



دانشگاه صنعتی شاهرود

دانشکده علوم ریاضی
گروه ریاضی کاربردی

پایان نامه

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

عنوان

یک روش هوش محاسباتی برای حل ردهای از مسائل مینی ماکس

استاد راهنما

دکتر علیرضا ناظمی

پژوهشگر

الهر رضایی

آبان ماه ۹۲

نام خانوادگی دانشجو: رضایی

نام: الهه

عنوان: یک روش هوش محاسباتی برای حل رده‌ای از مسائل مینی‌ماکس

استاد راهنما: دکتر علیرضا ناظمی

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

دانشگاه: دانشکده علوم ریاضی

دانشگاه صنعتی شاهرود

تعداد صفحات: ۸۳

تاریخ فارغ‌التحصیلی: آبان ماه ۹۲

واژگان کلیدی: شبکه عصبی، پایداری، لیاپانوف، نقطه تعادل، همگرایی سراسری، پرسپترون، غیرهموار

چکیده

در این پایان نامه ابتدا تعاریف مرتبط با نرم، ماتریس ژاکوبی، ماتریس هسین، بردار گرادین و تابع محدب را بیان نموده و در خصوص مسائل بهینه‌سازی به ذکر شرایط لازم و کافی بهینگی می‌پردازیم و مفاهیم پایداری و تابع انرژی در سیستم‌های دینامیکی را ارائه می‌دهیم. سپس ساختار شبکه عصبی و مدل ریاضی یک سلول عصبی را بیان می‌کنیم و به بیان تاریخچه‌ای از شبکه‌های عصبی در حل مسائل بهینه‌سازی و بیان مدل‌های ارائه شده پیشین برای حل مسائل بهینه‌سازی می‌پردازیم. در فصل سوم، ساختار یک مدل از شبکه‌های عصبی، بر پایه‌ی قضایای دوگانی، بهینه‌سازی، تجزیه و تحلیل توابع محدب، پایداری لیاپانوف و اصل تغییرناپذیری لازال برای حل مسائل برنامه‌ریزی غیرخطی محدب را ارائه می‌کنیم و ثابت می‌کنیم که نقطه تعادل شبکه‌ی عصبی پیشنهاد شده همان جواب بهینه مسأله برنامه‌ریزی غیرخطی محدب است. همچنین نشان می‌دهیم که شبکه عصبی پیشنهادی دارای پایداری به مفهوم لیاپانوف است و به طور دقیق همگرایی سراسری به یک جواب بهینه مسأله اصلی است. کارآمدی مدل پیشنهادی با ارائه چندین مثال نشان داده می‌شود. در فصل آخر یک تکنیک بهینه‌سازی را توضیح می‌دهیم که برای حل مسائل کلاس بهینه‌سازی غیر هموار کاربرد دارد و برای نشان دادن کارآمدی مدل، چند مثال را با این روش حل می‌کنیم.

تقدیم بہ ہمسفر عزیزم

کہ در این راہ بہترین یاورم بود.

الهی...۱

باز آمدیم با دو دست تهی چه باشد اگر مرحمی بر خستگان نهی.
الهی گرفتار آن دردم که تو دوی آنی و در آرزوی آن سوزم که تو سرانجام آنی.
الهی چراغ دل مریدانی و انس جان غریبانی، کریمآسایش سینه محبانی و نهایت همت قاصدانی.
الهی جرم من زیر حلم تو پنهان است و تو پرده عفو خود بر من گستران.
الهی عاجز و سرگردانم، نه آنچه دانم دارم و نه آنچه دارم دانم.
الهی چون به تو بنگریم شاهیم و تاج بر سر و چون بخود نگریم خاکیم و از خاک کمتر.
الهی هر کس تو را شناخت هرچه غیر تو بود بینداخت.
الهی! نام تو ما را جواز، مهر تو ما را جهاز، شناخت تو ما را امان، لطف تو ما را عیان.

سپاس گزارمی...

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 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 همسر عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودش، که در این سردترین روزگاران، بهترین پشتیبان من بود.

اله رضایی
آبان ماه ۹۲

فهرست مطالب

خ	لیست تصاویر
۳	۱ مقدمه و مفاهیم اولیه
۴	۱.۱ مقدمه
۴	۲.۱ مجموعه‌های آفین و محدب
۴	۳.۱ توابع آفین و محدب
۶	۴.۱ شرایط لازم و کافی برای مسائل نامقید
۸	۵.۱ شرایط لازم و کافی برای مسائل مقید
۹	۶.۱ پایداری
۱۲	۷.۱ تابع انرژی
۱۳	۸.۱ اصل تغییرناپذیری لازال
۱۵	۲ مقدمه‌ای بر شبکه‌های عصبی
۱۶	۱.۲ مقدمه
۱۹	۲.۲ مدل‌سازی نرون تنها
۲۳	۱.۲.۲ فراگیری در نرون‌های ساده
۲۶	۳.۲ پرسپترون چند لایه‌ای
۲۸	۴.۲ مدل‌های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه‌سازی
۳۶	۳ حل فرم کلی مسائل بهینه‌سازی محدب با یک مدل عصبی دینامیکی کارآمد
۳۷	۱.۳ مقدمه
۳۷	۲.۳ یک مدل شبکه عصبی
۴۱	۳.۳ تحلیل پایداری و همگرایی

۴۶	مثال عددی	۴.۳
۶۰		یک روش هوش مصنوعی برای حل رده‌ای از مسائل بهینه‌سازی غیر هموار	۴
۶۱	یک مدل شبکه عصبی	۱.۴
۶۲	مثال عددی	۲.۴
۷۴		مراجع	
۷۷		واژه‌نامه فارسی به انگلیسی	
۸۰		واژه‌نامه انگلیسی به فارسی	

لیست تصاویر

۱۰	پایداری لیاپانوف	۱.۱
۱۷	مشخصات اصلی یک نرون بیولوژیک.	۱.۲
	ناقل شیمیایی آزاد شده از شکاف سیناپس می‌گذرد و دریافت‌کننده‌های دندریت نرون	۲.۲
۱۸	دیگر را تحریک می‌کند.	
۱۹	مشخصات اصلی یک نرون بیولوژیک	۳.۲
۲۰	نمای مدل اصلی نرون	۴.۲
۲۲	تابع آستانه	۵.۲
۲۷	علامت منطقی یا حذفی	۶.۲
۲۷	پرسپترون چندلایه‌ای	۷.۲
۳۹	دیاگرام ساده شده برای شبکه‌ی عصبی (۵.۳)	۱.۳
	رفتار $x_5(t), x_3(t), x_2(t)$ و $x_7(t)$ با استفاده از مدل (۵.۳) در مثال (۱.۴.۳) با	۲.۳
۴۷	نقطه اولیه‌ی $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$	
	رفتار $x_6(t), x_4(t), x_1(t)$ و $x_8(t)$ با استفاده از مدل (۵.۳) در مثال (۱.۴.۳) با نقطه	۳.۳
۴۸	اولیه‌ی $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$	
	رفتار واگرای سیستم (۷.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ در	۴.۳
۴۹	مثال (۱.۴.۳).	
	رفتار واگرای سیستم (۱۰.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ در	۵.۳
۵۰	مثال (۱.۴.۳).	
۵۱	رفتار سیستم (۵.۳) با 5^0 نقطه اولیه تصادفی در مثال (۲.۴.۳).	۶.۳
۵۲	رفتار همگرای $\ x(t) - x^*\ ^2$ در مثال (۲.۴.۳).	۷.۳
۵۳	رفتار صفحه فاز (۵.۳) با ۱۱ نقطه اولیه مختلف در مثال (۳.۴.۳).	۸.۳

- ۹.۳ مسیر مدل پیشنهادی در (۵.۳) با 2° نقطه اولیه تصادفی در مثال (۴.۴.۳) ۵۴
- ۱۰.۳ رفتار همگرایی $\|y(t) - y^*\|^2$ با نقطه اولیه $y_0 = (1, -1, 1, -1, 1, -1, 1, -1)^T$ ۵۵
- در مثال (۴.۴.۳) ۵۵
- ۱۱.۳ نمایش شبکه‌ی عصبی (۵.۳) با ۸ نقطه اولیه تصادفی در مثال (۴.۴.۳) ۵۶
- ۱۲.۳ مسیر مدل پیشنهادی در (۵.۳) با ۵ نقطه اولیه تصادفی در مثال (۶.۴.۳) ۵۶
- ۱۳.۳ رفتار همگرایی $\|y(t) - y^*\|^2$ در مثال (۶.۴.۳) با نقطه اولیه $(-5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5)^T$ ۵۷
- ۱۴.۳ مسیر شبکه‌ی عصبی (۵.۳) با 10° نقطه اولیه تصادفی در مثال (۷.۴.۳) ۵۸
- ۱۵.۳ همگرایی $x_3 : x_2 : x_1$ و $x_3(t) + x_2(t)$ و $x_1(t)$ با نقطه اولیه $y_{1,0}$ در مثال (۸.۴.۳) ۵۸
- ۱۶.۳ همگرایی $x_3 : x_2 : x_1$ و $x_3(t) + x_2(t)$ و $x_1(t)$ با نقطه اولیه $y_{2,0}$ در مثال (۸.۴.۳) ۵۹
- ۱.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با 5° نقطه اولیه تصادفی با $\tau = 1$ در مثال (۱.۲.۴) ۶۳
- ۲.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با 5° نقطه اولیه تصادفی با $\tau = 0.1$ در مثال (۱.۲.۴) ۶۴
- ۳.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با 5° نقطه اولیه تصادفی با $\tau = 5$ در مثال (۱.۲.۴) ۶۵
- ۴.۴ بررسی تأثیرات متغیر τ بر شبکه‌ی عصبی (۵.۳) با مقدار خطای $\|x(t) - x^*\|^2$ در مثال (۱.۲.۴) ۶۶
- ۵.۴ رفتار شبکه‌های عصبی (۵.۳) با 5° نقطه اولیه تصادفی برای حل (۵.۴) ۶۷
- ۶.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۳.۴) با 15° نقطه اولیه تصادفی در مثال (۲.۲.۴) ۶۸
- ۷.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۴.۴) با 15° نقطه اولیه تصادفی در مثال (۲.۲.۴) ۶۹
- ۸.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۹.۴) با 3° نقطه اولیه تصادفی در مثال (۳.۲.۴) ۷۰
- ۹.۴ رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۲۰.۴) با 6° نقطه اولیه تصادفی در مثال (۳.۲.۴) ۷۱

پیشگفتار

در فصل اول به ارائه تعاریف و قضایایی می‌پردازیم که شامل مجموعه‌های محدب، آفینی، نرم، پایداری، تابع انرژی و قضایای مربوط به آنها و همچنین اصل تغییرناپذیری لازال است.

در فصل دوم ابتدا یک نرون بیولوژیکی بررسی شده و مدل‌سازی اولیه برای این نرون انجام شده است. فراگیری در نرون‌های ساده و پرسپترون چندلایه‌ای و الگوریتم مربوط به آنها و همچنین تاریخچه‌ای از شبکه‌های عصبی بیان شده است. و تعدادی از مدل‌های موجود از شبکه‌های عصبی پیشین برای حل مسائل بهینه‌سازی مورد بررسی قرار گرفته است.

در فصل سوم، یک مدل شبکه‌ی عصبی پیشنهادی بررسی شده است. همگرایی سراسری و پایداری آن بر مبنای قضایای دوگانی در بهینه‌سازی، پایداری لیاپانوف و اصل ناوردای لازال برای حل مسائل برنامه‌ریزی غیرخطی محدب، اثبات شده است. همچنین برای نشان دادن کارایی این مدل، آن را با سه مدل دیگر مقایسه کرده و مثال‌های عددی با استفاده از نرم‌افزار مطلب و شبکه‌عصبی پیشنهادی حل شده است.

در فصل چهارم به معرفی مسائل برنامه‌ریزی غیرخطی غیرهموار پرداخته‌ایم. با استفاده از دو تکنیک هموارسازی، این دسته از مسائل را به مسائل برنامه‌ریزی هموار تبدیل کردیم. سپس جواب بهینه این مسائل را با استفاده از شبکه عصبی پیشنهادی در فصل سوم به دست آوردیم.

فصل ۱

مقدمه و مفاهیم اولیه

۱.۱ مقدمه

در این فصل تعاریف مورد نیاز در فصل‌های بعدی به‌طور مختصر آورده شده است. ابتدا تعاریف مرتبط با مجموعه‌های آفین، مجموعه‌های محدب و همچنین تعاریف توابع آفین، توابع محدب، انواع مختلف نرم‌ها، بردارگرادیان، ماتریس هسین و ماتریس ژاکوبین را بیان نموده و در خصوص مسائل بهینه‌سازی مقید و نامقید به ذکر شرایط لازم و کافی بهینگی می‌پردازیم. سپس مفاهیم پایداری و تابع انرژی در سیستم‌های دینامیکی را ارائه می‌دهیم.

۲.۱ مجموعه‌های آفین و محدب

تعریف ۱.۲.۱. [۵] مجموعه $C \subset \mathbb{R}^n$ را آفین می‌نامیم، اگر خط گذرنده از هر دو نقطه متمایز در C در مجموعه C قرار گیرد. اگر C مجموعه‌ای آفین باشد، برای هر $x_1, x_2 \in C$ و $\theta \in \mathbb{R}$ خواهیم داشت:

$$\theta x_1 + (1 - \theta)x_2 \in C. \quad (1.1)$$

یک نقطه به‌صورت $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ که $\theta_1 + \theta_2 + \dots + \theta_k = 1$ را ترکیب آفین نقاط x_1, \dots, x_k می‌نامیم. در واقع هر مجموعه آفین شامل تمام ترکیبات آفین نقاط خودش است.

تعریف ۲.۲.۱. [۵] مجموعه $C \subset \mathbb{R}^n$ را محدب می‌نامیم، اگر پاره‌خط گذرنده از هر دو نقطه دلخواه C در مجموعه C قرار گیرد. اگر C مجموعه‌ای محدب باشد، برای هر $x_1, x_2 \in C$ و هر $0 \leq \theta \leq 1$ خواهیم داشت:

$$\theta x_1 + (1 - \theta)x_2 \in C. \quad (2.1)$$

نقطه x را ترکیب محدب نقاط x_1, \dots, x_k می‌نامیم، اگر $x = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ که در آن $\theta_1 + \theta_2 + \dots + \theta_k = 1$ و $0 \leq \theta_i \leq 1$ در واقع هر مجموعه محدب شامل تمام ترکیبات محدب نقاط خودش است.

۳.۱ توابع آفین و محدب

تعریف ۱.۳.۱. [۵] تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ را تابع خطی می‌گوییم، هرگاه حافظ عملیات خطی در فضای برداری \mathbb{R}^n باشد. یعنی:

$$f(x + y) = f(x) + f(y), \quad (۳.۱)$$

$$f(\alpha x) = \alpha f(x).$$

برای هر $x, y \in \mathbb{R}^n$ و هر $\alpha \in \mathbb{R}$.

تعریف ۲.۳.۱. تابع $f : \mathbb{R}^n \rightarrow \mathbb{R}^l$ را تابع آفین می‌گوییم، هرگاه به صورت مجموع یک تابع خطی و یک تابع ثابت باشد. به عنوان مثال $f(x) = Ax + b$ که $A \in \mathbb{R}^{l \times n}$ و $b \in \mathbb{R}^l$ تابع آفین است.

تعریف ۳.۳.۱. [۵] تابع $f : \mathbb{R}^n \rightarrow \mathbb{R}$ بر روی مجموعه نقاط واقع در مجموعه محدب $E \subset \mathbb{R}^n$ یک تابع محدب نامیده می‌شود، اگر:

$$\forall x, y \in E, \lambda \in (0, 1) : f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (۴.۱)$$

تابع f را روی E مقعر می‌گوییم، هرگاه $-f$ محدب باشد.

تعریف ۴.۳.۱. [۵]

اگر $\bar{x} \in X$ و ε -همسایگی از \bar{x} مثل $N_\varepsilon(\bar{x})$ موجود باشد به طوری که به ازای هر $x \in X \cap N_\varepsilon(\bar{x})$ داشته باشیم $f(\bar{x}) \leq f(x)$ ، آنگاه \bar{x} یک کمینه موضعی برای f است.

تعریف ۵.۳.۱. [۵] تابع $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ را تابع نرم می‌گوییم، هرگاه:

$$\begin{cases} \forall x, \|x\| \geq 0, \\ \|x\| = 0 \Leftrightarrow x = 0, \\ \|\alpha x\| = |\alpha| \|x\|, \\ \|x + y\| \leq \|x\| + \|y\|. \end{cases} \quad (۵.۱)$$

تعریف ۶.۳.۱. [۵] فرض کنید $f : \mathbb{R}^n \rightarrow \mathbb{R}$. گرادیان تابع f در x_0 برابر است با:

$$\nabla f(x_0) = \left(\frac{\partial f}{\partial x_1}(x_0), \dots, \frac{\partial f}{\partial x_n}(x_0) \right)^T. \quad (۶.۱)$$

تعریف ۷.۳.۱. [۲۳] فرض کنید $f : \mathbb{R}^n \rightarrow \mathbb{R}$ روی مجموعه $U \subseteq \mathbb{R}^n$ تابعی مشتق‌پذیر باشد. همچنین فرض کنید همه $\frac{\partial^2 f}{\partial x_i \partial x_j}$ موجود و روی U پیوسته باشند، در این صورت ماتریس هسین f را با H_f نشان می‌دهند و به صورت زیر تعریف می‌شود:

$$H_f = \nabla^2 f = \begin{bmatrix} \nabla \frac{\partial f}{\partial x_1} \\ \nabla \frac{\partial f}{\partial x_2} \\ \vdots \\ \nabla \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (۷.۱)$$

تعریف ۸.۳.۱. [۲۳] فرض کنید تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ دارای مولفه‌های f_1 و $f_2 \dots$ تا f_m باشد و هر مولفه $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ دارای مشتقات جزئی نسبت به هر x_j در x_0 باشند. در این صورت ماتریس ژاکوبین f در x_0 را با $J_f(x_0)$ نشان می‌دهند. بنابراین:

$$J_f(x_0) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x_0) & \dots & \frac{\partial f_1}{\partial x_j}(x_0) & \dots & \frac{\partial f_1}{\partial x_n}(x_0) \\ \vdots & \dots & \vdots & \dots & \vdots \\ \frac{\partial f_i}{\partial x_1}(x_0) & \dots & \frac{\partial f_i}{\partial x_j}(x_0) & \dots & \frac{\partial f_i}{\partial x_n}(x_0) \\ \vdots & \dots & \vdots & \dots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x_0) & \dots & \frac{\partial f_m}{\partial x_j}(x_0) & \dots & \frac{\partial f_m}{\partial x_n}(x_0) \end{bmatrix}. \quad (۸.۱)$$

تعریف ۹.۳.۱. 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 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 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 نامیده می‌شود، اگر عددی مثبت مانند α وجود داشته باشد به طوری که:

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

۴.۱ شرایط لازم و کافی برای مسائل نامقید

قضیه ۱۰.۴.۱. (شرط لازم مرتبه اول [۶]). فرض کنید X یک مجموعه باز و ناتهی در \mathbb{R}^n باشد. مسأله برنامه‌ریزی نامقید زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} && f(x), \\ & \text{subject to} && \\ & && x \in X, \end{aligned} \quad (۹.۱)$$

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

قضیه ۲.۴.۱. (شرط لازم مرتبه دوم [۶]). فرض کنید مشتق دوم $f : \mathbb{R}^n \rightarrow \mathbb{R}$ در \hat{x} موجود باشد. اگر \hat{x} یک کمینه موضعی باشد، آنگاه ماتریس هسین f در \hat{x} نیمه معین مثبت است.

قضیه ۳.۴.۱. (شرط کافی [۶]). 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 f است.

قضیه ۴.۴.۱. [۲۲]. فرض کنید مجموعه E به صورت زیر باشد:

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

اگر $g_k(x)$ ها برای $k = 1, \dots, m$ توابعی محدب و $h_p(x)$ ها برای $p = 1, \dots, l$ آفین باشند، آنگاه E یک مجموعه محدب است.

قضیه ۵.۴.۱. [۶]. اگر $f(x)$ بر مجموعه محدب E تابعی محدب باشد، آنگاه $f(x)$ دارای یک کمینه سراسری است و بر روی مجموعه محدب به دست آورده می شود.

دو قضیه بالا بیان می کنند که برای یک مسأله کلی کمینه سازی، اگر $f(x)$ و همه $g_k(x)$ ها برای $k = 1, \dots, m$ توابعی محدب و $h_p(x)$ ها برای $p = 1, \dots, l$ آفین باشند، آنگاه یک کمینه موضعی از $f(x)$ تحت این مجموعه از قیود، یک کمینه سراسری است.

قضیه ۶.۴.۱. [۶]. اگر $\gamma(x)$ کوچک ترین مقدار ویژه ماتریس $M(x)_{n \times n}$ باشد، آنگاه:

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

۵.۱ شرایط لازم و کافی برای مسائل مقید

قضیه ۱.۵.۱. (شرایط لازم کاروش-کان-تاکر^۱ (K.K.T), [۶]). فرض کنید X یک مجموعه باز در \mathbb{R}^n باشد. مسأله کمینه‌سازی مقید زیر را در نظر بگیرید:

$$\text{minimize } f(x),$$

subject to

$$g_k(x) \leq 0, \quad k = 1, \dots, m, \quad (10.1)$$

$$h_j(x) = 0, \quad j = 1, \dots, l,$$

$$x = (x_1, x_2, \dots, x_n)^T \in X.$$

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 $K = \{k: g_k(\hat{x}) = 0\}$. همچنین فرض کنید f و g_k برای $k \in K$ در \hat{x} مشتق‌پذیر و g_k برای $k \notin K$ در \hat{x} پیوسته و h_j برای $j = 1, \dots, l$ دارای مشتق پیوسته باشند. بعلاوه فرض کنید $\nabla g_k(\hat{x})$ برای $k \in K$ و $\nabla h_j(\hat{x})$ برای $j = 1, \dots, l$ مستقل خطی باشند. اگر \hat{x} کمینه موضعی برای (۱۰.۱) باشد، آنگاه اسکالره‌های یکتای $\hat{u}_k \geq 0$ برای $k \in K$ و $\hat{v}_j \geq 0$ برای $j = 1, \dots, l$ موجود هستند به طوری که:

$$\begin{cases} \nabla f(\hat{x}) + \sum_{k \in K} \hat{u}_k \nabla g_k(\hat{x}) + \sum_{j=1}^l \hat{v}_j \nabla h_j(\hat{x}) = 0, \\ \hat{u}_k \geq 0, \quad k \in K. \end{cases} \quad (11.1)$$

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

$$\begin{cases} \nabla f(\hat{x}) + \sum_{k=1}^m \hat{u}_k \nabla g_k(\hat{x}) + \sum_{j=1}^l \hat{v}_j \nabla h_j(\hat{x}) = 0, \\ \hat{u}_k g_k(\hat{x}) = 0, \quad k = 1, \dots, m, \\ \hat{u}_k \geq 0, \quad k = 1, \dots, m. \end{cases} \quad (12.1)$$

همچنین شرایط کاروش-کان-تاکر می‌تواند به فرم ماتریسی زیر نوشته شود:

$$\begin{cases} \nabla f(\hat{x}) + \nabla g(\hat{x})^T u + \nabla h(\hat{x})^T v = 0, \\ \hat{u}^T g(\hat{x}) = 0, \\ \hat{u} \geq 0. \end{cases} \quad (13.1)$$

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 می‌شود.

^۱Karush-Kuhn-Tucker

قضیه ۲.۵.۱. (شرایط کافی کاروش-کان-تاكر (K.K.T), [۶]). فرض کنید X یک مجموعه باز در \mathbb{R}^n باشد. مسأله (۱۰.۱) را در نظر بگیرید. فرض کنید \hat{x} یک جواب شدنی مسأله باشد و همچنین $K = \{k : g_k(\hat{x}) = 0\}$. اگر شرایط کاروش-کان-تاكر در \hat{x} برقرار باشد، یعنی اسکالرهایی $\hat{u}_k \geq 0$ برای $k \in K$ و $\hat{v}_j \geq 0$ برای $(j = 1, \dots, l)$ موجود باشند به طوری که:

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

آنگاه \hat{x} یک کمینه موضعی برای (۱۰.۱) خواهد بود.

قضیه ۳.۵.۱. [۲۱]. اگر در مسأله (۱۰.۱) توابع f و g_k برای $(k = 1, \dots, m)$ محدب باشند و توابع h_j برای $(j = 1, \dots, l)$ آفینی باشند، آنگاه شرایط کاروش-کان-تاكر، لازم و کافی اند.

