

دانشکده علوم ریاضی

گروه ریاضی کاربردی

پایان نامه

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

عنوان

حل مسأله ربات‌های چندانگشتی با استفاده از یک مدل شبکه عصبی بازگشتی

استاد راهنما

آقای دکتر علیرضا ناظمی

پژوهشگر

مهدی حسینی

مهر ۱۳۹۲

نام خانوادگی دانشجو: حسینی

نام: مهدی

عنوان: حل مسأله ربات‌های چندانگشتی با استفاده از یک مدل شبکه عصبی بازگشتی

استاد راهنما: آقای دکتر علیرضا ناظمی

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

دانشگاه: صنعتی شاهرود دانشکده علوم ریاضی

تاریخ فارغ‌التحصیلی: مهر ۱۳۹۲

تعداد صفحات: ۶۴

واژگان کلیدی: شبکه‌های عصبی، برنامه‌ریزی محدب، همگرایی، پایداری، ربات‌های چندانگشتی

چکیده

امروزه ربات‌های چندانگشتی به علت کاربردهای وسیع در صنعت و مهندسی مورد توجه زیادی قرار گرفته‌اند ولی استفاده از آنها مستلزم استفاده از آنالیز پیچیده‌ای است، زیرا لازم است با اعمال یک نیروی آزمند به جسم، بدون اینکه ربات جسم را رها کند و یا اینکه شدت فشار در محل اعمال نیرو باعث متلاشی شدن آن شود، جسم را جابجا کند. در این پایان‌نامه برای حل مسأله ربات‌های چندانگشتی از نوعی شبکه عصبی مصنوعی استفاده می‌شود. این شبکه عصبی مصنوعی بر پایه نظریه‌های دوگان و بهینه‌سازی، آنالیز محدب، قضیه پایداری لیاپانوف و اصل تغییرناپذیری لازل بنا شده است. مطابق قضیه نقطه بحرانی، ثابت خواهیم کرد که نقطه تعادل شبکه عصبی پیشنهاد شده، معادل با جواب بهینه مسأله بهینه‌سازی اکیدا محدب و درجه دوم مطرح شده است. با بکارگیری یک تابع لیاپانوف مناسب، نشان خواهیم داد که مدل شبکه عصبی پیشنهادی، پایدار به مفهوم لیاپانوف است و همچنین همگرای سراسری به تنها نقطه بهینه مسأله اصلی است. با ارائه مثالی شدنی بودن و کارائی شبکه عصبی پیشنهاد شده را نشان می‌دهیم.

تقدیم به همه آشنایی که

می خوانند بیشتر بدانند

خدایا...۱

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

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

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

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

اگر تنها ترین تنها شوم، باز خدا هست

او جان‌شین همه نداشتن هست...

فهرست مطالب

۱	مقدمه و مفاهیم اولیه	۱
۲	۱.۱ مقدمه	۱.۱
۲	۲.۱ مجموعه‌های آفین و محدب	۲.۱
۳	۳.۱ توابع آفین و محدب	۳.۱
۷	۴.۱ پایداری	۴.۱
۱۰	۵.۱ تابع انرژی	۵.۱
۱۱	۶.۱ اصل تغییرناپذیری لازال	۶.۱
۱۳	۲ مقدمه‌ای بر شبکه‌های عصبی	۲
۱۴	۱.۲ مقدمه	۱.۲
۱۵	۲.۲ مبنای پزشکی شبکه‌های عصبی	۲.۲
۱۶	۳.۲ اعصاب مصنوعی	۳.۲
۱۸	۴.۲ شبکه‌های عصبی مصنوعی	۴.۲
۲۰	۵.۲ علائم برداری و ماتریسی	۵.۲
۲۲	۶.۲ یادگیری و فراخوانی	۶.۲
۲۴	۷.۲ یادگیری تحت نظارت	۷.۲
۲۵	۸.۲ خصوصیات شبکه‌های عصبی مصنوعی	۸.۲
۲۶	۹.۲ تاریخچه کاربرد شبکه‌های عصبی مصنوعی در حل مسائل بهینه‌سازی	۹.۲
۲۸	۱۰.۲ مدل‌های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه‌سازی	۱۰.۲
۳۵	۳ ساختن یک مدل دینامیکی کارا برای حل مسائل برنامه‌ریزی غیرخطی محدب	۳
۳۶	۱.۳ مقدمه	۱.۳
۳۷	۲.۳ ساختن مدل دینامیکی	۲.۳

۳۸	۳.۳	بررسی پایداری و همگرایی
۴۲	۴.۳	شبیه‌سازی عددی
۴۹	۴	حل مسئله ربات‌های چندانگشتی با استفاده از یک مدل شبکه عصبی
۵۰	۱.۴	مقدمه
۵۲	۲.۴	بیان مسأله
۵۴	۳.۴	حل مسأله بهینه‌سازی نیروی آزمند در یک ربات سه‌انگشتی
۵۸		مراجع
۶۱		واژه‌نامه انگلیسی به فارسی

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^۱ و مجموعه‌های محدب^۲ و همچنین تعاریف توابع آفین^۳ و توابع محدب^۴، انواع مختلف نرم‌ها^۵، بردار گرادیان^۶، ماتریس هسین^۷ و ماتریس ژاکوبین^۸ را بیان نموده و در خصوص مسائل بهینه‌سازی مقید و نامقید به ذکر شرایط لازم و کافی بهینگی^۹ می‌پردازیم. سپس مفاهیم پایداری^{۱۰} و تابع انرژی در سیستم‌های دینامیکی را ارائه می‌دهیم.

۲.۱ مجموعه‌های آفین و محدب

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

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

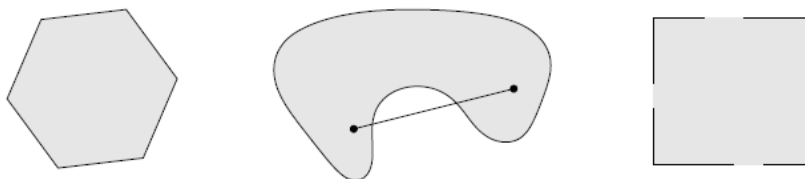
یک نقطه به‌صورت $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ که $\theta_1 + \theta_2 + \dots + \theta_k = 1$ را ترکیب آفین نقاط x_1, \dots, x_k می‌نامیم. در واقع هر مجموعه آفین شامل تمام ترکیبات آفین نقاط خودش است.

تعریف ۲.۲.۱. 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:

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

در واقع مجموعه محدب مجموعه‌ای است که هیچ مانعی در مسیر پاره‌خط عبوری از هر دو نقطه دلخواه آن وجود نداشته باشد. نقطه به‌صورت $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$ که $\theta_1 + \theta_2 + \dots + \theta_k = 1$ ، $0 \leq \theta_i \leq 1$ را ترکیب محدب نقاط x_1, \dots, x_k می‌نامیم. در واقع هر مجموعه محدب شامل تمام ترکیبات محدب نقاط خودش است. شکل (۱.۱) چند نمونه از مجموعه‌های محدب و نامحدب را نشان می‌دهد.

^۱ Affine sets
^۲ Convex sets
^۳ Affine functions
^۴ Convex functions
^۵ Norms
^۶ Gradient vector
^۷ Hessian matrix
^۸ Jacobian matrix
^۹ Optimality
^{۱۰} Stability



شکل ۱.۱: برخی مجموعه‌های محدب و نامحدب. شکل چپ یک شش ضلعی کراندار و محدب است. شکل میانی بدلیل آنکه پاره‌خط گذرنده از دو نقطه آن تماماً شامل مجموعه نیست، نامحدب است. شکل سمت راست همه نقاطش کراندار نیست، لذا نامحدب است.

۳.۱ توابع آفین و محدب

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

$$f(x + y) = f(x) + f(y),$$

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

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 به‌صورت 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. 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:

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

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

تعریف ۴.۳.۱. اگر $\bar{x} \in X$ و ε -همسایگی از \bar{x} مثل $N_\varepsilon(\bar{x})$ موجود باشد که به ازای هر $x \in X \cap N_\varepsilon(\bar{x})$ داشته باشیم $f(\bar{x}) \leq f(x)$ آنگاه، \bar{x} یک کمینه موضعی برای f است.

تعریف ۵.۳.۱. تابع $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$ را تابع نرم می‌گوییم، هرگاه برای هر $x, y \in \mathbb{R}^n$ و هر $\alpha \in \mathbb{R}$:

$$\begin{cases} \|x\| \geq 0, \\ \|x\| = 0 \Leftrightarrow x = 0, \\ \|\alpha x\| = |\alpha| \|x\|, \\ \|x + y\| \leq \|x\| + \|y\|. \end{cases}$$

و برای بردار $x \in \mathbb{R}^n$ داریم:

$$\begin{aligned}\|x\|_1 &= |x_1| + \dots + |x_n|, \\ \|x\|_2 &= \sqrt{x^T x} = \sqrt{x_1^2 + \dots + x_n^2}, \\ \|x\|_r &= \sqrt[r]{x_1^r + \dots + x_n^r}, \quad r \geq 1.\end{aligned}$$

تعریف ۶.۳.۱. فرض کنید $f: \mathbb{R}^n \rightarrow \mathbb{R}$ باشد. گرادیان تابع f در x برابر خواهد بود با:

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

تعریف ۷.۳.۱. 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 اینصورت ماتریس هسین f را با H_f نشان می‌دهند و به‌صورت زیر تعریف می‌شود:

$$H_f = \begin{bmatrix} \nabla \frac{\partial f}{\partial x_1} \\ \nabla \frac{\partial f}{\partial x_2} \\ \vdots \\ \nabla \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

تعریف ۸.۳.۱. 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 اینصورت ماتریس ژاکوبین f در x_0 را با $J_f(x_0)$ نشان می‌دهند. بنابراین:

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

تعریف ۹.۳.۱. زیرمجموعه E از فضای متریک X را یک مجموعه فشرده می‌گوییم، اگر هر پوشش باز E یک زیرپوشش باز متناهی داشته باشد.

شرایط لازم و کافی برای مسائل نامقید

قضیه ۱۰.۳.۱. (شرط لازم مرتبه اول [۴]). فرض کنید X یک مجموعه باز در \mathbb{R}^n باشد. مسأله برنامه‌ریزی نامقید زیر را در نظر بگیرید:

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

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

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$ در \hat{x} مشتق پذیر است. اگر \hat{x} یک کمینه موضعی باشد^{۱۱}، آنگاه $\nabla f(\hat{x}) = 0$.

تعریف ۱۱.۳.۱. 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^{۱۲} روی E نامیده می شود، اگر:

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

۲. معین مثبت^{۱۳} روی E نامیده می شود، اگر:

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

۳. معین مثبت قوی^{۱۴} روی E نامیم، اگر عددی مثبت مانند α وجود داشته باشد، به طوری که:

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

قضیه ۱۲.۳.۱. اگر $\gamma(x)$ کوچکترین مقدار ویژه ماتریس $M(X)_{n \times n}$ باشد، آنگاه:

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

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

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

قضیه ۱۳.۳.۱. (شرط لازم مرتبه دوم [۴]). فرض کنید مشتق دوم $f: \mathbb{R}^n \rightarrow \mathbb{R}$ در \hat{x} موجود باشد. اگر \hat{x} یک کمینه موضعی باشد، آنگاه $\nabla f(\hat{x}) = 0$ و ماتریس هسین f در \hat{x} نیمه معین مثبت است.

قضیه ۱۴.۳.۱. (شرط کافی [۴]). 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.

^{۱۱}Local minimum

^{۱۲}Positive semi-definite

^{۱۳}Positive definite

^{۱۴}Strong positive definite

قضیه ۱۵.۳.۱. ([۴]). 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 ۱۶.۳.۱. ([۴]). اگر $f(x)$ بر مجموعه محدب E تابعی محدب باشد، آنگاه $f(x)$ دارای یک کمینه موضعی است. اگر چنین کمینه‌ای یافت شود، یک کمینه سراسری^{۱۵} است و بر روی مجموعه محدب به دست آورده می‌شود.

دو قضیه بالا بیان می‌کنند که برای یک مسأله کلی کمینه‌سازی، 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.

شرایط لازم و کافی برای مسائل مقید

قضیه ۱۷.۳.۱. (شرایط لازم کاروش-کان-تاکر^{۱۶} (K.K.T), ([۴]). فرض کنید X یک مجموعه باز در \mathbb{R}^n باشد. مسأله کمینه‌سازی مقید زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (۳.۱)$$

$$\text{subject to } \begin{cases} g_k(x) \leq 0, & k = 1, \dots, m, \\ h_j(x) = 0, & j = 1, \dots, l, \\ x = (x_1, x_2, \dots, x_n)^T \in X. \end{cases} \quad (۴.۱)$$

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. به علاوه فرض کنید $\nabla g_k(\hat{x})$ برای $k \in K$ و $\nabla h_j(\hat{x})$ برای $j = 1, \dots, l$ مجموعاً مستقل خطی باشند. اگر \hat{x} کمینه موضعی برای (۴.۱)-(۳.۱) 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:

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

^{۱۵}Global minimum

^{۱۶}Karush-Kuhn-Tucker

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

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

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

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

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

قضیه ۱۸.۳.۱. (شرایط کافی کاروش-کان-تاگر (K.K.T), [۴]). فرض کنید X یک مجموعه باز در \mathbb{R}^n باشد، مسأله (۴.۱)-(۳.۱) 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-تاگر 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 (۴.۱)-(۳.۱) خواهد بود.

قضیه ۱۹.۳.۱. اگر در مسأله (۴.۱)-(۳.۱) توابع f و g_k برای $k = 1, \dots, m$ محدب باشند و توابع h_p برای $p = 1, \dots, l$ آفینی باشند، آنگاه شرایط کاروش-کان-تاگر، لازم و کافی اند.

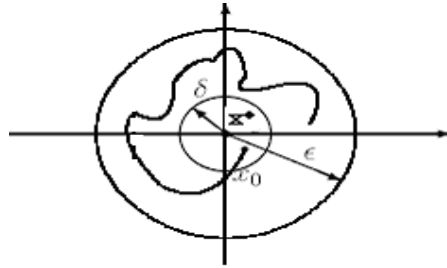
۴.۱ پایداری

سیستم دینامیکی زیر را در نظر بگیرید:

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

که در آن $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ است. x^* یک نقطه تعادل^{۱۷} برای (۵.۱) نامیده می‌شود، اگر $f(x^*) = 0$.

^{۱۷}Equilibrium point



شکل ۲.۱: پایداری لیاپانوف

تعریف ۱.۴.۱. فرض کنید $x(t)$ یک جواب (۵.۱) باشد. با توجه به شکل (۲.۱) نقطه تعادل x^* ، پایدار به مفهوم لیاپانوف^{۱۸} است، اگر برای هر $x(t_0) = x_0$ و هر $\epsilon > 0$ ، یک $\delta > 0$ وجود داشته باشد به طوری که اگر $\|x(t) - x_0\| \leq \delta$ ، آنگاه:

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

تعریف ۲.۴.۱. سیستم دینامیکی (۵.۱) همگرای سراسری^{۱۹} به مجموعه

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

گفته می شود اگر هر جواب دلخواه $x(t)$ از سیستم در رابطه زیر صدق کند:

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

که در آن

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

تعریف ۳.۴.۱. سیستم دینامیکی در نقطه تعادل یکتای x^* پایدار مجانبی سراسری^{۲۰} نامیده می شود، اگر x^* به مفهوم لیاپانوف پایدار باشد و در شرط زیر صدق کند:

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

تعریف ۴.۴.۱. مجموعه $M \subseteq \mathbb{R}^n$ ، یک مجموعه تغییرناپذیر^{۲۱} نسبت به سیستم (۵.۱) گفته می شود، اگر $x(t_0) \in M$ و $t_0 \geq 0$ ، آنگاه به ازای هر $t \geq t_0$ ، $x(t) \in M$ باشد.

^{۱۸} Lyapunov

^{۱۹} Global convergent

^{۲۰} Globally asymptotically stable

^{۲۱} Invariant set

تعریف ۵.۴.۱. نگاشت $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ در شرط لیپ شیتز^{۲۲} 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 موضعی 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 ۶.۴.۱. نگاشت $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ یکنوا^{۲۳} 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 F روی \mathbb{R}^n اکیداً یکنوا^{۲۴} است، اگر نامساوی فوق به صورت اکید برای هر $x \neq y$ برقرار باشد. همچنین نگاشت 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 ۷.۴.۱. ([۵۲]). فرض می‌کنیم که $F: K \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ ، یک تابع به طور پیوسته مشتق پذیر روی K باشد. اگر ماتریس ژاکوبین F که لزوماً متقارن نیست، نیمه معین مثبت (معین مثبت) باشد، آنگاه F یکنوا (یکنوای قوی) است.

قضیه ۸.۴.۱. ([۱۰]). در سیستم دینامیکی (۵.۱) اگر $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ یک تابع پیوسته باشد، آنگاه برای هر $t_0 > 0$ و $x_0 \in \mathbb{R}^n$ ، یک جواب موضعی^{۲۵} $x(t)$ به ازای $t \in [t_0, \tau)$ 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 $y^* = 0$ ، $\dot{y} = g(y)$ است. زیرا:

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

^{۲۲}Lipschitz

^{۲۳}Monotone

^{۲۴}Strictly monotone

^{۲۵}Local solution

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

۵.۱ تابع انرژی

قضیه ۱.۵.۱. ([۳۸]). فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم (۵.۱) و $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ یک تابع به طور پیوسته مشتق پذیر باشد، اگر:

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

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

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

آنگاه $x = 0$ نقطه پایداری سیستم خواهد بود و $E(x)$ را «تابع لیاپانوف» یا «تابع انرژی» برای سیستم (۵.۱) می‌نامیم.

قضیه ۲.۵.۱. ([۳۸]). فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم (۵.۱) و $E : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ یک تابع به طور پیوسته مشتق پذیر باشد، اگر:

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

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

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

آنگاه $x = 0$ پایدار مجانبی خواهد بود.

