصبی دوگانی، دوگان مساله تخصیص را حل کرد.

مراجع

- [۱] د. برقز-آ. گراهام. مقدمه‌ای بر نظریه کنترل و کنترل بهینه. موسسه چاپ و انتشارات دانشگاه فردوسی، بهمن ۱۳۷۲.
- [۲] دکتر علی کاوه و دکتر عباس ایران‌منش. شبکه‌های عصبی مصنوعی در بهینه‌سازی داده‌ها. مشاور مرکز تحقیقات ساختمان، ۱۳۸۷.
- [۳] نرگس طهماسبی. یک کاربرد از مدل‌های شبکه عصبی برای حل مسائل بهینه‌سازی سبد سرمایه. پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شاهرود، شهریور ۹۱.
- [۴] دکتر محمدباقر منهای. مبانی شبکه‌های عصبی. دانشگاه صنعتی امیر کبیر، ۱۳۷۷.
- [5] A. Rueda, J.L. Huertas A. Rodriguez-Vazquez, R. Dommnguer-Castro and E. Sahnchez-Sinencio. Nonlinear switched-capacitor neural network for optimization problems. *IEEE Transactions on Circuits and Systems*, 42:384–390, 1990.
- [6] A.J. Hoffman. *Cycling in the simplex algorithm*. National Bureau of Standards, Washington, 1953.
- [7] A.R. Nazemi, F. Omid. An efficient dynamic model for solving the shortest path problem. *Transportation Research Part C*, 26:1–19, 2013.

- [8] A. Bar-Noy, M. Bellare, M.M. Halldursson, H. Shachnai, and T. Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2):183–202, 1998.
- [9] S. Efati, A.R. Nazemi. Neural network models and its application for solving linear and quadratic programming problems. *Applied Mathematics and Computation*, 172:305–331, 2006.
- [10] G. Sierksma. *Linear and integer programming*. Inc, New York: Marcel Dekker, 2nd edition, 1996.
- [11] K. Giaro and M. Kubale. Edge-chromatic sum of trees and bounded cyclicity graphs. *Information Processing Letters*, 75:65–69, 2000.
- [12] G. Xing-Bao. Exponential stability of globally projected dynamic systems. *IEEE Transactions*, 14:426–431, 2003.
- [13] A.J. Hoffman and H.M. Markowitz. A note on shortest path, assignment, and transportation problems. *Naval Research Logistics Quart*, 10:375–380, 1963.
- [14] J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. *Biol*, 52:141–152, 1985.
- [15] M.S. Bazaraa, J.J. Jarvis and H. D. Sherali. *Linear Programming and Network Flows*. Wiley, 2nd edition, 1990.
- [16] J.C. Hull. *Options Futures and Other Derivatives*. New Jersey, 6th edition, 2006.
- [17] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 2nd edition, 1984.

- [18] M.P. Kennedy, L.O. Chua . Neural networks for nonlinear programming. *IEEE Transactions on Neural Networks*. 554-562, 35, 1988.
- [19] C.M. Shetty M.S. Bazaraa, H. D. Serali. *Nonlinear Programming Theory and Algorithms*. Wiley, New York, 2nd edition, 1993.
- [20] M. Małafiejski, K. Giaro, R. Janczewski and M. Kubale. Sum coloring of bipartite graphs with bounded degree, algorithmica. *Computer Science*, 40:235–244, 2004.
- [21] J.W. Suurballe, KT. Marshall. A note on cycling in the simplex method. *Naval Research Logistics Quarterly*, 16(1):121-37, 1969.
- [22] A.N. Michel, R.K. Miller. *Ordinary Differential Equations*. New York, 1982.
- [23] A.R. Nazemi. A dynamic system model for solving convex nonlinear optimization problems. *Communication in Nonlinear Science and Number Simulation*, 17:1696–1705, 2012.
- [24] A.R. Nazemi. Solving general convex nonlinear optimization problems by an efficient neurodynamic model. *Engineering Application of Artificial Intelligence*, 26:685–696, 2013.
- [25] T.M. Rheinboldt, W.C. Ortega. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
- [26] A. Quarteroni, R. Sacco, F. Saleri. *Numerical Mathematics*. Springer-Verlag, Berlin, 2007.
- [27] R. Fuchslin R. Pfeifer, D. Damian. *Neural Networks*. 2009.