۶.۱ پایداری

سیستم دینامیکی زیر را در نظر بگیرید:

$$\begin{cases} \dot{x} = f(x(t)), \\ x(t_0) = x_0 \in \mathbb{R}^n, \end{cases} \quad (۱۴.۱)$$

که در آن $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. x^* یک نقطه تعادل برای (۱۴.۱) نامیده می‌شود، اگر $f(x^*) = 0$.

تعریف ۱.۶.۱. فرض کنید $\Omega \in \mathbb{R}^n$ یک همسایگی باز از x^* باشد. تابع به طور پیوسته مشتق پذیر $w : \mathbb{R}^n \rightarrow \mathbb{R}$ را تابع لیاپانوف در x^* روی مجموعه Ω برای سیستم (۱۴.۱) گوئیم هرگاه:

$$w(x^*) = 0, \quad (۱۵.۱)$$

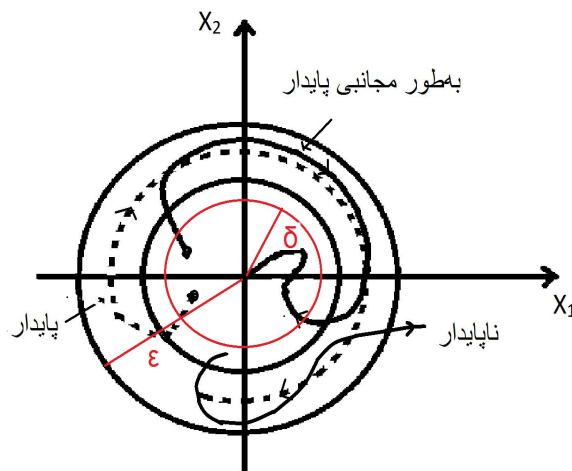
$$w(x) \geq 0, x \in \Omega - \{x^*\}, \quad (۱۶.۱)$$

$$\frac{dw(x)}{dt} \leq 0. \quad (۱۷.۱)$$

تعریف ۲.۶.۱. فرض کنید $x(t)$ یک جواب (۱۴.۱) باشد، در این صورت نقطه تعادل x^* پایدار به مفهوم لیاپانوف است، اگر برای هر $x_0 = x(t_0)$ و هر $\varepsilon > 0$ یک $\delta > 0$ وجود داشته باشد به طوری که اگر $\|x(t) - x_0\| \leq \delta$ ، آنگاه:

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

برای بررسی بیشتر شکل (۱.۱) را ببینید.



شکل ۱.۱: پایداری لیاپانوف

قضیه ۱.۶.۱ [۲۰]. نقطه تعادل x^* پایدار لیاپانوف است اگر یک تابع لیاپانوف روی یک همسایگی Ω^* از x^* موجود باشد.

تعریف ۳.۶.۱. سیستم دینامیکی (۱۴.۱) همگرای سراسری به مجموعه

$$\Omega^* = \{x \mid x \text{ جواب سیستم (۱۴.۱) است}\}$$

گفته می‌شود اگر هر نقطه آغازین دلخواه $x(t)$ از سیستم در رابطه زیر صدق کند:

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

که در آن

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

تعریف ۴.۶.۱. سیستم دینامیکی در نقطه تعادل یکتای x^* پایدار مجانبی سراسری نامیده می‌شود، اگر x^* به مفهوم لیاپانوف پایدار باشد و در شرط زیر صدق کند:

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

تعریف ۵.۶.۱. مجموعه $M \subseteq \mathbb{R}^n$ ، یک مجموعه تغییرناپذیر نسبت به سیستم (۱۴.۱) گفته می‌شود، اگر $x(t_0) \in M$ و $t_0 \geq 0$ ، آنگاه به ازای هر $t, t \geq t_0$ ، $x(t) \in M$ باشد.

تعریف ۶.۶.۱. نگاشت F در شرط لیپ شیتز صدق می‌کند اگر عدد ثابت L وجود داشته باشد، به طوری که برای هر دو نقطه $x, y \in \mathbb{R}^n$ داشته باشیم:

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

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

تعریف ۷.۶.۱. نگاشت $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ یکنوا گفته می‌شود، 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 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 $x, y \in \mathbb{R}^n$ عدد ثابت $\beta > 0$ وجود داشته باشد به طوری که:

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

قضیه ۲.۶.۱ [۷]. فرض می‌کنیم $F : K \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ ، 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 ۳.۶.۱ [۷]. فرض می‌کنیم $F : K \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ ، یک تابع به طور پیوسته مشتق پذیر روی مجموعه‌ی محدب و باز K شامل Ω باشد. در این صورت F یکنوا (یکنوای قوی) روی Ω است، اگر و تنها اگر ماتریس ژاکوبی F برای هر $x \in \Omega$ نیمه معین مثبت (معین مثبت) باشد.

قضیه ۴.۶.۱ [۷]. در سیستم دینامیکی (۱۴.۱) فرض می‌کنیم که $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ یک تابع پیوسته باشد، آنگاه برای هر $x_0 \in \mathbb{R}^n$ و $t_0 > 0$ ، یک جواب محلی

$x(t)$ به ازای $t \in [t_0, \tau)$ که $\tau > t_0$ وجود دارد. علاوه بر این اگر f در x_0 در شرط لیپ شیتز محلی صدق کند، آنگاه جواب یکتا خواهد بود و اگر f در \mathbb{R}^n در شرط لیپ شیتز صدق کند، آنگاه τ می‌تواند تا $+\infty$ توسعه داده شود.

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

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

در نتیجه نقطه تعادل سیستم جدید $y^* = \circ$ ، $y = g(y)$ است. زیرا:

$$g(\circ) = f(\circ + x^*) = f(x^*) = \circ.$$

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

۷.۱ تابع انرژی

قضیه ۱.۷.۱. [۲۴]. فرض کنید $x = \circ$ یک نقطه تعادل برای سیستم (۱۴.۱) باشد، همچنین تابع E پیوسته مشتق‌پذیر باشد و $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ اگر:

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

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

$$3. \quad \text{به ازای هر } x \in \Omega \subseteq \mathbb{R}^n \setminus \{\circ\}, \frac{d(E(x))}{dt} \leq \circ.$$

آنگاه $x = \circ$ نقطه پایداری سیستم خواهد بود و $E(x)$ را «تابع لیاپانوف» یا «تابع انرژی» برای سیستم (۱۴.۱) می‌نامیم.

قضیه ۲.۷.۱. [۲۴]. فرض کنید $x = \circ$ یک نقطه تعادل برای سیستم (۱۴.۱) باشد همچنین تابع E پیوسته مشتق‌پذیر باشد و $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ اگر:

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

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

$$3. \quad \text{به ازای هر } x \in \Omega \subseteq \mathbb{R}^n \setminus \{\circ\}, \frac{d(E(x))}{dt} < \circ.$$

آنگاه $x = \circ$ پایدار مجانبی خواهد بود.

قضیه ۳.۷.۱. [۲۴]. فرض کنید $x = \circ$ یک نقطه تعادل برای سیستم (۱۴.۱) باشد، همچنین تابع $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ به‌طور پیوسته مشتق‌پذیر باشد، اگر:

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

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

۳. به ازای هر $x \in \mathbb{R}^n \setminus \{0\}$ ، $\frac{d(E(x))}{dt} < 0$ ،

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

۸.۱ اصل تغییرناپذیری لازال

تعریف ۱.۸.۱. فرض کنید $x(t)$ جوابی از سیستم دینامیکی $\dot{x} = f(x)$ باشد. نقطه $p \in \mathbb{R}^n$ حد مثبت $x(t)$ نامیده می‌شود، اگر دنباله $(t_n)_{n \geq 0}$ موجود باشد به طوری که $\lim_{n \rightarrow \infty} t_n = \infty$ و $\lim_{n \rightarrow \infty} x(t_n) = p$. مجموعه حدود مثبت دنباله مسیره‌های $(x(t))_{t \geq 0}$ را با L^+ نمایش می‌دهیم.

تعریف ۲.۸.۱. مجموعه M را مجموعه تغییرناپذیر مثبت برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر:

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

تعریف ۳.۸.۱. مجموعه M را مجموعه تغییرناپذیر منفی برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر:

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

تعریف ۴.۸.۱. مجموعه M را مجموعه تغییرناپذیر اساسی برای سیستم دینامیکی $\dot{x} = f(x)$ می‌نامیم، اگر تغییرناپذیر مثبت و همچنین تغییرناپذیر منفی باشد.

تعریف ۵.۸.۱. دنباله مسیره‌های $(x(t))_{t \geq 0}$ پایدار مجانبی وابسته به M نامیده می‌شود، اگر $t \rightarrow \infty$ و برای هر $\varepsilon > 0$ وجود داشته باشد $T > 0$ ، به طوری که:

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

که در آن d یک متر است.

قبل از بیان اصل تغییرناپذیری لازال به بیان یک لم می‌پردازیم.

لم ۱.۸.۱ [۱۳]. فرض کنید تابع $f : D \rightarrow \mathbb{R}^n$ روی $D \subseteq \mathbb{R}^n$ پیوسته لیپ‌شیتس باشد که D مجموعه‌ای فشرده و شامل مبدأ است. همچنین فرض کنید تابع $V : D \rightarrow [0, \infty)$ به طور پیوسته مشتق‌پذیر و $x = x(t)$ جوابی برای سیستم دینامیکی $\dot{x} = f(x)$ باشد و این جواب‌ها برای هر $t \geq 0$ توسط مجموعه D کراندار باشند. در این صورت مجموعه حدود مثبت (L^+) ، ناتهی، فشرده و تغییرناپذیر مثبت است.

قضیه ۱.۸.۱. (اصل تغییر ناپذیری لازال، [۱۳]).

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 $V : D \rightarrow \mathbb{R}$ به‌طور 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 ۱.۸.۱. [۱۳]. فرض کنید مبدأ، نقطه تعادل سیستم دینامیکی $\dot{x} = f(x)$ باشد، همچنین تابع $V : D \rightarrow \mathbb{R}$ روی $D \subset \mathbb{R}^n$ پیوسته مشتق‌پذیر و شامل مبدأ باشد به‌طوری‌که روی مجموعه D داشته‌باشیم $\dot{V}(x) \leq 0$. همچنین مجموعه $E = \{x \in D \mid \dot{V}(x) = 0\}$ را در نظر می‌گیریم. اگر E فقط شامل نقطه $x(t) = 0$ باشد، آنگاه مبدأ به‌عنوان یک نقطه تعادل سیستم دینامیکی بالا، پایدار مجانبی خواهد بود.

فصل ۲

مقدمه‌ای بر شبکه‌های عصبی

۱.۲ مقدمه

در این فصل به بیان مقدمه‌ای از مدل‌های شبکه‌های عصبی می‌پردازیم. در ابتدا ساختار شبکه عصبی و مدل ریاضی یک سلول عصبی را بیان می‌کنیم و در ادامه به بیان تاریخچه‌ای از شبکه‌های عصبی در حل مسائل بهینه‌سازی و بیان مدل‌های ارائه شده پیشین برای حل مسائل بهینه‌سازی خواهیم پرداخت. می‌دانیم کامپیوترها می‌توانند بعضی از کارها که ما آنها را در مدت زمان قابل ملاحظه‌ای انجام می‌دهیم مانند انجام چهار عمل اصلی، در کمترین زمان انجام دهند و یا می‌توانند نام‌ها و آدرس‌ها را ماه‌ها بعد به‌درستی به‌یاد بیاورند. با این اوصاف آیا باز هم می‌توان گفت انسان‌ها از کامپیوترها باهوش‌ترند؟ و اینکه چرا کامپیوترها نمی‌توانند کارهایی را که ما انجام می‌دهیم، انجام دهند؟

اگر به درون مغز نگاه کنیم، بررسی اولیه ما چیزی جز مجموعه‌ای گره‌خورده از ماده‌ای خاکستری رنگ نشان نمی‌دهد. بررسی بیشتر روشن می‌کند که مغز از اجزایی ریز تشکیل شده است. لیکن این اجزاء به شیوه‌ای بی‌نهایت پیچیده، مرتب شده‌اند و هر جزء به هزاران جزء دیگر متصل است. شاید این تفاوت در شیوه ساختار، علت اصلی اختلاف بین مغز و کامپیوتر است. کامپیوترها طوری طراحی شده‌اند که یک عمل را بعد از عمل دیگر با سرعت بسیار زیاد انجام دهند. لیکن مغز با تعداد اجزای بیشتر اما با سرعت کمتر کار می‌کند. در حالیکه سرعت عملیات در کامپیوترها به میلیون‌ها محاسبه در ثانیه بالغ می‌شود، سرعت عملیات در مغز تقریباً بیشتر از ده بار در ثانیه نمی‌باشد. لیکن مغز در یک لحظه با تعداد زیادی اجزاء به‌طور هم‌زمان کار می‌کند، کاری که از عهده کامپیوتر بر نمی‌آید. مغز می‌تواند کامپیوتر را در مسائلی مانند دیدن و شنیدن که اعمال کاملاً موازیند و در آن‌ها داده‌های متضاد و متفاوت، هر کدام باعث اثرات و ظهور خاطرات متفاوتی در مغز می‌گردند شکست دهد.

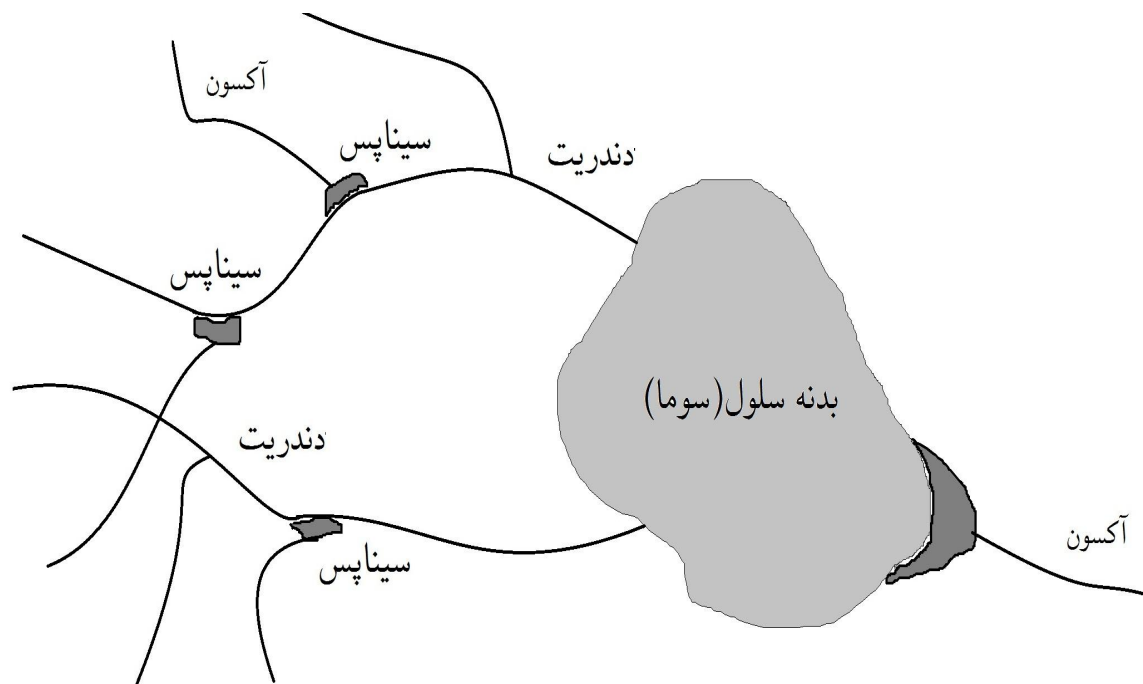
نرون^۱ عنصر اصلی مغز است و به تنهایی مانند یک واحد پردازش منطقی عمل می‌کند. مغز تقریباً دارای 10^{10} واحد پایه نرون است و هر نرون تقریباً به 10^4 نرون دیگر اتصال دارد. نرون‌ها دوناوع هستند، نرون‌های داخلی مغز که در فاصله‌های حدود 10^0 میکرون به یکدیگر متصل‌اند و نرون‌های خارجی که قسمت‌های مختلف مغز را به یکدیگر و مغز را به ماهیچه‌ها و اعضای حسی را به مغز متصل می‌کنند. نحوه عملیات نرون‌ها بسیار پیچیده است و هنوز در سطح میکروسکوپی چندان شناخته شده نیست. هر نرون ورودی‌های متعددی را پذیراست که با یکدیگر به طریقی جمع می‌شوند. اگر در یک لحظه تعداد ورودی‌های فعال نرون به حد کفایت برسد نرون نیز فعال شده و آتش می‌کند. در غیراین‌صورت نرون به صورت غیر فعال و آرام باقی می‌ماند.

بدنه‌ی نرون، سوما^۲ نامیده می‌شود. به سوما رشته‌های نامنظم طولانی متصل است که به آنها

^۱Neuron

^۲Soma

دندریت^۳ می‌گویند. قطر این رشته‌ها اغلب از یک میکرون نازک‌تر است و اشکال شاخه‌ای پیچیده‌ای دارند. شکل ظریف آن‌ها شبیه شاخه‌های درخت بدون برگ هستند که هر شاخه بارها و بارها به شاخه‌های نازک‌تر منشعب می‌شود.



شکل ۱۰۲: مشخصات اصلی یک نرون بیولوژیک.

دندریت‌ها نقش اتصالاتی را دارند که ورودی‌ها را به نرون‌ها می‌رسانند. یکی از عناصر متصل به هسته‌ی نرون، آکسون^۴ نامیده می‌شود. این عنصر برخلاف دندریت از نظر الکتریکی فعال است و بعنوان خروجی نرون عمل می‌کند. نمایشی از ویژگی‌های عمده نرون در شکل (۱۰۲) آمده است. رشته آکسون در نقطه‌ی تماس معینی به نام سیناپس^۵ قطع می‌شود و در این مکان به دندریت سلول دیگر وصل می‌گردد. در واقع این تماس به صورت اتصال مستقیم نیست بلکه از طریق ماده شیمیایی موقتی صورت می‌گیرد. سیناپس پس از آنکه پتانسیل آن از طریق پتانسیل‌های فعالیت دریافتی از طریق آکسون به اندازه کافی افزایش یافت از خود ماده شیمیایی به نام منتقل‌کننده عصبی^۶ ترشح می‌کند. یادگیری‌ها

^۳Dendrite

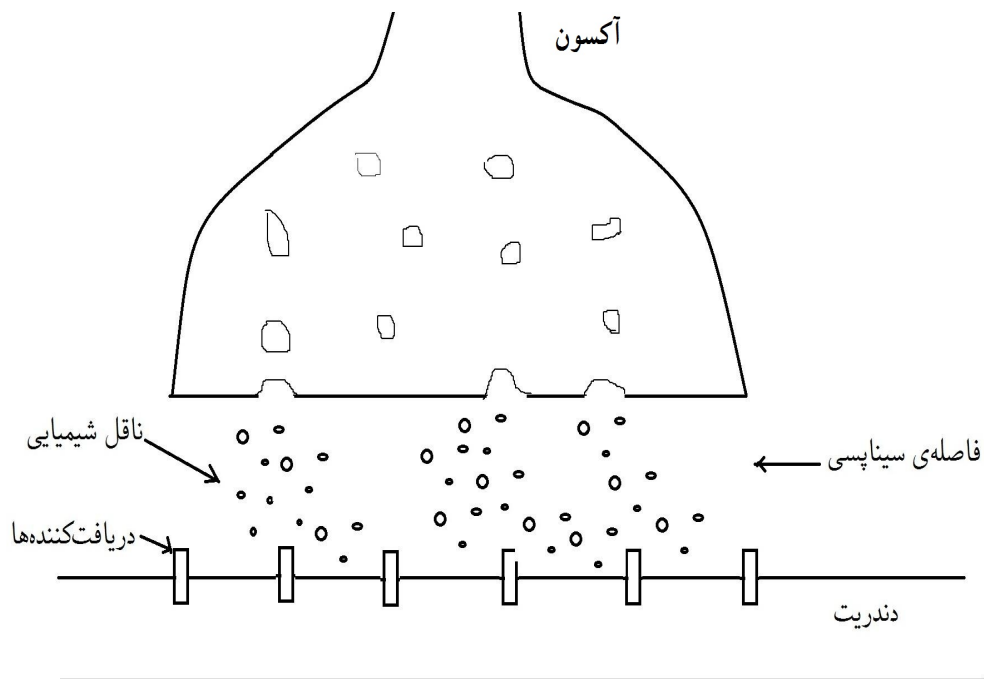
^۴axon

^۵synaps

^۶neurotransmitter

قدرت اتصال‌های سیناپسی را افزایش می‌دهند. تصور می‌شود یادگیری هنگامی صورت می‌گیرد که شدت اتصال یک سلول و سلول دیگر در محل سیناپس‌ها اصلاح می‌گردد. شکل (۲.۲) ویژگی‌های مهم سیناپس را با جزئیات بیشتر نشان می‌دهد.

به‌نظر می‌رسد که این مقصود از طریق ایجاد سهولت بیشتر در میزان آزاد شدن ناقل شیمیایی حاصل می‌گردد. این حالت باعث می‌شود که دروازه‌های بیشتری روی دندریتهای سمت مقابل باز شود و به این صورت باعث افزایش میزان اتصال دو سلول شود. تغییر میزان اتصال نرون‌ها به‌صورتی که باعث تقویت تماس‌های مطلوب شود از مشخصه‌های مهم در مدل‌های شبکه‌های عصبی است.



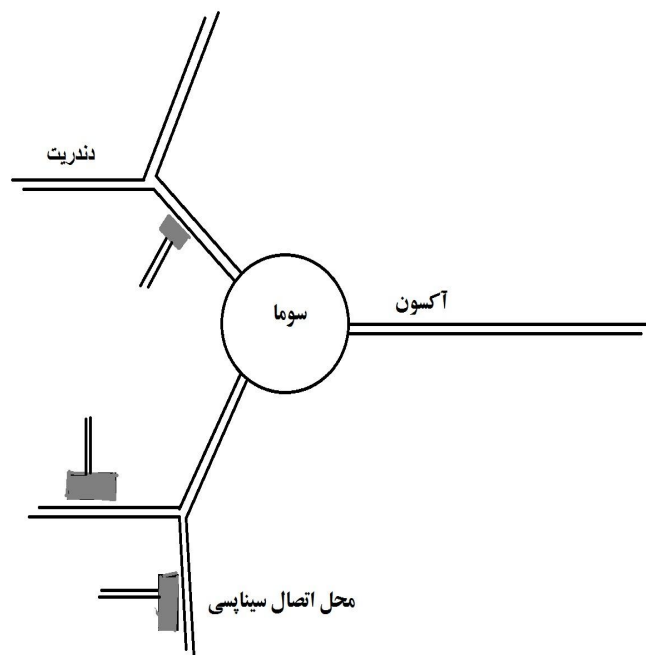
شکل ۲.۲: ناقل شیمیایی آزاد شده از شکاف سیناپس می‌گذرد و دریافت‌کننده‌های دندریت نرون دیگر را تحریک می‌کند.

گفتیم مغز را می‌توان به‌صورت مجموعه‌ای بسیار متصل و شبکه‌ای از عناصر پردازشی نسبتاً ساده در نظر گرفت. به مدلی نیاز داریم که بتواند ویژگی‌های مهم سیستم‌های عصبی را کسب کند. به این منظور که بتواند رفتار مشابهی را از خود بروز دهد. لیکن اگر بخواهیم این مدل، به اندازه کافی برای فهمیدن و به‌کارگیری، ساده باشد باید بسیاری از جزئیات را عمداً نادیده بگیریم. استخراج تعداد محدودی ویژگی‌های مهم و نادیده‌گرفتن بقیه ویژگی‌ها از ضروریتهای معمولی مدل‌سازی است. هدف مدل‌سازی،

اصولاً ایجاد نمونه ساده‌تری از سیستم است که رفتار عمومی سیستم را حفظ کرده و کمک کند که سیستم با سهولت بیشتر قابل درک باشد.

۲.۲ مدل‌سازی نرون تنها

شبکه‌های عصبی مصنوعی از یک سری واحدهای ساختمانی اولیه تشکیل می‌شوند. هر واحد دارای چندین ورودی است که با هم ترکیب شده و پس از انجام عملیات پردازش یک خروجی را به دست می‌دهند. این واحدهای اولیه به هم متصل هستند به طوری که خروجی هر واحد به عنوان ورودی واحدهای دیگر مورد استفاده قرار می‌گیرد. واحدهای ساختمانی را سلول عصبی، نرون یا گره نیز می‌نامند.