قضیه ۳.۵.۱. ([۳۸]). فرض کنید که $x = 0$ یک نقطه تعادل برای سیستم (۵.۱) و $E : \mathbb{R}^n \rightarrow \mathbb{R}$ یک تابع به طور پیوسته مشتق پذیر باشد، اگر:

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

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

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

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

۶.۱ اصل تغییرناپذیری لازال

تعریف ۱.۶.۱. فرض کنید $x(t)$ جوابی از سیستم دینامیکی $\dot{x} = f(x)$ باشد. نقطه $p \in \mathbb{R}^n$ حد مثبت^{۲۶} $x(t)$ نامیده می‌شود، اگر دنباله $(t_n)_{n \geq 0}$ موجود باشد به طوری که $\lim_{n \rightarrow \infty} t_n = \infty$ و همچنین $\lim_{n \rightarrow \infty} x(t_n) = p$. مجموعه حدود مثبت^{۲۷} دنباله مسیرهای $(x(t))_{t \geq 0}$ را با L^+ نمایش می‌دهیم.

تعریف ۲.۶.۱. مجموعه M را مجموعه تغییرناپذیر مثبت^{۲۸} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌گوییم هرگاه:

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

تعریف ۳.۶.۱. مجموعه M را مجموعه تغییرناپذیر منفی^{۲۹} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌گوییم هرگاه:

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

تعریف ۴.۶.۱. مجموعه M را مجموعه تغییرناپذیر اساسی^{۳۰} برای سیستم دینامیکی $\dot{x} = f(x)$ می‌گوییم اگر تغییرناپذیر مثبت و همچنین، تغییرناپذیر منفی باشد.

تعریف ۵.۶.۱. دنباله مسیرهای $(x(t))_{t \geq 0}$ پایدار مجانبی وابسته به M نامیده می‌شود، اگر 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^{۳۱} به بیان یک لم می‌پردازیم.

لم ۶.۶.۱. ([۱۹]). 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 $x(t)$ 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 دنباله مسیرهای $(x(t))_{t \geq 0}$:

• ناتهی است.

^{۲۶}Positive limit

^{۲۷}Positive limits set

^{۲۸}Positive invariant set

^{۲۹}Negative invariant set

^{۳۰}Functional invariant set

^{۳۱}LaSalle's invariance principle

● فشرده است.

● تغییرناپذیر مثبت است.

قضیه ۷.۶.۱. (اصل تغییرناپذیری لازال، [۱۹]). 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 ۸.۶.۱. ([۱۹]). فرض کنید مبدأ نقطه تعادل سیستم دینامیکی $\dot{x} = f(x)$ باشد و تابع $V : D \rightarrow \mathbb{R}$ روی $D \subset \mathbb{R}^n$ پیوسته مشتق‌پذیر و شامل مبدأ باشد به‌طوری‌که روی مجموعه D داشته باشیم $\dot{V}(x) \leq 0$. همچنین مجموعه $E = \{x \in D \mid \dot{V}(x) = 0\}$ را در نظر می‌گیریم. اگر E فقط شامل نقطه $x(t) \equiv 0$ باشد، آنگاه مبدأ 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 می‌پردازیم. در انتها به بیان مدل‌های ارائه شده پیشین برای حل مسائل بهینه‌سازی خواهیم پرداخت.

کار بر روی شبکه‌های عصبی مصنوعی، با الهام از عملکرد مغز انسان از آنجا شروع شد که دانشمندان دریافتند مغز بشر به گونه‌ای کاملاً متفاوت از کامپیوترهای دیجیتالی مرسوم محاسبات را انجام می‌دهد. مغز یک سیستم پردازش داده بسیار پیچیده است که از واحدهای ساختاری به نام سلول عصبی یا نرون^۱ تشکیل شده است. شبکه عصبی سیستمی پویا و غیرخطی است که از تعداد زیادی واحد پردازش (نرون‌ها) و اتصالات بین این واحدهای پردازش تشکیل می‌شود.

یک شبکه عصبی بر خلاف کامپیوتر که نیازمند دستورهای کاملاً صریح و مشخص است، به مدل‌های ریاضی محض نیازی ندارد، بلکه مانند انسان تجربه کسب کرده و سپس نتیجه این تجربیات را تعمیم می‌دهد. امروزه این شبکه‌ها تبدیل به ابزاری قدرتمند و عمومی شده‌اند که برای حل مسائل از قبیل تخمین (تقریب)، تشخیص الگو^۲، رباتیک، کنترل، و به‌طور کلی در هر جا که نیاز به یادگیری نگاشت خطی و یا غیرخطی باشد، به کار می‌روند [۱۸].

برای حل هر مسئله‌ای، شبکه‌های عصبی سه مرحله را طی می‌کنند: آموزش^۳، تعمیم^۴ و اجرا^۵. آموزش، فرآیندی است که طی آن شبکه می‌آموزد تا الگوی موجود در ورودی‌ها را (که به‌صورت 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 فرآیند درونیایی نیست به‌دست می‌آید. استفاده از شبکه برای انجام عملکردی که به آن منظور طراحی شده است را اجرا می‌گویند.

در اثر آموزش دیدن شبکه، وزن‌های داخلی^۶ (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

^۱Neuron^۲Pattern recognition^۳Training^۴Generalization^۵Operation^۶Internal weight

داخلی، هیچگونه تصویر روشنی از اعتبار جواب‌های مسأله به دست 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 به کار رفته است. از جمله آن‌ها می‌توان به مدل‌های جعبه سیاه^۷، مدل‌های تجربی، تقریب‌زن‌های عام^۸ و مدل‌های موازی^۹ اشاره نمود [۲۰].

۲.۲ مبنای پزشکی شبکه‌های عصبی

مغز انسان سیستم بسیار پیچیده‌ای است که قادر به تفکر، یادآوری و حل مسائل است. تاکنون تلاش‌های زیادی صورت گرفته است تا مدل‌های کامپیوتری از عملکرد مغز تقلید نمایند. اگرچه نتایج این تلاش‌ها بسیار تماشایی است ولی در مقایسه با عملکرد پیچیده مغز انسان کاملاً کمرنگ هستند. عصب یک واحد سلولی اساسی از سیستم عصبی مغز است. عصب یک عنصر پردازشگر ساده است که از طریق مسیرهای ورودی بنام دندریت‌ها^{۱۰} علائمی را از سایر اعصاب دریافت کرده و آن‌ها را با هم ترکیب می‌کند. اگر این علامت ورودی مرکب به اندازه کافی قوی باشد، عصب به اصطلاح شلیک می‌کند و یک علامت خروجی را در طول اکسون^{۱۱} که به دندریت سایر اعصاب متصل است ارسال می‌کند. شکل (۱.۲) طرحی از قسمت‌های مختلف یک عصب است. هر علامتی که در طول دندریت یک عصب فرستاده می‌شود از یک سیناپس^{۱۲} یا اتصال سیناپسی عبور می‌کند. این اتصال، یک شکاف بسیار کوچک در دندریت است که با نوعی مایع هادی عصبی^{۱۳} پر شده است که جریان بارهای الکتریکی را تندتر یا کندتر می‌کند. اصولاً عملکرد عصب‌ها بر پایه فعل و انفعالات شیمیایی است و این مایع هادی عصبی علائمی الکتریکی تولید می‌کند و به سوی هسته‌ی عصب می‌فرستد. تنظیم میزان مقاومت یا هدایت الکتریکی این فاصله سیناپسی، یک فرآیند بسیار مهم است. به علاوه همین تنظیم هدایت باعث حفظ کردن مطالب و یادگیری می‌شود. وقتی قدرت سیناپسی اعصاب تنظیم شود مغز یادگیری می‌نماید و اطلاعات را ذخیره می‌کند.

یکی از زمینه‌هایی که شبکه‌های عصبی مصنوعی در آن بسیار مفید واقع می‌شوند سرعت است. وقتی شخصی به سمت یک اتاق قدم برمی‌دارد حدود نیم ثانیه طول می‌کشد تا شخص دیگری را که در آن اتاق است تشخیص دهد. این فرآیند تشخیص، شامل حدود ۲۰۰ تا ۲۵۰ عمل جداگانه در مغز است. از نقطه نظر سرعت، این

^۷Black-Box models

^۸Universal approximators

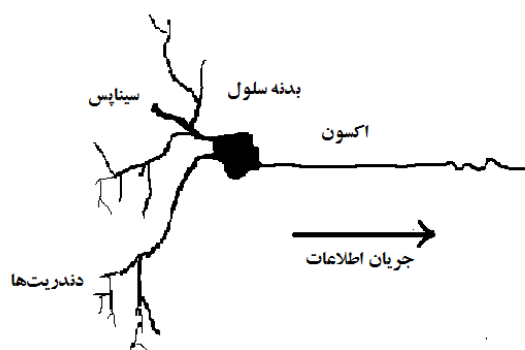
^۹Parallel models

^{۱۰}Dendrites

^{۱۱}Axon

^{۱۲}Synapse

^{۱۳}Neurotransmitter fluid



شکل ۱.۲: طرح قسمت‌های مختلف یک عصب بیولوژیکی

بدان معناست که مغز انسان با سرعتی حدود ۴۰۰ تا ۵۰۰ هرتز کار می‌کند. کامپیوترهای دیجیتالی جدید با سرعتی حدود ۱۰۰ تا ۲۰۰ مگاهرتز عمل می‌کنند که سرعت بسیار بالاتری نسبت به مغز دارند. ولی از سوی دیگر این مزیت به نحو محسوسی کاهش می‌یابد زیرا کامپیوترهای دیجیتالی به روش سری کار می‌کنند در حالی که مغز به روش موازی عمل می‌کند. البته در سال‌های اخیر تراشه شبکه عصبی ساخته شده که کامپیوترهای دیجیتالی را قادر می‌سازد به روش موازی عمل کنند.

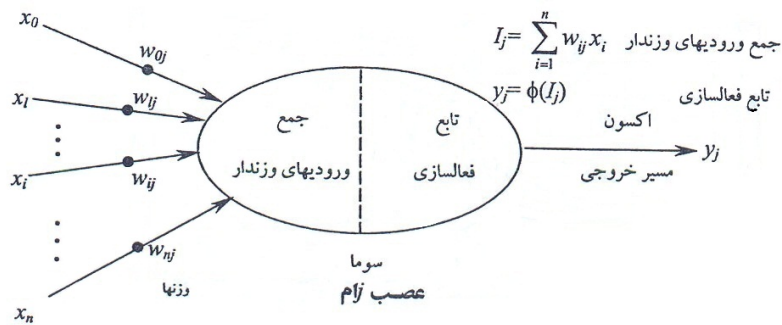
فهرست اصطلاحات در زمینه شبکه‌های عصبی هنوز استفاده نشده است. ممکن است بعضی کتاب‌ها یا مقالات موجود باشند که از شبکه‌های عصبی مصنوعی به عنوان سیستم‌های ارتباطی و از اعصاب مصنوعی به عنوان عناصر پردازشگر^{۱۴}، گره‌ها و به عبارت ساده‌تر اعصاب یاد شود. در این پایان نامه ما از اصطلاحات اعصاب و شبکه‌های عصبی استفاده می‌کنیم و از صفت مصنوعی صرف نظر می‌نماییم زیرا فقط با اعصاب مصنوعی سروکار خواهیم داشت.

۳.۲ اعصاب مصنوعی

یک عصب مصنوعی مدلی است که اجزاء آن شباهت مستقیمی به اجزاء واقعی دارند. شکل (۲.۲) نمایی از یک عصب مصنوعی است. علائم ورودی با x_0, x_1, \dots, x_n مشخص شده‌اند. این علائم، متغیرهایی پیوسته هستند و نه پالس‌های الکتریکی که در مغز رخ می‌دهند. هر یک از این مقادیر ورودی تحت تأثیر وزنی (که گاه وزن سیناپسی نامیده می‌شود) قرار می‌گیرند که تابع این وزن شبیه اتصال سیناپسی در یک عصب واقعی است. بسته به میزان هدایت یا مقاومت جریان علائم الکتریکی، این وزن‌ها می‌توانند مثبت یا منفی

^{۱۴}Processing elements

باشند. این عناصر پردازشگر از دو قسمت تشکیل شده‌اند. قسمت اول ورودی‌های وزن‌دار را با هم جمع می‌زند و کمیتی به نام I به دست می‌آورد؛ قسمت دوم یک صافی غیرخطی است که تابع فعالسازی^{۱۵} (تابع محرک) نامیده می‌شود و از طریق آن جریان‌های علائم ورودی ترکیب می‌شوند.



شکل ۲.۲: طرح شماتیک یک عصب مصنوعی

تابع فعالسازی می‌تواند مانند شکل (۳.۲) قسمت (الف) یک تابع آستانه‌ای باشد که فقط زمانی اطلاعات را عبور می‌دهد که خروجی I که مربوط به قسمت اول عصب مصنوعی است از مقدار آستانه‌ای T تجاوز کند. همچنین این تابع می‌تواند مانند شکل (۳.۲) قسمت (ب) یک تابع علامت^{۱۶} (که گاهی تابع اندازه‌گیری نامیده می‌شود) باشد که وقتی خروجی کمتر از مقدار آستانه‌ای T باشد اطلاعات منفی فرستاده و وقتی خروجی بیشتر از مقدار آستانه‌ای T باشد اطلاعات مثبت فرستاده می‌شود. در اغلب موارد، تابع فعالسازی یک تابع پیوسته است که به‌طور تدریجی بین دو مقدار مجانب مثلاً صفر و ۱ یا ۱- و ۱+ تغییر می‌کند و تابع S مانند^{۱۷} یا تابع سیگموئید نام دارد. پرکاربردترین تابع فعالسازی، تابع لجستیک^{۱۸} است که یکی از انواع توابع فعالسازی S مانند است که در شکل (۴.۲) نشان داده شده است و از رابطه زیر محاسبه می‌شود:

$$\phi(I) = \frac{1}{1 + e^{-\alpha I}} \quad (۱.۲)$$

که α ضریبی است که در تغییرات تابع بین دو مقدار مجانب، شیب تابع را مشخص می‌نماید. تابع فشار^{۱۹} نام گویاتری برای تابع فعالسازی است که نشان می‌دهد این تابع، مقادیر خروجی یک عصب مصنوعی را بین مقادیر مجانب، فشرده یا محدود می‌کند. این محدودسازی باعث می‌شود تا خروجی عناصر پردازشگر در یک محدوده‌ی پویا و قابل قبول قرار گیرد. البته در برخی موقعیت‌های مشخص و قطعی می‌توان

^{۱۵} Activation function

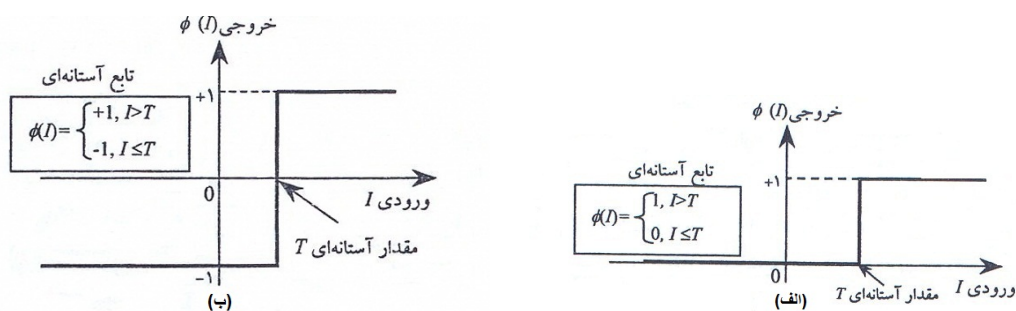
^{۱۶} Signum function

^{۱۷} Sigmoidal function

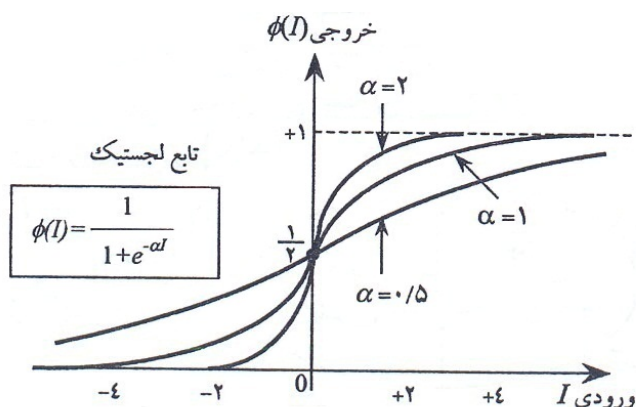
^{۱۸} Logistic function

^{۱۹} Squashing function

از یک رابطه خطی، گاهی فقط در نیم‌فضای سمت راست به عنوان تابع فعالسازی استفاده کرد. ولی باید به این نکته توجه کرد که استفاده از یک تابع خطی، خصوصیت غیرخطی بودن عصب مصنوعی را از بین می‌برد. یک شبکه عصبی بدون ویژگی غیرخطی بودن نمی‌تواند پدیده‌های غیرخطی را مدل‌سازی کند.



شکل ۳.۲: توابع انتقال برای اعصاب: (الف) تابع فعالسازی آستانه‌ای. (ب) تابع فعالسازی علامت



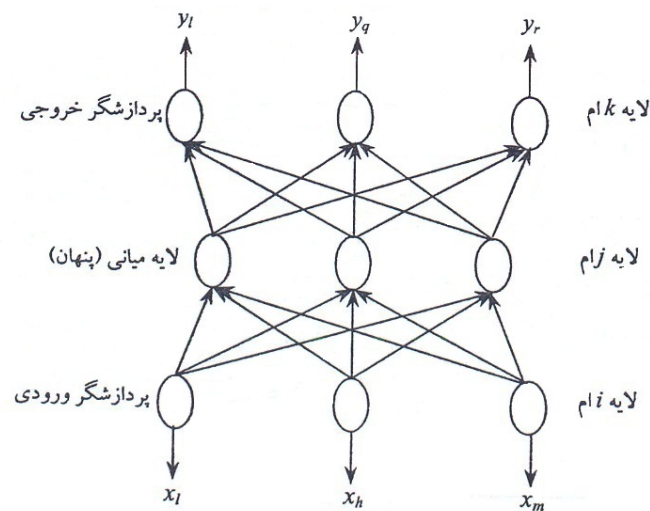
شکل ۴.۲: توابع فعالسازی لجستیک برای مقادیر $\alpha = 2$ و $\alpha = 1$ و $\alpha = 0.5$

۴.۲ شبکه‌های عصبی مصنوعی

یک شبکه عصبی نوعی سیستم پردازش داده‌هاست که از تعداد زیادی عناصر پردازشگر ساده و بسیار مرتبط با هم (یعنی همان اعصاب مصنوعی) تشکیل شده است و در ساختار آن از پوسته دماغی مغز الهام گرفته شده است.