-
- [28] S. Vinjamuri, I. Saul, SI. Gass. Cycling in linear programming problems. *Computers operations Research*, 31:303–311, 2004.
 - [29] SI. Gass. Encounters with degeneracy. *Annals of Operations Research*, 47, 1993.
 - [30] G. Simmons. *Differential Equations and Applications*. McGraw- Hill Education, India, 2006.
 - [31] J.M. Vegas, P.J. Zufiria. Generalized neural networks for spectral analysis: dynamics and liapunov functions. *Neural Networks*, 17(2):233–245, 2004.
 - [32] J. Wang. A recurrent neural network for solving the shortest path problem. *IEEE Transactions. Circuits Syst*, 43:482–486, 1996.
 - [33] J. Wang. Primal and dual assignment networks. *IEEE Transactions on Neural Network*, 8(3):784–790, May 1997.
 - [34] Y. Xia and J. Wang. Analysis and design of primal-dual assignment networks. *IEEE Transactions on Neural Networks*, 9:183–194, 1998.
 - [35] Y. Xia and J. Wang. Neural network for solving linear programming problems with bounded variables. *IEEE Transactions on Neural Networks*, 6:515–519, 1995.
 - [36] E.G. Gol'shtein, DB. Yudin. *Linear programming*. Israel Program Of Scientific Translations, Jerusalem, 1965.

واژه‌نامه فارسی به انگلیسی

Assignment Problem	مساله تخصیص
Alphabetic	الفبایی
Capacity	ظرفیت
Coefficient	ضریب
Complementarity	مکمل
Cycling	دوری
Cost	هزینه
Decision Variable	متغیر تصمیم
Degeneracy	تباہیدگی
Equilibrium Point	نقطه تعادل
Excitation Function	تابع تحریک
Feasible	شدنی
Global	سراسری
Globally Asymptotically Stable	پایداری مجانبی سراسری
Global Convergence	همگرایی سراسری
Global Minimum	کمینه سراسری
Invariant Set	مجموعه پایدار

Local Minimum	کمینه موضعی
Neural-network	شبکه عصبی
Neuron	نرون
Optimality	بهینگی
Perturbation	آشفستگی
Rank	رتبه
Rate of Convergance	نرخ همگرایی
Routing	مسیریابی
Selforganization	تابع انتقال
Shortest-path	خودسازمان دهی
Sinaps	دستگاه عصبی
Strictly Monotonic	اکیدا یکنوا
Symmetric	متقارن
Threshold	آستانه
Transfor Function	تابع انتقال
Theorem of Complementary Slackness	قضیه مکمل زاید
Transport	حمل و نقل
Weight	وزن

Surname: Ghezelsofly

Name: Ozra

Title: An application of neural network models for solving assignment problem

Supervisor: Dr. Alireza Nazemi

Degree: Master of Science

Subject: Applied Mathematics

Field: Operation Research

University of Shahrood

Faculty of Mathematical Sciences

Date: 2013

Number of pages: [76](#)

Keywords: Assignment, Neural network, Shortest-path, cycle

Abstract

Recently methods of optimization based on artificial intelligence approaches have been developed remarkable success in solving optimization problems efficiently acquired. Methods such as Genetic Algorithms, Tabu Search, refrigeration simulation and neural networks, their ability to solve large problems have good action. In this thesis, we tried one model of recursive neural network is presented to solve assignment and shortest path problems. The main idea is to replace the assignment problem with a linear programming (LP) problem. According to the saddle point theorem, optimization theory, convex analysis theory, Lyapunov stability theory and LaSalle invariance principle, the equilibrium point of the proposed neural network is proved to be equivalent to the optimal solution of the original problem. It is also shown that the proposed neural network model is stable in the sense of Lyapunov and it is globally convergent to an exact optimal solution of the assignment problem.



University of Shahrood
Faculty of Mathematical Sciences

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

An application of neural network models for solving assignment problem

Supervisor
Dr. Alireza Nazemi

by
Ozra Ghezelsofly

2013