



دانشکده علوم ریاضی

گروه ریاضی کاربردی

پایان نامه کارشناسی ارشد

# حل ردهای از مسائل کنترل بهینه تاخیر زمانی با استفاده از یک مدل از شبکه عصبی

انسه فیاضی

استاد راهنما

علیرضا ناظمی

مهر ۱۳۹۴

## تقدیم به پدر و مادر عزیزم

آنان که همواره بر کوتاهی و درشتی من، قلم عفو کشیده و گریانه از کنار غفلت هایم گذشته اند  
و در تمام عرصه های زندگی یار و یاور بی چشم داشت برای من بوده اند.

## تقدیم به برادرانم

که کنجینه های دلگرمی، مهربانی و محبت اند. کسانی که با فداکاری و گذشت مراد این امر خطیر  
همراهی و همیاری نمودند.

سپاس خدای را که به انسان توانایی و دانایی بخشید تا به بندگانش شفقت ورزد، مهربانی کند و در  
 حل مشکلاتشان یاری‌شان نماید. از راحت خویش بگذرد و آسایش هم نوحان را مقدم دارد،  
 با او معامله کند و در این خلوص انباز نکیرد و خوش باشد که پروردگار سمیع است و بصیر...  
 سپاس خدای را که پدر و مادری فداکار نسیم ساخته تا در سایه درخت پر بار وجودشان بیایم و از ریشه  
 آن ها شاخ و برگ گیریم و از سایه وجودشان در راه کسب علم و دانش تلاش نمایم. والدینی که بودندشان  
 تاج افتخاری است بر سرم و نامشان دلیلی است بر بودنم. چرا که این دو وجود، پس از پروردگار  
 مایه هستی ام بوده اند و راه رفتن را در این وادی پر از فراز و نشیب زندگی آموختند. آموزگارانی که  
 برایم زندگی، بودن و انسان بودن را معنا کردند...  
 سپاسگذار کسانی، هستم که سراغ از تولد من هستند، از یکی زاده می‌شوم و از دیگری جاودانه. استادی که  
 سپیدی را بر تخته سیاه زندگیم نگاشت و مادری که تار موئی از او پایی من سیاه نامند...

## تعمدنامه

اینجانب انسیه فیاضی دانشجوی کارشناسی ارشد رشته ریاضی کاربردی دانشکده علوم ریاضی دانشگاه شاهرود، نویسنده پایان نامه با عنوان حل رده‌ای از مسائل کنترل بهینه تاخیر زمانی با استفاده از یک مدل از شبکه عصبی، تحت راهنمایی علیرضا ناظمی متعهد می‌شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان نامه، تاکنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه شاهرود متعلق دارد، و مقالات مستخرج با نام “ دانشگاه شاهرود “ یا “ Shahrood University “ به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آوردن نتایج اصلی پایان نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده) شده است، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

انسه فیاضی  
مهر ۱۳۹۴

## مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان نامه بدون ذکر منبع مجاز نمی‌باشد.

## چکیده

در این پایان نامه به حل عددی رده‌ای از مسائل کنترل بهینه تاخیر زمانی با استفاده از شبکه عصبی می‌پردازیم. ابتدا در فصل اول به مقدمه و معرفی شبکه عصبی و تعاریف مورد نیاز می‌پردازیم. در فصل دوم یک مساله کنترل بهینه تاخیری از نوع بولزا را در نظر می‌گیریم که با استفاده از تقریب پاده به یک مساله بدون تاخیر زمانی تبدیل می‌شود. سپس با روش شبکه عصبی به حل عددی آن می‌پردازیم و در پایان فصل نتایج عددی را برای شش مساله بررسی خواهیم کرد. فصل سوم شامل مساله کنترل بهینه تاخیری افق نامتناهی است. آن را با استفاده از تقریب پاده از حالت تاخیر خارج می‌کنیم و با استفاده از تغییر متغیر مساله را به یک مساله افق متناهی تبدیل می‌کنیم. در ادامه با روش شبکه عصبی به حل عددی مساله بدست آمده می‌پردازیم و در پایان فصل سوم نتایج عددی را با حل دو مساله نشان می‌دهیم. تمامی مثال‌های پایان نامه با نرم افزار گمز حل شده و نمودارها در سرتاسر پایان نامه با استفاده از نرم افزار متلب رسم شده است.

**کلید واژه:** شبکه عصبی، تقریب پاده، مسائل کنترل بهینه تاخیر زمانی، مسائل افق نامتناهی

# لیست مقالات مستخرج از پایان نامه

۱. مقاله اول

۲. مقاله دوم

۳. مقاله سوم

# فهرست مطالب

|    |                                                               |
|----|---------------------------------------------------------------|
| خ  | لیست تصاویر                                                   |
| ۱  | لیست جداول                                                    |
| ۳  | ۱ مقدمات و مفاهیم اساسی                                       |
| ۳  | ۱.۱ مقدمه‌ای بر شبکه‌های عصبی                                 |
| ۳  | ۱.۱.۱ شبکه‌های عصبی بیولوژیکی                                 |
| ۴  | ۲.۱.۱ شبکه‌های عصبی مصنوعی                                    |
| ۱۰ | ۳.۱.۱ شبکه‌های عصبی پیش‌خورد                                  |
| ۱۲ | ۴.۱.۱ تاریخچه تکامل شبکه عصبی                                 |
| ۱۴ | ۵.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی |
| ۱۵ | ۶.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND                  |
| ۱۵ | ۷.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع OR                   |
| ۱۶ | ۸.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND NOT              |
| ۱۶ | ۹.۱.۱ شبکه عصبی هب                                            |
| ۱۷ | ۱۰.۱.۱ شبکه عصبی آدالاین                                      |
| ۲۰ | ۱۱.۱.۱ شبکه عصبی مادالاین                                     |
| ۲۲ | ۱۲.۱.۱ شبکه‌های حافظه پیوندی                                  |
| ۲۲ | ۱۳.۱.۱ شبکه‌های عصبی هاپفیلد                                  |
| ۲۷ | ۲.۱ مقدمه‌ای بر نظریه کنترل                                   |
| ۲۸ | ۳.۱ شکل کلی مساله کنترل بهینه                                 |
| ۲۹ | ۴.۱ شرط لازم مسائل کنترل بهینه                                |
| ۳۰ | ۵.۱ اصل حداقل‌یابی پونتریاگین                                 |
| ۳۲ | ۶.۱ معادلات دیفرانسیل تاخیری                                  |
| ۳۳ | ۱.۶.۱ تقریب پاده                                              |