این عناصر پردازشگر معمولاً در لایه‌ها یا صفحات منظمی قرار گرفته‌اند به طوری که بین لایه‌ها ارتباطات کامل یا تصادفی وجود دارد. شکل (۵.۲) این نظم و ترتیب را نشان می‌دهد، به طوری که لایه ورودی به

منزله پردازشگری که پس از پردازش داده‌های ورودی، آن‌ها را به شبکه ارائه می‌دهد. این لایه ورودی یک لایه عصبی محاسباتی نیست زیرا گره‌های آن نه وزن ورودی دارند و نه تابع فعالسازی. لایه فوقانی، لایه خروجی است که خروجی شبکه را در پاسخ به یک ورودی مشخص نشان می‌دهد. سایر لایه‌ها، لایه‌های میانی یا پنهان نامیده می‌شوند زیرا بین آن‌ها و دنیای خارجی ارتباطی وجود ندارد. معمولاً لایه‌های ورودی، پنهان و لایه‌های خروجی به ترتیب لایه‌های i ، m ، k و z نامیده می‌شوند. شبکه‌های عصبی مورد استفاده بر دو نوعند: شبکه‌های عصبی غیرشرکت‌پذیر^{۲۰} که در آن‌ها بردار خروجی با بردار ورودی متفاوت است، و شبکه‌های عصبی خودشرکت‌پذیر^{۲۱} که در آن‌ها بردار ورودی و بردار خروجی یکسانند. یک شبکه عصبی



شکل ۵.۲: نمونه‌ای از ساختار یک شبکه عصبی

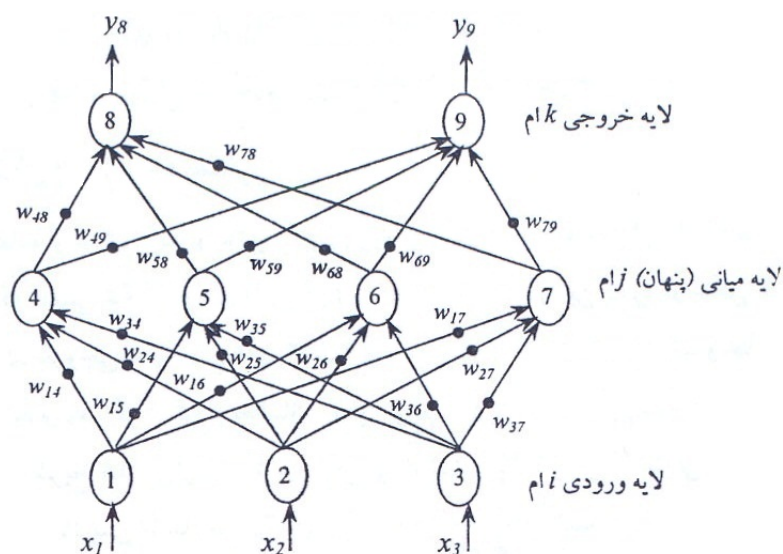
نوعی، شبکه‌ایست با ارتباطات کامل^{۲۲} و به این معناست که بین هر یک از عصب‌های هر لایه با هر یک از عصب‌های لایه بعد، یک ارتباط وجود دارد که نمونه‌ای از آن در شکل (۵.۲) نشان داده شده است. شبکه‌ای که در آن هیچگونه ارتباط افقی بین اعصاب یک لایه نباشد و ارتباطی به سوی لایه‌های قبلی نیز وجود نداشته باشد، شبکه پیش‌خور^{۲۳} نامیده می‌شود. شبکه‌هایی که ارتباطات بازخور (یعنی شبکه‌هایی که ارتباطاتی از یک لایه به سوی لایه قبلی و یا ارتباطات افقی بین اعصاب یک لایه وجود دارند) بین اعصاب برقرار است نیز شبکه‌های مفیدی هستند. در هر دو مورد، این ارتباطات دارای وزنی هستند که باید تحت آموزش قرار گیرد. همانطور که شکل (۶.۲) نشان می‌دهد هر یک از ارتباطات بین اعصاب، دارای یک وزن قابل تنظیم است. این شبکه عصبی ساده، یک شبکه پیش‌خور با ارتباطات کامل است که سه عصب در لایه ورودی،

^{۲۰} Heteroassociative neural network

^{۲۱} Autoassociative neural network

^{۲۲} Fully connected

^{۲۳} Feedforward network



شکل ۶.۲: شبکه عصبی پیش‌خور ساده

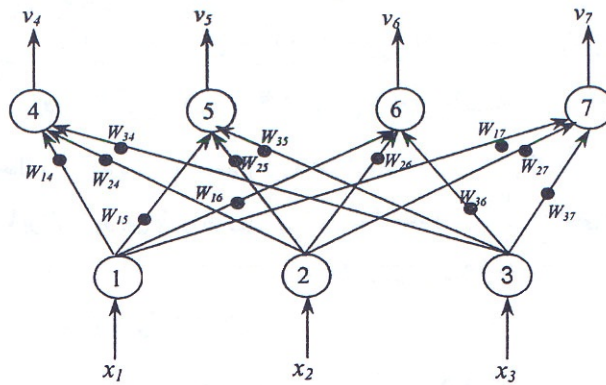
چهار عصب در لایه میانی یا پنهان و دو عصب در لایه خروجی دارد. هریک از وزن‌ها به صورت نقطه توپری به روی ارتباط رسم شده و با علامت w_{ij} نشان داده شده است. به عنوان مثال وزن w_{37} نشان‌دهنده وزنی است که روی ارتباط بین عصب‌های ۳ و ۷ قرار دارد. اکنون شبکه عصبی شکل (۶.۲) را در نظر می‌گیریم که در آن بردار ورودی X از عناصر x_3, x_2, x_1 تشکیل شده و بردار خروجی Y دارای عناصر y_9 و y_8 است. وقتی علامت x_1 به عصب شماره ۱ در لایه ورودی می‌رود خروجی x_1 به کلیه اعصاب مصنوعی در لایه میانی یا پنهان می‌رود و از وزن‌های $w_{14}, w_{15}, w_{16}, w_{17}$ عبور می‌کند. علائم ورودی x_2, x_3 نیز مسیر مشابهی را طی می‌کنند یعنی از طریق وزن‌های مربوطه که در شکل (۶.۲) نشان داده شده‌اند، به عصب‌های شماره ۴، ۵، ۶ و ۷ فرستاده می‌شوند.

۵.۲ علائم برداری و ماتریسی

استفاده از علائم بردار و ماتریس برای ورودی‌ها، خروجی‌ها و وزن‌ها بسیار مناسب است. شکل (۷.۲) در واقع برش شکل (۶.۲) از بالای لایه پنهان است. خروجی‌های عصب‌های ۴، ۵، ۶ و ۷ در بردار V_j نشان داده شده‌اند و عناصر V_j عبارتند از v_4, v_5, v_6 و v_7 . اگر ما توابع فعالسازی را به توابع خطی محدود کنیم روابط ریاضی را می‌توان به شکل ماتریسی نوشت و بنابراین بردار ستونی V_j معادل است با ضرب نقطه‌ای ماتریس وزن‌ها یعنی W_{ij} در بردار ورودی X_i . این رابطه را به صورت زیر می‌توان نشان داد:

$$\begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} w_{14} & w_{24} & w_{34} \\ w_{15} & w_{25} & w_{35} \\ w_{16} & w_{26} & w_{36} \\ w_{17} & w_{27} & w_{37} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (۲.۲)$$

$$V_j = W_{ij} \cdot X_i \quad (۳.۲)$$



شکل ۷.۲: قسمت تحتانی شبکه عصبی که از بالای لایه پنهان برش داده شده است.

به روشی مشابه با استفاده از نیمه فوقانی شبکه عصبی که در شکل (۸.۲) رسم شده می‌توان نشان داد که بردار خروجی Y_k با ضرب نقطه‌ای ماتریس اوزان W_{jk} و بردار ورودی V_j معادل است. این رابطه را به صورت زیر می‌توان نشان داد:

$$\begin{bmatrix} y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} w_{48} & w_{58} & w_{68} & w_{78} \\ w_{49} & w_{59} & w_{69} & w_{79} \end{bmatrix} \cdot \begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} \quad (۴.۲)$$

و یا:

$$Y_k = W_{jk} \cdot V_j \quad (۵.۲)$$

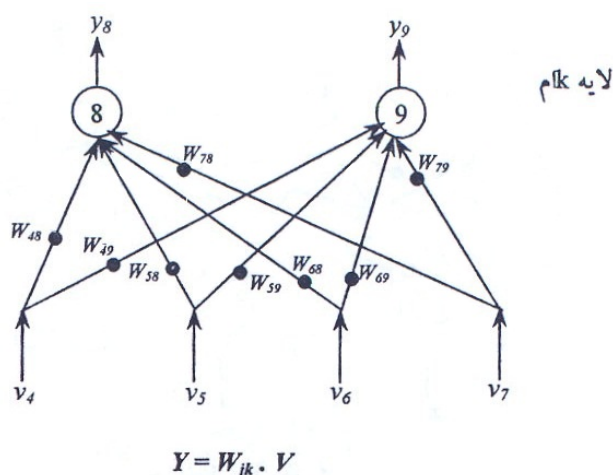
با ترکیب معادلات (۳.۲) و (۵.۲) مشخص می‌شود که بردار خروجی Y_k با ضرب نقطه‌ای دو ماتریس و بردار ورودی X_i به دست می‌آید.

$$\begin{bmatrix} y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} w_{48} & w_{58} & w_{68} & w_{78} \\ w_{49} & w_{59} & w_{69} & w_{79} \end{bmatrix} \cdot \begin{bmatrix} w_{14} & w_{24} & w_{34} \\ w_{15} & w_{25} & w_{35} \\ w_{16} & w_{26} & w_{36} \\ w_{17} & w_{27} & w_{37} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (6.2)$$

چون از ضرب دو ماتریس می‌توان یک ماتریس واحد بنام W_{ijk} به‌دست آورد پس نتیجه می‌گیریم که بردار خروجی Y با ضرب نقطه‌ای این ماتریس ترکیبی و بردار ورودی معادل است:

$$Y_k = W_{ij} \cdot W_{jk} \cdot X_i = W_{ijk} \cdot X_i \quad (7.2)$$

محدودیت خطی بودن توابع فعالسازی به این معناست که روابط موجود در معادلات (۲.۲) تا (۷.۲) به شدت محدود خواهند شد. به‌علاوه این نشان می‌دهد که یک پرسپترون سه لایه که توابع فعالسازی لایه میانی و خروجی آن خطی باشند را می‌توان با یک شبکه دو لایه جایگزین نمود که تابع فعالسازی خروجی آن خطی است. این فرآیند، مفهوم ماتریس‌های اوزان رامعرفی می‌کند که کاربردهای فراوانی دارد.



شکل ۸.۲: قسمت فوقانی شبکه عصبی که از بالای لایه پنهان برش داده شده است.

۶.۲ یادگیری و فراخوانی

شبکه‌های عصبی دو وظیفه عمده را انجام می‌دهند که عبارتند از یادگیری و فراخوانی^{۲۴}. یادگیری عبارت است از فرآیند تعدیل اوزان ارتباطی در یک شبکه عصبی مصنوعی است به‌طوری‌که شبکه بتواند به هنگام

^{۲۴} Learning and recall

دریافت بردار محرک توسط لایه ورودی، بردار خروجی دلخواه را به عنوان پاسخ تولید کند. فراخوانی، فرآیند پذیرش یک ورودی محرک و تولید یک پاسخ خروجی برحسب ساختار وزنی شبکه است. به طور کلی وقتی شبکه عصبی، محرک‌های وارد شده به لایه ورودی را پردازش می‌کند و یک پاسخ در لایه خروجی تولید می‌نماید، فراخوانی روی داده است. دو فرآیند یادگیری و فراخوانی مکمل یکدیگرند زیرا خروجی مطلوب یک شبکه باید با خروجی واقعی آن مقایسه شود تا یک تابع خطا به وجود آید.

قواعد یادگیری محاسبات عصبی نشان می‌دهد که چگونه اوزان ارتباطی در پاسخ به یک مثال یادگیری، تعدیل می‌شوند. در یادگیری تحت نظارت^{۲۵} شبکه عصبی مصنوعی آموزش داده می‌شود تا در پاسخ به یک محرک ورودی بخصوص، پاسخ دلخواه را ارائه نماید. در یادگیری درجه‌بندی شده^{۲۶} خروجی به صورت خوب یا بد در یک مقیاس عددی درجه‌بندی می‌شود و اوزان ارتباطی بر اساس این درجه تعدیل می‌شوند. یادگیری تحت نظارت، کاربرد وسیعی در مهندسی دارد. در این حالت محرکی وارد لایه ورودی (بردار ورودی) شده و محرکی نیز به لایه خروجی ارائه می‌شود که نشان‌دهنده پاسخ مطلوب به آن ورودی خاص است. تفاوت بین پاسخ واقعی شبکه و پاسخ مطلوب، خطایی ایجاد می‌کند که از این خطا برای تعدیل اوزان ارتباطی استفاده می‌شود. چندین الگوریتم یادگیری تحت نظارت وجود دارد که در شبکه‌های عصبی از آن‌ها استفاده می‌شود. قدیمی‌ترین آن‌ها یادگیری هب است که نام آن از دونالد هب^{۲۷} گرفته شده است. این فرد برای یادگیری بیولوژیکی مدلی ارائه داد که در آن اگر، هم بردار ورودی و هم بردار خروجی مطلوب، بزرگ باشند وزن ارتباطی افزایش می‌یابد. از قاعده دلتا^{۲۸} در یادگیری زمانی استفاده می‌شود که بخواهیم خطا (یعنی تفاوت بین پاسخ مطلوب و پاسخ واقعی شبکه) را به حداقل برسانیم که معمولاً این کار با استفاده از روش حداقل مربعات انجام می‌شود. در نوع دیگری از یادگیری بنام یادگیری تصادفی، تغییرات افزایشی و تصادفی در اوزان صورت می‌گیرد و بسته به اینکه خروجی بهبود می‌یابد یا نه، اوزان با همان مقادیر افزایش یافته، حفظ می‌شوند یا به مقادیر اولیه خود تنزل می‌یابند.

در فرآیند فراخوانی، یک شبکه عصبی، محرک وارد شده به لایه را می‌پذیرد و در لایه خروجی، پاسخی تولید می‌کند که این پاسخ توسط فرآیند آموزش شبکه تعیین شده است. ساده‌ترین شکل فراخوانی زمانی است که هیچگونه ارتباط بازخور از یک لایه به لایه دیگر و یا درون یک لایه وجود نداشته باشد.

^{۲۵}Supervised learning

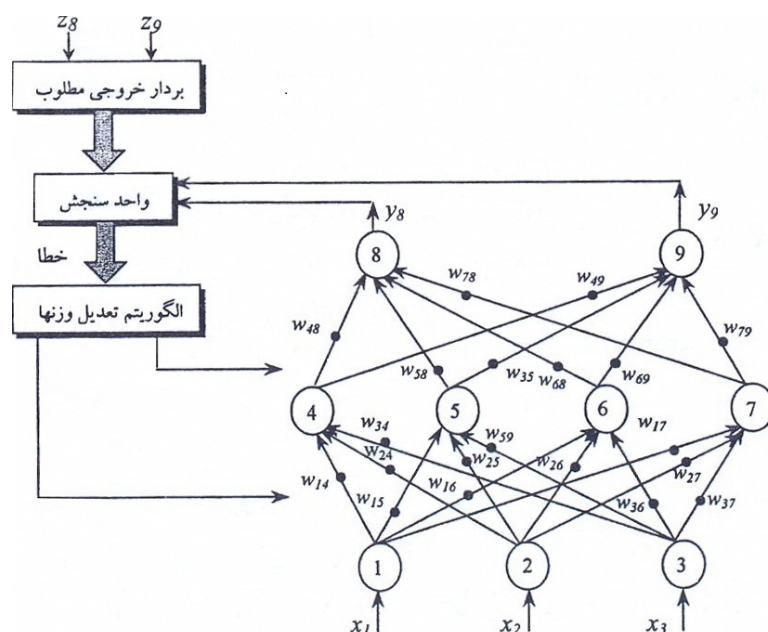
^{۲۶}Graded learning

^{۲۷}Donald Hebb

^{۲۸}Delta rule

۷.۲ یادگیری تحت نظارت

به منظور تشریح یادگیری تحت نظارت، یک الگوی خروجی مطلوب، یک واحد سنجش^{۲۹} و یک الگوریتم تعدیل در نظر می‌گیریم. این قسمت‌ها در شکل (۹.۲) نشان داده شده است که در آن خروجی مطلوب با بردار Z که دارای مولفه‌های z_8 و z_9 می‌باشد مشخص شده است. ورودی‌های واحد سنجش، خروجی مطلوب Z و خروجی واقعی Y هستند. خطایی که واحد سنجش تولید می‌کند تفاوت بین Z و Y است که از آن در الگوریتم تعدیل اوزان استفاده می‌شود تا میزان تعدیل در اوزان هر دو لایه مشخص شود.