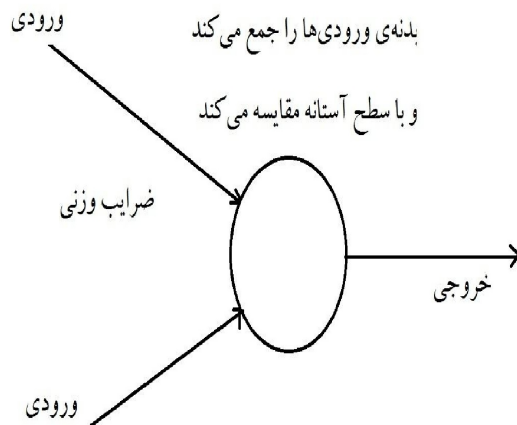
شکل ۳.۲: مشخصات اصلی یک نرون بیولوژیک

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

ابتدا مشخصات یک نرون تنها و نحوه مدل‌سازی آن را بررسی می‌کنیم. نقش اصلی یک نرون بیولوژیکی، عمل جمع ورودی‌های خود تا جایی است که مجموع ورودی‌ها از حدی که به آن آستانه^۷ می‌گوییم تجاوز نکند و آنگاه تولید یک خروجی است. ورودی‌های نرون از طریق دندریت‌ها که به خروجی‌های نرون‌های دیگر توسط نقاط اتصال (سیناپس‌ها) متصل است وارد می‌شوند. سیناپس‌ها کارایی سیگنال‌های دریافتی را تغییر می‌دهند. بدنه سلولی، کلیه‌ی ورودی‌ها را دریافت می‌کند و هنگامی که مجموع ورودی‌ها از حد آستانه تجاوز کرد سیگنالی را آتش می‌کند. این نرون بیولوژیکی ساده در شکل (۳.۲) نشان داده شده است.

مدلی که از نرون می‌سازیم باید مشخصه‌های زیر را داشته باشد. به‌طور خلاصه:

- خروجی یک نرون یا فعال است (یک) و یا غیرفعال است (صفر).
- خروجی تنها به ورودی‌ها بستگی دارد. میزان ورودی‌ها باید به حدی برسد که خروجی نرون را فعال سازد.



شکل ۴.۲: نمای مدل اصلی نرون

کارایی سیناپس‌ها در انتقال سیگنال‌های ورودی به بدنه سلول را می‌توان با استفاده از ضریبی که در ورودی‌های نرون ضرب می‌شود مدل‌سازی کرد. سیناپس‌های قوی‌تر که سیگنال بیشتری را منتقل می‌کنند دارای ضریب‌های بسیار بزرگتری هستند درحالی که سیناپس‌های ضعیف ضریب‌های کوچک‌تری

^۷Threshold

دارند. بدین صورت مدل ما به صورتی خواهد بود که در شکل (۴.۲) آمده است. این مدل ابتدا مجموع وزنی ورودی‌های خود را محاسبه کرده سپس آن را با سطح آستانه داخلی خود مقایسه می‌کند و چنانچه از آن تجاوز کرد فعال می‌شود. در غیر این صورت غیرفعال باقی می‌ماند. چون ورودی‌ها برای تولید خروجی از میان نرون عبور می‌کنند به این سیستم پیش‌خور^۸ می‌گوییم.

این عمل را باید به طریق ریاضی نشان دهیم. اگر تعداد ورودی‌ها n باشد آنگاه هر خط ورودی دارای یک ضریب وزنی مربوط به خود است. نرون مدل‌سازی شده ورودی خود را محاسبه می‌کند. ابتدا اولین ورودی را در ضریب وزنی مربوط به خط ارتباطی آن ورودی ضرب می‌کند. سپس همین عمل را برای ورودی دوم و سایر ورودی‌ها تکرار می‌کند در نهایت تمام مقادیر حاصل را جمع می‌کند. بطور خلاصه:

$$\sum_{i=1}^n = w_1x_1 + \dots + w_nx_n. \quad (۱.۲)$$

حاصل جمع فوق باید با مقدار آستانه نرون موردنظر مقایسه شود. در مقایسه با آستانه اگر حاصل جمع به دست آمده از میزان آستانه تجاوز کند آنگاه خروجی نرون مساوی ۱ خواهد بود و اگر حاصل جمع کمتر از آستانه باشد خروجی مساوی صفر می‌شود. از طرف دیگر می‌توان مقدار آستانه را ابتدا از حاصل جمع وزنی به دست آمده کم کرد و آنگاه مقدار حاصل را با صفر مقایسه کرد. اگر مقدار حاصل مثبت بود خروجی نرون ۱ و در غیر این صورت صفر خواهد بود. این دو حالت در نمودار (۵.۲) آمده است. در این شکل‌ها محور x نشان‌دهنده‌ی مجموع ورودی‌ها و محور y نشان‌دهنده‌ی خروجی نرون است.

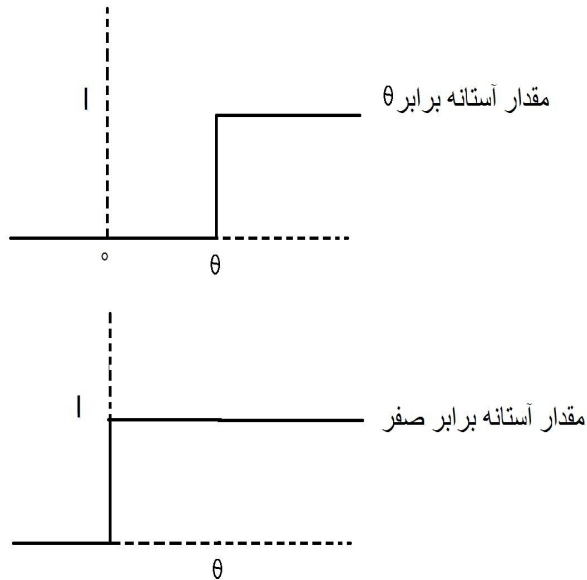
می‌بینیم کل آستانه تغییر نکرده است ولی پرش پلکانی در $x = 0$ صورت گرفته است. راه دیگر می‌تواند حذف کامل آستانه از بدنه نرون و در عوض اضافه کردن یک ورودی با مقدار ثابت ۱ باشد. این ورودی همیشه فعال باقی می‌ماند و ضریب وزنی خط ارتباطی آن برابر با مقدار منفی آستانه در نظر گرفته می‌شود. در این صورت به جای کم کردن مقدار آستانه از حاصل جمع ورودی نرون می‌توان ورودی ثابت اضافی ۱+ را در ضریب وزنی خود که برابر با مقدار منفی آستانه است ضرب کرد و به سایر ورودی‌ها اضافه کرد. به این کار احتساب تورش می‌گوییم. به همین صورت مقدار θ - را تورش یا یک‌سویی می‌نامیم. هر دو راه‌کار مشابه و قابل قبول است.

اگر خروجی را y بنامیم، رابطه‌ی زیر راه‌کار اول را بیان می‌کند.

$$y = f_h \left[\sum_{i=1}^n w_i x_i - \theta \right]. \quad (۲.۲)$$

^۸Feed forward

به تابع آستانه تابع پله‌ای و یا تابع هوی ساید هم اطلاق می‌گردد.



شکل ۵.۲: تابع آستانه

در حالی که f_h یک تابع پلکانی است (این تابع را تابع هوی ساید^۹ نیز می‌نامند). و

$$f_h(x) = 1 \quad \text{اگر } x > 0, \quad (3.2)$$

$$f_h(x) = 0 \quad \text{اگر } x \leq 0.$$

می‌بینیم خروجی تابع تنها مقادیر ۱ و ۰ است. به عبارت دیگر نرون یا فعال است یا غیرفعال. اگر از راه‌کار دوم استفاده کنیم، ورودی دیگری را با شماره‌ی ۰ انتخاب کرده و مقدار آن را همیشه برابر ۱ قرار می‌دهیم. در این صورت ضرب وزنی ورودی جدید برابر با مقدار تورش خواهد بود. تابع فوق به صورت زیر در خواهد آمد:

$$y = f_h \left[\sum_{i=0}^n w_i x_i \right]. \quad (4.2)$$

حد زیرین علامت زیگما از ۱ به صفر تغییر کرده و مقدار x همیشه برابر ۱+ خواهد بود. نرون‌های مدل، که به طریقی ساده به یکدیگر متصل‌اند، در سال ۱۹۶۲ توسط فرانک روزن‌بلات^{۱۰} بنام

^۹Heaviside

^{۱۰}Frank Rosenblatt

پرسپترون^{۱۱} نامگذاری شد. او برای نخستین بار نرون‌های مدل را در کامپیوترهای دیجیتال شبیه‌سازی کرد و آن‌ها را بطور رسمی تحلیل نمود. روزن بلات در کتاب خود «اصول دینامیسم عصبی»^{۱۲} پرسپترون‌ها را به صورت شبکه‌های ساده شده شرح داد که در آن‌ها برخی از ویژگی‌های سیستم‌های عصبی واقعی بطور اغراق‌آمیز به‌کار رفته و برخی دیگر از آنها نادیده گرفته شده است.

۱.۲.۲ فراگیری در نرون‌های ساده

برای حل هر مسأله‌ای، شبکه‌های عصبی سه مرحله را طی می‌کنند: آموزش^{۱۳}، تعمیم^{۱۴} و اجرا^{۱۵}. آموزش، فرایندی است که طی آن شبکه می‌آموزد تا الگوی موجود در ورودی‌ها را (که به صورت مجموعه داده‌های آموزشی است) بشناسد. برای این منظور هر شبکه عصبی از مجموعه‌ای از قوانین یادگیری که نحوه یادگیری را تعریف می‌کنند استفاده می‌کند. تعمیم، توانایی شبکه برای ارائه جواب قابل قبول در قبال ورودی‌هایی است که در مجموعه آموزشی نبوده‌اند. یعنی پس از آنکه مثال‌های اولیه به شبکه آموزش داده شد، شبکه می‌تواند در قبال یک ورودی آموزش داده نشده قرار گیرد و یک خروجی مناسب را ارائه نماید. این خروجی براساس مکانیسم تعمیم که همانا چیزی جز فرایند درونیابی نیست به دست می‌آید. استفاده از شبکه برای انجام عملکردی که به آن منظور طراحی شده است را اجرا می‌گویند. در اثر آموزش، وزن‌های داخلی (که بر روی ورودی‌های هر سلول اعمال می‌شود) تغییر می‌کند و به وضعیت مناسب می‌رسند. یکی از نقاط ضعف شبکه‌های عصبی این است که نتایج آموزش، یعنی وزن‌های داخلی، هیچگونه تصویر روشنی از اعتبار جواب‌های مسأله بدست نمی‌دهد. این وزن‌ها کاملاً قابل درک نیستند. با این وجود جواب‌های تولید شده توسط شبکه، اغلب صحیح است و با شرایط کمی حاکم بر محیط سازگاری دارد. این صحت جوابها و صدق شرایط کمی حاکم بر محیط، گاهی مهم‌تر از توضیح‌پذیر بودن آن است. تاکنون اسامی متفاوتی برای شبکه‌های عصبی به‌کار رفته است. از جمله آنها می‌توان به مدل‌های جعبه سیاه^{۱۶}، مدل‌های تجربی، تقریب زن‌های عام و مدل‌های موازی^{۱۷} اشاره نمود.

ما به شیوه‌ای برای فراگیری در مدل‌های نرون خود نیازمندیم. اتصال این نرون‌ها به یکدیگر شاید شبکه‌هایی را ایجاد کند که بتوانند کاری را انجام دهند، لیکن برای انجام کاری مفید باید بتوانیم به طریقی

^{۱۱}Perceptron

^{۱۲}Principle of neurodynamics

^{۱۳}Traning

^{۱۴}Generalization

^{۱۵}Operation

^{۱۶}Black-Box models

^{۱۷}Parallel models

آن را آموزش دهیم و آنچه این مدل‌ها را قابل استفاده می‌کند توانایی آن‌ها در فراگیری است. همچنین برای سهولت درک مدل‌ها، روش‌های فراگیری باید تا حد امکان ساده باشند. به همان صورت که در اغلب محاسبات عصبی معمول است، منبع الهام ما سیستم‌های عصبی واقعی خواهد بود. کودکان اغلب برای کسب نتایج خوب تشویق می‌شوند، و برای عبور از خیابان بدون توجه به اطراف، سرزنش می‌گردند. همین شیوه را می‌توان در شبکه‌های مصنوعی نیز به‌کار گرفت. باید رفتارهای مطلوب را تشویق و رفتارهای نامطلوب را تضعیف کرد. اگر دو گروه از اشیاء داشته باشیم مثلاً گروهی از شکل‌های نوشته شده مختلف A و گروهی از شکل‌های نوشته شده مختلف B . شاید مایل باشیم نرون مورد نظر ما A را از B تمیز دهد. شاید بخواهیم نرون ما با مشاهده یک A عدد ۱ را و با دیدن B عدد صفر را بیرون دهد.

اگر در ابتدا ضرایب وزنی خطوط ارتباطی نرون را به‌طور تصادفی تعیین کنیم یعنی در واقع حالت شروع بوده و نرون هیچ نمی‌داند، آنگاه می‌توانیم یک حرف A را به نرون وارد کنیم، نرون مجموع وزنی ورودی‌های خود را محاسبه می‌کند و با مقدار آستانه مقایسه می‌کند. چنانچه مقدار محاسبه شده از مقدار آستانه بیشتر باشد نرون جواب ۱ و در غیر این صورت خروجی صفر خواهد داد. احتمال این‌که به‌طور تصادفی جواب صحیح باشد 5% است، فرض کنید نرون جواب صحیح بدهد. در این صورت نیاز به هیچ اقدامی نیست زیرا مدل موفق بوده است. ولی اگر جواب صفر بود باید مجموع وزنی را افزایش دهیم به‌صورتی که بار دیگر که با حرف A روبه‌رو شد جواب صحیح ۱ بدهد. این عمل را با افزایش ضرایب وزنی خطوط ارتباطی نرون انجام می‌دهیم. برای حرف B مایلیم که نرون عدد صفر را تولید کند. بنابراین هرگاه نرون با حرف B روبه‌رو شد مایل خواهیم بود که ضرایب وزنی آن را کاهش دهیم تا مجبور گردد در آینده با مشاهده‌ی حرف B عدد صفر را تولید نماید.

این بدان معنی است که برای فراگیری شبکه باید زمانی که مایلیم نرون فعال باشد ضرایب وزنی را افزایش داده و آنگاه که مایلیم نرون غیر فعال باشد ضرایب را کاهش دهیم. این مقصود با اضافه کردن مقدار ورودی‌ها به ضرایب مربوطه هنگامی که مایلیم نرون فعال باشد و با کسر کردن مقدار ورودی‌ها از ضرایب هنگامی که مایلیم نرون غیرفعال باشد حاصل می‌گردد. این قاعده‌ی فراگیری ما خواهد بود.

الگوریتم فراگیری پرسپترون

روش فراگیری که در بالا به آن اشاره شد را می‌توان به‌صورت الگوریتم زیر نشان داد.

- ۱- ضرایب و مقادیر اولیه آستانه را تعیین کنید برای $0 \leq i \leq n$ ، $w_i(t)$ را بعنوان ضریب وزنی ورودی i در زمان t و θ را بعنوان مقدار آستانه خروجی در نظر بگیرید. مقدار w را برابر $-\theta$

و مقدار x_0 را همیشه برابر ۱ قرار دهید. $w_i(0)$ را برابر مقادیر تصادفی کوچک قرار دهید، بدین‌صورت تمام وزن‌ها و آستانه‌ها را به حالت شروع درآورید.

۲- ورودی و خروجی مطلوب را ارائه دهید. ورودی‌های $x_0, x_1, x_2, \dots, x_n$ و خروجی $d(t)$ را به مدل ارائه دهید.

۳- خروجی واقعی را محاسبه کنید.

$$y(t) = f_h \left[\sum_{i=0}^n w_i(t) x_i(t) \right]. \quad (5.2)$$

۴- ضرایب وزنی را تبدیل کنید.

- $w_i(t+1) = w_i(t)$ اگر خروجی صحیح بود.
- $w_i(t+1) = w_i(t) + x_i(t)$ اگر خروجی واقعی صفر و خروجی مطلوب ۱ بود.
- $w_i(t+1) = w_i(t) - x_i(t)$ اگر خروجی واقعی ۱ و خروجی مطلوب صفر بود.

این الگوریتم اصلی پرسپترون است. لیکن اصلاحات متعددی در این الگوریتم پایه پیشنهاد شده است. اولین اصلاح وارد کردن عامل ضربی کوچکتر از یک در فرمول تغییر ضرایب وزنی است. این عمل باعث کندشدن سرعت تغییر ضرایب وزنی می‌شود و بدین صورت شبکه در گام‌های کوتاه‌تری به جواب نزدیک‌تر می‌شود. این اصلاح، قدم چهارم الگوریتم را به‌صورت زیر تغییر می‌دهد:

۴- ضرایب وزنی را تعدیل کنید (شکل اصلاح شده).

- $w_i(t+1) = w_i(t)$ اگر خروجی صحیح بود.
- $w_i(t+1) = w_i(t) + \eta x_i(t)$ اگر خروجی واقعی صفر و خروجی مطلوب ۱ بود.
- $w_i(t+1) = w_i(t) - \eta x_i(t)$ اگر خروجی واقعی ۱ و خروجی مطلوب صفر بود.

الگوریتم مشابهی توسط ویدرو و هاف^{۱۸} ارائه شده است. آنها به این نکته پی برده بودند که بهتر است هنگامی که اختلاف خروجی واقعی و خروجی مطلوب زیاد است ضرایب وزنی به میزان بیشتر و هنگامی که این اختلاف جزئی است به مقدار کمتر تعدیل گردد. آن‌ها قاعده‌ای را برای فراگیری به‌نام

^{۱۸}Widrow and Hoff

قاعده‌ی دلتای^{۱۹} ویدرو و هاف پیشنهاد کردند. این قاعده اختلاف جمع وزنی و مقدار خروجی مطلوب را محاسبه می‌کند و آن را خطا می‌نامد. تعدیل ضرایب وزنی آن‌گاه به تناسب این خطا انجام می‌شود. مقدار خطای Δ را می‌توان به صورت زیر نوشت:

$$\Delta = d(t) - y(t). \quad (۶.۲)$$

$d(t)$ خروجی مطلوب سیستم و $y(t)$ خروجی واقعی است. چنانچه تصمیم صحیح باشد وزن‌ها تغییر نمی‌کند زیرا $d(t) - y(t) = 0$. الگوریتم فراگیری اصولاً مشابه پرسپترون اولیه است. تنها گام ۴ الگوریتم پرسپترون به صورت زیر تغییر می‌یابد.

۴- ضرایب وزنی زیر را تعدیل کنید. (قاعده‌ی دلتای ویدرو- هاف)

$$\Delta = d(t) - y(t),$$

$$w_i(t+1) = w_i(t) + \eta \Delta x_i(t),$$

$$d(t) = \begin{cases} 1 & \text{اگر ورودی مطلوب ۱ باشد} \\ 0 & \text{اگر ورودی مطلوب ۰ باشد} \end{cases}$$

جایی که $(0 \leq \eta \leq 1)$ باشد عامل بازیابی مثبت است که سرعت تعدیل را کنترل می‌کند.

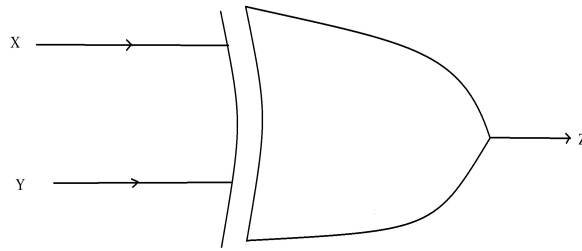
۳.۲ پرسپترون چند لایه‌ای

متأسفانه توانایی پرسپترون محدود است. پرسپترون در صورتی جواب را فرا می‌گیرد که اصولاً جوابی وجود داشته باشد. ملاحظه می‌کنید که پرسپترون به دنبال خطی است که طبقه‌ها را تفکیک کند، لیکن حالت‌های فراوانی است که جدایی طبقات بسیار پیچیده‌تر است. مثلاً مورد یای حذفی^{۲۰} را در نظر بگیرید. تابع منطقی یای حذفی همان‌طور که در شکل (۶.۲) آمده دارای دو ورودی و یک خروجی است. خروجی آن تنها زمانی فعال است که تنها یکی از ورودی‌ها فعال باشد. لیکن اگر هر دو فعال یا هر دو خاموش باشند خروجی خاموش خواهد بود. این نوع الگوها را جدایی‌ناپذیر خطی می‌گوییم.

چگونه باید مشکل عدم توانایی حل مسأله‌های جدایی‌ناپذیر خطی را برطرف کرد؟ در مرحله اول شاید راه حل مشکل استفاده از تعداد بیش‌تری از یک پرسپترون باشد به طوری که هر پرسپترون قسمت کوچکی از فضا را تفکیک کرده و از جمع‌آوری آن‌ها طبقات جدایی‌ناپذیر خطی با موفقیت تفکیک شوند.

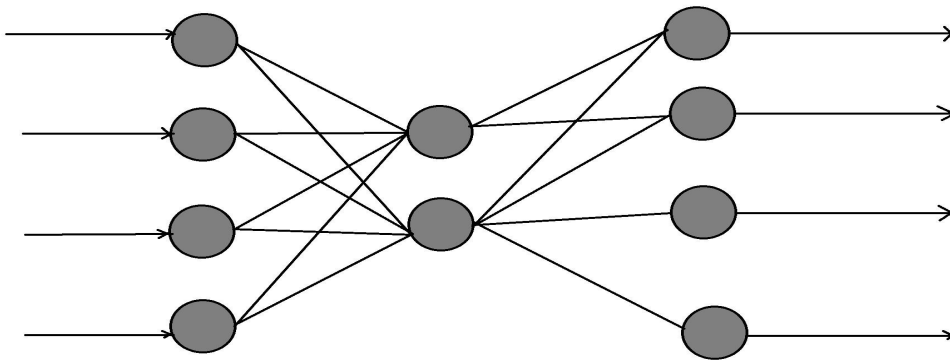
^{۱۹}Delta rule

^{۲۰}Exclusive OR(XOR)



شکل ۶.۲: علامت منطقی یای حذفی

در نرون‌ها لایه‌ی اول ورودی‌ها همان ورودی‌های اصلی شبکه خواهند بود، ولی ورودی‌های لایه‌ی دوم خروجی‌های لایه اول می‌باشند. این بدان معناست که پرسپترون‌های لایه دوم نمی‌دانند که کدام یک از ورودی‌های اصلی فعال و کدام خاموش‌اند. آن‌ها نسبت به ورودی‌های خود که در واقع خروجی‌های لایه اول هستند آگاهی ندارند. پرسپترون‌های جدید به صورت لایه‌ای منظم شده‌اند. طبیعتاً به آن‌ها پرسپترون چندلایه‌ای اطلاق می‌شود این مدل در شکل (۷.۲) آمده است.



شکل ۷.۲: پرسپترون چندلایه‌ای

قاعده فراگیری پرسپترون چندلایه را (قاعده‌ی کلی دلتا) یا (قاعده‌ی پس انتشار^{۲۱}) گویند. این

^{۲۱}Backpropagation

عناوین در سال ۱۹۸۶ توسط رومل هارت^{۲۲}، مککلند^{۲۳} و ویلیامز^{۲۴} پیشنهاد شد و این آغاز تولد دوباره‌ی شبکه‌های عصبی بود. بعدها معلوم شد که نتایج مشابهی نیز قبلاً در سال ۱۹۸۲ توسط پارکر^{۲۵} منتشر شده و همچنین وربس^{۲۶} در سال ۱۹۷۴ کار مشابهی انجام داده است. نحوه عمل پرسپترون چندلایه‌ای مشابه پرسپترون تک‌لایه‌ای است. بدین صورت که الگویی به شبکه عرضه می‌شود و خروجی آن محاسبه می‌گردد، مقایسه خروجی واقعی و خروجی مطلوب باعث می‌گردد که ضرایب وزنی شبکه تغییر یابد به طوری که در دفعات بعد خروجی درست‌تری حاصل شود. قاعده‌ی فراگیری، روش میزان کردن ضرایب وزنی شبکه را بیان می‌کند.