|    |     |                                               |
|----|-----|-----------------------------------------------|
| ۳۵ | ۲   | حل عددی مسائل کنترل بهینه تاخیری              |
| ۳۵ | ۱.۲ | مقدمه                                         |
| ۳۵ | ۲.۲ | معرفی مساله                                   |
| ۳۶ | ۳.۲ | تبدیل مساله تاخیری به مساله بدون تاخیر        |
| ۳۹ | ۴.۲ | شبکه عصبی                                     |
| ۴۳ | ۵.۲ | مثال‌های عددی                                 |
| ۵۹ | ۳   | حل عددی مسائل کنترل بهینه تاخیری افق نامتناهی |
| ۵۹ | ۱.۳ | مقدمه                                         |
| ۵۹ | ۲.۳ | معرفی مساله                                   |
| ۶۰ | ۳.۳ | تبدیل مساله تاخیری به یک مساله بدون تاخیر     |
| ۶۲ | ۴.۳ | تبدیل مساله افق نامتناهی به مساله افق متناهی  |
| ۶۴ | ۵.۳ | شبکه عصبی                                     |
| ۶۷ | ۶.۳ | مثال‌های عددی                                 |
| ۷۹ |     | مراجع                                         |
| ۸۷ |     | واژه‌نامه فارسی به انگلیسی                    |
| ۸۹ |     | واژه‌نامه انگلیسی به فارسی                    |



# لیست تصاویر

|    |       |                                                            |
|----|-------|------------------------------------------------------------|
| ۴  | ..... | ۱.۱ ساختار یک نرون.                                        |
| ۵  | ..... | ۲.۱ مدل ریاضی یک نرون.                                     |
| ۶  | ..... | ۳.۱ ساختار یک نرون با لایه ورودی، لایه پنهان و لایه خروجی. |
| ۸  | ..... | ۴.۱ شبکه عصبی مثال (۱.۱.۱)                                 |
| ۹  | ..... | ۵.۱ تابع پله‌ای                                            |
| ۹  | ..... | ۶.۱ تابع پله‌ای دوقطبی                                     |
| ۹  | ..... | ۷.۱ تابع سیگموئید                                          |
| ۱۰ | ..... | ۸.۱ تابع سیگموئید قطبی                                     |
| ۱۱ | ..... | ۹.۱ شبکه‌های عصبی دو لایه بازگشتی.                         |
| ۱۱ | ..... | ۱۰.۱ ساختار شبکه عصبی مصنوعی (پرسپترون)                    |
| ۱۲ | ..... | ۱۱.۱ پرسپترون چندلایه                                      |
| ۱۵ | ..... | ۱۲.۱ نرون مک کلاچ برای تابع AND                            |
| ۱۶ | ..... | ۱۳.۱ نرون مک کلاچ برای تابع OR                             |
| ۱۶ | ..... | ۱۴.۱ نرون مک کلاچ برای تابع AND NOT                        |
| ۱۷ | ..... | ۱۵.۱ شبکه عصبی هب                                          |
| ۱۸ | ..... | ۱۶.۱ الگوریتم هب برای مثال (۲.۱.۱)                         |
| ۱۹ | ..... | ۱۷.۱ الگوریتم هب برای مثال (۲.۱.۱)                         |
| ۲۰ | ..... | ۱۸.۱ شبکه عصبی آدالین                                      |
| ۲۱ | ..... | ۱۹.۱ شبکه عصبی مادالین                                     |
| ۲۲ | ..... | ۲۰.۱ شبکه عصبی دیگر پیوندی                                 |
| ۲۲ | ..... | ۲۱.۱ شبکه عصبی خود پیوندی                                  |
| ۲۳ | ..... | ۲۲.۱ شبکه عصبی تشخیص الگو                                  |
| ۲۳ | ..... | ۲۳.۱ ساختار شبکه هاپفیلد با سه نرون                        |
| ۲۵ | ..... | ۲۴.۱ مدل الکتریکی شبکه هاپفیلد                             |
| ۲۵ | ..... | ۲۵.۱ یک واحد از نرون شبکه هاپفیلد                          |
| ۴۴ | ..... | ۱.۲ نمودار کنترل $u(t)$ برای مثال ۱.۵.۲                    |