شکل ۹.۲: یک شبکه عصبی با یادگیری تحت نظارت

برای شروع فرآیند، اجازه دهید همه وزن‌های موجود در شبکه عصبی شکل (۹.۲) را به صورت مقادیر کوچک تصادفی تعدیل نماییم و سپس جفت‌های آموزش X و Z را به ترتیب با اجزای x_1, x_2, x_3 و z_8, z_9 در نظر بگیریم. وقتی بردار X روی شبکه عصبی اعمال می‌شود بردار خروجی Y را تولید می‌کند که با بردار Z مقایسه می‌شود تا خطای بین آن‌ها بوجود آید. سپس الگوریتم تعدیل اوزان، وزن‌های ارتباطی را به گونه‌ای تغییر می‌دهد که این خطا کاهش یابد. وقتی بردار ورودی X مجدداً بر روی شبکه اعمال می‌شود یک Y جدید تولید می‌کند که با Z مقایسه می‌شود و الگوریتم تعدیل اوزان دوباره از خطای بین آن‌ها برای تعدیل وزن‌ها استفاده می‌کند. این فرآیند بارها و بارها تکرار می‌شود تا خطا به حد مشخصی تنزل یابد و یا مقداری

^{۲۹}Comparator

کوچک و غیرقابل تقلیل پیدا کند. در این نقاط، بردار خروجی Y و بردار خروجی مطلوب Z ، ذاتا با هم معادلند و شبکه عصبی، آموزش یافته تا بردار ورودی X را بر بردار خروجی مطلوب Z منطبق کند. این فرآیند، پایه و اساس روش یادگیری تحت نظارت است.

۸.۲ خصوصیات شبکه‌های عصبی مصنوعی

به‌طور کلی شبکه‌های عصبی دارای چهار خصوصیت عمده هستند:

- آن‌ها از طریق مثال یادگیری می‌کنند.
- آن‌ها یک حافظه توزیع شده و شرکت‌پذیر تشکیل می‌دهند.
- از معایب چشم‌پوشی می‌کنند.
- قادر به تشخیص الگو هستند.

یکی از مهمترین خصوصیات شبکه‌های عصبی مصنوعی این است که از مثال‌های گرفته شده از داده‌ها استفاده می‌کند و این اطلاعات را به شکلی سودمند سازماندهی می‌نماید. به عنوان مثال این شکل سودمند، مدلی را تشکیل می‌دهد که بیانگر رابطه بین متغیرهای ورودی و خروجی است.

حافظه یک شبکه عصبی هم توزیع شده و هم شرکت‌پذیر است. منظور از حافظه توزیع شده این است که اطلاعات در بین همه اوزانی که در طی فرآیند یادگیری تعدیل شده‌اند توزیع شده است. این اوزان ارتباطی، واحدهای حافظه شبکه عصبی را تشکیل می‌دهند و مقادیر این اوزان، وضعیت فعلی دانش شبکه را نشان می‌دهد. حافظه یک شبکه عصبی مصنوعی شرکت‌پذیر نیز هست. این بدان معناست که اگر یک ورودی ناقص به یک شبکه آموزش دیده، ارائه شود شبکه عصبی، نزدیکترین تطبیق به آن ورودی را در حافظه انتخاب کرده و خروجی تولید می‌کند که با یک ورودی کامل تطابق داشته باشد.

شبکه‌های عصبی از معایب ورودی خود چشم‌پوشی می‌کنند و علت این امر آن است که اطلاعات ذخیره شده، بین همه اوزان توزیع می‌شود. به‌طور کلی میزان انحراف خروجی شبکه از خروجی تعریف شده آن، معادل کسر اوزانی است که تخریب شده‌اند. از این گذشته، حتی وقتی که تعداد زیادی از اوزان تخریب شده‌اند عملکرد شبکه عصبی به تدریج ضعیف می‌شود و سیستم ناگهان از کار نمی‌افتد زیرا همه اطلاعات در یکجا مستقر نشده، بلکه در همه جای شبکه توزیع شده است. اگر از شبکه‌های عصبی مصنوعی در سخت‌افزارها استفاده شود از معایب بسیار زیادی چشم‌پوشی خواهند کرد. به همین دلیل شبکه‌های عصبی برای استفاده در محیط‌های دنیای واقعی، جاییکه به تشخیص الگو با چشم‌پوشی زیاد از معایب، نیاز است و ممکن است داده‌های ورودی، تحریف شده یا مغشوش باشند نوید بسیار بزرگی هستند. چند نمونه از این

محیط‌ها عبارتند از: نیروگاه‌های هسته‌ای، سیستم‌های راهنمای شلیک، سیستم اکتشاف فضا و یا هر سیستمی که دسترسی به آن برای تعمیر، غیرممکن بوده و یا جائیکه انجام عملیات پیوسته، بسیار حیاتی است. برای تشخیص الگو لازم است تا شبکه عصبی حجم زیادی از اطلاعات ورودی را به‌طور همزمان با هم تطبیق دهد و یک خروجی کلی یا دسته‌بندی شده تولید کند به‌طوری‌که این خروجی، پاسخ معقولی به داده‌های ناقص یا مغشوش ورودی باشد. شبکه‌های عصبی مصنوعی الگوها را بخوبی تشخیص می‌دهند، حتی اگر اطلاعات تشکیل دهنده الگو، مغشوش، پراکنده یا ناقص باشند. در یک سیستم پیچیده که سنسورهای زیادی دارد و انواع مختلفی از خرابی‌ها ممکن است برای آن رخ دهد، جهت رسیدن به پاسخ واقعی باید یک تعامل شدید بین کاربران و سیستم‌های خبره صورت گیرد. از آنجاکه آموزش یک شبکه عصبی ممکن است طولانی باشد، شبکه یک‌بار آموزش داده می‌شود تا شرایط و وضعیت‌های گوناگون این سیستم پیچیده مشخص شود، بنابراین برای تشخیص یک وضعیت خاص، فقط به یک مرحله آموزش شبکه عصبی نیاز داریم.

۹.۲ تاریخچه کاربرد شبکه‌های عصبی مصنوعی در حل مسائل بهینه‌سازی

مسائل بهینه‌سازی درطیف وسیعی از کاربردهای علوم و مهندسی تحلیل پردازش تصویر^{۳۰}، شناسایی سیستم‌ها^{۳۱}، حل معادلات، طراحی فیلتر^{۳۲}، تقریب توابع^{۳۳}، آنالیز رگرسیون^{۳۴} و... پدیدار می‌شود. در بسیاری از کاربردهای علوم و مهندسی جواب زمان واقعی^{۳۵} مسائل بهینه‌سازی بسیار مورد نیاز می‌شود. الگوریتم‌های قدیمی برای کامپیوترهای دیجیتال چندان موثر نیستند زیرا زمان مورد نیاز برای به‌دست آوردن جواب، وابستگی زیادی به بُعد^{۳۶} و ساختار^{۳۷} مسأله دارد، این یک رهیافت ممکن و بسیار امیدوارکننده کاربرد شبکه‌های عصبی مصنوعی است.

به واسطه ذات شبکه‌های عصبی که می‌توانند حجم عظیمی از کارهای موازی را با هم انجام دهند رهیافت شبکه عصبی می‌تواند مسائل بهینه‌سازی در زمان واقعی را با سرعت بیشتری نسبت به سایر الگوریتم‌های بهینه‌سازی موجود حل کند. هاپفیلد^{۳۸} و تنک^{۳۹} [۱۵] ابتدا شبکه عصبی را برای حل مسائل برنامه‌ریزی خطی پیشنهاد دادند. کار بنیادین آنها الهام بخش تحقیقات زیادی برای ارائه شبکه‌های عصبی دیگر برای حل

^{۳۰} Signal processing

^{۳۱} System identification

^{۳۲} Filter design

^{۳۳} Function approximation

^{۳۴} Regression analysis

^{۳۵} Real time

^{۳۶} Dimension

^{۳۷} Structure

^{۳۸} Hopfield

^{۳۹} Tank

مسائل برنامه‌ریزی خطی و غیرخطی شد. بدین ترتیب شبکه‌های عصبی بهینه‌سازی زیادی گسترش یافتند. به‌عنوان مثال کندی^{۴۰} و چاو^{۴۱} [۲۱] یک شبکه عصبی برای حل مسائل برنامه‌ریزی غیرخطی پیشنهاد کردند که از روش گرادیان و روش تابع جریمه استفاده می‌نمودند و نقاط تعادلشان متناظر با تقریبی از جواب‌های بهینه بود. با استفاده از روش گرادیان و تصویر، بازردهوم^{۴۲} و پاتیسن^{۴۳} [۶] یک شبکه عصبی برای حل مسائل بهینه‌سازی درجه دوم تنها با متغیرهای کراندار معرفی کردند. این شبکه عصبی تحت شرایط خاصی به تنها جواب بهینه همگرا بود.

بر پایه روش‌های گرادیان و تصویر، ژیا^{۴۴} و همکارانش [۴۰، ۴۹] چندین شبکه عصبی برای حل مسائل برنامه‌ریزی درجه دوم و خطی با جواب‌های بهینه غیریکتا معرفی کردند و همگرایی سراسری آن‌ها را به جواب‌های دقیق ثابت نمودند.

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

۱. همگرایی شبکه عصبی به ازای هر وضعیت دلخواه داده شده تضمین شود.

۲. در طراحی شبکه ترجیحاً هیچ پارامتر مجهولی وجود نداشته باشد.

۳. نقاط تعادل شبکه، متناظر با جواب‌های دقیق یا تقریبی باشد.

از دیدگاه ریاضی این خصوصیات مربوط به تکنیک‌های بهینه‌سازی هستند که برای طراحی مدل‌های شبکه‌های عصبی بهینه‌سازی به کار می‌روند.

در مجموع برای فرمول‌بندی یک مسأله بهینه‌سازی برحسب یک شبکه عصبی دو نوع روش وجود دارد. یک رهیافت که عموماً در ایجاد یک شبکه عصبی بهینه‌سازی استفاده می‌شود این است که در ابتدا مسأله بهینه‌سازی را به یک مسأله بهینه‌سازی نامقید تبدیل کرده و سپس یک شبکه عصبی طراحی می‌کنند که مسأله نامقید را با استفاده از روش گرادیان حل کند. روش گرادیان یک مزیت برای آن دسته از شبکه‌های عصبی است که مستقیماً از مشتق تابع انرژی استفاده می‌کنند اما عیب آن‌ها این است که همگرایی در این موارد تضمین نمی‌شود به‌ویژه درحالتی که مجموعه‌های جواب کراندار نیستند.

رهیافت دیگر این است که یک مجموعه از معادلات دیفرانسیل ایجاد می‌کنند به طوری که نقاط تعادل آن‌ها متناظر با جواب مطلوب باشد و بعد یک تابع لیاپانوف مناسب پیدا کرده، به طوری که همه مسیرهای سیستم به نقاط تعادل همگرا باشند. به طور کلی شبکه‌های عصبی نقاط قوت و ضعف بسیاری دارند و اگر بخواهیم

^{۴۰} Kennedy

^{۴۱} Chua

^{۴۲} Bouzerdoum

^{۴۳} Pattison

^{۴۴} Xia

از آن‌ها به‌طور صحیح استفاده کنیم باید این نقاط را مدنظر قرار دهیم. شبکه‌های عصبی اطلاعات را به روش مبتنی بر حافظه^{۴۵} ذخیره می‌کنند که با ذخیره اطلاعات در یک جدول جستجو، متفاوت بوده و از آن انعطاف‌پذیرتر است. در یک شبکه عصبی درست مانند مغز، اطلاعات در همه جای شبکه توزیع می‌شوند. بنابراین در این روش، اطلاعات در قسمت‌های مختلف شبکه و به‌طور جداگانه ذخیره می‌شوند، اگرچه این کار دشواری است ولی این قابلیت بسیار مهم را در شبکه‌های عصبی به‌وجود می‌آورد که بتوانند نتایج را تعمیم دهند.

۱۰.۲ مدل‌های شبکه عصبی ارائه شده پیشین برای حل مسائل بهینه‌سازی

بحث و بررسی در مورد کاربرد شبکه‌های عصبی (شبکه‌های عصبی مصنوعی) در بهینه‌سازی از اوایل سال ۱۹۸۰ میلادی آغاز شده است. نتایج پژوهش‌های انجام شده از آن زمان تاکنون، مواردی چون برنامه‌ریزی خطی، برنامه‌ریزی درجه دوم، برنامه‌ریزی هندسی و برنامه‌ریزی غیرخطی را در برمی‌گیرد. 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 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 به‌دست 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 به‌صورت زیر بیان می‌شود:

برای یک نقطه آغازین دلخواه، نقطه تعادل دستگاه معادلات دیفرانسیل غیرخطی به‌دست آمده جواب بهینه مسأله اصلی است و برعکس.

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

^{۴۵}Memory-based storage

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.

مدل اول

مسأله برنامه‌ریزی خطی زیر را در نظر بگیرید:

$$\text{minimize } f(x) = a^T x \quad (۸.۲)$$

$$\text{subject to } \begin{cases} g(x) = Dx - b = 0, \\ x \geq 0. \end{cases} \quad (۹.۲)$$

که در آن D یک ماتریس $m \times n$ و $\text{Rank}(D) = m$ و $a \in \mathbb{R}^m$ ، $x, b \in \mathbb{R}^n$ است. دوگان (۹.۲)-(۸.۲) به صورت زیر بیان می‌شود:

$$\text{maximize } \bar{f}(y) = b^T y \quad (۱۰.۲)$$

$$\text{subject to } \begin{cases} \bar{g}(y) = \bar{D}^T y - a \leq 0, \\ y \in \mathbb{R}^n. \end{cases} \quad (۱۱.۲)$$

در [۴۰] مدل متناظر با (۱۱.۲)-(۸.۲) به صورت زیر نمایش داده شده است:

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

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

مدل دوم

مسأله برنامه‌ریزی درجه دوم زیر را در نظر بگیرید:

$$\text{minimize } f(x) = \frac{1}{2} x^T A x + a^T x \quad (۱۲.۲)$$

$$\text{subject to } \begin{cases} Dx - b = 0, \\ x \geq 0, \end{cases} \quad (۱۳.۲)$$

که در آن A یک ماتریس متقارن معین مثبت است. دوگان (۱۳.۲)-(۱۲.۲) به صورت زیر است:

$$\text{maximize } \bar{f}(y) = b^T y - \frac{1}{2} y^T A y \quad (۱۴.۲)$$

$$\text{subject to } \bar{D}y - Ax - a \leq 0. \quad (15.2)$$

در [۴۰] مدل متناظر با (۱۵.۲)-(۱۲.۲) به صورت زیر است:

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

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

مدل سوم

مسئله برنامه‌ریزی غیرخطی زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (16.2)$$

$$\text{subject to } g(x) = [g_1(x), g_2(x), \dots, g_m(x)]^T \leq 0, \quad (17.2)$$

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 (۱۷.۲)-(۱۶.۲) هاپفیلد^{۴۶} و تنک^{۴۷} [۱۵] یک مدل شبکه عصبی به صورت زیر ارائه دادند:

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

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

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

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

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

مدل چهارم

برای حل (۱۷.۲)-(۱۶.۲)، کندی و چوآ [۲۱] مدل دیگری به صورت زیر ارائه داده‌اند:

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

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

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

^{۴۶}Hopfield

^{۴۷}Tank

مدل پنجم

مجدداً برای حل (۱۷.۲)–(۱۶.۲) توسط رودریگز-واسکوئز مدلی [۳۶] به صورت زیر ارائه شده است:

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

که در آن u_x متناظر اندیس‌های شدنی x است چنان‌که

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

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

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

همان‌طور که ملاحظه می‌کنید، مدل‌های بیان شده به‌طور متوالی اصلاح شده‌اند و این به دلیل مشکلاتی بوده است که هرکدام از این مدل‌ها داشته‌اند.

مدل ششم

برنامه‌ریزی محدب درجه دوم^{۴۸} زیر را در نظر بگیرید:

$$\text{minimize } f(x) = \frac{1}{\nu} x^T Q x + D^T x \quad (18.2)$$

$$\text{subject to } \begin{cases} g(x) = Ax - b \leq 0, \\ h(x) = Ex - f = 0, \end{cases} \quad (19.2)$$

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. دوگان برنامه‌ریزی محدب درجه دوم (۱۹.۲)–(۱۸.۲) به صورت زیر نوشته می‌شود:

^{۴۸}Convex quadratic programming

$$\text{maximize } \bar{f}(x, u, v) = f(x) + u^T(Ax - b) + v^T(Ex - f) \quad (20.2)$$

$$\text{subject to } \begin{cases} \bar{g}(x, u, v) = Qx + D + A^T u + E^T v = 0, \\ u \geq 0. \end{cases} \quad (21.2)$$

بر طبق شرایط کاروش-کان-تاگر برای (21.2)-(20.2)، $x^* \in \mathbb{R}^n$ ، جواب بهینه (19.2)-(18.2) است اگر و تنها اگر بردارهای $u^* \in \mathbb{R}^m$ و $v^* \in \mathbb{R}^l$ وجود داشته باشند به طوری که در رابطه زیر صدق کند:

$$\begin{cases} u^* \geq 0, Ax^* - b \leq 0, u^{*T}(Ax^* - b) = 0, \\ Qx^* + D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \quad (22.2)$$

برای حل مسأله بالا تابع زیر را که معروف به تابع فیشر-برمیستر^{۴۹} است، معرفی می‌کنیم:

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

قضیه ۱.۱۰.۲. ([۳۵]).

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

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

۳. $\phi(a, b)$ همه جا به جز در مبدأ دو بار به طور پیوسته مشتق پذیر است، ولی در مبدأ قویاً هموار^{۵۰} است.

تابع ϕ که در شرط (۱) قضیه (۱.۱۰.۲) صدق کند، تابع NCP^{۵۱} نامیده می‌شود.