۴.۲ مدل‌های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه‌سازی

بحث و بررسی در مورد کاربرد شبکه‌های عصبی مصنوعی در بهینه‌سازی از اوایل سال ۱۹۸۰ میلادی آغاز شده است. نتایج پژوهش‌های انجام شده از آن زمان تاکنون، مواردی چون برنامه‌ریزی خطی، برنامه‌ریزی درجه دوم، برنامه‌ریزی هندسی و برنامه‌ریزی غیرخطی را در برمی‌گیرد. 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 بیان‌کننده مدل‌های شبکه‌های عصبی مصنوعی متناظر با مسائل بهینه‌سازی هستند. سیستم دینامیکی بیان شده معمولاً یک دستگاه معادلات دیفرانسیل غیرخطی مرتبه اول است. انتظار می‌رود که برای یک نقطه آغازین نقطه پایداری دستگاه معادلات دیفرانسیل غیرخطی به دست آمده، جواب بهینه مسأله بهینه‌سازی اصلی باشد. یک اصل اساسی در استفاده از تابع انرژی این است که برای همگرایی دستگاه معادلات دیفرانسیل به نقطه پایداری آن، تابع انرژی متناظر با آن حتماً باید نامنفی باشد. البته 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 تعادل دستگاه معادلات دیفرانسیل غیرخطی به دست آمده جواب بهینه مسأله اصلی است و برعکس.“

مدل‌های مطرح شده متناظر مسائل مختلف بهینه‌سازی را می‌توان به دو قسمت مدل‌های دوگانی و

^{۲۲}Rumelhart

^{۲۳}McClelland

^{۲۴}Williams

^{۲۵}Parker

^{۲۶}Werbos

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 نظریه توابع جریمه‌ای معمولاً از مدل‌های گرادیانی برای معرفی مدل مورد نظر استفاده می‌شود، ولی در نظریه دوگانی از مدل‌های اولیه - دوگان برای حل این مسائل استفاده می‌شود. اگر بتوانیم متناظر با هر مسأله بهینه‌سازی، مدل دینامیکی مشخصی را ارائه دهیم که نقطه‌ی تعادل مدل ارائه شده برای حل آن مسأله شرایط لازم و کافی را برآورده سازد، آنگاه می‌توانیم 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.

لازم به ذکر است که مطالب بیان شده در این فصل تا اینجا از [۱] آورده شده است.

مدل اول

مسأله برنامه‌ریزی خطی زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} && f(x) = a^T x \\ & \text{subject to} && \\ & && g(x) = Dx - b = 0, \\ & && x \geq 0, \end{aligned} \tag{۷.۲}$$

که در آن $D \in \mathbb{R}^{m \times n}$ و $\text{Rank}(D) = m$ و $a \in \mathbb{R}^n$ ، $x, b \in \mathbb{R}^m$ است. دوگان (۷.۲) به صورت زیر بیان می‌شود:

$$\begin{aligned} & \text{minimize} && \bar{f}(y) = b^T y \\ & \text{subject to} && \\ & && \bar{g}(y) = \bar{D}^T y - a \leq 0, \\ & && y \in \mathbb{R}^n. \end{aligned} \tag{۸.۲}$$

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

$$\begin{cases} \frac{dx}{dt} = -[D^T(Dx - b) - \beta(\bar{D}^T y - a)], \\ \frac{dy}{dt} = -\beta[(Dx - \bar{D}^T y - a)^+ - b], \end{cases} \tag{۹.۲}$$

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

مدل دوم

مسئله برنامه‌ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} \quad f(x) = \frac{1}{\gamma} x^T A x + a^T x \\ & \text{subject to} \quad (10.2) \\ & \quad \quad \quad D x - b = 0, \\ & \quad \quad \quad x \geq 0, \end{aligned}$$

که در آن A یک ماتریس متقارن معین مثبت است. دوگان (۱۰.۲) به صورت زیر است:

$$\begin{aligned} & \text{minimize} \quad \bar{f}(y) = b^T y - \frac{1}{\gamma} x^T A x \\ & \text{subject to} \quad (11.2) \\ & \quad \quad \quad \bar{D} y - A x - a \leq 0. \end{aligned}$$

در [۱۱] مدل متناظر برای حل (۱۰.۲) به صورت زیر نمایش داده شده است:

$$\begin{cases} \frac{dx}{dt} = -\{D^T(x-b) + \gamma(-D^T y + Ax + a) + \gamma A[x - (x + D^T y - Ax - a)^+]\}, \\ \frac{dy}{dt} = -\gamma\{Dx - b + D[(x + D^T y - Ax - a)^+ - x]\}, \end{cases} \quad (12.2)$$

که در آن $\gamma = \|(x + D^T y - Ax - a)^+ - x\|_2^{-1}$ و $(x, y)^T \in \{(x, y)^T \mid x \geq 0\}$.

مدل سوم

مسئله برنامه‌ریزی غیرخطی زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} \quad f(x), \\ & \text{subject to} \quad (13.2) \\ & \quad \quad \quad g(x) = [g_1(x), g_2(x), \dots, g_m(x)]^T \leq 0, \quad i = 1, \dots, m. \end{aligned}$$

که در آن $f(x), g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}^1$ و $f(x), g(x) \in C^2$. برای حل (۱۳.۲) هاپفیلد و تنک^{۲۷} در [۱۲] یک مدل شبکه عصبی به صورت زیر ارائه دادند:

$$\dot{x} = C^{-1} \{-\nabla f(x) - \nabla g(x) g^+(x) - \frac{1}{s} Q^{-1} x\} s, \quad (14.2)$$

^{۲۷}Hopfield and Tank

که در آن C یک ماتریس قطری $n \times n$ بیان‌کننده ظرفیت هر نرون و Q یک ماتریس قطری $n \times n$ است که اعضای آن به صورت زیر تعریف می‌شوند:

$$q_{ij} = \frac{1}{\frac{1}{\rho_i} + \sum_{j=1}^m -d_{ji}}, \quad (15.2)$$

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:

$$E_{\setminus}(x) = f(x) + \sum_{j=1}^m [g_j^+]^2 + \sum_{i=1}^n \frac{x_i^2}{2sr_{ii}}. \quad (16.2)$$

مدل چهارم

برای حل (۱۳.۲) کندی و چوآ^{۲۸} در [۱۵] مدلی دیگر به صورت زیر ارائه داده‌اند:

$$\dot{x} = C^{-1}[-\nabla f(x) - \nabla g(x)g^+(x)], \quad (17.2)$$

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

$$E_{\setminus}(x) = f(x) + \frac{s}{\gamma} \sum_{j=1}^m [g_j^+]^2. \quad (18.2)$$

مدل پنجم

مجدداً برای حل (۱۳.۲) توسط رودریگز واسکوئز^{۲۹} [۱۴] مدلی به صورت زیر ارائه شده است:

$$\dot{x} = -u_x \nabla f(x) - s \nabla g(x)g^+(x), \quad (19.2)$$

که در آن u_x متناظر اندیس‌های شدنی x است چنان‌که

$$u_x = \begin{cases} 1 & g(x) \leq 0, \\ 0 & o.w. \end{cases} \quad (20.2)$$

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

$$E_{\setminus}(x) = -u_x f(x) + \frac{s}{\gamma} \sum_{j=1}^m [g_j^+(x)]^2. \quad (21.2)$$

^{۲۸}Kennedy and Chua

^{۲۹}Rodriguez-Vazquez

همان طور که ملاحظه می‌کنید، مدل‌های بیان شده برای حل مسأله (۱۳.۲) به‌طور متوالی اصلاح شده‌اند و این به‌دلیل مشکلاتی بوده است که هرکدام از این مدل‌ها داشته‌اند.

مدل ششم

برنامه‌ریزی محدب درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} && f(x) = \frac{1}{2}x^T Qx + D^T x \\ & \text{subject to} && \\ & && g(x) = Ax - b \leq 0, \\ & && h(x) = Ex - f = 0. \end{aligned} \quad (22.2)$$

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 کاروش-کان-تاکر، $x^* \in \mathbb{R}^n$ جواب بهینه (۲۲.۲) است اگر و تنها اگر بردارهای $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ وجود داشته باشند به‌طوری‌که در رابطه زیر صدق کنند:

$$\begin{cases} u^* \geq 0, Ax^* - b \leq 0, u^{*T}(Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (23.2)$$

برای حل مسأله (۲۲.۲) تابع زیر را که معروف به تابع فیشر-برمیستر^{۳۰} [۲] است، معرفی می‌کنیم:

$$\begin{cases} \phi : \mathbb{R}^2 \rightarrow \mathbb{R}^1, \\ \phi(a, b) = \sqrt{a^2 + b^2} - a - b. \end{cases} \quad (24.2)$$

قضیه ۱.۴.۲. [۱۷] :

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 به‌جز در مبدأ دو بار به‌طور پیوسته مشتق پذیر است، ولی در مبدأ قویاً هموار است.

تابع ϕ که در شرط (۱) قضیه (۱.۴.۲) صدق کند، تابع مکمل غیرخطی نامیده می‌شود.

^{۳۰}Fisher-Burmister

متناظر با مسائل (۲۲.۲) تابع زیر را در نظر می‌گیریم:

$$\phi(x, u, v) = \begin{bmatrix} \phi(\alpha, Qx + D + A^T u + E^T v) \\ \phi(u, Ax - b) \\ \phi(\beta, Ex - f) \end{bmatrix}, \quad (25.2)$$

که در آن $\alpha = (\alpha_1, \dots, \alpha_m)^T > 0$ ، $u = (u_1, \dots, u_m)$ ، $\beta = (\beta_1, \dots, \beta_l)^T > 0$ است. مدل شبکه عصبی متناظر با (۲۵.۲) به صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = -\tau \nabla E(y(t)), & \tau > 0, \\ y(t_0) = y_0 \in \mathbb{R}^{n+m+l}, \end{cases} \quad (26.2)$$

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 دینامیکی (۲۵.۲) است. تابع انرژی متناظر این مدل به صورت زیر انتخاب شده است:

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

مدل هفتم

برای طراحی مدل هفتم، مجدداً شرایط کاروش-کان-تاکر را برای حل مسائل (۲۲.۲) به صورت زیر در نظر می‌گیریم:

$$\begin{cases} (u + Ax - b)^+ = u, \\ Qx + D + A^T u + E^T v = 0, \\ Ex - f = 0. \end{cases} \quad (27.2)$$

آنگاه برای حل (۲۲.۲) دستگاه زیر تعریف می‌شود:

$$U(x, u, v) = \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}. \quad (28.2)$$

مدل شبکه عصبی برای حل (۲۲.۲) در [۲] به صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = \kappa U(y), \\ y(t_0) = y_0. \end{cases} \quad (29.2)$$

که در آن κ ضریب همگرایی سیستم است.

مدل هشتم

مسئله برنامه ریزی درجه دوم (۲۲.۲) را در نظر بگیرید. که در آن فرض می‌کنیم $Q \in \mathbb{R}^{n \times n}$ یک ماتریس نیمه معین مثبت است. برای حل مسئله (۲۲.۲) در [۱۰] یک مدل شبکه عصبی به صورت زیر ارائه می‌شود:

$$\frac{dy}{dt} = (I + M^T) \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}, \quad (۳۰.۲)$$

که در آن κ نرخ همگرایی است و

$$M = \begin{bmatrix} -Q & A^T & E^T \\ A & \circ_{m \times m} & \circ_{m \times l} \\ E & \circ_{l \times m} & \circ_{l \times l} \end{bmatrix}. \quad (۳۱.۲)$$

مدل نهم

مسئله برنامه‌ریزی درجه دوم زیر را در نظر بگیرید:

$$\begin{aligned} &\text{minimize} && f(x) = \frac{1}{2} x^T A x + a^T x \\ &\text{subject to} && \\ &&& D x \leq b, \\ &&& x \geq \circ. \end{aligned} \quad (۳۲.۲)$$

دوگان آن عبارت است از:

$$\begin{aligned} &\text{minimize} && \bar{f}(w) = b^T w - \frac{1}{2} x^T A x \\ &\text{subject to} && \\ &&& \bar{D}^T w - A x \leq a, \\ &&& w \geq \circ, \end{aligned} \quad (۳۳.۲)$$

که در آن متقارن و نیمه معین مثبت است، $A \in \mathbb{R}^{n \times n}$ ، $a \in \mathbb{R}^n$ ، $b \in \mathbb{R}^m$ و $D \in \mathbb{R}^{m \times n}$ است. با توجه به شرایط کاروش-کان-تاکر برای مسائل برنامه‌ریزی محدب، $(x^{*T}, w^{*T})^T$ به ترتیب یک

جواب بهینه برای (۳۲.۲) است اگر و فقط اگر $(x^{*T}, w^{*T})^T$ در شرایط کاروش-کان-تاکر زیر صدق کند:

$$\begin{cases} w^* \geq 0, Dx^* - b \leq 0, w^{*T}(Dx^* - b) = 0, \\ Ax^* + a + D^T w^* \geq 0, x \geq 0, \\ x^{*T}(Ax^* + a + D^T w^*) = 0. \end{cases} \quad (34.2)$$

مدل متناظر با (۳۲.۲) و (۳۳.۲) در [۴] به صورت زیر بیان شده است:

$$\dot{u} = B(I + M^T)\{(u - (Mu + q))^+ - u\}, \quad (35.2)$$

که در آن

$$u = \begin{bmatrix} x \\ w \end{bmatrix}, q = \begin{bmatrix} a \\ b \end{bmatrix}, M = \begin{bmatrix} A & D^T \\ -D & 0 \end{bmatrix}. \quad (36.2)$$

M یک ماتریس نیمه معین مثبت است زیرا $x^T Ax \geq 0$ و $u^T Mu = x^T Ax \geq 0$ همچنین $B = \lambda I$ است که $\lambda > 0$ و با افزایش λ سرعت همگرایی افزایش می‌یابد.

فصل ۳

حل فرم کلی مسائل بهینه‌سازی محدب با یک
مدل عصبی دینامیکی کارآمد

۱.۳ مقدمه

در این فصل ساختار یک مدل از شبکه‌های عصبی، بر پایه قضایای دوگانی، بهینه‌سازی، تجزیه و تحلیل توابع محدب، پایداری لیاپانوف و اصل تغییرناپذیری لازال برای حل مسائل برنامه‌ریزی غیرخطی محدب ارائه می‌شود و ثابت می‌شود که نقطه‌ی تعادل شبکه‌ی عصبی پیشنهاد شده همان جواب بهینه‌ی مسأله برنامه‌ریزی غیرخطی محدب است. همچنین نشان داده می‌شود که شبکه‌ی عصبی پیشنهادی دارای پایداری به مفهوم لیاپانوف است و به‌طور دقیق همگرایی سراسری به یک جواب بهینه مسأله اصلی است. کارآمدی مدل پیشنهادی با ارائه چندین مثال نشان داده می‌شود.

۲.۳ یک مدل شبکه عصبی

مسأله برنامه‌ریزی غیرخطی محدب زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && \\ & && g(x) \leq 0, \\ & && h(x) = 0, \end{aligned} \tag{۱.۳}$$

که در آن $x \in \mathbb{R}^n$ و $f: \mathbb{R}^n \rightarrow \mathbb{R}$ و $g(x) = (g_1(x), \dots, g_m(x))^T$ یک بردار m بعدی است. توابع $g_1(x), \dots, g_m(x)$ و $f(x)$ محدب و دوبار مشتق‌پذیرند، تابع $h(x) = Ax - b$ که $b \in \mathbb{R}^l$ و $A \in \mathbb{R}^{l \times n}$ ($0 < l < n$, $\text{Rank}(A) = l$) است. فرض می‌کنیم مسأله (۱.۳) دارای جواب بهینه متناهی است و در شرایط اسلاتر^۱ [۲] صدق می‌کند. یعنی وجود دارد $x^\circ \in \mathbb{R}^n$ که

$$g(x^\circ) \leq 0, h(x^\circ) = 0.$$

در این قسمت با استفاده از روش‌های استاندارد، یک مدل دینامیکی متناظر مسأله (۱.۳) به دست می‌آوریم.

از [۵] می‌بینیم که $x^* \in \mathbb{R}^n$ جواب بهینه مسأله (۱.۳) است اگر و تنها اگر $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ موجود باشند به طوری که $(x^{*T}, u^{*T}, v^{*T})^T$ در شرایط کاروش-کان-تاکر زیر صدق کند:

^۱Slater

$$\begin{aligned} u^* &\geq 0, g(x^*) \leq 0, u^{*T}g(x^*) = 0, \\ \nabla f(x^*) + \nabla g(x^*)^T u^* + \nabla h(x^*)^T v^* &= 0, \\ h(x^*) &= 0. \end{aligned} \quad (۲.۳)$$

x^* نقطه‌ی کاروش-کان-تاکر مسأله (۱.۳) و زوج مرتب $(u^{*T}, v^{*T})^T$ بردار ضرایب لاگرانژین مربوط به x^* نامیده می‌شود. همچنین اگر $f(x)$ و $g_k(x)$ ($k = 1, \dots, m$) توابعی محدب باشند، در این صورت x^* جواب بهینه (۱.۳) است اگر و تنها اگر x^* نقطه کاروش-کان-تاکر (۱.۳) باشد. فرض کنید $u(\cdot)$ ، $x(\cdot)$ و $v(\cdot)$ متغیرهای وابسته به زمان باشند. هدف، طراحی یک سیستم دینامیکی پیوسته زمانی است آن چنان که منجر به یافتن نقطه کاروش-کان-تاکر مسأله برنامه‌ریزی غیرخطی محدب (۱.۳) و دوگانش شود. در اینجا یک مدل شبکه‌ی عصبی برای حل مسأله (۱.۳) و دوگان آن به صورت زیر ارائه می‌شود:

$$\begin{aligned} \frac{dx}{dt} &= -(\nabla f(x) + \nabla g(x)^T(u + g(x))^+ + \nabla h(x)^T v), \\ \frac{du}{dt} &= (u + g(x))^+ - u, \\ \frac{dv}{dt} &= h(x), \end{aligned} \quad (۳.۳)$$

با نقطه اولیه $y_0 = (x_0^T, u_0^T, v_0^T)^T$ که در آن

$$\begin{aligned} (u + g(x))^+ &= ([u_1 + g_1(x)]^+, [u_2 + g_2(x)]^+, \dots, [u_m + g_m(x)]^+), \\ [u_k + g_k(x)]^+ &= \max\{u_k + g_k(x), 0\}, \quad k = 1, 2, \dots, m. \end{aligned}$$

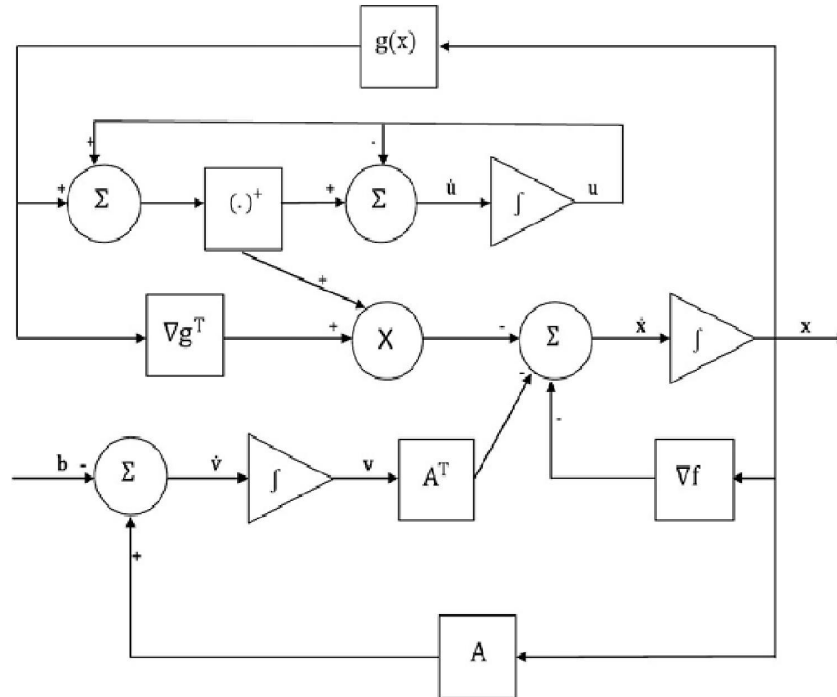
برای سادگی بحث، نقطه بهینه مسأله (۱.۳) را با $y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ و مجموعه جواب اولیه و دوگان آن را با نماد D^* نشان می‌دهیم. $\psi(y)$ را به صورت زیر تعریف می‌کنیم:

$$\psi(y) = \begin{bmatrix} -(\nabla f(x) + \nabla g(x)^T(u + g(x))^+ + \nabla h(x)^T v) \\ (u + g(x))^+ - u \\ h(x) \end{bmatrix}. \quad (۴.۳)$$

بنابراین شبکه‌ی عصبی (۳.۳) را می‌توان به صورت زیر نوشت:

$$\begin{aligned} \frac{dy}{dt} &= \tau \psi(y), \\ y(t_0) &= y_0, \end{aligned} \quad (۵.۳)$$

که در آن τ یک پارامتر مقیاس است و نشان دهنده نرخ همگرایی شبکه‌ی عصبی (۵.۳) است. برای ساده‌تر شدن تجزیه و تحلیل فرض می‌کنیم $\tau = 1$. شکل (۱.۳) نمایش مداری شبکه‌ی عصبی (۵.۳) را نشان می‌دهد.



شکل ۱.۳: دیاگرام ساده شده برای شبکه‌ی عصبی (۵.۳)

برای اینکه بینیم مدل شبکه‌ی عصبی ارائه شده در (۵.۳) چگونه می‌تواند مسائل برنامه‌ریزی غیرخطی محدب را حل نماید آن را با سه مدل شبکه‌ی عصبی دیگر مقایسه می‌کنیم. ابتدا مسأله زیر را در نظر بگیرید:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && \\ &&& g(x) \leq 0, \\ &&& x \in \Omega, \end{aligned} \tag{۶.۳}$$

که در آن $x \in \mathbb{R}^n$ و $\Omega = \{x \in \mathbb{R}^n : l_i \leq x_i \leq v_i, i = 1, 2, \dots, n\}$ و ممکن است که v_i (یا $-l_i$) برابر $+\infty$ باشد. همچنین $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T \in \mathbb{R}^m$ و توابع

$f(x)$ و $g_1(x), \dots, g_m(x)$ توابعی محدب روی Ω و دوبار به‌طور پیوسته مشتق‌پذیر روی بازه محدب $\varepsilon \subseteq \mathbb{R}^n$ شامل Ω می‌باشند.

در [۸] از مدل شبکه‌ی عصبی زیر برای حل (۶.۳) استفاده شده است.

$$\begin{aligned} \frac{dx}{dt} &= -(x - P_\Omega(x - \nabla f(x) - \nabla g(x)^T u)), \\ \frac{du}{dt} &= -(u - (u + g(x))^+), \end{aligned} \quad (۷.۳)$$

که در آن $\Omega \subset \mathbb{R}^n$ یک مجموعه محدب بسته و $P_\Omega : \mathbb{R}^n \rightarrow \Omega$ عملگر تصویر است [۹] که به‌صورت زیر تعریف می‌شود:

$$P_\Omega = \operatorname{argmin}_{v \in \Omega} \|x - v\|. \quad (۸.۳)$$

البته هیچ تضمینی برای پایداری مدل دینامیکی (۷.۳) برای نگاشت‌های یکنوا و متقارن وجود ندارد [۲۶]. بنابراین دستگاه دینامیکی مطرح‌شده در (۷.۳) نمی‌تواند در بعضی موارد، مسأله (۶.۳) را حل کند. برای توضیحات بیشتر، مثال‌های (۱.۴.۳) و (۲.۴.۳) را ببینید.

نوع دیگری از مدل شبکه عصبی، مدل گرادینانی می‌باشد. در مدل شبکه‌ی عصبی گرادینانی، یک مسأله بهینه‌سازی مقید را می‌توان بوسیله یک مسأله بهینه‌سازی نامقید تقریب زد. سپس تابع انرژی را با استفاده از روش تابع جریمه ساخت. لازم به ذکر است که مدل شبکه‌ی عصبی گرادینانی نسبت به مدل‌هایی که با استفاده از تعریف مستقیم مشتقات تابع انرژی بیان شده‌اند دارای مزیت است. اما نقطه ضعف آن در عدم تضمین همگرایی، به‌خصوص در مجموعه جواب‌های بی‌کران می‌باشد [۳۰]. علاوه بر این، مدل شبکه‌ی عصبی بر مبنای تابع جریمه نیاز به یک پارامتر قابل تنظیم^۲ به نام پارامتر جریمه دارد. تقریب مسأله مقید (۱.۳) با استفاده از مسأله نامقید بهینه‌سازی زیر، یک نمونه از کاربرد روش تابع جریمه است.

$$\text{minimize } E_\gamma(x) = f(x) + \frac{\gamma}{2} \left\{ \sum_{k=1}^m [g_k^+(x)]^2 + \sum_{p=1}^l h_p^+(x) \right\},$$

که در آن γ پارامتر جریمه است. مدل شبکه‌ی عصبی گرادینان با روش زیر به‌دست می‌آید:

$$\frac{dx}{dt} = -\nabla E_\gamma(x) = -(\nabla f(x) + \gamma[\nabla g(x)g^+(x) + \nabla h(x)h(x)]). \quad (۹.۳)$$

^۲Adjustable

سیستم ارائه شده در (۹.۳) به مدل کندی و چاو معروف^۳ است که در [۹] بیان شده است. این شبکه‌ی عصبی به علت دارا بودن پارامتر جریمه‌ای متناهی، توانایی یافتن یک جواب بهینه دقیق را ندارد و اگر پارامتر جریمه‌ای خیلی بزرگ باشد، یافتن یک جواب بهینه دقیق دشوار است [۱۶]. بنابراین این شبکه‌ی عصبی زمانی همگرا به یک جواب تقریبی برای مسأله (۱.۳) است که پارامتر جریمه متناهی باشد. همچنین می‌توان نشان داد که شبکه‌ی عصبی (۹.۳) در بعضی مسائل برنامه‌ریزی محدب، همگرای سراسری به یک جواب بهینه دقیق نیست. برای توضیح بیشتر مثال‌های (۲.۴.۳) و (۳.۴.۳) و مثال ۵.۱ در [۲۱] را ببینید.