|    |       |                                                          |      |
|----|-------|----------------------------------------------------------|------|
| ۴۵ | ..... | نمودار مسیرهای $x_1(t)$ و $x_2(t)$ برای مثال ۱.۵.۲       | ۲.۲  |
| ۴۵ | ..... | نمودار مسیرهای $x_3(t)$ و $x_4(t)$ برای مثال ۱.۵.۲       | ۳.۲  |
| ۴۷ | ..... | نمودار کنترل $u(t)$ و مسیر $x_1(t)$ برای مثال ۲.۵.۲      | ۴.۲  |
| ۴۸ | ..... | نمودار کنترل $u(t)$ برای مثال ۳.۵.۲                      | ۵.۲  |
| ۴۹ | ..... | نمودار مسیرهای $x_1(t)$ و $x_2(t)$ برای مثال ۳.۵.۲       | ۶.۲  |
| ۴۹ | ..... | نمودار مسیرهای $x_3(t)$ و $x_4(t)$ برای مثال ۳.۵.۲       | ۷.۲  |
| ۵۱ | ..... | نمودار کنترل $u(t)$ برای مثال ۴.۵.۲                      | ۸.۲  |
| ۵۱ | ..... | نمودار مسیر $x_1(t)$ برای مثال ۴.۵.۲                     | ۹.۲  |
| ۵۲ | ..... | نمودار مسیرهای $x_3(t)$ و $x_4(t)$ برای مثال ۴.۵.۲       | ۱۰.۲ |
| ۵۳ | ..... | نمودار کنترل $u(t)$ برای مثال ۵.۵.۲                      | ۱۱.۲ |
| ۵۴ | ..... | نمودار مسیر $x_1(t)$ برای مثال ۵.۵.۲                     | ۱۲.۲ |
| ۵۴ | ..... | نمودار مسیرهای $x_2(t)$ و $x_3(t)$ برای مثال ۵.۵.۲       | ۱۳.۲ |
| ۵۶ | ..... | نمودار کنترل $u(t)$ برای مثال ۶.۵.۲                      | ۱۴.۲ |
| ۵۶ | ..... | نمودار مسیر $x_1(t)$ برای مثال ۶.۵.۲                     | ۱۵.۲ |
| ۵۷ | ..... | نمودار مسیرهای $x_2(t)$ و $x_3(t)$ برای مثال ۶.۵.۲       | ۱۶.۲ |
| ۷۰ | ..... | نمودار کنترل $v(\tau)$ برای مثال ۱.۶.۳                   | ۱.۳  |
| ۷۰ | ..... | نمودار مسیرهای $y_1(\tau)$ و $y_2(\tau)$ برای مثال ۱.۶.۳ | ۲.۳  |
| ۷۱ | ..... | نمودار مسیرهای $y_3(\tau)$ و $y_4(\tau)$ برای مثال ۱.۶.۳ | ۳.۳  |
| ۷۳ | ..... | نمودار کنترل $v(\tau)$ برای مثال ۲.۶.۳                   | ۴.۳  |
| ۷۴ | ..... | نمودار مسیرهای $y_1(\tau)$ و $y_2(\tau)$ برای مثال ۲.۶.۳ | ۵.۳  |
| ۷۴ | ..... | نمودار مسیرهای $y_3(\tau)$ و $y_4(\tau)$ برای مثال ۲.۶.۳ | ۶.۳  |
| ۷۵ | ..... | نمودار مسیرهای $y_5(\tau)$ و $y_6(\tau)$ برای مثال ۲.۶.۳ | ۷.۳  |

# لیست جداول

|     |                 |   |
|-----|-----------------|---|
| ۱.۱ | توابع فعال سازی | ۷ |
|-----|-----------------|---|



# فصل ۱

## مقدمات و مفاهیم اساسی

### ۱.۱ مقدمه‌ای بر شبکه‌های عصبی

#### ۱.۱.۱ شبکه‌های عصبی بیولوژیکی

واحد اصلی دستگاه عصبی، سلولی خاص به نام نرون می‌باشد و بدون شک نحوه کار مغز و راز شعور آدمی در آن نهفته است. نرون‌ها با وجود تفاوت‌های زیاد از نظر اندازه و شکل ظاهری، مشخصه‌های مشترکی دارند. همان‌طور که در شکل (۱.۱) نشان داده شده است از تنه سلولی نرون تعدادی شاخک کوتاه خارج می‌شود که دندریت<sup>۱</sup> نام دارند. دندریت‌ها و تنه سلولی، سیگنال‌ها را از نرون‌های مجاور دریافت می‌کنند و از طریق یک لوله باریک به نام آکسون<sup>۲</sup> به نرون‌های دیگر منتقل می‌گردد. آکسون در انتهای خود به تعدادی رشته جانبی باریک تقسیم می‌گردد که پایانه‌های آکسونی نام دارد و با دندریت‌های سایر نرون‌ها مرتبط است. ارتباط بین آکسون یک نرون و دندریت نرون دیگر را سیناپس<sup>۳</sup> می‌نامند. یک سیناپس مرکب از پایانه قبل از سیناپسی<sup>۴</sup>، شکاف سیناپسی<sup>۵</sup> و پایانه بعد از سیناپسی<sup>۶</sup> می‌باشد. همه سیگنال‌های جمع‌بندی شده، در تنه نرون ترکیب می‌شوند و اگر وسعت سیگنال‌های ترکیب شده به آستانه نرون برسد، مرحله تحریک شدن<sup>۷</sup> فعال می‌شود و یک سیگنال خروجی تولید می‌شود. این سیگنال به صورت یک پالس منفرد یا بخشی از پالس‌ها در یک میزان خاص به موازات آکسون به پایانه‌های سیناپسی انتقال می‌یابد و موجب ترشح موادی به نام عصب-رسانه می‌شود. عصب-رسانه در داخل شکاف سیناپسی پخش می‌شود و نرون بعدی را تحریک می‌کند و به این ترتیب یک سیگنال از یک نرون به دیگری انتقال می‌یابد. تعداد بسیار زیادی آکسون از نرون‌های مختلف با دندریت‌های یک