متناظر با دستگاه (21.2)-(18.2) تابع زیر را در نظر می‌گیریم:

$$\phi(x, u, v) = \begin{bmatrix} \phi(\alpha, Qx + D + A^T u + E^T v) \\ \phi(u, Ax - b) \\ \phi(\beta, Ex - f) \end{bmatrix}, \quad (23.2)$$

که در آن $\alpha = (\alpha_1, \dots, \alpha_m)^T > 0$ ، $u = (u_1, \dots, u_m)$ و $\beta = (\beta_1, \dots, \beta_l)^T > 0$ است. مدل شبکه عصبی متناظر با (23.2) در [۱۰] به صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = -\tau \nabla E(y(t)), \quad \tau > 0, \\ y(t_0) = y_0 \in \mathbb{R}^{n+m+l}. \end{cases} \quad (24.2)$$

^{۴۹}Fisher-Bermister

^{۵۰}Strongly smooth

^{۵۱}Nonlinear Complementarity Problem

که در آن $y = (x^T, u^T, v^T)^T$ و τ ضریبی برای افزایش سرعت همگرایی و کاهش تعداد تکرارها در روش عددی انتخاب شده برای حل دستگاه (۲۳.۲) 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 (۲۱.۲)-(۱۸.۲) به صورت زیر در نظر می‌گیریم:

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

آنگاه برای حل (۱۹.۲)-(۱۸.۲) دستگاه زیر تعریف می‌شود:

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

مدل شبکه عصبی برای حل (۱۹.۲)-(۱۸.۲) در [۱۰] به صورت زیر بیان شده است:

$$\begin{cases} \frac{dy}{dt} = \kappa U(y), \\ y(t_0) = y_0. \end{cases} \quad (26.2)$$

مدل هشتم

مسئله برنامه ریزی درجه دوم (۱۹.۲)-(۱۸.۲) را در نظر بگیرید، که در آن $Q \in \mathbb{R}^{n \times n}$ یک ماتریس نیمه معین مثبت است. برای حل مسئله (۱۹.۲)-(۱۸.۲) یک مدل شبکه عصبی در [۳] به صورت زیر ارائه می‌شود:

$$\frac{dy}{dt} = \kappa(I + M^T) \begin{bmatrix} -(Qx + D + A^T u + E^T v) \\ (u + Ax - b)^+ - u \\ (Ex - f) \end{bmatrix}, \quad (27.2)$$

که در آن κ نرخ همگرایی است و

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

مدل نهم

مسئله برنامه‌ریزی درجه دوم زیر را در نظر بگیرید:

$$\text{minimize } f(x) = \frac{1}{2}x^T Ax + a^T x \quad (28.2)$$

$$\text{subject to } \begin{cases} Dx \leq b, \\ x \geq 0. \end{cases} \quad (29.2)$$

دوگان آن عبارت است از:

$$\text{maximize } \bar{f}(w) = b^T w - \frac{1}{2}x^T Ax \quad (30.2)$$

$$\text{subject to } \begin{cases} \bar{D}^T w - Ax \leq a, \\ w \geq 0, \end{cases} \quad (31.2)$$

که در آن $A \in \mathbb{R}^{n \times n}$ متقارن و نیمه معین مثبت است، $a \in \mathbb{R}^n$ ، $b \in \mathbb{R}^m$ و $D \in \mathbb{R}^{m \times n}$ است. با توجه به شرایط کاروش-کان-تاکر برای مسائل برنامه‌ریزی محدب، $(x^*, w^*)^T$ به ترتیب یک جواب بهینه برای مسائل (28.2)-(31.2) است، اگر و فقط اگر $(x^*, w^*)^T$ در شرایط کاروش-کان-تاکر زیر صدق کند:

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

مدل متناظر با (28.2)-(31.2) در [۹] به صورت زیر بیان شده است:

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

که در آن

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

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

فصل ۳

ساختن یک مدل دینامیکی کارا برای حل
مسائل برنامه‌ریزی غیرخطی محدب

۱.۳ مقدمه

در این فصل مدلی از یک شبکه عصبی بازگشتی را برای حل مسأله برنامه‌ریزی غیرخطی محدب^۱ ارائه خواهیم کرد. استفاده از این مدل مستلزم این است که تابع هدف مسأله برنامه‌ریزی غیرخطی محدب، تابعی محدب باشد و سایر توابعی که تشکیل قیود مسأله را می‌دهند توابعی اکیدا محدب باشند و یا اینکه این قیود، محدب بوده و تابع هدف مسأله برنامه‌ریزی غیرخطی، تابعی اکیدا محدب باشد. همچنین اثبات خواهیم کرد که مدل شبکه عصبی ارائه شده، پایدار به مفهوم لیاپانوف و همگرایی سراسری به جواب بهینه مسأله اصلی است و با به‌کار بردن مثال‌های متنوع، همگرایی مدل ارائه شده را نشان خواهیم داد.

برنامه‌ریزی محدب در مسائل مختلف ریاضیات، مهندسی، اقتصاد و غیره ظاهر می‌شود [۴]. مقالات بسیاری وجود دارند که در آن‌ها برای حل این مسائل از الگوریتم‌های عددی استفاده شده است [۱، ۲، ۴، ۱۴] از آنجائیکه پیچیدگی زمانی به بعد و ساختار مسأله وابسته است، الگوریتم‌های عددی کلاسیک برای حل مسائل با ابعاد بزرگ و مسائل بهینه‌سازی وابسته به زمان^۲، عملاً مفید نخواهند بود. از سوی دیگر شبکه‌های عصبی مصنوعی^۳ به دلیل انجام محاسبات به‌صورت موازی، سرعت همگرایی بالایی دارند، لذا استفاده از شبکه‌های عصبی مصنوعی برای حل مسائل بهینه‌سازی وابسته به زمان و مسائل با ابعاد بزرگ، مفید خواهد بود.

ماهیت شبکه‌های عصبی برای حل مسائل بهینه‌سازی، بر پایه ساختن یک تابع انرژی (تابع نامنفی) و یک مدل دینامیکی استوار است [۲۷]. مدل دینامیکی غالباً به‌صورت یک دستگاه معادلات دیفرانسیل معمولی است که انتظار داریم با هر نقطه شروع دلخواه، مدل دینامیکی ارائه شده، همگرا به نقطه تعادلی باشد که این نقطه تعادل همان جواب بهینه مسأله بهینه‌سازی اصلی است. همچنین برای بررسی همگرایی سیستم دینامیکی به نقطه تعادل، تابع انرژی باید تابعی اکیدا نزولی باشد. امروزه، شبکه‌های عصبی به‌طور گسترده برای حل مسائل مختلف بهینه‌سازی مورد استفاده قرار گرفته و مطالعات و تحقیقات بسیاری در این زمینه در حال انجام است.

بنابر آنچه در بالا گفته شد، در این فصل، براساس شرایط بهینگی کاروش-کان-تاکر^۴ برای مسائل برنامه‌ریزی محدب، مدلی از شبکه عصبی به منظور حل مسأله (۲.۳)-(۱.۳) مطرح خواهد شد. همچنین، پایدار مجانبی بودن شبکه عصبی مطرح شده را به‌وسیله‌ی ساختن یک تابع لیاپانوف نشان داده و اثبات خواهیم کرد که نقطه تعادل مدل دینامیکی همان جواب بهینه مسأله اصلی است.

در بخش دوم، مدل شبکه عصبی را خواهیم ساخت. در بخش سوم، پایداری نقطه تعادل و همگرایی به

^۱Convex nonlinear programming (CNLP)

^۲Real-time optimization problems

^۳Artificial neural networks (ANN)

^۴Karush Kuhn Tucker

جواب بهینه را مورد بررسی قرار خواهیم داد و در ادامه مسائل مختلفی مطرح شده و کارائی شبکه عصبی ارائه شده، نشان داده خواهد شد.

۲.۳ ساختن مدل دینامیکی

مسئله برنامه‌ریزی غیرخطی محدب زیر را در نظر بگیرید:

$$\text{minimize } f(x) \quad (۱.۳)$$

$$\text{subject to } \begin{cases} g(x) \leq 0, \\ h(x) = 0, \end{cases} \quad (۲.۳)$$

که در آن $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ، $x \in \mathbb{R}^n$ و $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$ تابعی برداری، پیوسته $-m$ بعدی و n متغیره است. توابع f, g_1, \dots, g_m محدب و دارای مشتق مرتبه دوم و $h(x) = Ax - b$ تابعی آفینی است به طوری که $A \in \mathbb{R}^{l \times n}$ ، $\text{rank}(A) = l$ ($0 \leq l < n$) و $b \in \mathbb{R}^l$ است. همچنین فرض می‌کنیم مسئله (۲.۳)–(۱.۳) دارای جواب بهینه یکتاست.

مطابق [۱۶] تابع لاگرانژ مسئله بالا را به صورت زیر تعریف می‌کنیم:

$$L(x, u, v) = f(x) + \frac{1}{\nu} \sum_{k=1}^m u_k g_k(x) + \sum_{p=1}^l v_p h_p(x), \quad (۳.۳)$$

که $u \in \mathbb{R}^m$ و $v \in \mathbb{R}^l$ ضرایب لاگرانژ نامیده می‌شوند. همانطور که در [۴] آمده $x^* \in \mathbb{R}^n$ جواب بهینه (۲.۳)–(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 کنند:

$$\begin{cases} u^* \geq 0, & g(x^*) \leq 0, & u^{*T} g(x^*) = 0, \\ \nabla f(x^*) + \nabla g(x^*)^T u^* + \nabla h(x^*)^T v^* = 0, \\ h(x^*) = 0. \end{cases} \quad (۴.۳)$$

x^* یک نقطه KKT مسئله (۲.۳)–(۱.۳) و $(u^{*T}, v^{*T})^T$ بردار ضرایب لاگرانژ وابسته به نقطه x^* نامیده می‌شود.

لم ۱.۲.۳. ([۴]). اگر f و g_k برای $k = 1, \dots, m$ توابعی محدب باشند، آنگاه x^* جواب بهینه مسئله (۲.۳)–(۱.۳) است اگر و تنها اگر x^* یک نقطه KKT مسئله (۲.۳)–(۱.۳) باشد.

اکنون فرض کنید $x(\cdot)$ ، $u(\cdot)$ و $v(\cdot)$ متغیرهایی وابسته به زمان باشند. هدف، ساختن یک مدل دینامیکی است که نسبت به زمان پیوسته بوده و همچنین به نقطه KKT مسأله (۲.۳)–(۱.۳) همگرا باشد. در حقیقت هدف ما ساختن یک شبکه عصبی است که به نقطه بحرانی تابع لاگرانژ (۳.۳) همگرا باشد. شبکه عصبی مسأله (۲.۳)–(۱.۳) را به صورت مدل دینامیکی زیر نمایش می‌دهیم:

$$\frac{dx}{dt} = -\nabla_x L(x, u, v) = -\left(\frac{\partial f}{\partial x_i} + \frac{1}{\nu} \sum_{k=1}^m u_k \frac{\partial g_k}{\partial x_i} + \sum_{p=1}^l v_p \frac{\partial h_p}{\partial x_i} \right), \quad i = 1, \dots, n, \quad (5.3)$$

$$\frac{du}{dt} = \nabla_u L(x, u, v) = \text{diag}(u_1, \dots, u_m)g(x), \quad (6.3)$$

$$\frac{dv}{dt} = \nabla_v L(x, u, v) = h_p(x), \quad p = 1, \dots, l. \quad (7.3)$$

نقطه $(x_0^T, u_0^T, v_0^T)^T$ که $u_k(t_0) \neq 0$ نقطه شروع مدل دینامیکی بالا در نظر می‌گیریم. به منظور سادگی در محاسبات، قرار می‌دهیم $y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ و D^* را مجموعه نقاط بهینه مسأله (۲.۳)–(۱.۳) و دوگان آن در نظر گرفته و قرار می‌دهیم:

$$\Phi(x, u, v) = \begin{bmatrix} -\left(\nabla f(x) + \frac{1}{\nu} \nabla g(x)^T u + \nabla h(x)^T v \right) \\ \text{diag}(u_1, \dots, u_m)g(x) \\ h(x) \end{bmatrix}. \quad (8.3)$$

بنابراین مدل شبکه عصبی (۷.۳)–(۵.۳) را می‌توان به صورت زیر نوشت:

$$\frac{dy}{dt} = \kappa \Phi(y), \quad (9.3)$$

$$y(t_0) = y_0, \quad u(t_0) \neq 0. \quad (10.3)$$

که در آن κ عدد ثابتی است که پارامتر جریمه نامیده می‌شود. برای سادگی در انجام محاسبات $\kappa = 1$ در نظر گرفته می‌شود.

۳.۳ بررسی پایداری و همگرایی

در این قسمت، همگرایی و پایداری شبکه عصبی (۱۰.۳)–(۹.۳) را بررسی می‌کنیم.

قضیه ۱۰.۳.۳. اگر $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل شبکه عصبی (۱۰.۳)–(۹.۳) باشد، آنگاه x^* یک نقطه KKT مسأله (۲.۳)–(۱.۳) است.

برهان. با مقایسه شرایط کاروش-کان-تاگر (۴.۳) با شبکه عصبی ارائه شده در (۱۰.۳)-(۹.۳) فقط کافی است نشان دهیم، $g_k(x^*) \leq 0$ و $u_k g_k(x^*) = 0$ برای $k = 1, \dots, m$. به سادگی می‌توان نشان داد که ضرایب u_k در (۶.۳) به صورت زیر است:

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

که در آن $u_k(t_0) \neq 0$ است. فرض کنید همه $g_k(x^*) \leq 0$ نباشند (یعنی حداقل یک k موجود باشد که $g_k(x) > 0$)، بنابراین ضریب $u_k(t)$ وابسته به آن به صورت نمایی افزایش می‌یابد، لذا $\frac{1}{\gamma} u_k^\gamma g_k(x)$ در تابع لاگرانژ (۳.۳) با زمان افزایش می‌یابد. این بدان معنی است که جریمه افزوده شده با گذشت زمان به طور بی‌وقفه افزایش می‌یابد تا اینکه مسیر $x(t)$ در ناحیه شدنی قرار بگیرد. بنابراین، بعد از گذشت زمانی متناهی مسیر $x(t)$ به سمت ناحیه شدنی حرکت می‌کند. این عمل به طور متناوب تکرار می‌شود که این با نقطه تعادل بودن x^* در تناقض است. پس برای هر k باید داشته باشیم $g_k(x^*) \leq 0$. از آنجائیکه $(x^T, u^T, v^T)^T$ در نهایت به نقطه $(x^{*T}, u^{*T}, v^{*T})^T$ همگرا می‌شود لذا ضریب $u_k(t)$ وابسته به محدودیت‌های غیرفعال باید به صفر همگرا شده و بقیه ضرایب مربوط به متغیرهای فعال به عددی ثابت همگرا خواهند شد. \square

لم ۲.۳.۳. نقطه تعادل شبکه عصبی (۱۰.۳)-(۹.۳) منحصربفرد است.

برهان. چون مسأله مینیم‌سازی (۲.۳)-(۱.۳) دارای جواب بهینه منحصربفرد x^* است، پس مطابق [۴] لازم و کافی است که شرایط کاروش-کان-تاگر (۴.۳) دارای جواب منحصربفرد $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ باشد. همچنین، در قضیه (۱.۳.۳) نشان دادیم که نقطه تعادل مدل شبکه عصبی ارائه شده در (۱۰.۳)-(۹.۳) در شرایط کاروش-کان-تاگر صدق می‌کند، لذا نقطه تعادل شبکه عصبی ارائه شده منحصربفرد است. \square

لم ۳.۳.۳. در مدل شبکه عصبی (۱۰.۳)-(۹.۳)، برای هر نقطه شروع $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ جواب منحصربفرد و پیوسته $y(t) = (x(t)^T, u(t)^T, v(t)^T)^T$ وجود دارد.

برهان. چون $\nabla f(x)$ ، $\nabla g_k(x)$ برای $k = 1, \dots, m$ و $\nabla h_p(x)$ برای $p = 1, \dots, l$ همگی روی مجموعه محدب و باز $D \subseteq \mathbb{R}^{n+m+l}$ به طور پیوسته مشتق پذیرند، پس $\nabla f(x) + \frac{1}{\gamma} \nabla g(x)^T u^\gamma + \sum_{p=1}^l v_p \nabla h_k(x)$ برای $k = 1, \dots, m$ و $h(x)$ پیوسته لیپ‌شیتس محلی‌اند. لذا مطابق قضیه وجود و یکتایی جواب معادلات دیفرانسیل معمولی در [۳۱] مدل شبکه عصبی (۱۰.۳)-(۹.۳) دارای جواب منحصربفرد و پیوسته $y(t)$ است و $t \in [t_0, \tau)$ که $\tau > t_0$ و می‌تواند تا $+\infty$ افزایش یابد. \square

در ادامه به اثبات قضیه‌ای مهم خواهیم پرداخت.

قضیه ۴.۳.۳. اگر $\nabla^2 f(x)$ معین مثبت و $\nabla^2 g_k(x)$ برای $k = 1, \dots, m$ ، نیمه معین مثبت و یا $\nabla^2 f(x)$ نیمه معین مثبت و $\nabla^2 g_k(x)$ برای $k = 1, \dots, m$ معین مثبت باشند، آنگاه مدل شبکه عصبی ارائه شده در (۱۰.۳)–(۹.۳) پایدار سراسری به مفهوم لیاپانوف و همگرای سراسری به نقطه $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ خواهد بود که در آن x^* جواب بهینه مسأله (۲.۳)–(۱.۳) و $(u^{*T}, v^{*T})^T$ جواب بهینه مسأله دوگان متناظر آن است.