با شرط محدب اکید بودن $f(x)$ و محدب بودن $g_k(x)$ ($k = 1, \dots, m$)، یا محدب بودن $f(x)$ و محدب اکید بودن $g_k(x)$ ($k = 1, \dots, m$)، مدل شبکه عصبی گرادیانی زیر [۲۱] برای حل (۱.۳) ارائه می‌شود:

$$\begin{aligned} \frac{dx}{dt} &= -(\nabla f(x) + \frac{1}{\rho} \nabla g(x)^T u^2 + \nabla h(x)^T v), \\ \frac{du}{dt} &= \text{diag}(u_1, \dots, u_m) g(x), \\ \frac{dv}{dt} &= h(x), \end{aligned} \quad (10.3)$$

با نقطه اولیه $(x^T, u^T, v^T)^T$ و $u_k(t_0) \neq 0$ ($k = 1, \dots, m$) این مدل نمی‌تواند بعضی از مسائل بهینه‌سازی محدب را حل نماید. توضیحات بیشتر در مثال (۱.۴.۳) بیان شده است. برای مقایسه این مدل‌ها با مدل پیشنهادی در (۵.۳)، می‌توان به همگرایی سراسری بودن مدل پیشنهادی برای حل مسائل (۱.۳) اشاره کرد.

۳.۳ تحلیل پایداری و همگرایی

در این قسمت پایداری و همگرایی مدل (۵.۳) بیان می‌گردد.

قضیه ۱.۳.۳. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل شبکه‌ی عصبی (۵.۳) باشد آنگاه y^* نقطه‌ی کاروش-کان-تاکر مسأله (۱.۳) است. برعکس اگر $x^* \in \mathbb{R}^n$ جواب بهینه مسأله (۱.۳) باشد، آنگاه $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ وجود دارند به طوری که نقطه $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل مدل (۵.۳) می‌باشد.

برهان. فرض کنید $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل شبکه‌ی عصبی (۵.۳) باشد، آنگاه $\frac{dx^*}{dt} = 0$ و $\frac{du^*}{dt} = 0$ و $\frac{dv^*}{dt} = 0$. در نتیجه داریم:

^۳Kennedy and Chua

$$\nabla f(x^*) + \nabla g(x^*)^T (u^* + g(x^*))^+ + \nabla h(x^*)^T v^* = \circ, \quad (11.3)$$

$$(u^* + g(x^*))^+ = u^*, \quad (12.3)$$

$$h(x^*) = \circ. \quad (13.3)$$

واضح است که $(u^* + g(x^*))^+ = u^*$ اگر و تنها اگر

$$u^* \geq \circ, \quad g(x^*) \leq \circ, \quad u^{*T} g(x^*) = \circ. \quad (14.3)$$

با جایگزینی (۱۲.۳) در (۱۱.۳) داریم:

$$\nabla f(x^*) + \nabla g(x^*)^T u^* + \nabla h(x^*)^T v^* = \circ. \quad (15.3)$$

از (۱۳.۳) - (۱۵.۳) می‌فهمیم که $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ در شرایط کاروش - کان - تاکر صدق می‌کند.

□

اثبات برعکس واضح است.

لم ۱.۳.۳. برای هر نقطه اولیه $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ ، یک جواب پیوسته منحصر بفرد $y(t) = (x(t)^T, u(t)^T, v(t)^T)^T$ برای سیستم (۵.۳) موجود است.

برهان. از آنجا که $\nabla f(x)$ ، $\nabla g_k(x)$ ($k = 1, \dots, m$) و $\nabla h_p(x)$ ($p = 1, \dots, l$) به‌طور پیوسته بر روی مجموعه‌ی باز $D \subseteq \mathbb{R}^{n+m+l}$ مشتق‌پذیرند، در نتیجه $\nabla f(x) + \nabla g(x)^T (u + g(x))^+ + \nabla h(x)^T v$ ، $h(x)$ پیوسته لیپ‌شیتز موضعی می‌باشند. با توجه به قضیه وجود مشتقات موضعی [۲۰]، سیستم (۵.۳) دارای جواب منحصر بفرد پیوسته $y(t)$ است که در آن $t \in [t_0, \eta]$ برای هر $\eta > t_0$ وقتی $\eta \rightarrow \infty$.

□

لم ۲.۳.۳. ماتریس ژاکوبی $\nabla \psi(y)$ از نگاشت ψ تعریف شده در (۴.۳) نیمه معین منفی است.

برهان. بدون کاستن از کلیت مسأله، فرض می‌کنیم $\circ < p < m$ وجود دارد به‌طوری که

$$(u + g)^+ = \left(u_1 + g_1(x), u_2 + g_2(x), \dots, u_p + g_p(x), \underbrace{\circ, \circ, \dots, \circ}_{m-p} \right) \geq \circ.$$

با محاسبات ساده می‌توان نشان داد که

$$\nabla \psi(y) = \begin{bmatrix} -(\nabla^{\vee} f(x) + \sum_{k=1}^p ((u_k + g_k) \nabla^{\vee} g_k^p(x)) + \nabla g^p(x)^T \nabla g^p(x)) & -\nabla g^p(x)^T & -\nabla h(x)^T \\ \nabla g^p(x) & S_{m \times m} & O_{m \times l} \\ \nabla h(x) & O_{l \times m} & O_{l \times l} \end{bmatrix},$$

که در آن

$$\nabla g^p(x) = \begin{bmatrix} U_{p \times n} \\ O_{(m-p) \times n} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_{p-1}} & \frac{\partial g_1}{\partial x_p} & \frac{\partial g_1}{\partial x_{p+1}} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \cdots & \frac{\partial g_2}{\partial x_{p-1}} & \frac{\partial g_2}{\partial x_p} & \frac{\partial g_2}{\partial x_{p+1}} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_p}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_{p-1}} & \frac{\partial g_p}{\partial x_p} & \frac{\partial g_p}{\partial x_{p+1}} & \cdots & \frac{\partial g_p}{\partial x_n} \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \end{bmatrix},$$

و $\nabla^2 g_k^p(x)$ برای $(k = 1, 2, \dots, p)$ عبارت است از:

$$\nabla^2 g_k^p(x) = \begin{bmatrix} \frac{\partial^2 g_k}{\partial x_1^2} & \cdots & \frac{\partial^2 g_k}{\partial x_1 \partial x_{p-1}} & \frac{\partial^2 g_k}{\partial x_1 \partial x_p} & \frac{\partial^2 g_k}{\partial x_1 \partial x_{p+1}} & \cdots & \frac{\partial^2 g_k}{\partial x_1 \partial x_n} \\ \frac{\partial^2 g_k}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 g_k}{\partial x_2 \partial x_{p-1}} & \frac{\partial^2 g_k}{\partial x_2 \partial x_p} & \frac{\partial^2 g_k}{\partial x_2 \partial x_{p+1}} & \cdots & \frac{\partial^2 g_k}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 g_k}{\partial x_p \partial x_1} & \cdots & \frac{\partial^2 g_k}{\partial x_p \partial x_{p-1}} & \frac{\partial^2 g_k}{\partial x_p \partial x_p} & \frac{\partial^2 g_k}{\partial x_p \partial x_{p+1}} & \cdots & \frac{\partial^2 g_k}{\partial x_p \partial x_n} \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \\ \circ & \cdots & \circ & \circ & \circ & \cdots & \circ \end{bmatrix},$$

و

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

که در آن O ماتریس صفر است. ماتریس $\nabla g^p(x)^T \nabla g^p(x)$ نیمه معین مثبت است [۲۳]. چون توابع f, g_1, \dots, g_m محدب و دوبار مشتق پذیر فرض شده اند، بنابراین ماتریس های هسین $\nabla^2 f(x)$ و $\nabla^2 g_k(x)$ برای $(k = 1, 2, \dots, p)$ ، ماتریس های نیمه معین مثبت هستند. بعلاوه از نیمه معین مثبت بودن $\nabla^2 g_k(x)$ می توان گفت که ماتریس های $\nabla^2 g_k^p(x)$ نیمه معین مثبت هستند. همچنین واضح است که ماتریس $S_{m \times m}$ ، نیمه معین منفی است.

طبق توضیحات گفته شده، ما می توانیم نتیجه بگیریم ماتریس ژاکوبی $\nabla \psi(y)$ ، نیمه معین منفی است.

اگر $p = m$ یعنی $(u + g)^+ = (u_1 + g_1(x), u_2 + g_2(x), \dots, u_m + g_m(x))$ ، آنگاه داریم:

$$\nabla\psi(y) = \begin{bmatrix} -(\nabla^2 f(x) + \sum_{k=1}^m ((u_k + g_k) \nabla^2 g_k(x)) + \nabla g(x)^T \nabla g(x)) & -\nabla g(x)^T & -\nabla h(x)^T \\ \nabla g(x) & O_{m \times m} & O_{m \times l} \\ \nabla h(x) & O_{l \times m} & O_{l \times l} \end{bmatrix},$$

که در آن $\nabla g(x) = \nabla g^m(x)$ ماتریس ژاکوبی تابع برداری $g(x)$ و $\nabla^2 g_k(x) = \nabla^2 g_k^m(x)$ ماتریس هسین $g_k(x)$ برای $k = 1, 2, \dots, m$ است. مشابه حالت قبلی می‌توان به سادگی نشان داد که $\nabla\psi(y)$ ماتریس نیمه معین منفی است.

سرانجام، اگر $p = 0$ ، یعنی $(u + g)^+ = (\underbrace{0, 0, \dots, 0}_m)$ ، آنگاه داریم:

$$\nabla\psi(y) = \begin{bmatrix} -\nabla^2 f(x) & O_{n \times m} & -\nabla h(x)^T \\ O_{m \times n} & -I_{m \times m} & O_{m \times l} \\ \nabla h(x) & O_{l \times m} & O_{l \times l} \end{bmatrix}.$$

در این حالت هم، می‌توان نشان داد که $\nabla\psi(y)$ ماتریس نیمه معین منفی است، این اثبات را کامل می‌کند. \square

قضیه ۲.۳.۳. مدل شبکه‌های عصبی (۵.۳) پایدار سراسری به مفهوم لیاپانوف و همگرایی سراسری به $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ است که جواب بهینه (۱.۳) است.

برهان. تابع لیاپانوف زیر را در نظر بگیرید:

$$E(y) = \|\psi(y)\|^2 + \frac{1}{\gamma} \|y - y^*\|^2. \quad (۱۶.۳)$$

از (۴.۳)، داریم:

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

بنابراین

$$\begin{aligned} \frac{dE(y(t))}{dt} &= \left(\frac{d\psi}{dt}\right)^T \psi + \psi^T \left(\frac{d\psi}{dt}\right) + (y - y^*)^T \frac{dy(t)}{dt} \\ &= \psi^T (\nabla\psi(y)^T + \nabla\psi(y))\psi + (y - y^*)^T \psi(y). \end{aligned}$$

با به کار بردن لم (۲.۳.۳) داریم:

$$\psi^T(y) (\nabla\psi(y)^T + \nabla\psi(y))\psi(y) \leq 0, \quad \forall y \neq y^*. \quad (۱۷.۳)$$

با استفاده از تعریف (۷.۶.۱) و لم (۳.۶.۱) داریم:

$$(y - y^*)^T(\psi(y) - \psi(y^*)) = (y - y^*)^T \psi(y) \leq 0, \quad \forall y \neq y^*.$$

بنابراین

$$\frac{dE(y(t))}{dt} \leq 0. \quad (18.3)$$

یعنی شبکه عصبی (۵.۳)، پایدار سراسری به مفهوم لیاپانوف است. در ادامه چون

$$E(y) \geq \frac{1}{\rho} \|y - y^*\|^2, \quad (19.3)$$

زیردنباله‌های همگرای

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

وقتی $t_k \rightarrow \infty$ از $k \rightarrow \infty$ وجود دارد، به طوری که

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

و $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ در رابطه زیر صدق می‌کند:

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

این نشان می‌دهد که $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ ، یک نقطه $-\omega$ حدی از $\{(x(t)^T, u(t)^T, v(t)^T)^T | t \geq t_0\}$ است. با استفاده از اصل تغییر ناپذیری لازال (۱.۸.۱) داریم $\{(x(t)^T, u(t)^T, v(t)^T)^T \rightarrow M\}$ برای $t \rightarrow \infty$ که در آن M بزرگترین مجموعه ناوردا در $\{(x(t)^T, u(t)^T, v(t)^T)^T | \frac{dE(y(t))}{dt} = 0\}$ است. از (۵.۳) و (۱۸.۳) داریم

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

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

با جایگذاری $x^* = \bar{x}$ ، $u^* = \bar{u}$ و $v^* = \bar{v}$ در (۱۶.۳)، تابع لیاپانوف دیگری به صورت زیر تعریف می‌کنیم:

$$\bar{E}(y) = \|\psi(y)\|^2 + \frac{1}{\rho} \|y - \bar{y}\|^2. \quad (20.3)$$

آنگاه $\bar{E}(y)$ به طور پیوسته مشتق پذیر است و $\bar{E}(\bar{y}) = 0$. از طرفی

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

از اینرو داریم

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

بنابراین به ازای هر $\epsilon > 0$ ، $q > 0$ وجود دارد به طوری که برای هر $t \geq t_q$ ، داریم $\bar{E}(y(t)) < \epsilon$. به طور مشابه می‌توانیم به دست بیاوریم $\frac{d\bar{E}(y(t))}{dt} \leq 0$ برای $t \geq t_q$ داریم:

$$\frac{1}{4} \|y(t) - \bar{y}\|^2 \leq \bar{E}(y(t)) \leq \epsilon.$$

در نتیجه $\lim_{t \rightarrow \infty} \|y(t) - \bar{y}\| = 0$ یا $\lim_{t \rightarrow \infty} y(t) = \bar{y}$.

بنابراین شبکه عصبی پیشنهاد شده در (۵.۳) به نقطه تعادل $\bar{y} = (\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ همگرای سراسری است که در آن \bar{x} جواب بهینه (۱.۳) است. □

از قضیه (۲.۳.۳)، نتیجه‌ی زیر به دست می‌آید.

نتیجه ۱.۳.۳. اگر $D^* = \{(x^{*T}, u^{*T}, v^{*T})^T\}$ ، آنگاه شبکه عصبی (۵.۳) برای حل مسأله (۱.۳)، پایدار مجانبی سراسری به نقطه تعادل یکتای $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ است.

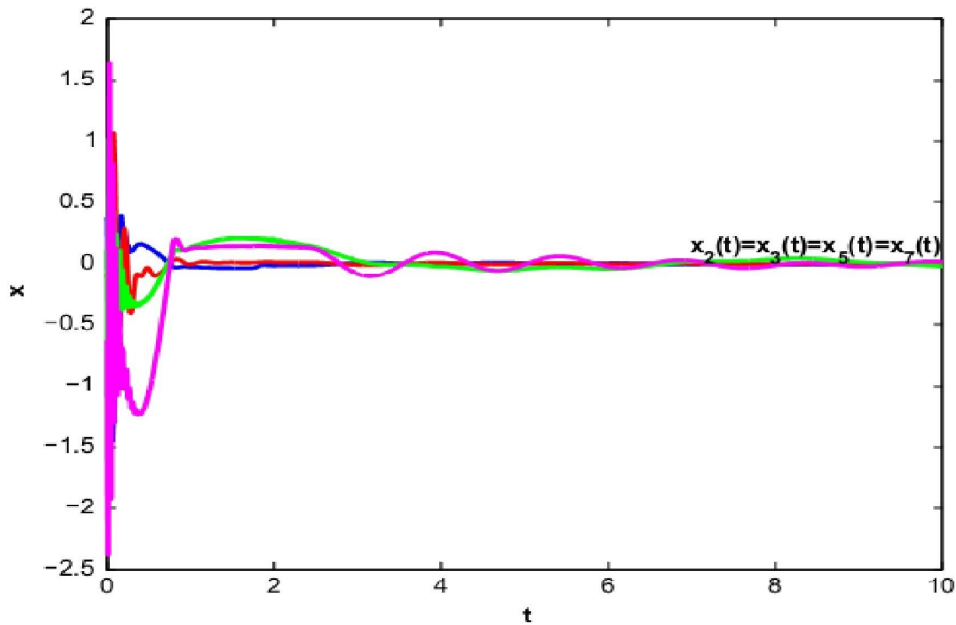
۴.۳ مثال عددی

به منظور نشان دادن کارایی و تأثیر شبکه عصبی پیشنهاد شده (۵.۳)، در این قسمت، ما به ذکر چندین مثال عددی می‌پردازیم. برای برخی مثال‌ها، مقایسه عملکرد عددی از شبکه عصبی پیشنهادی را با مقادیر گوناگون از حالت اولیه $y(0)$ داریم. برای برخی دیگر از مسائل ما مقایسه عملکرد عددی از شبکه عصبی پیشنهاد شده (۵.۳) با مقادیر گوناگون از نرخ همگرایی κ و مقادیر مختلف $\|x(t) - x^*\|^2$ با نقطه آغازین مشخص y_0 انجام دادیم. و مدل‌های موجود در [۱۵]، [۸] و [۲۱] را در بعضی از مثال‌ها با هم مقایسه کردیم. نتایج شبیه‌سازی در پکیج Ode45s، نرم افزار Matlab انجام شده و جواب واقعی مسائل نیز توسط نرم افزار Lingo ۱۱ محاسبه شده است.

مثال ۱.۴.۳.

$$\begin{aligned} \text{minimize} \quad & -\frac{3}{4}x_4 + 2x_5 - \frac{1}{4}x_6 + 6x_7 \\ \text{subject to} \quad & \begin{cases} x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0, \\ x_2 + \frac{1}{4}x_4 - 12x_5 - \frac{1}{4}x_6 + 3x_7 = 0, \\ x_3 + x_6 - 1 = 0, \\ x \geq 0. \end{cases} \end{aligned}$$

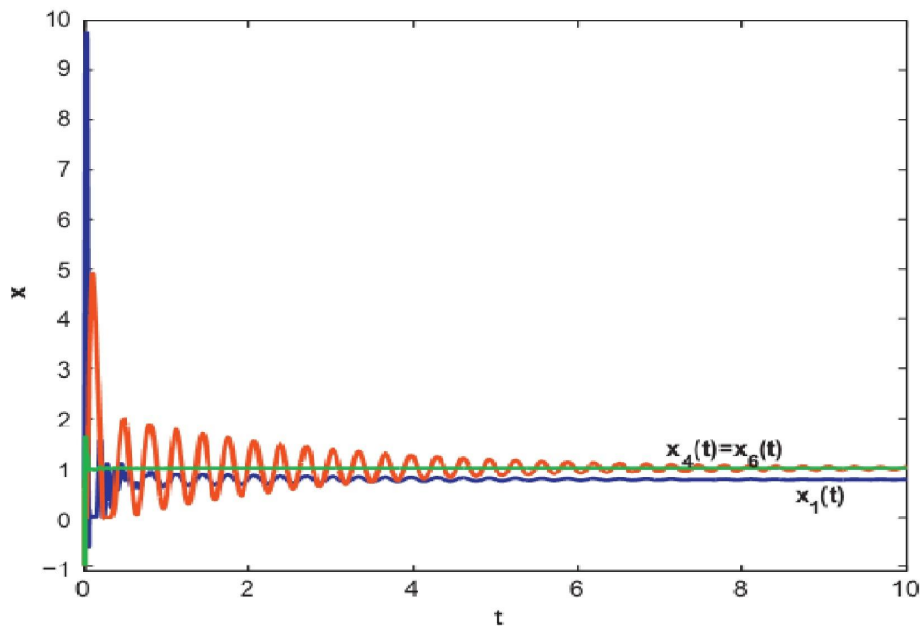
این مثال، یک مسأله برنامه‌ریزی خطی است و به جواب بهینه $x^* = (0.75, 0, 0, 1, 0, 1, 0)^T$ همگرا است. نمودار (۲.۳) و (۳.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ مسیره‌های مدل شبکه‌ی عصبی (۵.۳) را نشان می‌دهد که همگرا به جواب بهینه است. برای مقایسه، این مثال با استفاده از مدل شبکه‌ی عصبی (۷.۳) حل شده است. واضح است که این مدل با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ پایدار نیست. نمودار (۴.۳) نشان‌دهنده رفتار آن است. این مثال با استفاده از مدل شبکه‌ی عصبی ارائه شده در (۱۰.۳) حل شده است. که در آن نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ است. نمودار (۵.۳) نشان‌دهنده رفتار این مدل است، با توجه به این نمودار می‌بینیم که مدل (۱۰.۳) برای حل مسائل برنامه‌ریزی خطی مناسب نیست.



شکل ۲.۳: رفتار $x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t), x_7(t)$ با استفاده از مدل (۵.۳) در مثال (۱.۴.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$.

مثال ۲.۴.۳. [۹]

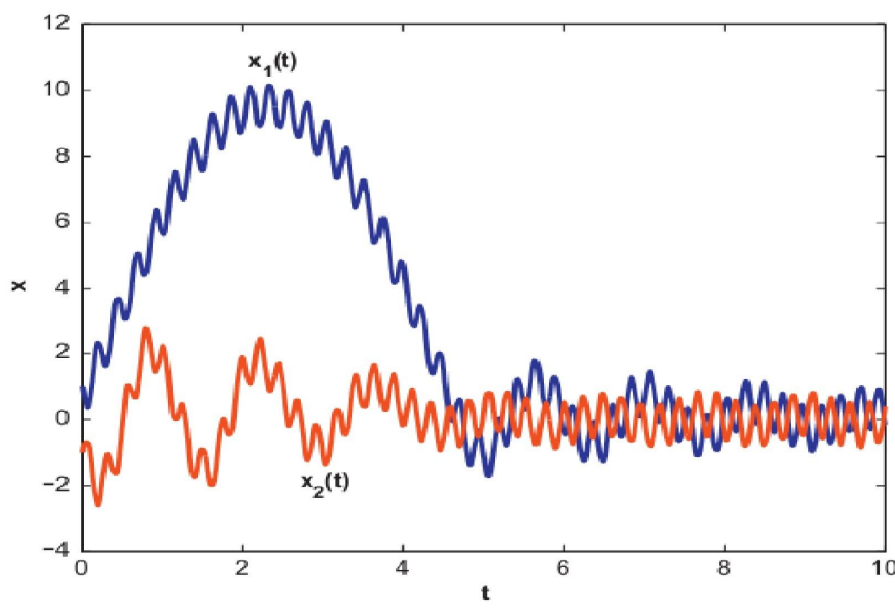
$$\begin{aligned} & \text{minimize} && \frac{1}{4}(x_1 + x_2)^4 - 16x_2 \\ & \text{subject to} && \begin{cases} -x_1 + x_2 \leq 0, \\ -3 \leq x_i \leq 3, \quad i = 1, 2. \end{cases} \end{aligned}$$



شکل ۳.۳: رفتار $x_1(t)$, $x_4(t)$ و $x_6(t)$ با استفاده از مدل (۵.۳) در مثال (۱.۴.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$

جواب بهینه این مسأله عبارت است از $x^* = (1, 1)^T$. شبکه‌ی عصبی (۵.۳) را برای حل این مسأله به‌کار می‌بریم. نتایج شبیه‌سازی نشان می‌دهد که مسیرهای حل با هر نقطه‌ی اولیه، همگرا به نقطه $y^* = (x^{*T}, u^{*T})^T$ است. نمودار (۶.۳) رفتار $x(t)$ بر مبنای شبکه‌ی عصبی (۵.۳) را با 5° نقطه‌ی اولیه‌ی تصادفی نشان می‌دهد. همچنین نمودار (۷.۳) نشان‌دهنده‌ی خطای نرم l_2 بین x و x^* با 1° نقطه‌ی اولیه‌ی تصادفی است.

برای مقایسه، مسئله فوق با استفاده از مدل دینامیکی (۷.۳) در مثال ۴ در [۹] حل شده است و با بررسی آن می‌بینیم که مدل شبکه‌های عصبی (۷.۳) پایدار نیست. بعلاوه مثال فوق در [۹] با مدل‌کنندگی و چاو حل شده است. نتایج به‌دست آمده نشان می‌دهد که مدل شبکه‌ی عصبی (۹.۳) به نقطه‌ی تعادل $(1 + \frac{4}{\gamma}, 1 - \frac{4}{\gamma})^T$ همگراست که می‌تواند به‌عنوان یک جواب تقریبی از مسأله فوق باشد. هنگامی که γ بزرگ باشد، می‌توان نشان داد که این نقطه تعادل، شدنی نیست.