---

<sup>۱</sup>Dendrite

<sup>۲</sup>Axon

<sup>۳</sup>Synapse

<sup>۴</sup>Presynaptic Terminal

<sup>۵</sup>Synaptic Cleft

<sup>۶</sup>Postsynaptic Terminal

<sup>۷</sup>Firing

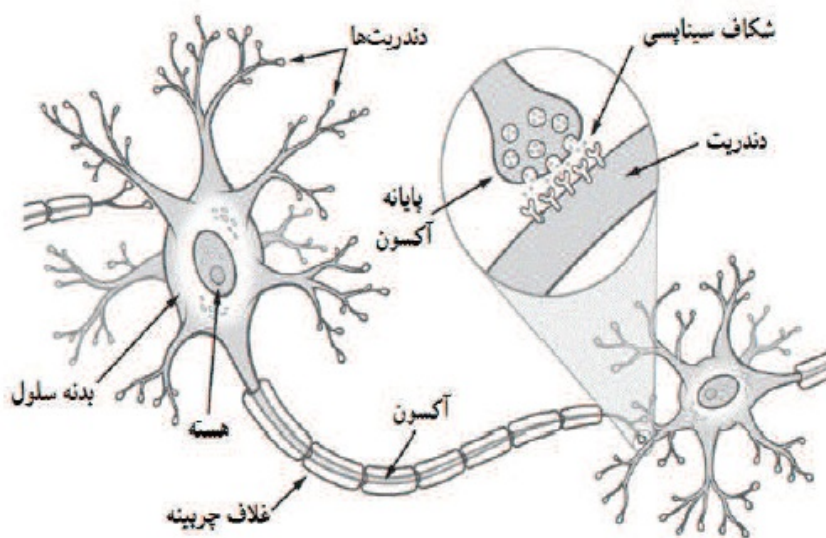
نرون به این صورت ارتباط برقرار می‌کنند. نرون‌ها با توجه به عملکردشان به سه دسته تقسیم می‌شوند:

(۱) **نرون‌های حسی:** این نرون‌ها پیام‌ها را از گیرنده‌ها به دستگاه عصبی مرکزی منتقل می‌کنند. گیرنده سلول خاصی است که در اندام‌های حسی، عضلات، پوست و ... قرار دارد و تغییرات فیزیکی یا شیمیایی را در می‌یابد، این تغییرات را به سیگنال‌های عصبی تبدیل می‌کند.

(۲) **نرون‌های حرکتی:** این نرون‌ها پیام‌ها را از مغز یا طناب نخاعی به اندام‌های عمل کننده می‌رسانند.

(۳) **نرون‌های رابط:** این نرون‌ها پیام‌ها را از نرون‌های حسی می‌گیرند و به یک نرون رابط دیگر انتقال می‌دهند و یا پیام‌ها را به یک نرون حرکتی می‌رسانند. نرون‌های رابط فقط در مغز، چشم و طناب نخاعی یافت می‌شوند.

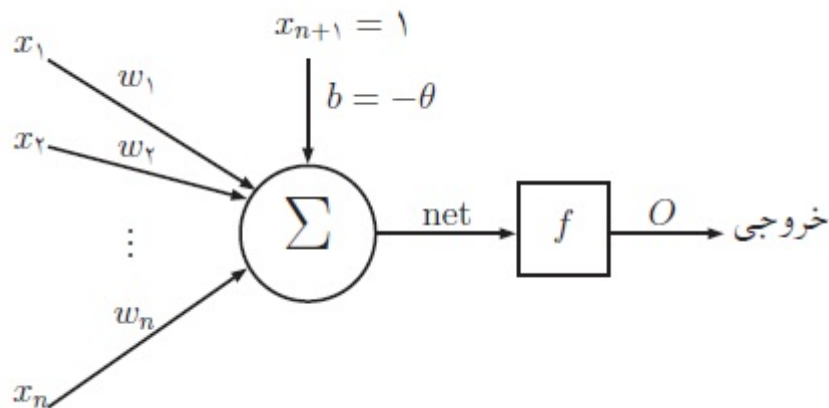
حال منظور از یادگیری عبارت است از تغییری نسبتاً ساختاری در سیناپس (بعد از تحریک)، که این تغییر ساختاری باعث افزایش کارایی سیناپس می‌شود.



شکل ۱۰۱: ساختار یک نرون.

## ۲.۱.۱ شبکه‌های عصبی مصنوعی

بعد از اشاره‌ای گذرا بر شبکه عصبی بیولوژیکی و نشان دادن گوشه‌ای بسیار کوچک از این جهان مرموز و فوق‌العاده پیچیده به موضوع اصلی یعنی شبکه‌های عصبی مصنوعی می‌پردازیم. شبکه‌های عصبی مصنوعی مدل‌های سخت افزاری یا نرم افزاری الهام گرفته از ساختار و رفتار نرون‌های بیولوژیکی و سیستم‌های عصبی انسان هستند که شامل عناصر پردازشی (به نام نرون‌ها) و ارتباطات میان آنها با ضرایب (وزن‌ها) اختصاص یافته به ارتباطات است، که ساختار شبکه عصبی را تشکیل می‌دهند. در این پایان‌نامه ما شبکه‌های عصبی مصنوعی را یک مدل ریاضی فرض خواهیم کرد و به جای استفاده از عبارت شبکه‌های عصبی مصنوعی به طور خلاصه عبارت شبکه‌های عصبی را به کار خواهیم برد.



شکل ۲.۱: مدل ریاضی یک نرون.

مدل شبیه‌سازی شده یک نرون، در شکل (۲.۱) نمایش داده شده است. یک نرون در حالت کلی یک مجموعه از  $n + 1$  ورودی  $x_j$  ( $j = 1, \dots, n + 1$ ) دارد. این ورودی‌ها همان سیگنال‌های عصبی می‌باشند که از محیط یا از یک نرون دیگر فرستاده می‌شوند.  $w_j$  ( $j = 1, \dots, n + 1$ ) که روی آکسون‌ها واقع هستند نقش ساختار سیناپسی را ایفا می‌کنند. این‌ها هستند که باید تغییر کنند تا فرآیند یادگیری اتفاق افتد حال این‌که این تغییر بر چه قاعده‌ای استوار باشد بستگی به الگوریتم آموزش یا تعلیم دارد. وزن آخرین ورودی یعنی  $\theta -$  را با  $b$  نشان می‌دهند و آن را بایاس<sup>۸</sup> می‌نامند. سیگنال‌های عصبی  $x_1$  و  $x_2$  و ... و  $x_{n+1}$  به ترتیب در  $w_1$  و  $w_2$  و ... و  $w_{n+1}$  ضرب شده و مجموع این حاصل ضرب‌ها که در شکل آن را با  $\text{net}$  نشان می‌دهیم، وارد نرون می‌شوند. در این مرحله نرون همانند یک تابع روی  $\text{net}$  عمل می‌کند و حاصل کار خود (برد تابع) را از طریق آکسون‌های طرف راست به نرون‌های دیگر می‌فرستد. این تابع را تابع فعال‌سازی<sup>۹</sup> می‌نامند.