برهان. تابع لیاپانوف زیر را در نظر می‌گیریم:

$$E(y) = \|\Phi(y)\|^2 + \frac{1}{\nu} \|y - y^*\|^2. \quad (11.3)$$

واضح است که $E(y) \geq 0$ و $E(y^*) = 0$ است. همچنین

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

که در آن ماتریس ژاکوبین نگاشت Φ به صورت زیر است:

$$\nabla \Phi(y) = \begin{bmatrix} -\nabla^2 f(x) - \frac{1}{\nu} \sum_{k=1}^m u_k \nabla^2 g_k(x) & -\nabla g(x)^T S & -\nabla h(x)^T \\ S^T \nabla g(x) & K & O_{m \times l} \\ \nabla h(x) & O_{l \times m} & O_{l \times l} \end{bmatrix}, \quad (12.3)$$

که در آن O ماتریس صفر، $S = \text{diag}(u_1, \dots, u_m)$ و $K = \text{diag}(g_1(x), \dots, g_m(x))$ است. با محاسبه مشتق $E(y(t))$ و در نظر گرفتن مدل (۱۰.۳)–(۹.۳) و استفاده از ماتریس ژاکوبین (۱۲.۳) خواهیم داشت:

$$\begin{aligned} \frac{dE(y(t))}{dt} &= \left(\frac{d\Phi}{dt} \right)^T \Phi + \Phi^T \left(\frac{d\Phi}{dt} \right) + (y - y^*)^T \frac{dy(t)}{dt} \\ &= \Phi^T (\nabla \Phi(y)^T + \nabla \Phi(y)) \Phi + (y - y^*)^T \Phi(y) \\ &= -\nu \left(\nabla f(x) + \frac{1}{\nu} \nabla g(x)^T u + \nabla h(x)^T v \right)^T \left(\nabla^2 f(x) + \frac{1}{\nu} \sum_{k=1}^m u_k \nabla^2 g_k(x) \right) \\ &\quad + \nu \sum_{k=1}^m (u_k g_k)^T g_k + (y - y^*)^T \Phi(y) \\ &\leq \nu \sum_{k=1}^m (u_k g_k)^T g_k + (y - y^*)^T \Phi(y). \end{aligned}$$

از فرض مسأله می‌دانیم که $\left(\nabla^2 f(x) + \frac{1}{\nu} \sum_{k=1}^m u_k \nabla^2 g_k(x) \right)$ معین منفی است. حال اگر برای هر k داشته باشیم $g_k(x) \leq 0$ ، آنگاه $\sum_{k=1}^m (u_k g_k)^T g_k \leq 0$. همچنین از آنجائیکه ماتریس

$\nabla\Phi(y)$ معین منفی است لذا طبق قضیه (۷.۴.۱) $-\Phi(y)$ یکنوای اکید است. از طرفی چون y^* نقطه تعادل است داریم $\Phi(y^*) = 0$ بنابراین

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

همچنین به دلیل آنکه $g_k(x) \leq 0$ برای هر $k = 1, \dots, m$ ، بنابراین $\sum_{k=1}^m (u_k g_k)^2 g_k \leq 0$ لذا:

$$\frac{dE(y(t))}{dt} \leq 0. \quad (۱۳.۳)$$

اکنون فرض کنیم حداقل یک $k \in \{1, \dots, m\}$ وجود داشته باشد که $g_k(x(t)) > 0$ باشد. در این حالت مطابق قضیه (۱.۳.۳)، u_k همگرا به صفر خواهد شد و لذا با انجام محاسبات مشابه آنچه در بالا گفته شد خواهیم داشت $\frac{dE(y(t))}{dt} \leq 0$. بنابراین، مدل شبکه عصبی ارائه شده پایدار سراسری به مفهوم لیپانوف است. از طرفی چون

$$E(y) \geq \frac{1}{4} \|y - y^*\|^2,$$

زیردنباله‌ای همگرا به صورت زیر وجود دارد

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

به طوری که وقتی $k \rightarrow \infty$ داریم $t_k \rightarrow \infty$ و $(x(t_k)^T, u(t_k)^T, v(t_k)^T)^T = (\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ به طوری که $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ در $\frac{dE(y(t))}{dt} = 0$ صدق می‌کند. واضح است که $(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ حدمثبت M است. بنابر نتیجه اصل تغییرناپذیری لازال (۸.۶.۱) مجموعه M وجود دارد که وقتی $t \rightarrow \infty$ داریم $\{(x(t)^T, u(t)^T, v(t)^T)^T \rightarrow M$ و بزرگترین مجموعه تغییرناپذیر مثبت در $\{(x(t)^T, u(t)^T, v(t)^T)^T \mid \frac{dE(y(t))}{dt} = 0\}$ است. اکنون با توجه (۱۰.۳)–(۹.۳) و نامعادله (۱۴.۳) واضح است که:

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

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

با قراردادن $x^* = \bar{x}$ ، $u^* = \bar{u}$ و $v^* = \bar{v}$ در (۱۱.۳) تابع لیپانوف دیگری به صورت زیر در نظر

می‌گیریم:

$$\bar{E}(y) = \|\Phi(y)\|^2 + \frac{1}{4} \|y - \bar{y}\|^2. \quad (۱۴.۳)$$

چون تابع $\bar{E}(y)$ مشتق‌پذیر پیوسته بوده و $\bar{E}(\bar{y}) = 0$ و همچنین

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

بنابراین $\lim_{k \rightarrow \infty} \bar{E}(x(t_k)^T, u(t_k)^T, v(t_k)^T)^T = \bar{E}(\bar{x}^T, \bar{u}^T, \bar{v}^T)^T = 0$ است. لذا برای هر $\varepsilon > 0$ وجود دارد $q > 0$ به طوری که برای هر $t \geq t_q$ داریم $\bar{E}(y(t)) < \varepsilon$. به همین ترتیب می‌توان نشان داد $\frac{d\bar{E}(y(t))}{dt} \leq 0$. بنابراین برای هر $t \geq t_q$ خواهیم داشت:

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

در نتیجه $\lim_{t \rightarrow \infty} \|y(t) - \bar{y}\| = 0$ پس $\lim_{t \rightarrow \infty} y(t) = \bar{y}$. بنابراین شبکه عصبی (۱۰.۳)-(۹.۳) همگرایی سراسری به نقطه تعادل $\bar{y} = (\bar{x}^T, \bar{u}^T, \bar{v}^T)^T$ می‌باشد که در آن \bar{x} جواب بهینه مسأله (۲.۳)-(۱.۳) است. \square

نتیجه ۵.۳.۳. فرض کنید شرایط قضیه (۴.۳.۳) برقرار باشد. اگر $D^* = \{(x^{*T}, u^{*T}, v^{*T})^T\}$ آنگاه، شبکه عصبی (۱۰.۳)-(۹.۳) برای حل مسأله (۲.۳)-(۱.۳) پایدار مجانبی سراسری به نقطه تعادل منحصر بفرد $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ است.

۴.۳ شبیه‌سازی عددی

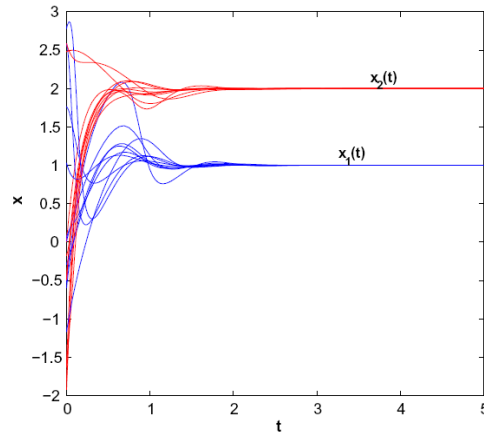
در این قسمت، به منظور نشان دادن کارایی و عملکرد شبکه عصبی ارائه شده، ۶ مثال مختلف را با به کار بردن شبکه عصبی (۱۰.۳)-(۹.۳) حل خواهیم کرد. محاسبات با استفاده از نرم‌افزار متلب و دستور ode45 حاصل شده‌اند.

مثال ۱.۴.۳. ([۹]).

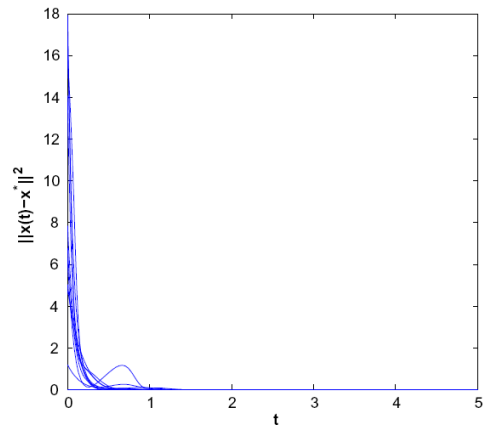
$$\text{minimize } x_1^2 + 2x_2^2 + 2x_1x_2 - 10x_1 - 12x_2$$

$$\text{subject to } \begin{cases} x_1 + 3x_2 - 8 \leq 0, \\ x_1^2 + x_2^2 + 2x_1 - 2x_2 - 3 \leq 0. \end{cases}$$

جواب بهینه این مسأله نقطه $x^* = (1, 2)^T$ است. مدل شبکه عصبی (۱۰.۳)-(۹.۳) را برای حل مسأله فوق به کار می‌بریم. نتایج نشان می‌دهند که با هر نقطه شروع دلخواهی مدل ارائه شده همواره همگراست. به عنوان مثال، رفتار همگرایی $x(t)$ بر پایه مدل (۱۰.۳)-(۹.۳) با ۱۰ نقطه شروع تصادفی در شکل (۱.۳) نشان داده شده است. همچنین نرم خطای $x(t)$ و x^* در شکل (۲.۳) نشان داده شده است.



شکل ۱.۳: همگرایی شبکه عصبی (۱۰.۳)-(۹.۳) با ۱۰ نقطه شروع تصادفی برای مثال (۱.۴.۳).



شکل ۲.۳: همگرایی $\|x(t) - x^*\|^2$ در مثال (۱.۴.۳).

مثال ۲.۴.۳. (۵۰).

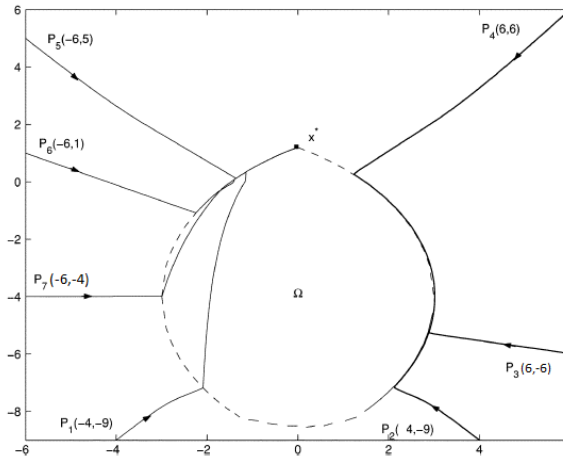
$$\text{minimize } x_1^2 + x_2^2 + 2x_1x_2 + (x_1 - 1)^2 + (x_2 - 3)^2$$

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

جواب بهینه این مسأله $x^* = (0, 1/196)^T$ است. شکل (۴.۳) نمودار صفحه فاز متغیرهای $(x_1(t), x_2(t))^T$ را با ۱۱ نقطه شروع تصادفی نشان می‌دهد. این بردارها به جواب بهینه x^* همگرای سراسری هستند. به منظور مقایسه، مسأله بالا را با استفاده از مدل کندی-چاو^۵ که به صورت زیر است حل می‌کنیم.

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

برای حل مسأله با استفاده از این مدل از ۱۱ نقطه شروع و پارامتر جریمه $s = 5000$ استفاده شده است [۵۰]. مطابق شکل (۳.۳) دیده می‌شود که مدل شبکه عصبی کندی-چاو به جواب بهینه x^* همگرای سراسری نمی‌باشد.



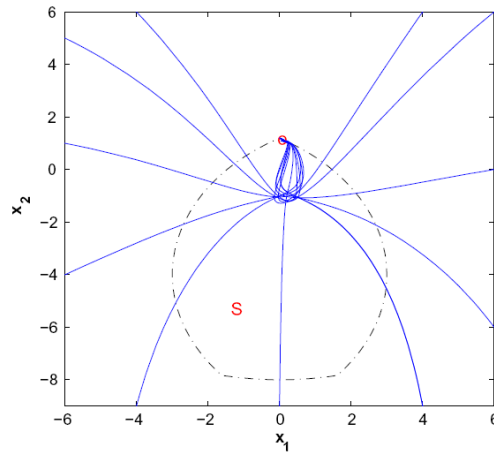
شکل ۳.۳: رفتار همگرایی مثال (۲.۴.۳) با ۷ نقطه شروع تصادفی و به کار بردن مدل کندی-چاو.

مثال ۳.۴.۳. ([۵۴]).

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

جواب بهینه این مسأله $x^* = (0, 0, 1)^T$ است. شکل (۵.۳) همگرایی مثال بالا را با ۳۰ نقطه شروع دلخواه و به کار بردن مدل شبکه عصبی (۱۰.۳)-(۹.۳) نشان می‌دهد. به آسانی می‌توان نشان داد که با هر نقطه شروع (داخل و یا خارج ناحیه شدنی)، شبکه عصبی ارائه شده به جواب بهینه x^* همگراست.

^۵Kennedy-Chua



شکل ۴.۳: نمودار صفحه فاز مثال (۲.۴.۳) با استفاده از مدل شبکه عصبی (۱۰.۳)-(۹.۳) با ۱۱ نقطه شروع مختلف.

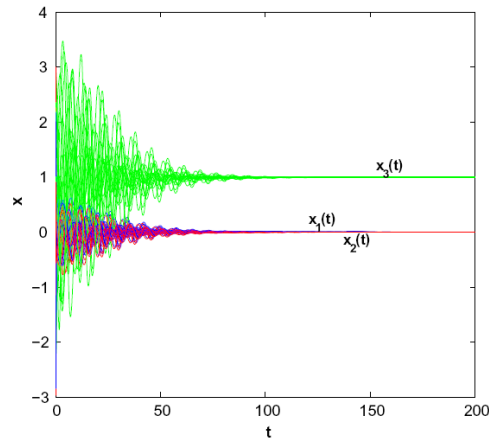
مثال ۴.۴.۳. ([۵۵]).

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

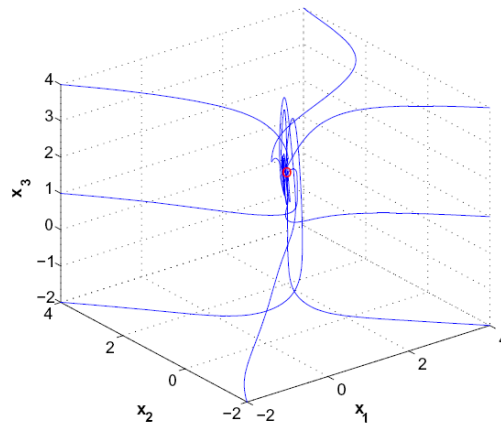
بررسی‌ها نشان می‌دهند که مسیر $x(t)$ به جواب بهینه $x^* = (1/109, 0/922, 1/954)^T$ همگراست. به عنوان مثال، شکل (۶.۳) همگرایی مسأله را با ۸ نقطه شروع تصادفی نشان می‌دهد.

مثال ۵.۴.۳. ([۱۹]).

$$\begin{aligned} & \text{minimize} \quad 0/4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0/5x_3^2 + 0/5x_4^2 + \frac{1}{30}x_1^3 \\ & \text{subject to} \quad \begin{cases} -x_1 + x_2 - x_3 \leq 2, \\ 3x_1 + x_2 - x_3 - x_4 \leq 18, \\ \frac{1}{4}x_1 + x_2 - x_4 = 2, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{cases} \end{aligned}$$



شکل ۵.۳: همگرایی مثال (۳.۴.۳) با استفاده از مدل شبکه عصبی (۱۰.۳)-(۹.۳) با ۳۰ نقطه شروع تصادفی.



شکل ۶.۳: نمودار صفحه فاز مسأله (۴.۴.۳) با ۸ نقطه شروع تصادفی.

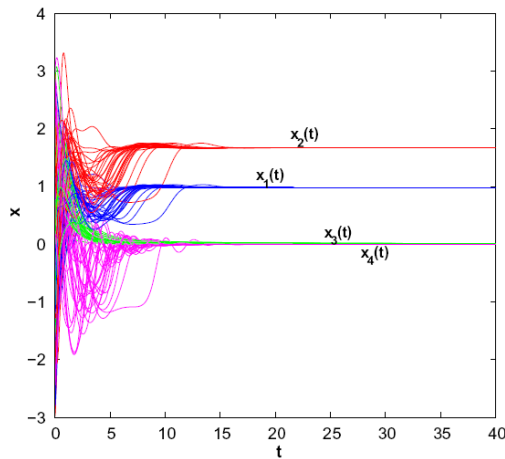
شکل (۷.۳) همگرایی شبکه عصبی ارائه شده را با ۳۰ نقطه شروع تصادفی نشان می‌دهد. همه مسیرهای شبکه به نقطه $x^* = (0.982, 1.672, 0, 0)$ همگرا می‌شوند. به علاوه، اگر نقطه شروع، نقطه شدنی مسأله باشد باز هم مدل ارائه شده به x^* همگراست.

مثال ۶.۴.۳. ([۴۲]).

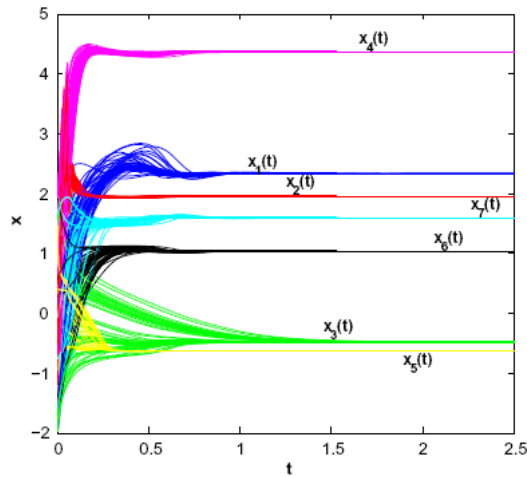
$$\begin{aligned} \text{minimize} \quad & (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^6 - 4x_6x_7 \\ & - 10x_6 - 8x_7 \end{aligned}$$

$$\text{subject to } \begin{cases} 2x_1^2 + 3x_2^2 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0, \\ 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0, \\ 23x_1 + x_2^2 + 6x_3^2 - 8x_4 - 196 \leq 0, \\ 4x_1^2 + x_2^2 + 2x_3^2 - 3x_1x_2 + 5x_4 - 11x_5 \leq 0. \end{cases}$$

این مسأله دارای جواب بهینه $x^* = (2/330, 1/951, -0/4775, 4/366, -0/625, 1/038, 1/594)^T$ است. شکل (۸.۳) همگرایی $x(t)$ را با ۵۰ نقطه شروع تصادفی نشان می‌دهد.