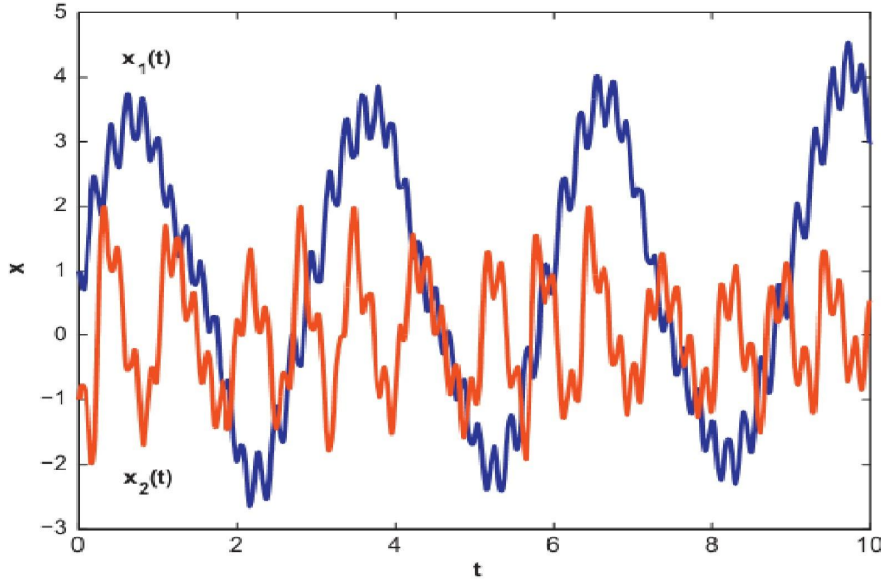
شکل ۴.۳: رفتار واگرای سیستم (۷.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ در مثال (۱.۴.۳).

مثال ۳.۴.۳. [۲۷]

$$\begin{aligned} & \text{minimize} \quad x_1^2 + x_2^2 + 2x_1x_2 + (x_1 - 1)^4 + (x_2 - 3)^4 \\ & \text{subject to} \quad \begin{cases} x_1^2 + x_2^2 - 64 \leq 0, \\ (x_1 + 3)^2 + (x_2 + 4)^2 - 36 \leq 0, \\ (x_1 - 3)^2 + (x_2 + 4)^2 - 36 \leq 0, \\ x_1 \geq 0, \quad x_2 \geq 0. \end{cases} \end{aligned}$$

این مسأله دارای جواب یکتای $x^* = (0, 1/196)^T$ است. قضیه‌ی (۲.۳.۳) و نتیجه‌ی (۱.۳.۳) تضمین می‌کند که مدل (۵.۳) همگرایی سراسری به x^* است. نمودار (۸.۳) رفتار $(x_1(t), x_2(t))$ را با ۱۱ نقطه اولیه مختلف به تصویر می‌کشد، که در آن S نشانگر ناحیه شدنی است. این بردارها دقیقاً به جواب x^* همگرایی سراسری هستند. به‌خصوص سه نقطه $p_1 = (4, -9)$ ، $p_2 = (6, 6)$ و $p_3 = (6, -6)$ در حالی‌که این سه نقطه در مدل کندی و چاو (۹.۳) [۹]، همگرا به جواب بهینه نمی‌باشند.

برای مقایسه، مسأله فوق را با استفاده از مدل شبکه‌ی عصبی کندی و چاو (۹.۳) با ۷ نقطه اولیه



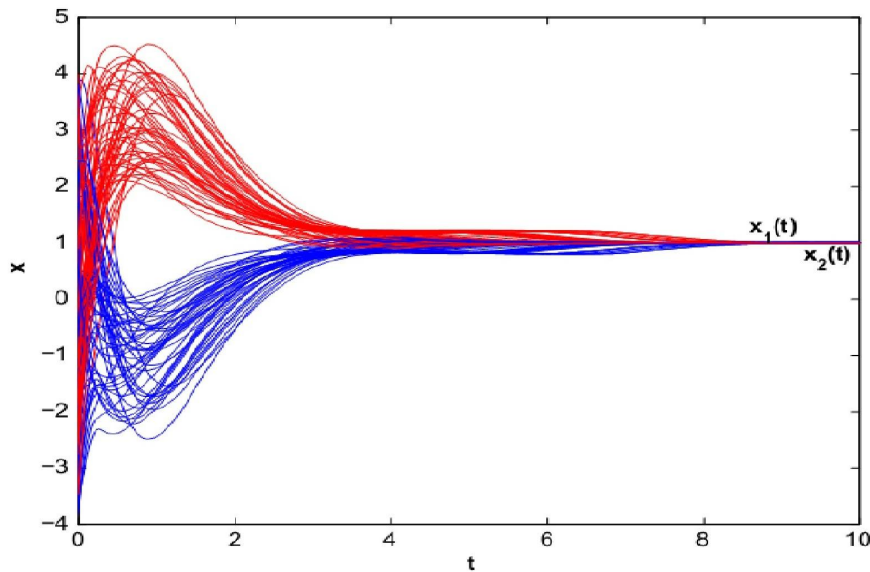
شکل ۵.۳: رفتار واگرای سیستم (۱۰.۳) با نقطه اولیه $x_0 = (1, -1, 1, -1, 1, -1, 1)^T$ در مثال (۱.۴.۳).

مختلف و پارامتر جریمه $\gamma = 5000$ در [۲۷] حل شده است. نتیجه به دست آمده آن است که مدل کندی و چاو به x^* همگرایی سراسری نیست.

مثال ۴.۴.۳. [۳۰]

$$\begin{aligned} & \text{minimize} && (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2 \\ & \text{subject to} && \begin{cases} x_1^2 - 6x_2 - 4x_3 - 3 < 0, \\ x_1 + x_2 + x_3 - 1 = 0, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{cases} \end{aligned}$$

جواب بهینه این مسأله و دوگان آن عبارت است از $x^* = (0, 0, 1)^T$ ، $u^* = (0, 0, 4, 0)^T$ و $v^* = -2$. شکل (۹.۳) مسیر شبکه‌ی عصبی پیشنهادی در (۵.۳) را با 20 نقطه اولیه تصادفی برای حل مثال فوق نشان می‌دهد. اکنون تأثیر پارامتر τ را در شبکه‌ی عصبی (۵.۳) روی مقدار $\|y(t) - y^*\|^2$ بررسی می‌کنیم. شکل (۱۰.۳) این بررسی را با τ های مختلف نشان می‌دهد. با بررسی این نمودار می‌بینیم که با نقطه‌ی اولیه‌ی $y_0 = (1, -1, 1, -1, 1, -1, 1, -1)^T$ ، زمانی که $\tau = 0.1$ باشد میزان



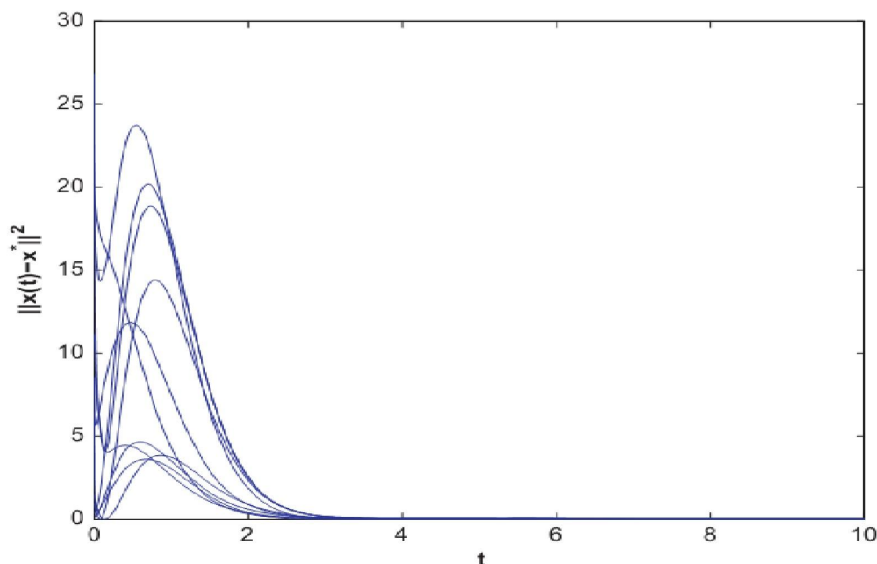
شکل ۶.۳: رفتار سیستم (۵.۳) با $\tau = 50^\circ$ نقطه اولیه تصادفی در مثال (۲.۴.۳).

خطای $\|y(t) - y^*\|^2$ در شبکه‌ی عصبی (۵.۳) با سرعت کمتر و زمانی که $\tau = 10^\circ$ باشد با سرعت بیشتر کاهش می‌یابد. یعنی بزرگی τ همگرایی بهتری را برای خطای $\|y(t) - y^*\|^2$ در پی دارد.

مثال ۵.۴.۳. [۳۲]

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

نتایج شبیه‌سازی شده نشان می‌دهد که خروجی مسیر $x(t)$ از مدل پیشنهادی، به نقطه بهینه x^* همگراست که $x^* = (1/109, 0/922, 1/954)$ نمودار (۱۱.۳) نشان‌دهنده‌ی همگرایی با ۸ نقطه اولیه تصادفی است.



شکل ۷.۳: رفتار همگرایی $\|x(t) - x^*\|^2$ در مثال (۲.۴.۳).

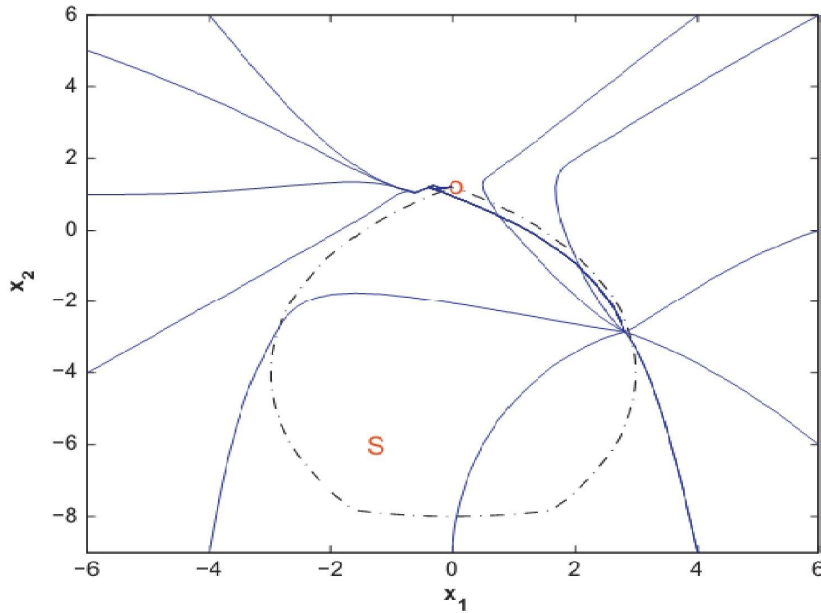
مثال ۶.۴.۳. [۲۵]

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

جواب بهینه برای این مسأله و دوگان آن عبارت است از

$$\begin{aligned} x^* &= (2.330, 1.951, -0.4775, 4.366, -0.625, 1.038, 1.594)^T, \\ u^* &= (1.1398, 0.0000, 0.0000, 0.3686). \end{aligned}$$

نمودار (۱۳.۳) نشان‌دهنده همگرایی موضعی خطای $\|y(t) - y^*\|^2$ با مقادیر مختلف τ با نقطه اولیه $y_0 = (-5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5)^T$ (۱۲.۳) نشان‌دهنده



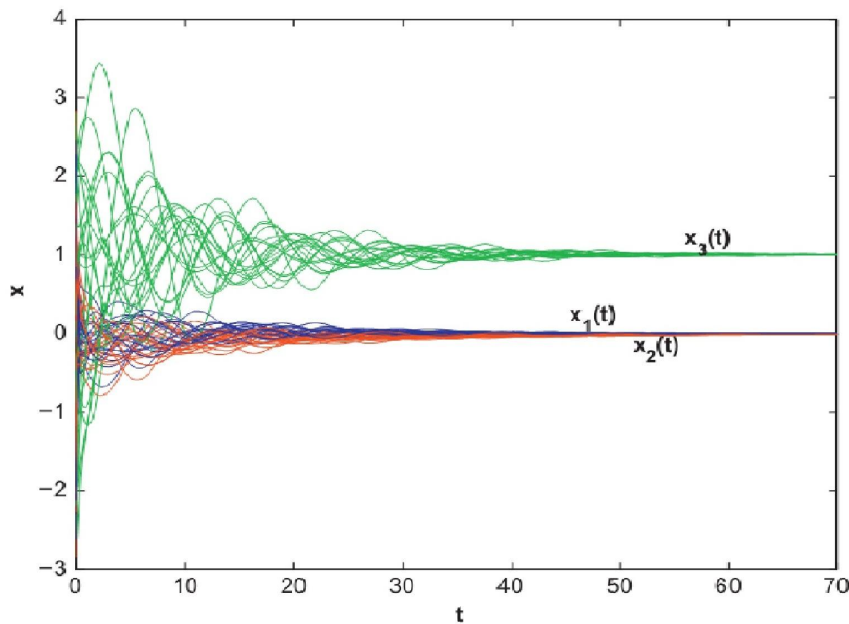
شکل ۸.۳: رفتار صفحه فاز (۵.۳) با ۱۱ نقطه اولیه مختلف در مثال (۳.۴.۳).

همگرایی $x(t)$ با ۵ نقطه اولیه تصادفی است. واضح است که افزایش τ نرخ همگرایی بهتری را موجب می‌گردد.

مثال ۷.۴.۳. [۱۸]

$$\begin{aligned} & \text{minimize} && \frac{3}{4}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) + 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_4 \\ & \text{subject to} && \begin{cases} -x_1 + x_2 + x_3 + x_4 \leq 2, \\ x_1 + x_2 - x_4 \leq 12, \\ 2x_1 - x_2 + x_3 \leq 18, \\ 2x_1 + x_2 - x_3 + x_4 \geq 2, \\ 0 \leq x_1 \leq 10, \\ -2 \leq x_2 \leq 10, \\ 0.1 \leq x_3 \leq 10, \\ 0.1 \leq x_4 \leq 10. \end{cases} \end{aligned}$$

این مسأله دارای جواب بهینه یکتای $x^* = (2.786, -2, 0, 1, 0.9213)^T$ است. نمودار (۱۴.۳) نمایش همگرایی مسیرهای خروجی $x(t)$ با ۱۰۰ نقطه اولیه تصادفی است.



شکل ۹.۳: مسیر مدل پیشنهادی در (۵.۳) با x_0 نقطه اولیه تصادفی در مثال (۴.۴.۳).

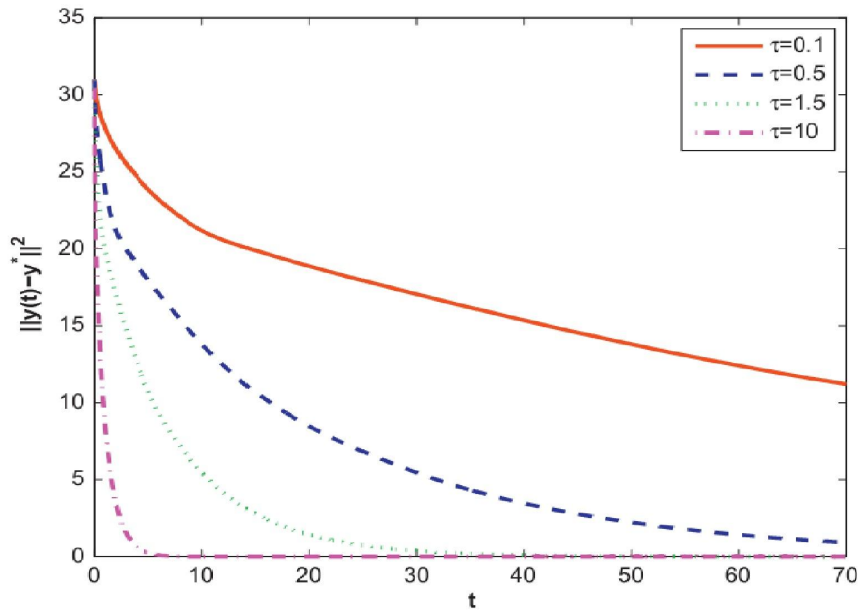
مثال ۸.۴.۳. [۲۹]

$$\begin{aligned} & \text{minimize} \quad 2x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{4}x_3^2 + x_1x_2 + 2x_1x_3 - 2x_1 - x_2 - x_3 \\ & \text{subject to} \quad \begin{cases} 5 \leq x_1 + x_2 + x_3 \leq 7, \\ 5 \leq 2x_2 + x_3 \leq 7, \\ 1 \leq x_1 \leq 3, \\ 2 \leq x_2 \leq 5, \\ 0 \leq x_3 \leq 2. \end{cases} \end{aligned}$$

این مسأله، برنامه‌ی محدب درجه‌ی دو تباهیده است. مجموعه‌ی شامل جواب‌های این مسأله عبارت است از

$$E_N = \{(x_1, x_2, x_3)^T \mid x_1 = 1, x_2 + x_3 = 4, 2 \leq x_2 \leq 3/5, x_3 \geq 0\}.$$

همه‌ی نتایج مشابه‌سازی نشان می‌دهد که $x(t)$ در مدل شبکه‌ی عصبی پیشنهادی همگرا به جواب بهینه مسأله برنامه‌ریزی محدب درجه دو تباهیده است. نمودار (۱۵.۳) و (۱۶.۳) نشان‌دهنده همگرایی با دو



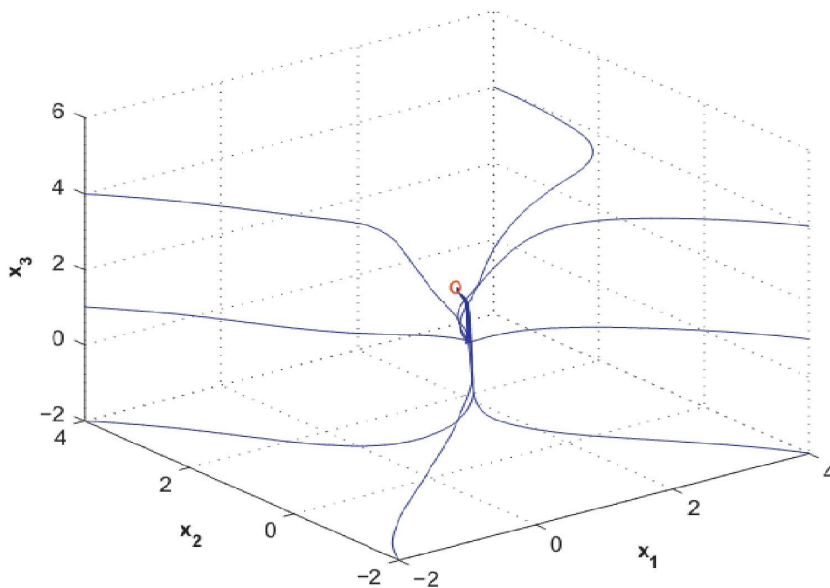
شکل ۴.۳: رفتار همگرایی $\|y(t) - y^*\|^2$ با نقطه اولیه $y_0 = (1, -1, 1, -1, 1, -1, 1, -1)^T$ در مثال (۴.۴.۳).

نقطه اولیه تصادفی است که عبارت‌اند از

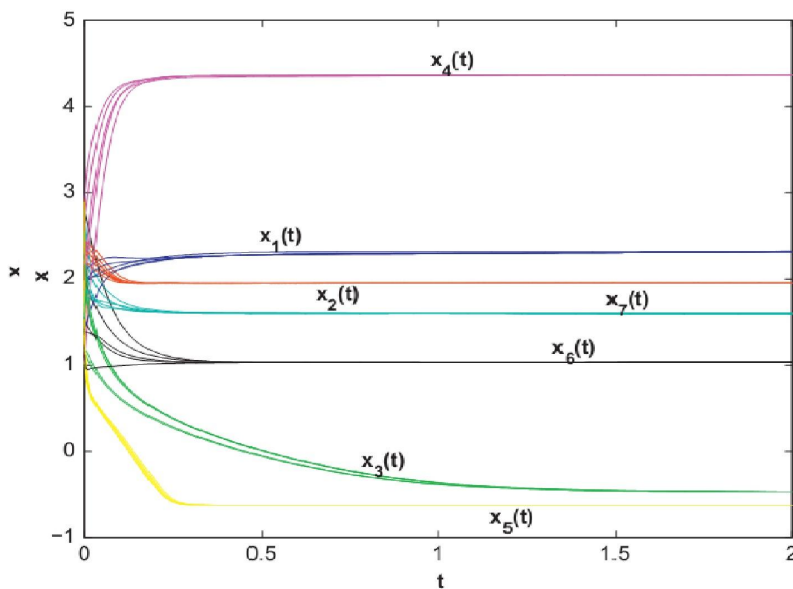
$$y_{1,0} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T,$$

$$y_{2,0} = (-1, 2, -3, 4, -5, 6, -7, 8, -9, 10, -11, 12, -13)^T.$$

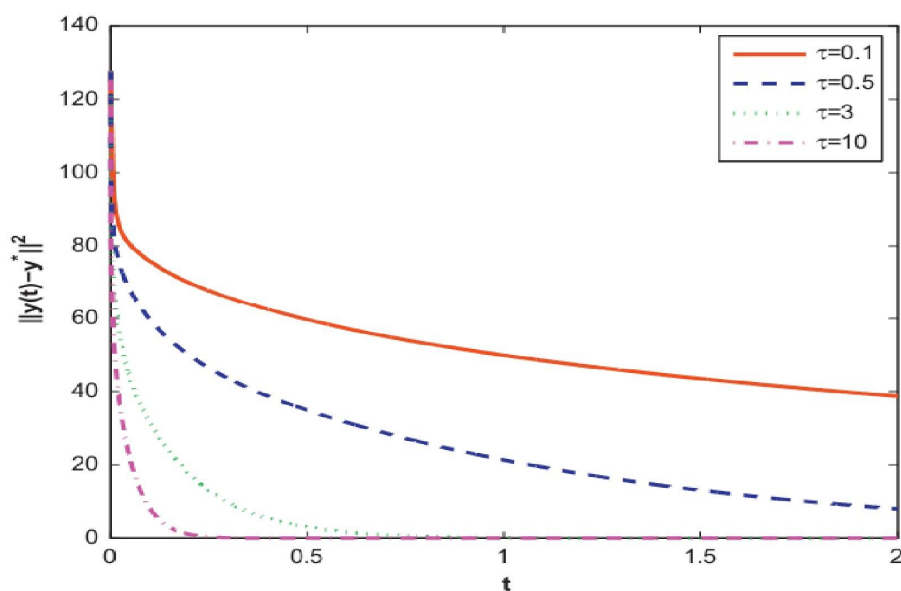
با توجه به همگرایی $x_1(t), x_2(t), x_3(t)$ و $x_1(t) + x_2(t)$ نتیجه می‌گیریم که خروجی‌های مدل شبکه‌ی عصبی (۵.۳) به یکی از عناصر E_N همگراست.



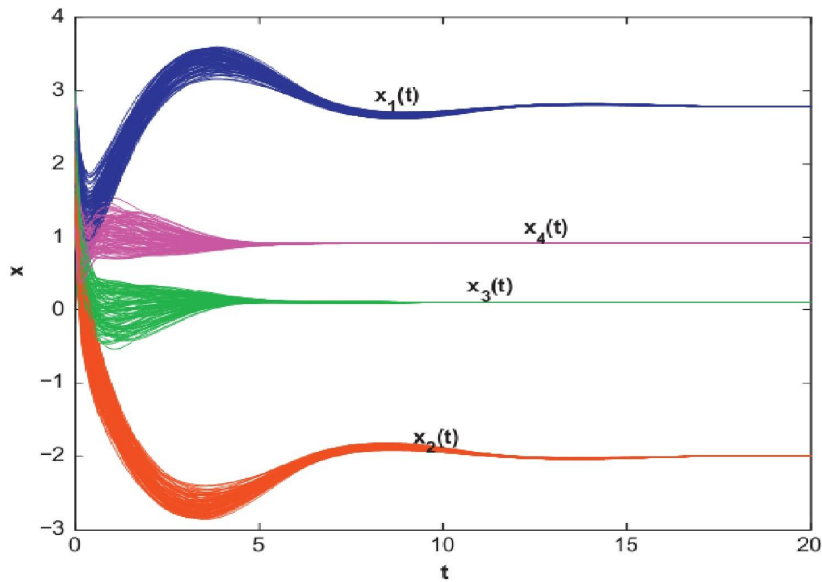
شکل ۱۱.۳: نمایش شبکه‌ی عصبی (۵.۳) با ۸ نقطه اولیه تصادفی در مثال (۴.۴.۳).