البته خاطر نشان می‌کنیم که اگر نرون به عنوان تابع فرض شود این تابع از  $\mathbb{R}^n$  به  $\mathbb{R}$  تعریف می‌گردد. یعنی سیگنال خروجی از نرون در تمام آکسون‌های طرف راست یکسان می‌باشد که روی شکل با  $O$  نمایش داده‌ایم. همان‌طور که قبلاً اشاره شد نرون‌ها می‌توانند حسی، ارتباطی و یا حرکتی باشند. اطلاعات از طریق نرون‌های حسی دریافت می‌شود و از طریق نرون‌های ارتباطی انتقال می‌یابد و بالاخره نتیجه کار شبکه عصبی و یا واکنش شبکه عصبی به اطلاعات دریافت شده از طریق نرون‌های حرکتی به خارج منتقل می‌گردد (شکل (۳.۱)).

نرون‌های حسی را لایه ورودی، نرون‌های حرکتی را لایه خروجی و نرون‌های ارتباطی را لایه پنهان نیز می‌گویند. لایه مخفی می‌تواند از چندین لایه تشکیل گردد البته در شکل (۳.۱) لایه مخفی را تک لایه در نظر گرفته شده است.

نحوه‌ی اتصالات نرون‌ها می‌تواند به گونه‌ای باشد که شبکه تک لایه یا چندلایه بوجود آورد. شبکه‌های

<sup>۸</sup>Bias

<sup>۹</sup>Activation Function

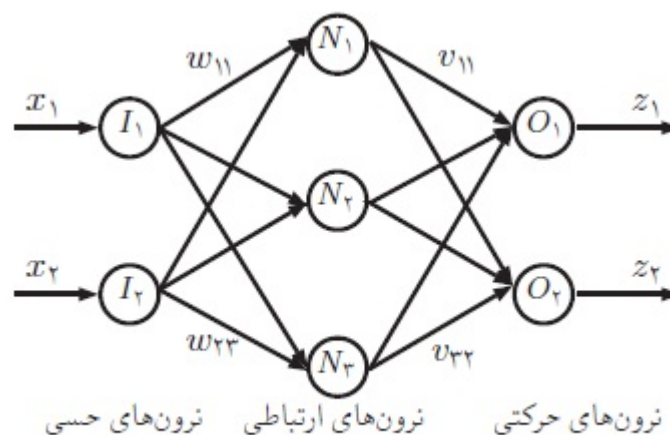
چند لایه از لایه ورودی که الگوهای ورودی به آن اعمال می‌شود و لایه خروجی که خروجی شبکه را تعیین می‌کند و لایه پنهان که خود متشکل از چند لایه است و وظیفه‌ی آن ارتباط دادن لایه‌ی ورودی و لایه‌ی خروجی است، تشکیل می‌شود.

مشخصه‌های اصلی یک شبکه عصبی که در طراحی آن باید مدنظر قرار گیرند عبارت‌اند از :

**معماری شبکه عصبی:** نحوه اتصالات بین نرون‌ها، تعداد آن‌ها و تعداد لایه‌های تشکیل دهنده بخش نرون‌های ارتباطی را معماری شبکه عصبی می‌گوییم.

**تابع‌های فعال‌سازی:** این‌که چه تابعی روی نرون قرار بگیرد تا بر ورودی‌های نرون اثر کرده و خروجی نرون را تولید کند تابع فعال‌سازی آن نرون گویند. در جدول (۱.۱) تعدادی از مهم‌ترین توابع فعال‌سازی نشان داده شده است.

**الگوریتم آموزش:** روشی که وزن‌های روی آکسون‌ها بر اساس آن تغییر می‌کنند ( $w_{ij}$  ها و  $v_{jk}$  ها در شکل (۳.۱)) تا خروجی شبکه عصبی به حالت مطلوب درآید را الگوریتم آموزش می‌گوییم.



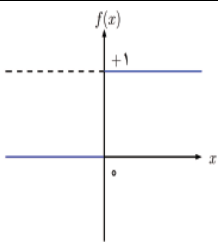
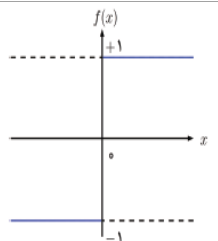
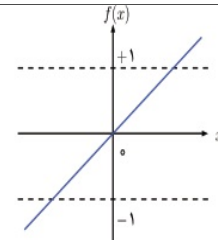
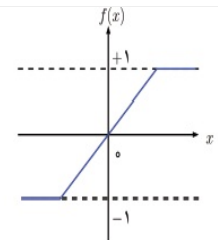
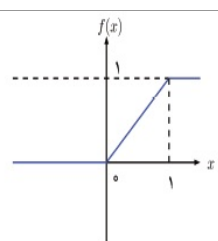
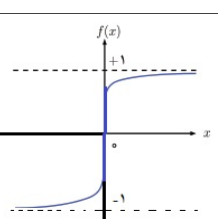
شکل ۳.۱: ساختار یک نرون با لایه ورودی، لایه پنهان و لایه خروجی.