شکل ۷.۳: همگرایی مثال (۵.۴.۳) با استفاده از مدل شبکه عصبی (۱۰.۳) - (۹.۳) با ۳۰ نقطه شروع تصادفی.



شکل ۸.۳: همگرایی مثال (۶.۴.۳) با استفاده از مدل شبکه عصبی (۱۰.۳) - (۹.۳) با ۵۰ نقطه شروع تصادفی.

نتیجه ۷.۴.۳. در این فصل، مدلی از شبکه عصبی برای حل مسائل برنامه‌ریزی غیرخطی محدب ارائه داده شد. همگرایی سراسری این شبکه عصبی با ساختن یک تابع لیاپانوف اثبات گردید. در مقایسه با بعضی از مدل‌های شبکه‌های عصبی، مدل ارائه شده علاوه بر اینکه ساختار ساده‌ای دارد، نیاز به هیچگونه پارامتری برای تعادل‌پذیری ندارد. از ویژگی‌های منحصر بفرد این مدل اینست که نقطه شروع لزومی ندارد که در ناحیه شدنی مسأله واقع شود. همچنین مدل ارائه شده را می‌توان برای حل مسائل بهینه‌سازی با قیود محدب به کار برد. مثال‌هایی که در قسمت قبل مطرح گردید، همگی همگرایی مدل ارائه شده را نشان دادند.

فصل ۴

حل مسئله ربات‌های چندانگشتی با استفاده از
یک مدل شبکه عصبی

۱.۴ مقدمه

امروزه ربات‌های چندانگشتی^۱ به علت کاربردهای وسیع در صنعت و مهندسی مورد توجه زیادی قرار گرفته‌اند. در ساخت و طراحی این ربات‌ها مشکلاتی وجود دارد و بکارگیری آن‌ها مستلزم استفاده از آنالیز پیچیده‌ای است، زیرا لازم است با اعمال یک نیروی آزمند^۲ به جسم، بدون اینکه ربات جسم را رها کند و یا اینکه شدت فشار باعث شکستن و متلاشی شدن آن شود، جسم را جابجا کند.

یکی از پیامدهای مهم کنترل بهینه^۳ در زمینه این نوع ربات‌ها، بهینه‌سازی این نیروی آزمند است. مسأله ربات‌های چندانگشتی در واقع تعیین کردن حداقل نیروی آزمندی است که این نیرو از جانب هر انگشت به جسمی که در فضای سه‌بعدی قرار دارد اعمال می‌شود تا جسم بدون لغزش و سقوط جابجا گردد. بهینه‌سازی این نیروی آزمند، بیشترین اهمیت را در زمینه طراحی ربات‌های چندانگشتی دارد. در واقع هدف، کمینه کردن این نیروی آزمند از جانب هر انگشت است، تا جسم، بدون لغزش و خردشدن و همچنین با غلبه بر نیروهای خارجی از قبیل جاذبه، شتاب و دیگر نیروهای اعمال شده به جسم، در حالت تعادل قرار گیرد. تاکنون روش‌های مختلفی برای بهینه‌سازی این نیرو مطرح شده است. ناکامورا^۴ و همکارانش [۳۲] با استفاده از روش ضرایب لاگرانژ، مقدار بهینه این نیرو را محاسبه کردند. سینها و آبل^۵ [۳۹] برای حل این مسأله، مدل برنامه‌ریزی درجه دوم با قیود غیرخطی را مطرح نمودند. کومار و والدرون^۶ [۲۴] برای بهینه‌سازی این مسأله الگوریتم‌هایی ارائه نمودند. این روش‌ها به دلیل وجود قیود غیرخطی نیاز به محاسبات پیچیده‌ای دارند. بنابراین مسأله بهینه‌سازی این نیروی آزمند، با روش‌هایی مانند خطی‌سازی و استفاده از مسائل برنامه‌ریزی مرتبه دوم قابل حل می‌باشد [۲۲، ۲۳، ۲۹]. در هر صورت، خطی‌سازی برای یافتن این نیروی آزمند در مسائلی که شامل قیود غیرخطی‌اند بدلیل افزایش بعد، اغلب محاسبات را پیچیده و حجم آن را افزایش می‌دهد. باس^۷ و همکارانش [۷] مسأله بهینه‌سازی این نیروی آزمند را مجدداً فرمول‌بندی نموده و آن‌را به عنوان مسأله برنامه‌ریزی نیمه‌معین^۸ مطرح کردند. هان^۹ و همکارانش [۱۳] قیود غیرخطی را در ماتریسی از نامعادلات غیرخطی قرار داده و مسأله موردنظر را با بکارگیری الگوریتم‌های نقطه درونی^{۱۰} حل نمودند. همچنین، لیو^{۱۱} و همکاران او [۲۸] یک روش مینی‌ماکس^{۱۲} به منظور تولید نقطه درونی، به صورت خودکار

^۱ Multifingered robot^۲ Grasping force^۳ Optimal control^۴ Nakamura^۵ Sinha and Abel^۶ Kumar and Waldron^۷ Buss^۸ Semidefinite programming^۹ Han^{۱۰} Interior point algorithm^{۱۱} Liu^{۱۲} Min-max method

ارائه کردند. به منظور کاربردی کردن این مسأله، جواب‌های به دست آمده باید در قانون کنترل-حرکت^{۱۳} صدق کرده و قابل تعمیم به مسائل وابسته به زمان باشند.

در دو دهه اخیر، شبکه‌های عصبی بازگشتی برای حل مسائل مختلف بهینه‌سازی و کاربردهای مهندسی در زمینه مسائل بهینه‌سازی به‌طور گسترده مورد بررسی قرار گرفته‌اند. به‌عنوان مثال شبکه‌های عصبی برای حل مسائل برنامه‌ریزی خطی، غیرخطی، نامعادلات برداری، مسائل مینی‌ماکس و مسائل مکمل خطی و غیرخطی مورد استفاده قرار گرفته‌اند. اگرچه انواع مختلفی از شبکه‌های عصبی برای انجام محاسبات گوناگون ارائه شده‌اند اما تنها یک نوع از این شبکه‌ها برای حل مسأله ربات چندانگشتی پیشنهاد شده است. ژیا^{۱۴} و همکارانش [۵۲] مدلی از شبکه عصبی را برای حل مسأله ربات‌های چندانگشتی مورد بررسی قرار دادند که پایداری سراسری مدل ارائه شده را با استفاده از شرایط لیاپانوف به اثبات رساندند و توانستند تقریبی از جواب بهینه را پیدا کنند. به هر حال شبکه عصبی مطرح شده در [۵۲] نسبتاً پیچیده است و نیاز به ساده‌سازی بیشتری دارد، بنابراین ارائه یک شبکه عصبی کارا با ساختاری ساده، پایدار و همگرا به جواب، برای حل مسأله ربات‌های چندانگشتی ضروری و مفید خواهد بود.

بنابر آنچه گفته شد، در این پایان‌نامه، نوعی شبکه عصبی مصنوعی به منظور حل مسأله ربات چندانگشتی ارائه خواهد شد. با استفاده از یک تکنیک هموارسازی، محدودیت مخروط مرتبه دوم^{۱۵} مسأله (۳.۴) - (۱.۴) به محدودیتی هموار (مشتق‌پذیر) تبدیل خواهد شد که این امر مسأله را به یک مسأله برنامه‌ریزی محدب^{۱۶} تبدیل می‌کند. براساس شرایط بهینگی کاروش-کان-تاکر برای برنامه‌ریزی محدب، مدلی از شبکه عصبی به منظور حل این مسأله پیشنهاد شده است. همچنین اثبات خواهیم کرد که نقطه تعادل شبکه عصبی ارائه شده، معادل با جواب بهینه مسأله برنامه‌ریزی مخروط مرتبه دوم محدب^{۱۷} می‌باشد. علاوه بر آن وجود و یکتایی نقطه تعادل شبکه عصبی ارائه شده، مورد بررسی قرار خواهد گرفت و با معرفی یک تابع لیاپانوف مناسب، پایدار مجانبی سراسری بودن تنها نقطه پایدار شبکه عصبی ارائه شده به دست خواهد آمد. همچنین شبکه عصبی ارائه شده برای حل مسائل مینی‌ماکس و حداکثر جریان^{۱۸} کاربرد دارد.

در این فصل مباحث زیر را مورد بررسی قرار خواهیم داد. در بخش دوم، مسأله ربات چندانگشتی را به یک مسأله برنامه‌ریزی محدب تبدیل خواهیم کرد. در بخش سوم، پایداری نقطه تعادل و همگرایی به جواب بهینه را بررسی می‌کنیم. در بخش چهارم، با بیان یک مثال کارایی و شدنی بودن شبکه عصبی ارائه شده مورد بررسی قرار خواهیم داد و نتیجه‌گیری کلی را در بخش آخر ارائه می‌کنیم.

^{۱۳} Motion-control law

^{۱۴} Xia

^{۱۵} Second-order cone constraint

^{۱۶} Convex nonlinear programming

^{۱۷} Convex second order cone programming

^{۱۸} Maximum flow

۲.۴ بیان مسأله

ربات چندانگشتی شی‌ای در فضای سه‌بعدی با m نقطه تماس بین شی برداشته شده و انگشتان است. مسأله بهینه‌سازی نیروی آزمند، در حقیقت، یافتن نیروی تماس بین جسم موردنظر و انگشتان ربات است به‌طوری‌که ربات، جسم را با غلبه بر نیروهای خارجی وارد شده بر آن در مکان موردنظر قرار دهد. بنابراین نیروی آزمند x_i از جانب هر انگشت به جسم وارد می‌شود تا از لغزش آن جلوگیری کرده و به نیروهای خارجی وارد شده بر جسم غلبه کند. به منظور اطمینان از عدم لغزش جسم، نیروی آزمند x_i می‌بایست در نامعادله $x_i^2 + x_i^3 \leq \mu x_{i1}$ صدق کند که $\mu > 0$ ضریب اصطکاک و x_{i1}, x_{i2}, x_{i3} مولفه‌های نیروی تماسی x_i هستند. همچنین به منظور غلبه بر نیروهای خارجی وارد شده به جسم، هر انگشت باید نیروی آزمند $x_i = [x_{i1}, x_{i2}, x_{i3}]^T$ را به‌گونه‌ای به جسم اعمال کند که برآیند نیروهای خارجی وارد شده به جسم صفر باشد یعنی $Gx = -f_{ext}$ ، که در آن $G \in \mathbb{R}^{6 \times 3m}$ ماتریس تبدیل آزمند^{۱۹} و $x = [x_1, \dots, x_m]^T \in \mathbb{R}^{3m}$ بردار نیروی آزمند^{۲۰} و f_{ext} برآیند نیروهای خارجی وارد شده به جسم نامیده می‌شوند. از طرفی به منظور جلوگیری از متلاشی شدن جسم، نیروی وارد شده به جسم در نقطه تماس باید کمترین مقدار ممکن را داشته باشد. فرض می‌کنیم تابع هدف، تابع مرتبه دوم باشد، لذا مسأله بهینه‌سازی نیروی آزمند، مسأله‌ای با قيود خطی و غیرخطی به‌صورت زیر است:

$$\text{minimize } f(x) = \frac{1}{4} x^T Q x + q^T x \quad (1.4)$$

$$\text{subject to } \begin{cases} \sqrt{x_{i2}^2 + x_{i3}^2} \leq \mu x_{i1}, & (i = 1, \dots, m) \\ Gx = -f_{ext}. \end{cases} \quad (2.4)$$

که در آن $Q \in \mathbb{R}^{3m \times 3m}$ یک ماتریس متقارن و معین مثبت است و $G \in \mathbb{R}^{l \times n}$ [۲۹]. تابع هدف مرتبه دوم مسأله بالا، حداقل نرم وزن‌دار شده بردارهای گشتاور نیروی الحاقی را نمایش می‌دهد و زمانیکه $q = 0$ باشد این تابع، حداقل نرم این نیروها خواهد بود. اکنون برای به‌دست آوردن شبکه عصبی مورد نظر قرار می‌دهیم:

$$g^\varepsilon = \sqrt{\varepsilon^2 + x_{i2}^2 + x_{i3}^2} - \mu x_{i1}, \quad h(x) = Gx + f_{ext}.$$

مطابق آنچه در فصل قبل مطرح شد، به منظور یافتن یک مدل شبکه عصبی برای حل مسأله (۲.۴)–(۱.۴)

تابع لاگرانژ زیر را در نظر می‌گیریم:

^{۱۹} Grasp transformation matrix

^{۲۰} Grasping force vector

$$L(x, u, v) = f(x) + \frac{1}{\nu} \sum_{k=1}^m u_k^\nu g_k^\varepsilon(x) + \sum_{p=1}^l v_p h_p(x), \quad (۳.۴)$$

که در آن $u \in \mathbb{R}^m$ و $v \in \mathbb{R}^l$ ضرایب لاگرانژ نامیده می‌شوند. می‌دانیم جواب بهینه مسأله (۲.۴)–(۱.۴) در شرایط کاروش–کان–تاکر به صورت زیر صدق می‌کند:

$$\begin{cases} u^* \geq 0, \quad g^\varepsilon(x^*) \leq 0, \quad u^{*T} g^\varepsilon(x^*) = 0, \\ Qx^* + q + \nabla g^\varepsilon(x^*)^T u^* + G^T v^* = 0, \\ Gx^* + f = 0, \end{cases} \quad (۴.۴)$$

که در آن $g_i^\varepsilon(x) = \sqrt{\varepsilon^2 + x_{i\nu}^2 + x_{i\mu}^2} - \mu x_{i1}$ و ε عدد ثابت کوچکی است که تقریباً 10^{-6} در نظر گرفته می‌شود. در حقیقت با این روش محدودیت‌های ناهموار به محدودیت‌های هموار تبدیل می‌شوند. واضح است که تابع f و تابع هموار g_ε محدب و دارای مشتق مرتبه دوم هستند. اکنون فرض می‌کنیم $x(\cdot)$ ، $u(\cdot)$ و $v(\cdot)$ متغیرهایی وابسته به زمان باشند. مطابق فصل قبل، هدف ما ساختن مدل دینامیکی است که نسبت به زمان پیوسته بوده و به جواب بهینه مسأله (۲.۴)–(۱.۴) همگراست. حال مدل شبکه عصبی زیر را برای حل مسأله به کار می‌گیریم:

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

نقطه $y_0 = (x_0^T, u_0^T, v_0^T)^T$ را که $u_k(t_0) \neq 0$ برای $k = 1, \dots, m$ به عنوان نقطه شروع در نظر می‌گیریم. به منظور ساده کردن محاسبات فرض می‌کنیم $y = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$ و D^* را مجموعه جواب‌های بهینه مسأله (۲.۴)–(۱.۴) و دوگان آن در نظر می‌گیریم. قرار می‌دهیم:

$$\eta(x, u, v) = \begin{bmatrix} -(Qx + q + \frac{1}{\nu} \nabla g^\varepsilon(x)^T u^\nu + G^T v) \\ \text{diag}(u_1, \dots, u_m) g^\varepsilon(x) \\ Gx + f_{ext} \end{bmatrix}. \quad (۵.۴)$$

بنابراین مدل شبکه عصبی بالا را می‌توانیم به صورت زیر بنویسیم:

$$\frac{dy}{dt} = \eta(y), \quad (۶.۴)$$

$$y(t_0) = y_0, \quad u(t_0) \neq 0. \quad (۷.۴)$$

مشابه آنچه در قضیه (۱.۳.۳) نشان دادیم، اگر $y^* = (x^{*T}, u^{*T}, v^{*T})^T$ نقطه تعادل شبکه عصبی (۷.۴) - (۶.۴) باشد، آنگاه x^* جواب بهینه مسأله (۲.۴) - (۱.۴) است. همچنین، مطابق آنچه در لم (۲.۳.۳) و لم (۳.۳.۳) گفته شد، نقطه تعادل شبکه عصبی ارائه شده یکتاست و برای هر نقطه شروع دلخواهی مانند $y = (x(t)^T, u(t)^T, v(t)^T)^T$ جواب منحصربفرد و پیوسته $y(t_0) = (x(t_0)^T, u(t_0)^T, v(t_0)^T)^T$ وجود دارد.

اثبات پایداری و همگرایی سراسری مدل ارائه شده مشابه آنچه در قضیه (۴.۳.۳) آمده است خواهد بود.