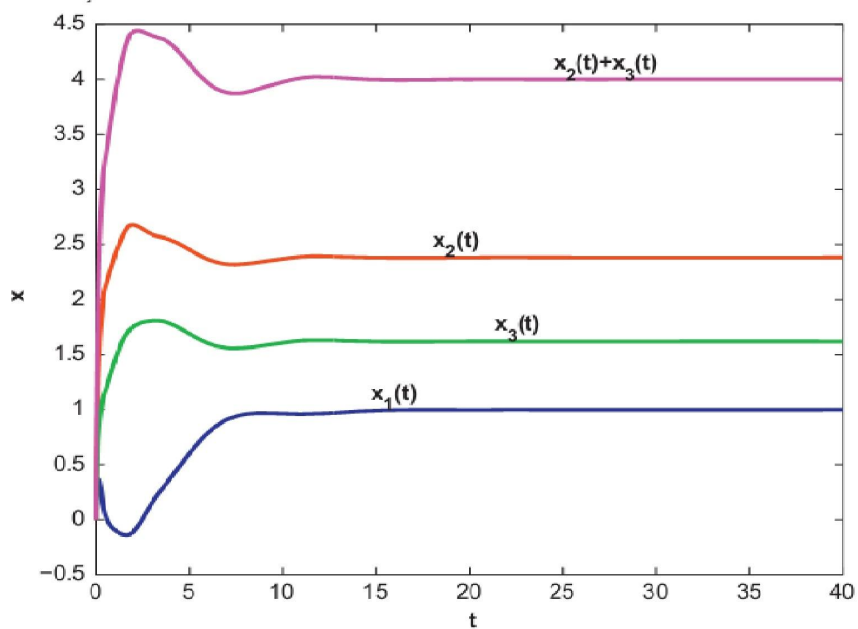
شکل ۱۲.۳: مسیر مدل پیشنهادی در (۵.۳) با ۵ نقطه اولیه تصادفی در مثال (۶.۴.۳).



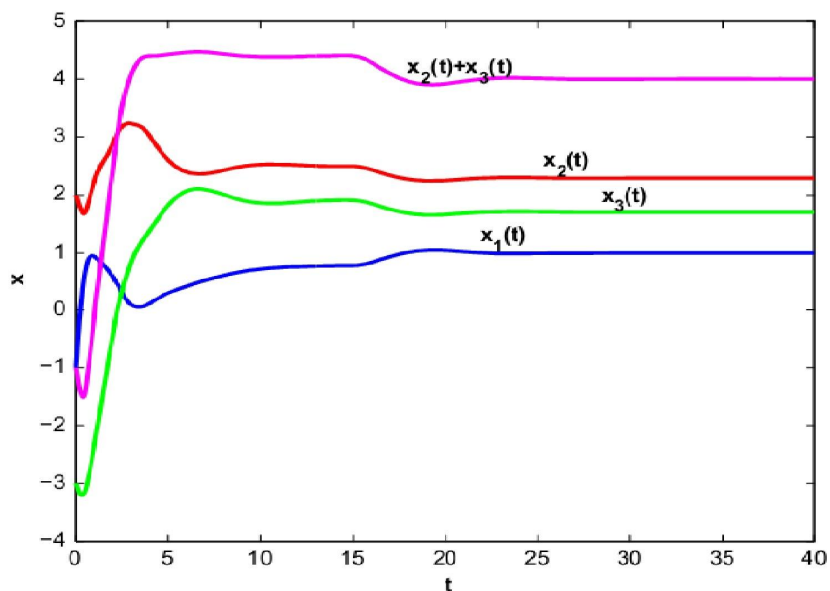
شکل ۱۳.۳: رفتار همگرایی $\|y(t) - y^*\|^2$ در مثال (۶.۴.۳) با نقطه اولیه $(-5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5)^T$.



شکل ۱۴.۳: مسیر شبکه‌ی عصبی (۵.۳) با 10° نقطه اولیه تصادفی در مثال (۷.۴.۳).



شکل ۱۵.۳: همگرایی $x_3 : x_2 : x_1$ و $x_1(t) + x_2(t)$ با نقطه اولیه $y_{1,0}$ در مثال (۸.۴.۳).



شکل ۱۶.۳: همگرایی $x_3 : x_2 : x_1$ و $x_1(t) + x_2(t)$ با نقطه اولیه y_2 در مثال (۸.۴.۳).

فصل ۴

یک روش هوش مصنوعی برای حل رده‌ای از
مسائل بهینه‌سازی غیر هموار

در این فصل، یک تکنیک بهینه‌سازی را توضیح می‌دهیم که برای حل مسائل کلاس بهینه‌سازی غیر هموار کاربرد دارد. ایده اصلی، جایگزینی تابع ماکزیمم غیرهموار با یک تابع هموار است. با این تکنیک، مسأله بهینه‌سازی غیرهموار، به یک مسأله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر تبدیل می‌گردند. مسأله‌ی بهینه‌سازی غیرهموار زیر را در نظر بگیرید:

$$\begin{aligned} & \text{minimize } F(x) = \text{maximize}_{1 \leq i \leq s} \{F_i(x)\} \\ & \text{subject to} \\ & G_k(x) = 0, \quad k = 1, \dots, m, \end{aligned} \tag{۱.۴}$$

که در آن $F_i(x)$ ، $x \in \mathbb{R}^n$ ($i = 1, \dots, s$) توابعی پیوسته، محدب و دوبار مشتق‌پذیرند. و توابع $G_k(x)$ ($k = 1, \dots, m$)، توابعی آفینی می‌باشند. اگرچه توابع $F_i(x)$ ($i = 1, \dots, s$) پیوسته و مشتق‌پذیرند، لیکن تابع هدف، یعنی تابع $F(x) = \text{maximize}_{1 \leq i \leq s} \{F_i(x)\}$ یک تابع غیرهموار است. بنابراین مسائل (۱.۴) در رده مسائل بهینه‌سازی غیر هموار قرار دارند. فرض بر این است که مسأله (۱.۴) دارای جواب بهینه منحصر بفرده است.

۱.۴ یک مدل شبکه عصبی

به آسانی می‌توان دید که مسأله (۱.۴) با مسأله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر زیر معادل است:

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to} \\ & F_i(x) \leq z, \quad i = 1, \dots, s, \\ & G_k(x) = 0, \quad k = 1, \dots, m, \end{aligned} \tag{۲.۴}$$

که در آن $z = \text{maximize}\{F_i(x), i = 1, 2, \dots, s\}$. واضح است که مسأله (۲.۴) مشتق‌پذیر است. لیکن تعداد محدودیت‌ها و پیچیدگی محاسباتی آن زیاد است. برای رفع این مشکل، می‌توانیم از تابع آنتروپی $F_p(x)$ برای حل مسأله بهینه‌سازی غیرهموار (۱.۴) استفاده کنیم [۲۸].
قرار دهید

$$F_p(x) = \frac{1}{p} \ln \sum_{i=1}^s \exp(pF_i(x)), \tag{۳.۴}$$

که p یک عدد مثبت است. با این تعریف مسأله (۱.۴) به مسأله معادل زیر تبدیل می‌شود [۳۱].

$$\begin{aligned} \text{minimize } F_p(x) &= \text{minimize } \frac{1}{p} \ln \sum_{i=1}^s \exp(pF_i(x)) \\ \text{subject to} \end{aligned} \quad (4.4)$$

$$G_k(x) = 0, \quad k = 1, \dots, m.$$

می‌توان نشان داد که جواب بهینه تقریبی مسأله بهینه‌سازی غیرهموار (۱.۴) با حل مسأله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر (۴.۴) معادل است که در آن p به اندازه کافی بزرگ است [۳۱]. از تحلیل فوق، می‌توان به این نتیجه رسید که مسأله (۱.۴) را می‌توان به مسأله برنامه‌ریزی محدب هموار تبدیل کرد.

۲.۴ مثال عددی

مثال ۱.۲.۴. [۳۳]

$$\begin{aligned} \text{minimize } f(x) &= \text{maximize } \{4(x_1^2 + 2x_2^2), 1 - x_1 - x_2\} \\ \text{subject to} \end{aligned} \quad (5.4)$$

$$x_1 + 2x_2 = \frac{3}{4}.$$

جواب بهینه این مسأله عبارت است از $x^* = (0.2500, 0.2500)^T$. مسأله (۵.۴)، با معرفی تابع آنتروپی

$$F(x) = \frac{1}{p} \ln \sum_{i=1}^2 \exp(pf_i(x)),$$

به یک مسأله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر به صورت زیر تبدیل می‌شود:

$$\begin{aligned} \text{minimize } F(x) &= \text{minimize } \frac{1}{p} \ln \sum_{i=1}^2 \exp(pf_i(x)) \\ \text{subject to} \end{aligned} \quad (6.4)$$

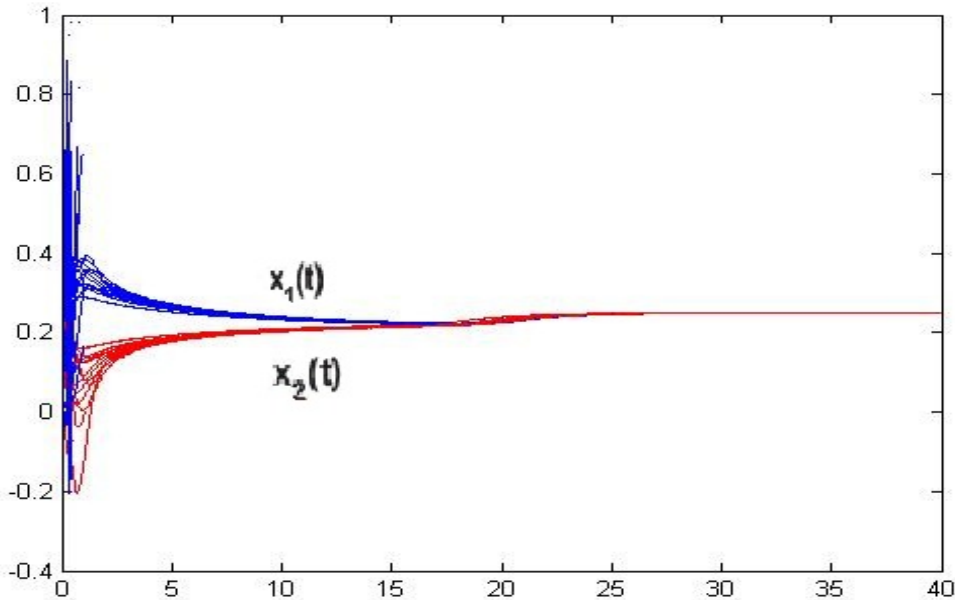
$$x_1 + 2x_2 = \frac{3}{4},$$

که در آن

$$f_1(x) = 4(x_1^2 + 2x_2^2), \quad (7.4)$$

$$f_2(x) = 1 - x_1 - x_2. \quad (8.4)$$

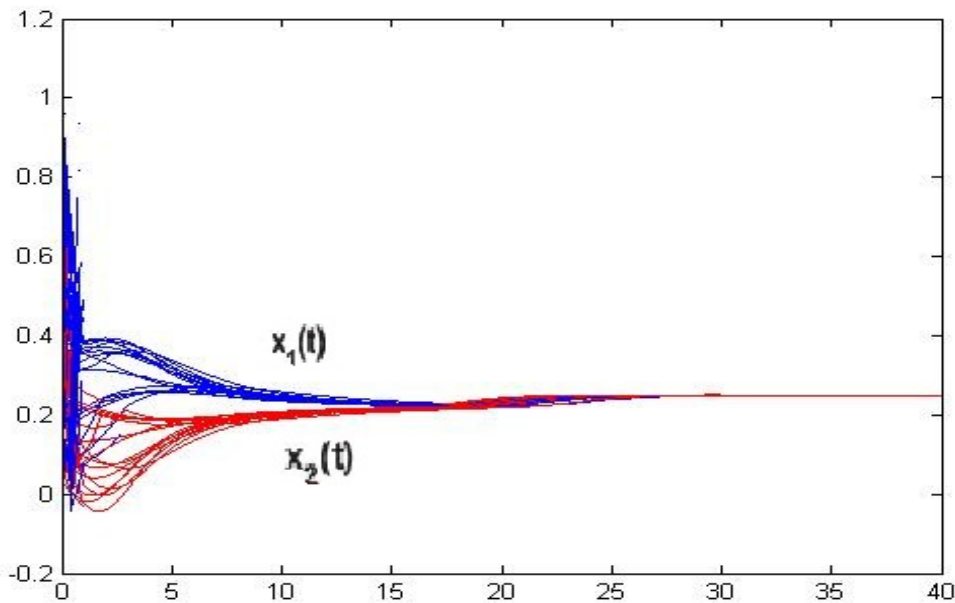
شکل‌های (۱.۴)، (۲.۴) و (۳.۴) نشان می‌دهند که با به‌کاربردن مدل (۵.۳) با $p = 10^2$ و τ های مختلف برای حل مسأله (۶.۴) با 50° نقطه شروع متفاوت، مسیر حل همگرا به نقطه تعادل x^* است.



شکل ۱.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با 50° نقطه اولیه تصادفی با $\tau = 1$ در مثال (۱.۲.۴).

برای مسأله (۶.۴) با $p = 10^2$ ، تأثیرات متغیر τ را در شبکه‌ی عصبی (۵.۳) با مقدار خطای $\|x(t) - x^*\|^2$ بررسی می‌کنیم. با توجه به نمودارهای (۴.۴) می‌بینیم هنگامی که $\tau = 1$ است آنگاه خطای $\|x(t) - x^*\|^2$ در شبکه‌ی عصبی (۵.۳) دیرتر به سمت صفر میل می‌کند، درحالی‌که اگر $\tau = 5$ باشد، خطای $\|x(t) - x^*\|^2$ با نقطه اولیه تصادفی زودتر به سمت صفر میل می‌کند. می‌بینیم بزرگی مقدار τ همگرایی سریع‌تری از خطای $\|x(t) - x^*\|^2$ را موجب می‌شود. در اینجا این مثال را بدون استفاده از تابع آنتروپی $F_p(x)$ و تنها با استفاده از (۲.۴) حل می‌کنیم. قرار می‌دهیم:

$$z = \text{maximize}\{4(x_1^2 + 2x_2^2), 1 - x_1 - x_2\}.$$



شکل ۲.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با $\tau = 0.1$ و نقطه اولیه تصادفی با $\tau = 0.1$ در مثال (۱.۲.۴).

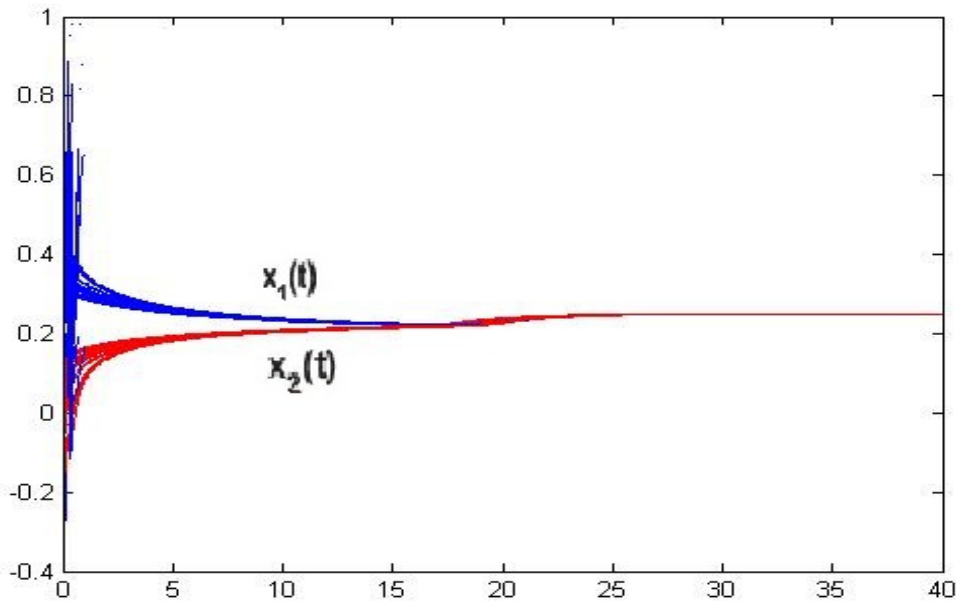
مسأله (۵.۴) به مسأله زیر تبدیل می‌شود:

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \begin{cases} 4(x_1^2 + 2x_2^2) - z \leq 0, \\ 1 - x_1 - x_2 - z \leq 0, \\ x_1 + 2x_2 = \frac{3}{4}. \end{cases} \end{aligned} \quad (9.4)$$

نمودار (۵.۴) نشان‌دهنده همگرایی مدل (۵.۳) برای حل (۵.۴) به نقطه بهینه $(0.2500, 0.2500)$ است.

مثال ۲.۲.۴. [۳۳]

$$\begin{aligned} & \text{minimize } f(x) = \text{maximize}\{f_1(x), \dots, f_n(x)\} \\ & \text{subject to } \begin{cases} x_1 + x_2 = \frac{1}{\lambda}, \\ x_2 + x_3 = \frac{1}{\lambda}, \end{cases} \end{aligned} \quad (10.4)$$



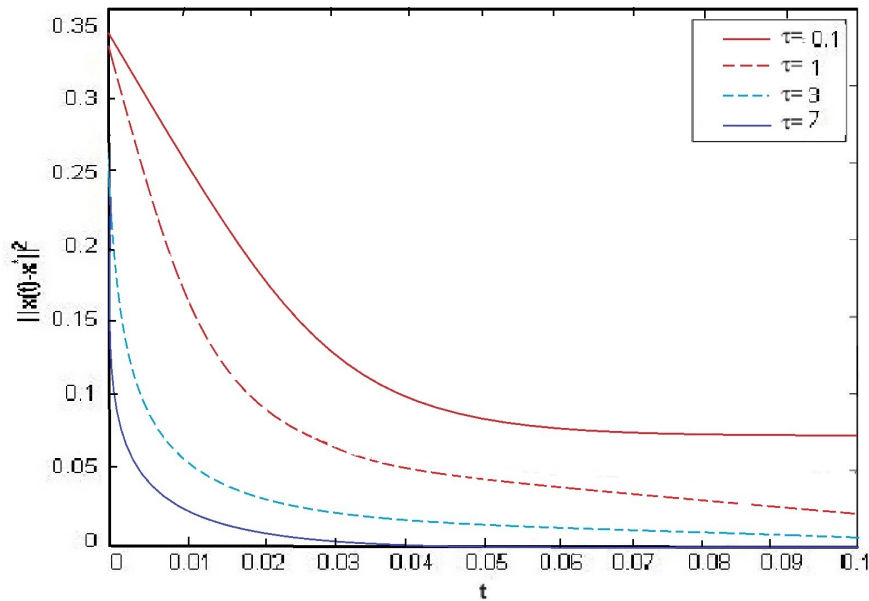
شکل ۳.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۶.۴) با $\tau = 5$ نقطه اولیه تصادفی با $\tau = 5$ در مثال (۱.۲.۴).

که در آن

$$\begin{cases} f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1, \\ f_2(x) = x_1 + x_2 - x_3 + 1, \\ f_3(x) = x_1^2 + x_2^2 + (x_3 - 2)^2, \\ f_4(x) = 2x_1^2 + 6x_2^2 + 2(5x_3^2 - x_1 + 1)^2, \\ f_5(x) = x_1 + x_2 - x_3 - 1, \\ f_6(x) = x_1^2 - 9x_3. \end{cases} \quad (11.4)$$

جواب بهینه این مسأله عبارت است از $x^* = (0.3354, -0.0021, 0.5271)^T$ قرار می‌دهیم:

$$z = \text{maximize}\{f_1(x), \dots, f_6(x)\}. \quad (12.4)$$



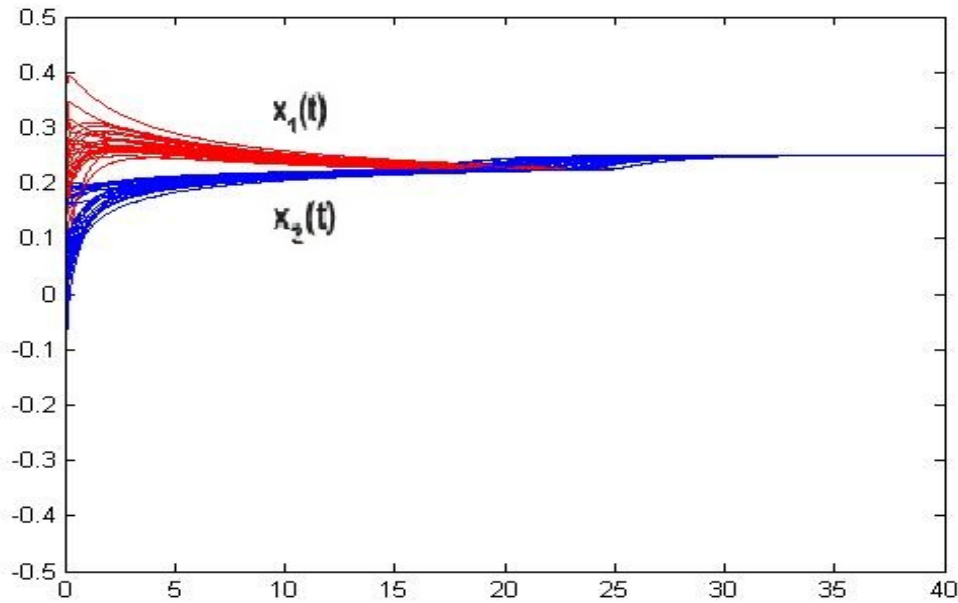
شکل ۴.۴: بررسی تأثیرات متغیر τ بر شبکه‌ی عصبی (۵.۳) با مقدار خطای $\|x(t) - x^*\|_2$ در مثال (۱.۲.۴).

در این صورت مسأله (۱۰.۴) به مسأله زیر تبدیل می‌گردد:

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \begin{cases} x_1^2 + x_2^2 + x_3^2 - 1 - z \leq 0, \\ x_1 + x_2 - x_3 + 1 - z \leq 0, \\ x_1^2 + x_2^2 + (x_3 - 2)^2 - z \leq 0, \\ 2x_1^2 + 6x_2^2 + 2(5x_3^2 - x_1 + 1)^2 - z \leq 0, \\ x_1 + x_2 - x_3 - 1 - z \leq 0, \\ x_1^2 - 9x_3 - z \leq 0, \\ x_1 + x_2 = \frac{1}{3}, \\ x_2 + x_3 = \frac{1}{8}. \end{cases} \quad (13.4) \end{aligned}$$

نمودار (۶.۴) نشان‌دهنده همگرایی مدل (۵.۳) برای حل مسأله (۱۳.۴) است که به نقطه بهینه همگرایی استفاده شده است. در این مسأله از مقدار $\tau = 10$ برای نشان‌دادن بهتر همگرایی استفاده شده است. با معرفی تابع آنتروپی

$$F(x) = \frac{1}{p} \ln \sum_{i=1}^2 \exp(pf_i(x)).$$



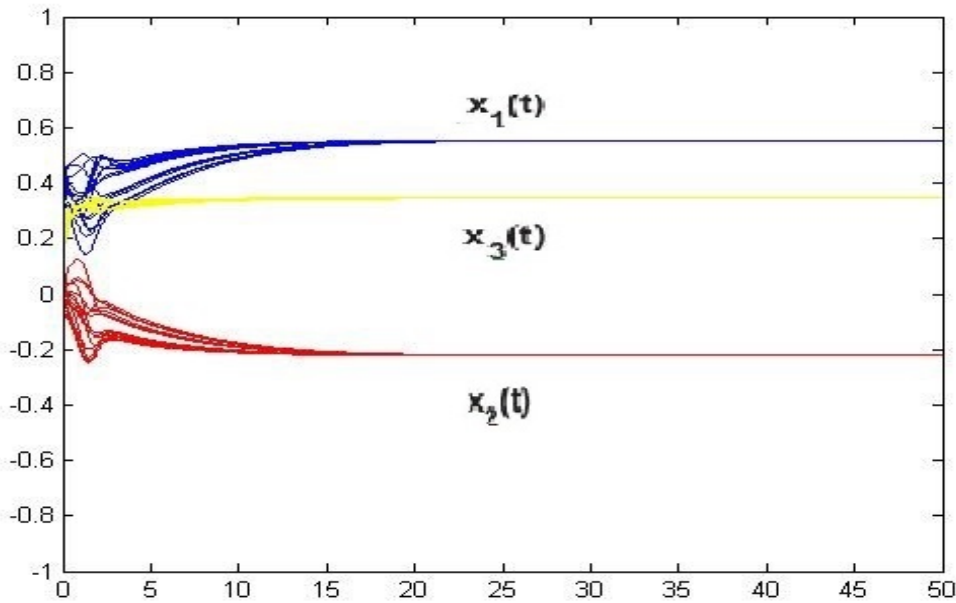
شکل ۵.۴: رفتار شبکه‌های عصبی (۵.۳) با 5° نقطه اولیه تصادفی برای حل (۵.۴).