با توجه به مطالب قبل می‌توان برای شبکه‌های عصبی مصنوعی خصوصیات عمومی زیر را نسبت داد:

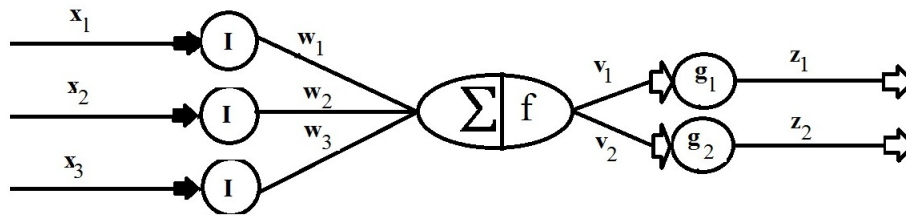
۱. پردازش اطلاعات در عناصر بسیار ساده به نام نرون (سلول عصبی) انجام می‌گیرد.
۲. سیگنال‌های عصبی بین سلول‌های عصبی از طریق اتصالات بین آن‌ها مبادله می‌شوند.
۳. به هر خط اتصال (آکسون) یک وزن نسبت داده می‌شود که در شبکه‌های عصبی معمولی این وزن‌ها در سیگنال‌های منتقل شده ضرب می‌گردند.
۴. هر نرون دارای یک تابع فعال‌سازی می‌باشد که این تابع در اکثر موارد غیرخطی است.



جدول ۱.۱: توابع فعال‌سازی

| ردیف | نام                        | تعریف تابع                                                                                | شکل تابع                                                                            |
|------|----------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| ۱    | آستانه‌ای دو مقداره        | $f(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$                            |    |
| ۲    | آستانه‌ای دو مقداره متقارن | $f(x) = \begin{cases} -1 & x < 0, \\ 1 & x \geq 0. \end{cases}$                           |    |
| ۳    | خطی                        | $f(x) = x$                                                                                |   |
| ۴    | آستانه‌ای خطی متقارن       | $f(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$ |  |
| ۵    | آستانه‌ای خطی              | $f(x) = \begin{cases} 0, & x < 0, \\ x, & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$    |  |
| ۶    | تانژانت هایپربولیک         | $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$                                                |  |

مثال ۱.۱.۱.۱. می‌خواهیم خروجی شبکه زیر را تعیین کنیم:  
 شبکه‌های عصبی مربوط به شکل (۴.۱) را به صورت ریاضی بیان می‌کنیم. این شبکه در ۳ لایه ارائه



شکل ۴.۱: شبکه عصبی مثال (۱.۱.۱.۱)

شده است. نرون‌های حسی سیگنال‌های دریافتی را عیناً انتقال می‌دهند این سیگنال‌ها در وزن‌های موجود در روی آکسون‌ها ضرب می‌شوند و در ورودی نرون ارتباطی با هم جمع می‌شوند.

$$w_1x_1 + w_2x_2 + w_3x_3,$$

اگر خروجی این لایه را با  $y$  نشان دهیم خواهیم داشت:

$$y = f(w_1x_1 + w_2x_2 + w_3x_3) = f\left(\sum_{i=1}^3 w_i x_i\right),$$

این خروجی هنگام عبور از آکسون‌های لایه بعدی باید در وزن‌های موجود در آن ضرب شود و بعد از گذر از لایه‌ی خارجی سیگنال‌های زیر به عنوان سیگنال حرکتی از شبکه عصبی خارج می‌گردند.

$$z_1 = g_1(v_1y), \quad z_2 = g_2(v_2y).$$

## توابع فعال سازی

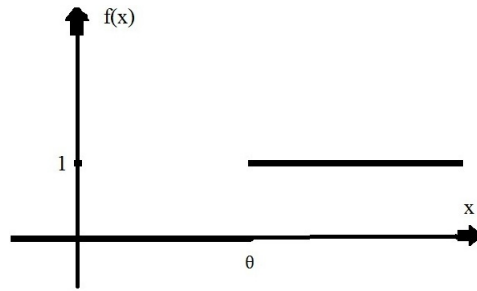
چند نمونه از این توابع را در ذیل بیان می‌کنیم

تابع پله‌ای: این تابع با حد آستانه‌ای  $\theta$  است و به صورت زیر می‌باشد:

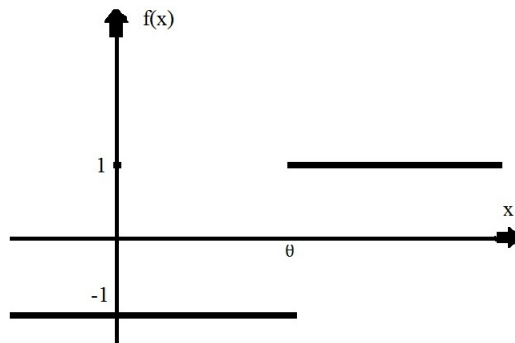
$$f(x) = \begin{cases} 1 & x \geq \theta, \\ 0 & x < \theta. \end{cases}$$

تابع پله‌ای دوقطبی: این تابع با حد آستانه‌ای  $\theta$  است و به صورت زیر تعریف می‌شود

$$f(x) = \begin{cases} 1 & x \geq \theta, \\ -1 & x < \theta. \end{cases}$$



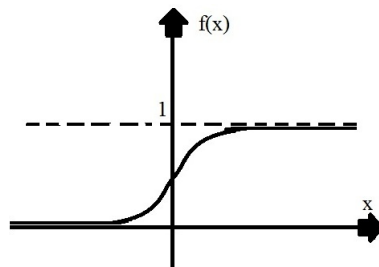
شکل ۵.۱: تابع پله‌ای



شکل ۶.۱: تابع پله‌ای دوقطبی

تابع سیگموئید: تابع سیگموئید<sup>۱</sup>، تابعی یکنوا، کراندار و مشتق پذیر است و می‌توان مشتق تابع را به کمک خود تابع بیان کرد و به شکل زیر است:

$$f(x) = \frac{1}{1 + e^{-x}}, \quad f'(x) = f(x)(1 - f(x)).$$



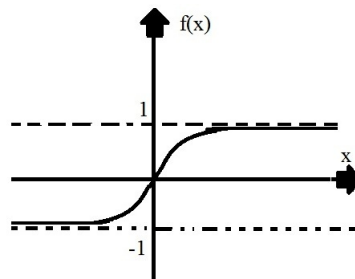
شکل ۷.۱: تابع سیگموئید

---

<sup>۱</sup> Sigmoid

تابع سیگموئید قطبی: این تابع همان خواص تابع سیگموئید را دارد و به صورت زیر تعریف می شود:

$$g(x) = 2f(x) - 1 = 2 \left( \frac{1}{1 + e^{-x}} \right) - 1 = \frac{1 - e^{-x}}{1 + e^{-x}}, \quad g'(x) = \frac{1}{4}(1 + g(x))(1 - g(x)).$$



شکل ۸.۱: تابع سیگموئید قطبی

تابع همانی: این تابع بیشتر برای نرون های ورودی مورد استفاده قرار می گیرد و به صورت زیر بیان می شود:

$$f(x) = x.$$

چند تابع فعال سازی مهم دیگر را در جدول (۱.۱) آورده ایم.