۳.۴ حل مسأله بهینه‌سازی نیروی آزمند در یک ربات سه‌انگشتی

در اینجا مسأله بهینه‌سازی نیروی آزمند را برای یک ربات سه‌انگشتی در نظر می‌گیریم. هدف ما پیدا کردن حداقل نیرویی است که از جانب انگشتان ربات به جسم وارد می‌شود تا آنرا جابجا کند. مسأله ذکر شده را می‌توان به صورت زیر فرمول‌بندی کرد:

$$\text{minimize } \frac{1}{4} \|x\|^2 \quad (۸.۴)$$

$$\text{subject to } \begin{cases} \sqrt{x_{i2}^2 + x_{i3}^2} \leq \mu x_{i1}, & (i = 1, 2, 3) \\ Gx = -f_{ext}, \end{cases} \quad (۹.۴)$$

که در آن $x = [x_{11}, x_{12}, \dots, x_{33}] \in \mathbb{R}^9$ بردار نیروی آزمند، $G \in \mathbb{R}^{6 \times 9}$ ماتریس تبدیلات آزمند، f_{ext} برآیند نیروهای خارجی و μ ضریب اصطکاک در نقطه تماس می‌باشند. فرض می‌کنیم جسم مورد نظر یک چندوجهی با وزن $M = 0.1 \text{ Kg}$ و نقاط تماس $[0, 1, 0]^T$ ، $[1, 0.5, 0]^T$ و $[0, -1, 0]^T$ باشد که توسط یک ربات سه‌انگشتی جابجا می‌گردد. شکل (۱.۴) ربات و جسم موردنظر را نشان می‌دهد. ضریب اصطکاک هر انگشت $\mu = 0.6$ است. همچنین ماتریس تبدیلات آزمند به صورت زیر است:

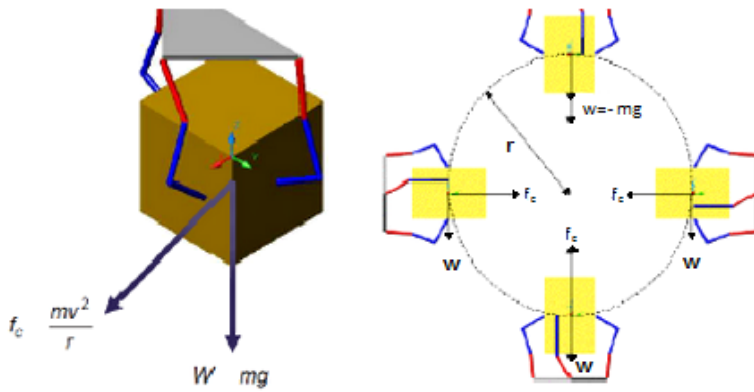
$$G = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0.5 & 0 & -1 & 0 & 1 & 0 \end{bmatrix}.$$

اکنون فرض می‌کنیم ربات جسم را در یک مسیر دایره‌ای با شعاع $r = 0.2 \text{ m}$ و با سرعت ثابتی معادل $v = 0.4 \pi \frac{\text{m}}{\text{s}}$ همانند شکل (۱.۴) جابجا می‌کند. نیروی خارجی وارد شده به مرکز جرم جسم در هر لحظه

از رابطه زیر قابل محاسبه است:

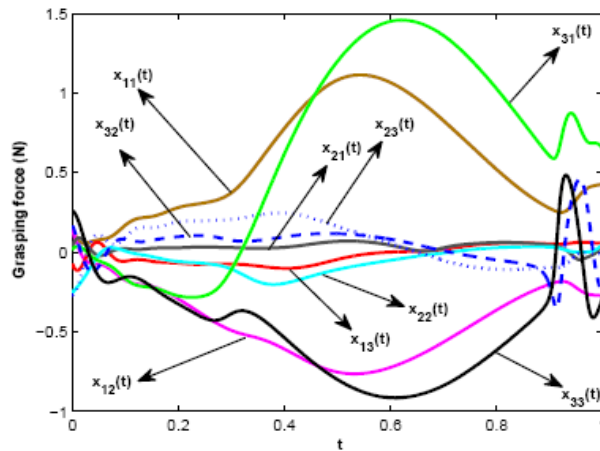
$$f_{ext} = [0, f_c \sin \theta(t), -Mg + f_c \cos \theta(t), 0, 0, 0]^T$$

که در آن شتاب مرکزگرا، $g = 9.8 \frac{m}{s^2}$ ، نیروی مرکزگرا (جانب مرکز) و $\theta(t) = \frac{vt}{r} \in [0, 2\pi]$ است. همگرایی نیروی آزمند وابسته به زمان که با به‌کار بردن مدل شبکه عصبی (۶.۴)–(۷.۴) به‌دست آمده

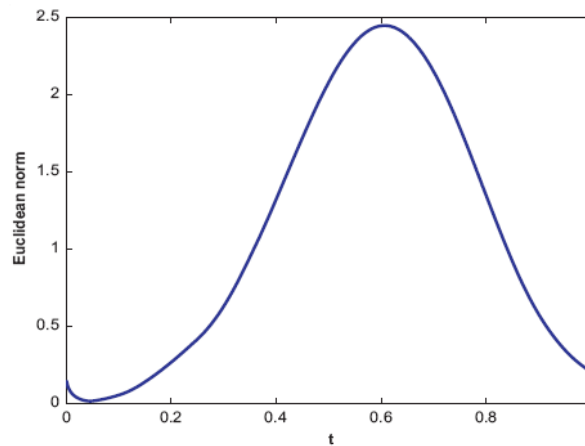


شکل ۱.۴: حرکت ربات سه‌انگشتی در مسیر دایره‌ای (شکل سمت راست). ربات سه‌انگشتی و نیروهای خارجی اعمال شده (شکل سمت چپ).

است در شکل (۲.۴) نشان داده شده است. همچنین، شکل (۳.۴) همگرایی نرم اقلیدسی نیروی بهینه را نشان می‌دهد.



شکل ۲.۴: همگرایی نیروی بهینه ربات سه‌انگشتی



شکل ۳.۴: همگرایی نرم اقلیدسی نیروی بهینه

نتایج و پیشنهاداتی برای کارهای آینده

در این پایان‌نامه ابتدا مروری بر شبکه‌های عصبی داشته و با استفاده از یک مدل شبکه عصبی کارا به حل مسأله یافتن نیروی آزمند در ربات‌های چندانگشتی در فضای سه بعدی پرداختیم. همچنین پایداری و همگرایی سراسری به جواب بهینه را مورد بررسی قرار دادیم. از جمله مزیت‌های مدل ارائه شده می‌توان به موارد زیر

اشاره کرد:

- مدل ذکر شده همگرایی تضمین شده دارد.
 - این مدل نیاز به هیچگونه پارامتر جریمه ندارد.
 - با وجود سادگی مدل، در رده بسیاری از مسائل بهینه‌سازی محدب قابل استفاده است.
 - این مدل هیچگونه وابستگی به نقطه شروع نداشته و این نقطه می‌تواند خارج از ناحیه شدنی و نقطه‌ای تصادفی باشد.
 - این مدل می‌تواند قادر به حل مسائل وابسته به زمان و مسائل با ابعاد بزرگ باشد.
- از جمله تحقیقات دیگری که می‌توان در این زمینه انجام داد می‌توان به توسعه مدل ارائه شده برای حل مسائل ربات‌های با بیشتر از سه انگشت و همچنین توسعه مدل برای حل مسأله کنترل بازوهای ربات و مسائل پردازش تصویر اشاره کرد.

مراجع

- [1] Agrawal SK, Fabien BC. Optimization of dynamic systems. Netherlands: Kluwer Academic Publishers; 1999.
- [2] Avriel M. Nonlinear programming: Analysis and Methods. Englewood Cliffs, NJ: Prentice-Hall; 1976.
- [3] Azarbak S, Nazemi AR, Shahsavani D, "An Application of Neural networks for solving linear and quadratic programming problems", 3th International Conference of Iranian Operations Research; 2010.
- [4] Bazaraa MS, Sherali HD, Shetty CM, "Nonlinear Programming-Theory and Algorithms", 2nd ed. New York: Wiley, 1993.
- [5] Berry MJA, Linoff G, "Data Mining Techniques", John wiley and sons, 450-460, 1997.
- [6] Bouzerdoun A, pattison TR, "Neural network for quadratic optimization with bound constraints", IEEE Trans. on Neural Networks, 4, 293-304, 1993.
- [7] Buss M, Hashimoto H, Moore JB, "Dextrous hand grasping force optimization," IEEE Trans. Robot. Automat., vol. 12, pp. 406-417, June 1996.
- [8] Cheng FT, Orin DE, "Efficient algorithm for optimal force distribution- The compact dual LP method, " IEEE Trans. Robot. Automat., vol. 6, pp. 178-187, Apr, 1990.
- [9] Effati S, Ghomashi A, Nazemi AR, "Application of projection neural network in solving convex programming problems", Appl. Math. Comput, 188, 1103-1114, 2007.
- [10] Effati S, Nazemi AR, "Neural network models and its application for solving linear and quadratic programming problems", Appl. Math. Comput, 172, 305-331, 2006.
- [11] Fletcher R. Practical methods of optimization. New York: Wiley; 1981.
- [12] Gao X. A novel neural network for nonlinear convex programming. IEEE Trans Neural Netw ; 15:613-21, 2004.
- [13] Han L, Trinkle JC, Li ZX, "Grasp analysis as linear matrix inequality problems," IEEE Trans. Robot. Automat., vol. 16, pp. 663-674, Dec. 2000.
- [14] He BS, Zhou J. A modified alternating direction method for convex minimization problems. Appl Math Lett ; 13:123-30, 2000.
- [15] Hopfield JJ, Tank DW, "Simple neural optimization network:An A/D converter, signal decision circuit, and a linear programming pircuit", IEEE Trans on Circuits and Systems, 33, 533-541, 1986.
- [16] Huang YC. A novel method to handle inequality constraints for convex programming neural network. Neural Process Lett ; 16:17-27, 2002.

- [17] Hu X, Balasubramaniam P, Recurent neural networks, ISBN: 978-953-7619-08-4, pp. 255-388, I-Tech, Vienna, Austria, 2008.
- [18] Hull JC, "Options,Futures,And Other Derivatives", 6th Edition, Perntice Hall,New Jersey, 2006.
- [19] Hu X, Wang J. Design of general projection neural network for solving monotone linear variational inequalities and linear and quadratic programming problems. IEEE Trans Syst Man Cybernet Part B: Cybernet ; 37:1414-21, 2007.
- [20] Jones CP, "Investments: Analysis and Management", New York: Wiley, 2001.
- [21] Kennedy MP, Chua LO, "Neural networks for nonlinear programming", IEEE Trans. on Neural Networks, 35, 554-562, 1988.
- [22] Kerr J, Roth B, "Analysis of multifingered hands," Int. J. Robot. Res., vol. 8, no. 4, pp. 3-17, 1986.
- [23] Klein CA, Kittivatcharapong S, "Optimization force distribution for the legs of a walking machine with friction cone constraints," IEEE Trans. Robot. Automat., vol. 6, pp. 73-85, Feb. 1990.
- [24] Kumar V, Waldron K, "Suboptimal algorithms for force distribution in multifingered grippers," IEEE Trans. Robot. Automat., vol. 8,pp. 7-22, Feb. 1992.
- [25] Leung Y, Chen K, Gao X. A high-performance feedback neural network for solving convex nonlinear programming problems. IEEE Trans Neural Network ; 14:1469-77, 2003.
- [26] Liang XB, Wang J. A recurent neural network for nonlinear optimization with a continuously differentiable objective function and bounded constraints. IEEE Trans Neural Netw ; 11:1251-62, 2000.
- [27] Liao L, Qi H, Qi L. Solving nonlinear complementarity problems with neural networks: a reformulation method approach. J Comput App Math ; 131 :343-59, 2001.
- [28] Liu GF, Xu JJ, Li ZX, "Automatc real-time grasping-force determination for multifingered manipulation: Theory and experiments," in Proc. IROS, pp. 1676-1680, 2002.
- [29] Liu YH, "Qualitative test and force optimization of 3-D frictional form-closure grasp using linear programming,"IEEE Trans. Robot. Automat., vol. 15,pp. 163-173, Apr. 1999.
- [30] Malek A, Hosseinipour-Mahani N, Ezazipour S. Efficient recurrent neural network model for the solution of general nonlinear optimization problems. Opt Methods Softw ; 25:1-18, 2009.
- [31] Miller RK, Michel AN. Ordinary differential equations. New York: Academic Press; 1982.
- [32] Nakamura Y, Nagai K, Yoshikawa T, "Dynamics and stability in coordination of multiple robotic mechanisms,"Int. J. Robot. Res., vol. 8, no. 2, pp. 44-61, 1989.
- [33] Ortega TM, Rheinboldt WC. Iterative solution of nonlinear equations in several variables. New York: Academic; 1970.
- [34] Popescua M, Dumitracheb A. Stabilization of feedback control and stabilizability optimal solution for nonlinear quadratic problems. Commun Nonlinear Sci Numer Simul ; 16: 2319-27, 2011.
- [35] Qi L, Sun J, "A nonsmooth version of Newton method, Math Program, 58, 353-368, 1993.
- [36] Rodriguez-Vazquez A, Dommnguer-Castro R, Rueda A, Huertas J. L and Sahnchez-Sinencio E, "Nonlinear switched-capacitor neural network for optimization problems", IEEE Trans on Circuits and Systems, 384-390, 1990.

- [37] Shelly B, Stephenson S, "The use of artificial neural network in completion simulation and design", *Computer and Geosciences*, 26, 941-951, 2000.
- [38] Simmons G, "Differential Equations and Applications", McGraw-Hill Education (India), 2006.
- [39] Sinha PR, Abel JM, "A contact stress model for multifingered grasp of rough objects," *IEEE Trans. Robot. Automat.* vol. 8, pp. 7-22, Feb. 1992.
- [40] Wu X, Xia Y, Li J and Chen W, "A High Performance Neural Network for solving Linear and Quadratic Programming Problems", *IEEE Trans. On Neural Networks*, 7, 3, 643-651, 1996.
- [41] Xia Y, "A neural network for solving linear and quadratic programming problems", *IEEE Trans. on Neural Networks*, 7, 1544-1547, 1996.
- [42] Xia Y, Feng G. A new neural network for solving nonlinear projection equations. *Neural Netw* ; 20:577-89, 2007.
- [43] Xia Y, Feng G, "An improved network for convex quadratic optimization with application to real-time beam forming", *Neurocomputing*, 64, 359-374, 2005.
- [44] Xia Y, Feng G, Wang J, "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equation", *Neural Networks*, 17, 1003-1015, 2004.
- [45] Xia Y, Leung H, Wang J, "A projection neural network and its application to constrained optimization problems", *IEEE Transactions On Circuits And Systems*, 44, 4, 447-458, 2002.
- [46] Xia Y, Leng H, Wang J. A projection neural network and its application to constrained optimization problems. *IEEE Trans Circuits Syst* ; 49:447-58, 2002.
- [47] Xia Y, Wang J, "A recurrent neural network for solving linear projection equation", *Neural Networks*, 13, 337-350, 2000.
- [48] Xia Y, Wang J, "A general projection neural network for solving monotone variational inequality and related optimization problems" , *IEEE Trans. On Neural Networks*, 15, 318-328, 2004.
- [49] Xia Y, Wang J. A recurrent neural network for solving linear projection equations. *Neural Netw* ; 13:337-50, 2000.
- [50] Xia Y, Wang J. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Trans Circuits Syst* ; 51:447-58, 2004.
- [51] Xia Y, Wang J, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 147-154, Jan. 2001.
- [52] Xia Y, Wang J, "Grasping-Force Optimization for Multifingered Robotic Hands Using a Recurrent Neural Network", *IEEE Transactions on robotics and Automation*, vol. 20, No. 3, june 2004.
- [53] King-Bao, "Exponential Stability of Globally Projected Dynamic Systems", *IEEE Trans. On Neural Networks*, 14, 2 426-431, 2003.
- [54] Yang Y, Cao J. A feedback neural network for solving convex constraint optimization problems. *Appl Math Comput*; 201:340-50, 2008.
- [55] Yang Y, Xu X. The projection neural network for solving convex nonlinear programming. In: Huang D-S, Heutte L, Loog M, editors. *ICIC 2007, LNAI*, vol. 4682. Heidelberg: Springer-Verlag; p. 174-81, 2007.

واژه‌نامه انگلیسی به فارسی

Artificial	مصنوعی
Convex function	تابع محدب
Convex programming	برنامه‌ریزی محدب
Equilibrium point	نقطه تعادل
Global convergance	همگرای سراسری
Global minimum	می‌نیمم سراسری
Globally asymptotically stable	پایدار مجانبی سراسری
Grasping force	نیروی آزمند
Invariant set	مجموعه پایدار
Jacobian matrix	ماتریس ژاکوبین
Local minimum	می‌نیمم موضعی
Multifingered robot	ربات چندانگشتی
Monotone	یکنوا
Motion-Control Law	قانون کنترل حرکت
Neural network	شبکه عصبی
Neuron	نرون
Optimal control	کنترل بهینه
Optimality	بهینگی
Positive limit set	مجموعه حدود مثبت
Positive semi-definite	نیمه معین مثبت
Stability	پایداری
Sigmoidal function	تابع سیگموئید

Surname: Hoseini

Name: Mehdi

Title: A Recurrent Neural Network Model for Solving Grasping-Force Optimization for Multifingered Robotic Hands.

Supervisor: Dr. A.R. Nazemi

Degree: Master of Science

Subject: Applied Mathematics

Field: Operations Research

University of Shahrood

Faculty Of Mathematical Sciences

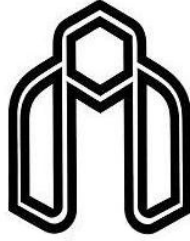
Date: October 2013

Number of pages: 64

Keywords: Neural network, Convex programming, Convergent, Stability, Multifingered robotic hands.

Abstract

Today multifingered robotic hands have become of great interest in robotics due to their advantages over conventional grippers in tasks requiring dexterous manipulation. Robotichands are more complex in construction and require more sophisticated analysis to determine the grasping force to be exerted on an object without dropping or breaking the object. In this thesis a neural network is constructed on the basis of the duality theory, optimization theory, convex analysis theory, Lyapunov stability theory and Lasalle invariance principle to solve Multifingered Robotic Hands (MRH) problems. According to the Saddle point theorem, the equilibrium point of the proposed neural network is proved to be equivalent to the optimal solution of the strictly convex quadratic optimization problem. By employing Lyapunov function approach, it is also shown that the proposed neural network model is stable in the sense of Lyapunov and it is globally convergent to an exact optimal solution of the original optimization problem. Simulation results show that the proposed neural network is feasible and efficient.



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 Recurrent Neural Network Model for Solving Grasping-Force Optimization for Multifingered Robotic Hands.

Supervisor
Dr. A.R. Nazemi

by
Mehdi Hoseini

October 2013