مسئله (۱۰.۴) به یک مسئله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر به صورت زیر تبدیل می‌شود:

$$\begin{aligned} \text{minimize } F(x) &= \text{minimize } \frac{1}{p} \ln \sum_{i=1}^6 \exp(p f_i(x)) \\ \text{subject to} & \\ x_1 + x_2 &= \frac{1}{3}, \\ x_2 + x_3 &= \frac{1}{8}, \end{aligned} \tag{۱۴.۴}$$

که در آن

$$\begin{cases} f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1, \\ f_2(x) = x_1 + x_2 - x_3 + 1, \\ f_3(x) = x_1^2 + x_2^2 + (x_3 - 2)^2, \\ f_4(x) = 2x_1^2 + 6x_2^2 + 2(5x_3^2 - x_1 + 1)^2, \\ f_5(x) = x_1 + x_2 - x_3 - 1, \\ f_6(x) = x_1^2 - 9x_3. \end{cases} \tag{۱۵.۴}$$



شکل ۶.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۳.۴) با ۱۵ نقطه اولیه تصادفی در مثال (۲.۲.۴).

شکل (۷.۴) نشان می‌دهد که با به‌کاربردن مدل (۵.۳) با $p = 10^3$ برای حل مسأله‌ی (۱۴.۴) با ۱۵ نقطه شروع متفاوت، مسیر حل همگرا به نقطه‌ی تعادل x^* است.

مثال ۳.۲.۴. [۳۳]

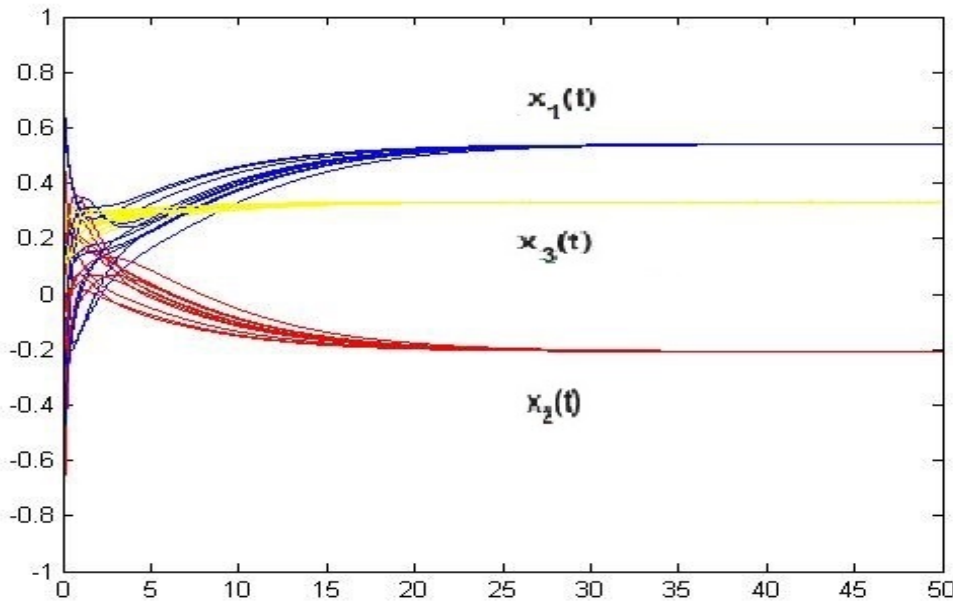
$$\begin{aligned} & \text{minimize } f(x) = \text{maximize}\{f_1(x), \dots, f_4(x)\} \\ & \text{subject to } \begin{cases} 2x_1 + 2x_2 - x_3 - 3x_4 = 3, \\ x_1 - 2x_2 + 3x_3 + 2x_4 = 2, \end{cases} \end{aligned} \quad (16.4)$$

که در آن

$$\begin{cases} f_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 + 5x_1 - 5x_2 - 21x_3 + 7x_4, \\ f_2(x) = 11x_1^2 + 11x_2^2 + 12x_3^2 + 11x_4^2 + 5x_1 - 15x_2 - 11x_3 - 3x_4 - 80, \\ f_3(x) = 11x_1^2 + 21x_2^2 + 12x_3^2 + 21x_4^2 + 5x_1 - 5x_2 - 21x_3 - 3x_4 - 100, \\ f_4(x) = 11x_1^2 + 11x_2^2 + 12x_3^2 + x_4^2 + 15x_1 - 15x_2 - 21x_3 - 3x_4 - 50. \end{cases} \quad (17.4)$$

جواب بهینه این مسأله عبارت است از $(0.0551, 0.8511, 1.9026, -1.0301)^T$. قرار می‌دهیم:

$$z = \text{maximize}\{f_1(x), \dots, f_4(x)\}. \quad (18.4)$$



شکل ۷.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۴.۴) با ۱۵ نقطه اولیه تصادفی در مثال (۲.۲.۴).

در این صورت مسأله (۱۶.۴) به مسأله زیر تبدیل می‌گردد:

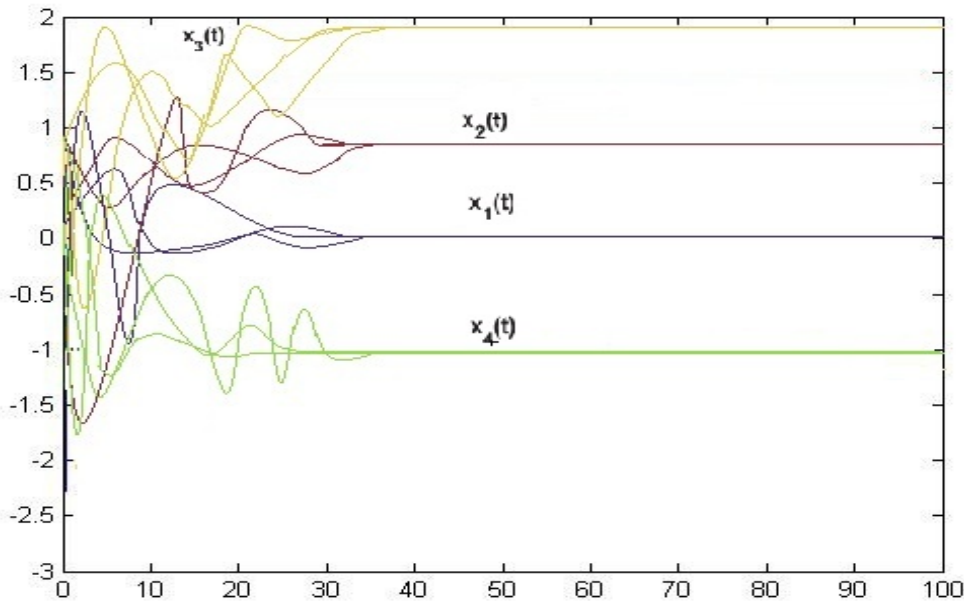
$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \begin{cases} x_1^2 + x_2^2 + 2x_3^2 + x_4^2 + 5x_1 - 5x_2 - 21x_3 + 7x_4 - z \leq 0, \\ 11x_1^2 + 11x_2^2 + 12x_3^2 + 11x_4^2 + 5x_1 - 15x_2 - 11x_3 - 3x_4 - 80 - z \leq 0, \\ 11x_1^2 + 21x_2^2 + 12x_3^2 + 21x_4^2 + 5x_1 - 5x_2 - 21x_3 - 3x_4 - 100 - z \leq 0, \\ 11x_1^2 + 11x_2^2 + 12x_3^2 + x_4^2 + 15x_1 - 15x_2 - 21x_3 - 3x_4 - 50 - z \leq 0, \\ x_1 + x_2 = \frac{1}{3}, \\ x_2 + x_3 = \frac{1}{8}. \end{cases} \quad (19.4) \end{aligned}$$

نمودار (۸.۴) نشان‌دهنده همگرایی مدل (۵.۳) برای حل مسأله (۱۹.۴) است که به نقطه بهینه $(0.3354, -0.0021, 0.5271)^T$ همگرا است. در این مسأله از مقدار $\tau = 10$ برای نشان‌دادن بهتر همگرایی استفاده شده است.

با معرفی تابع آنتروپی

$$F(x) = \frac{1}{p} \ln \sum_{i=1}^p \exp(pf_i(x)).$$

مسأله (۱۶.۴) به یک مسأله برنامه‌ریزی غیرخطی محدب مشتق‌پذیر به صورت زیر تبدیل می‌شود:



شکل ۸.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۱۹.۴) با ۳ نقطه اولیه تصادفی در مثال (۳.۲.۴).

$$\begin{aligned} \text{minimize } F(x) = & \text{minimize } \frac{1}{p} \ln \sum_{i=1}^4 \exp(p f_i(x)) \\ \text{subject to} & \end{aligned} \quad (20.4)$$

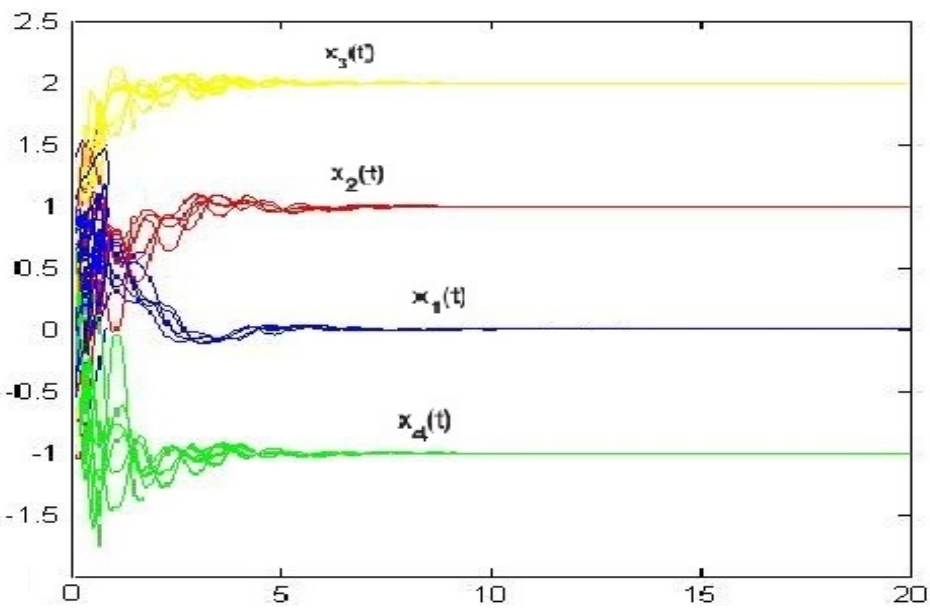
$$2x_1 + 2x_2 - x_3 - 3x_4 = 3,$$

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

که در آن

$$\begin{cases} f_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 + 5x_1 - 5x_2 - 21x_3 + 7x_4, \\ f_2(x) = 11x_1^2 + 11x_2^2 + 12x_3^2 + 11x_4^2 + 5x_1 - 15x_2 - 11x_3 - 3x_4 - 80, \\ f_3(x) = 11x_1^2 + 21x_2^2 + 12x_3^2 + 21x_4^2 + 5x_1 - 5x_2 - 21x_3 - 3x_4 - 100, \\ f_4(x) = 11x_1^2 + 11x_2^2 + 12x_3^2 + x_4^2 + 15x_1 - 15x_2 - 21x_3 - 3x_4 - 50. \end{cases} \quad (21.4)$$

شکل (۹.۴) نشان می‌دهد که با به‌کاربردن مدل (۵.۳) با $p = 10$ برای حل مسأله (۲۰.۴) با ۶ نقطه شروع متفاوت، مسیر حل همگرا به نقطه تعادل x^* است.



شکل ۹.۴: رفتار شبکه‌های عصبی (۵.۳) برای حل مسأله (۲۰.۴) با ۶ نقطه اولیه تصادفی در مثال (۳.۲.۴).

نتیجه‌گیری

در این پایان نامه یک شبکه‌ی عصبی کارا برای حل مسائل بهینه‌سازی محدب و همچنین حل مسائل بهینه‌سازی غیرهموار ارئه شده است.

- در این پایان نامه بین مدل شبکه عصبی پیشنهادی و سه مدل دیگر مقایسه‌ای انجام دادیم. در نگاه اول به نظر می‌رسد که این مدل‌ها دارای پیچیدگی زمانی کمتری هستند. اگرچه تفاوت عملکرد محاسباتی با بررسی چند مسأله برنامه‌ریزی محدب در آنها مشهود است.
- در مدل پیشنهادی، نقطه اولیه چه در داخل ناحیه شدنی باشد و چه خارج ناحیه شدنی، در یافتن جواب بهینه تأثیری ندارد. دلیل آن هم همگرایی سراسری مدل پیشنهادی است.
- مدل‌های (۱۰.۳)، (۹.۳) و (۷.۳) به‌طور کلی نمی‌توانند مسائل برنامه‌ریزی محدب که در آن تابع هدف و همه‌ی محدودیت‌ها، فقط محدب هستند، مانند مسائل برنامه‌ریزی خطی، مسائل برنامه‌ریزی محدب درجه ۲ تباهیده و ... را حل نماید ولی مدل پیشنهادی قادر به حل آن است.
- در این پایان‌نامه تأثیرات پارامتر τ در مدل پیشنهادی را بر خطای $\|x(t) - x^*\|^2$ و همگرایی بررسی کرده‌ایم و به این نتیجه رسیدیم که زمانی که τ بزرگ است خطا زودتر به سمت صفر میل می‌کند.

به عنوان پیشنهادات برای کارهای آتی، می‌توان به موارد زیر اشاره کرد:

- از مدل ارائه شده می‌توان برای حل مسائل چندسطحی و چندهدفه استفاده کرد. همچنین می‌توان آن را برای حل نامساوی‌های وردشی به‌کار برد.
- اگر قیود تساوی در مسئله (۱۰.۳) مشتق‌ناپذیر باشند یا زمانی که قیود تساوی، مخروط مرتبه دوم باشد نیز می‌توان از این روش استفاده کرد.

مراجع

- [۱] آشنایی با شبکه‌های عصبی، جکسون. تی، بیل. آر، ۱۳۸۳، موسسه انتشارات علمی دانشگاه صنعتی شریف، چاپ دوم.
- [2] Alishahi. M, 2011, On dynamic coloring of graphs, Applied Mathematics and Computation 159 (2-3): 152–156.
- [3] Avriel. M, 1976, Nonlinear Programming: Analysis and Methods, Prentice-Hall, Englewood Cliffs, New Jersey.
- [4] Barnoy. A, Bellare. M, Halldórsson. M, Shachnai. H, 1988, On chromatic sums and distributed resource allocation, Information and Computation 140, 183–202.
- [5] Bazaraa. M. S, Sherali. H. D, Jarvis. J. J, 1990, Linear Programming and Network Flows, 2nd ed. Wiley, New York.
- [6] Bazaraa. M. S, Sherali. H. D, Shetty. C. M, 1993, Nonlinear Programming—Theory and Algorithms, 2nd ed, Wiley, New York.
- [7] Effati. S, Nazemi. A. R, 2006, Neural network models and its application for solving linear and quadratic programming problems, Applied Mathematics and Computation 172, 305–331.
- [8] Friesz. T. L, Bernstein. D. H, Mehta. N. J, Tobin. R. L, GaNew Jerseylizadeh. S, 1994, Day-today dynamic network disequilibria and idealized traveler information systems, Operations Research 42, 1120–1136.
- [9] Gao. X, 2004, A novel neural network for nonlinear convex programming, IEEE Transactions, Neural Networks 15, 613–621.
- [10] Giaro. K, Kubale. M, 1999, On the edge chromatic sum of a graph, János Bolyai Math. Soc, Budapest , 05C75, MR1901884, 58.
- [11] Giaro. K, Kubale. M, 2000, Edge-chromatic sum of trees and bounded cyclicity graphs, Information Processing Letters, 75, 1-2, 65–69.
- [12] Hopfield. J. J, Tank. D.W, 1985, Neural computation of decisions in optimization problems. Biol, 52:141–152.

- [13] Hull. J. C, 2006, Options Futures And Other Derivatives, 6nd ed, Perntice Hall, New Jersey.
- [14] Rueda. A, Huertas. J. L, Rodriguez-Vazquez. A, Dommnguer-Castro. R and Sahnchez-Sinencio. E, 1990, Nonlinear switched-capacitor neural network for optimization problems. IEEE Transactions on Circuits and Systems, 42:384–390.
- [15] Kennedy. M. P, Chua. L. O, 1988, Neural networks for nonlinear programming, IEEE Transactions, circuits systems. 35, 554–562.
- [16] Lillo. W. E, Loh. M. H, Hui. S, Za'k. S. H, 1993, On solving constrained optimization problems with neural networks: a penalty method approach, IEEE Transactions, Neural Networks 4, 931–939.
- [17] Małafiejski. M, Giaro. K, Janczewski. R, Kubale. M, 2004, Sum coloring of bipartite graphs with bounded degree, Algorithmica 40, 235–244.
- [18] Malek. A, Hosseinipour-Mahani. N, Ezazipour. S, 2010, Efficient recurrent neural network model for the solution of general nonlinear optimization problems, optimization Methods and Software 25, 1–18.
- [19] Meng. X, Miao. L, Gong. Z, Su. B, 2006, The conditional coloring numbers of pseudo-Harlin graphs, Ars Combinatoria, 54.
- [20] Miller. R. K, Michel. A. N, 1982, Ordinary Differential Equations, Academic Press, New York.
- [21] Nazemi. A. R, 2012, A dynamic system model for solving convex nonlinear optimization problems, Commun, Nonlinear Sci, Numer, Simulation 17, 1696–1705.
- [22] Noga. A, Spencer. J, 2009, The probabilistic method, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley and Sons Inc, Hoboken, New Jersey, 4–7
- [23] Quarteroni. A, Sacco. R, Saleri. F, 2007, Numerical mathematics, Springer-Verlag, Berlin.
- [24] Simmons. G. F, 1991, Differential equations with applications and historical notes, McGraw-Hill, New York.
- [25] Xia. Y, Feng. G, 2007, A new neural network for solving nonlinear projection equations, Neural Networks 20, 577–589.
- [26] Xia. Y, Wang. J, 2000, A recurrent neural network for solving linear projection equations, Neural Networks 13, 337–350.
- [27] Xia. Y, Wang. J, 2004, A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints, IEEE Transactions, Circuits System, 51, 447–458.

-
- [28] Xingsia. L, 1992, An entropy-based aggregate method for minimax optimization, *Engineering Optimization* 18, 277-285.
- [29] Xue. X, Bian. W, 2007, A project neural network for solving degenerate convex quadratic program, *Neural Networks* 70, 2449–2459.
- [30] Yang. Y, Cao. J, 2008, A feedback neural network for solving convex constraint optimization problems, *Applied Mathematics and Computation*, 201, 340–350.
- [31] Yang. Y, Cao. J, 2010, The optimization technique for solving a class of non-differentiable programming based on neural network method, *Nonlinear Analysis* 11, 1108-1114.
- [32] Yang. Y, Xu. X, 2007, The projection neural network for solving convex nonlinear programming, in: Huang, D.-S., Heutte, L., Loog, M. (Eds.), *ICIC 2007, Lecture Notes in Artificial Intelligence*, vol, 4682, Springer-Verlag, Berlin, Heidelberg, pp. 174–181.
- [33] Yongqing. Y, Jinde. C, Xianyum. X, Jiao. L, A generalized neural network for solving a class of minimax optimization problems with linear constraints, 2012, *Applied Mathematics and Computation* 218, 7528- 7537.

واژه‌نامه فارسی به انگلیسی

Bias	احتساب تورش
LaSalle's invariance principle	اصل تغییرناپذیری لازال
Strictly monotonic	اکیداً یکنوا
Gradient vector	بردار گرادیان
Convex quadratic programming	برنامه‌ریزی محدب درجه دوم
Optimality	بهینگی
Non-smooth optimization	بهینه‌سازی غیر هموار
Penalty parameter	پارامتر جریمه
Scale parameter	پارامتر مقیاس
Globally asymptotically stable	پایدار مجانبی سراسری
Stability	پایداری
Affine function	تابع آفینی
Lyapunov	تابع لیاپانوف
Convex function	تابع محدب
Nonlinear complementarity problem(NCP)	تابع مکمل غیرخطی
Universal Approximators	تقریب زن‌های عام
Linearly separable	جدایی‌ناپذیر خطی

Local solution	جواب محلی
Multi objective	چندسطحی
Multilayer	چندلایه‌ای
Multi level	چندهدفه
Positive limit	حد مثبت
Lipschitz	لیپ شیتز
Jacobian matrix	ماتریس ژاکوبین
Hessian matrix	ماتریس هسین
Affine set	مجموعه آفینی
Invariant set	مجموعه تغییرناپذیر
Functional invariant set	مجموعه تغییرناپذیر اساسی
Positive invariant set	مجموعه تغییرناپذیر مثبت
Negative invariant set	مجموعه تغییرناپذیر منفی
Positive limit set	مجموعه حدود مثبت
Convex set	مجموعه محدب
Positive definite	معین مثبت
Global minimum	مینیمم سراسری
Norm	نرم
Equilibrium point	نقطه تعادل
Positive semi-definite	نیمه معین مثبت
Monotone	یکنوا
Variational inequalities	وردشی
Internal Weight	وزن‌های داخلی

Global convergance همگرایی سراسری

Strongly smooth قویاً هموار

واژه‌نامه انگلیسی به فارسی

Affine function	تابع آفینی
Affine set	مجموعه آفینی
Bias	احتساب تورش
Convex function	تابع محدب
Convex quadratic programming	برنامه‌ریزی محدب درجه دوم
Convex set	مجموعه محدب
Equilibrium point	نقطه تعادل
Functional invariant set	مجموعه تغییرناپذیر اساسی
Globally asymptotically stable	پایدار مجانبی سراسری
Global convergance	همگرایی سراسری
Global minimum	مینیمم سراسری
Gradient vector	بردار گرادیان
Hessian matrix	ماتریس هسین
Invariant set	مجموعه تغییرناپذیر
Internal Weight	وزن‌های داخلی
Jacobian matrix	ماتریس ژاکوبین
LaSalle's invariance principle	اصل تغییرناپذیری لازال

Linearly separable	جدایی ناپذیر خطی
Lipschitz	لیپ شیتز
Local solution	جواب محلی
Lyapunov	تابع لیاپانوف
Monotone	یکنوا
Multilayer	چندلایه‌ای
Multi level	چندهدفه
Multi objective	چندسطحی
Negative invariant set	مجموعه تغییرناپذیر منفی
Nonlinear complementarity problem(NCP)	تابع مکمل غیرخطی
Non-smooth optimization	بهینه‌سازی غیر هموار
Norm	نرم
Optimality	بهینگی
Penalty parameter	پارامتر جریمه
Positive definite	معین مثبت
Positive invariant set	مجموعه تغییرناپذیر مثبت
Positive limit	حد مثبت
Positive semi-definite	نیمه معین مثبت
Scale parameter	پارامتر مقیاس
Stability	پایداری
Strictly monotonic	اکیداً یکنوا
Strongly smooth	قویاً هموار

Surname: Rezaei

Name: Elahe

Title: A computational intelligent method for solving a class of min-max problems.

Supervisor: Dr. Alireza Nazemi

Degree: Master of Science

Subject: Applied Mathematics

Field: Differential Equations

Technology University of Shahrood

Faculty Of Mathematical Sciences

Date: 92/8/8

Number of pages: 83

Keywords: Neural network model, Stability, Lyapunov, Equilibrium point, Global convergence, Perceptron, Non-smooth

Abstract

In this thesis, we explain about norm, Jacobian matrix, Hessian matrix, Gradient vector and convex function; next pay attention to optimality with the requirements and show concepts of stability and energy function in dynamical system. then the structure of the neural network model and mathematical model of neurons represent. we want to state short history of neural network model to solve optimization problems and previous models to solve optimization problems. The third chapter, we show one model of neural network to solve convex nonlinear optimization problems based on two cases are likely, optimality, analysis of convex function, Lyapunov's stability and Lasalle's invariance principal and stable equilibrium point of the neural network, we proposed the optimal solution is a convex nonlinear programming problems. we also show the proposed neural network is stable in the concept of Lyapunov and globally convergent to an exact optimal solution to the original problem. several examples are given to show the effectiveness of the proposed model. The last chapter, we will describe an optimization technique that class of non-smooth optimization problems are used. To demonstrate the efficiency of the model, some examples are solved using this model.



Technology 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

**A computational intelligent method for
solving a class of min-max problems.**

Supervisor

Dr. Alireza Nazemi

by

Elahe Rezaei

92/8/8