### شبکه های عصبی بازگشتی

در شبکه های عصبی بازگشتی<sup>۱۱</sup>، حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون های همان لایه و یا لایه قبل وجود دارد. چنین شبکه ای حافظه ای را نگه می دارد و حالت بعد نه تنها به سیگنال های ورودی بلکه به حالات قبل شبکه نیز وابسته است. این شبکه های عصبی معمولاً در مدل ریاضی به صورت معادلات دیفرانسیل مرتبه اول ظاهر می شوند.

### ۳.۱.۱ شبکه های عصبی پیش خورده

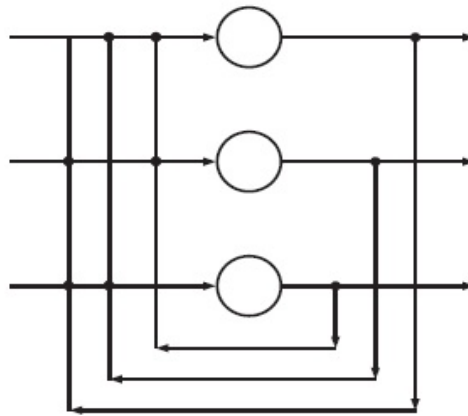
شبکه های عصبی مصنوعی<sup>۱۲</sup> یک گرایش هیجان انگیز از هوش مصنوعی است که شبیه به فرآیند یادگیری مغز انسان به منظور استخراج الگوهایی از داده های تاریخی عمل می کند [۳۲]. برای سال های بسیاری این تکنولوژی بطور موفقیت آمیز برای گونه های گسترده ای از کاربردهای دنیای واقعی بکار گرفته شده است [۳۰]. پرسپترون<sup>۱۳</sup> برای اولین بار توسط روزنبلات<sup>۱۴</sup> در سال ۱۹۶۲ معرفی شد [۲۴] (شکل

<sup>۱۱</sup>Feedback (Recurrent) Neural Network

<sup>۱۲</sup>Artificial Neural Networks

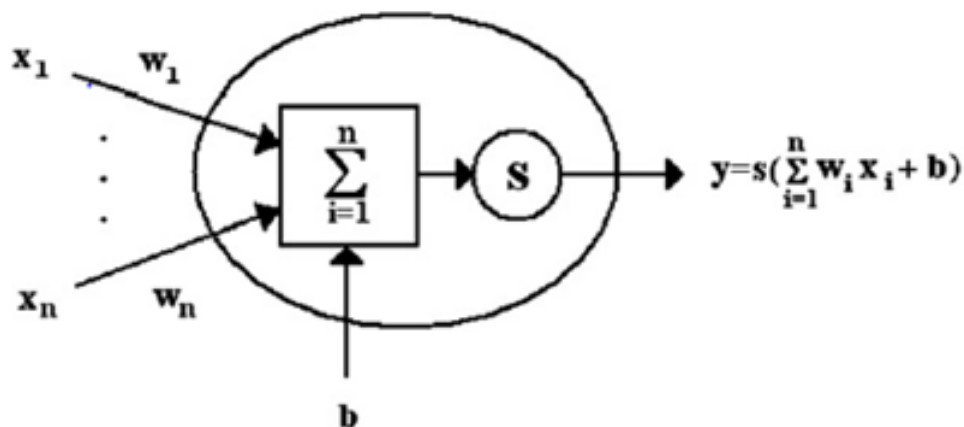
<sup>۱۳</sup>Perceptron

<sup>۱۴</sup>Rosenblatt



شکل ۹.۱: شبکه‌های عصبی دو لایه بازگشتی.

(۱۰.۱) رابینید). پرسپترون ساده به یک یاددهنده احتیاج دارد که به شبکه بگوید که چه خروجی مطلوبی باید داشته باشد. آن‌ها شبکه‌های تحت نظارت هستند. در یک شبکه بدون نظارت، شبکه سازگار با پاسخ به ورودی آن است. شکل (۱۱.۱) مثالی از پرسپترون سه لایه را نشان می‌دهد. پرسپترون چندلایه با لایه‌های بیشتر از سه‌تا، از لایه‌های پنهان شده استفاده می‌کنند [۴۰، ۳۴].



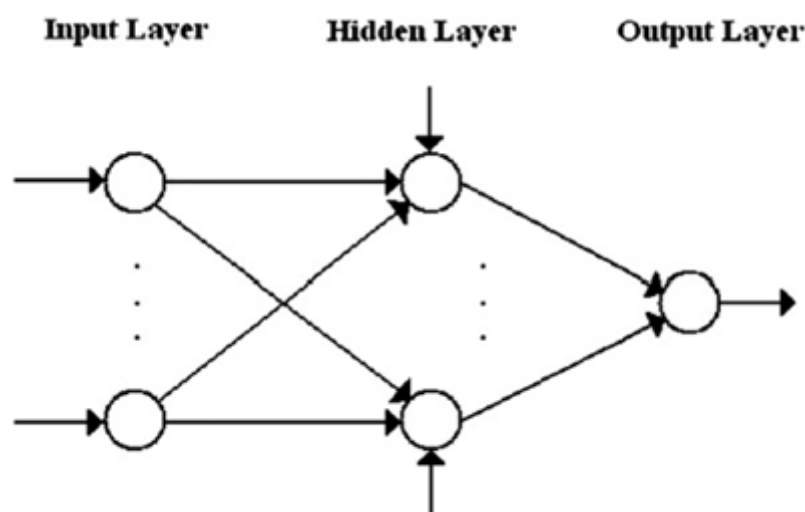
شکل ۱۰.۱: ساختار شبکه عصبی مصنوعی (پرسپترون)

در پرسپترون‌های چندلایه واحدهای ورودی به واحدهای خروجی توسط یک نگاشت غیرخطی بهم متصل می‌شوند [۱۹]. مهمترین کاربرد پرسپترون چندلایه توانایی آنها در تقریب تابع است. از قضیه وجودی کولوموگروف<sup>۱۵</sup> می‌دانیم که یک پرسپترون سه لایه با  $n(2n+1)$  گره می‌تواند از هر تابع پیوسته با  $n$  متغیر محاسبه شود [۲۳، ۳۳]. دقت تقریب به تعداد نرون‌ها<sup>۱۶</sup> در لایه‌های پنهان شده بستگی دارد و به تعداد لایه‌های پنهان شده بستگی ندارد [۸]. شبکه‌های عصبی چندلایه پیش‌خورد<sup>۱۷</sup> قابل یادگیری هستند. در برخی موارد مساله‌ی آموزشی پیش‌خورد بدحالت است [۲۰].

<sup>۱۵</sup>Existence Colomogorov Theorem

<sup>۱۶</sup>Neuron

<sup>۱۷</sup>FeedForward



شکل ۱۱.۱: پرسپترون چندلایه

شبکه‌های عصبی یک شکل از سیستم کامپیوتر چندپردازنده با عناصر ساده پردازش، ارتباطات درجه بالاتر، مقیاس پیام ساده<sup>۱۸</sup> و عکس العمل تطبیقی بین عناصر هستند. یک نرون بیولوژیکی<sup>۱۹</sup> ممکن است  $10^4$  ورودی متفاوت داشته باشد و ممکن است خروجی آن را به بسیاری از نرون‌های دیگر ارسال کند. شبکه عصبی مصنوعی هرچیزی را که با روش‌های محاسبات سنتی نمی‌توان انجام داد، می‌تواند انجام دهد. آنها می‌توانند برخی از کارهای محاسباتی که بسیار مشکل است را انجام دهند. از شبکه‌های عصبی پیش‌خورد بصورت یک مدل (تابع با متغیرهای زیاد) از داده‌های آموزشی آنها (داده‌های ورودی) به تنهایی استفاده می‌کنیم.

#### ۴.۱.۱ تاریخچه تکامل شبکه عصبی

در سال ۱۹۴۳ یک نرولوژیست به نام وارن مک کلاچ<sup>۲۰</sup> به همراه یک متخصص آمار به نام والتر پیتز<sup>۲۱</sup> اولین مدل ریاضی نرون را ارائه دادند [۴۱]. این نرون ساده یک عنصر محاسبه‌گر بود که عناصری به نام ورودی را در مقادیر ثابتی به نام وزن ضرب می‌کرد و آنها را از یک عملگر خطی عبور می‌داد و حاصل خروجی نرون را تشکیل می‌داد. در این مدل وزن‌ها مقادیر ثابتی بودند. در سال ۱۹۴۹ دونالد هب<sup>۲۲</sup> [۱۵] کتابی درباره نحوه یادگیری در مغز انسان نوشت. بر این مبنا قانون یادگیری برای نرون را ارائه داد که به قانون هب معروف است و بعدها در تعلیم شبکه‌های عصبی از آن استفاده شد. در سال ۱۹۵۴ فارلی و کلارک<sup>۲۳</sup> [۲] مدلی از نرون‌ها را به‌طور تصادفی به هم وصل کردند و قانون هب را پیاده‌سازی

<sup>۱۸</sup>Simple Message Scale

<sup>۱۹</sup> Biological Neural Network

<sup>۲۰</sup>Warren McCulloch

<sup>۲۱</sup>Walter Pitts

<sup>۲۲</sup>Donald Hebb

<sup>۲۳</sup>Farley and Clark

نمودند و نشان دادند که می‌توان دو الگوی ورودی را با این قانون تعلیم از هم تشخیص داد. با پیشرفت علم کامپیوتر اولین مدل از نرون عصبی مصنوعی توسط ناتانیال روچستر<sup>۲۴</sup> محقق شرکت IBM در سال ۱۹۵۸ با استفاده از کامپیوتر شبیه‌سازی شد. روزنبلات [۱۷، ۱۸] در سال ۱۹۵۸ نرون ساده مک‌کلاچ پیتر را اصلاح کرد و به آن قابلیت یادگیری و سازگاری را اضافه کرد و این نرون را پرسپترون نامید. قانون تعلیم پرسپترون<sup>۲۵</sup> اولین قانون رسمی برای شبکه‌های عصبی است.

برنارد ویدرو<sup>۲۶</sup> و مارسین هاف<sup>۲۷</sup> و همکارانشان یک مدل سه سطحی از نرون‌ها را ابداع کردند که آدلاین<sup>۲۸</sup> نام‌گذاری شد و مختصر شده حروف "عصر خطی تطبیقی"<sup>۲۹</sup> بود [۹، ۱۰، ۱۱، ۱۲]. و برای آدلاین یک قانون تعلیم بر مبنای مشتق‌گیری بیان نمودند. این قانون تعلیم به  $\alpha - LMS$  معروف است. از متصل کردن چند آدلاین به یکدیگر مدل‌های<sup>۳۰</sup> ساخته شد. ویدرو و هاف از شبکه عصبی مدل‌های برای حذف اکو از خطوط تلفن استفاده نمودند.

سال ۱۹۶۹ آغاز یک دوره رکود برای شبکه عصبی بود. پاپرت و مینسکی<sup>۳۱</sup> [۲۶] کتابی به نام پرسپترون نوشتند و نشان دادند که اگرچه گیت‌های OR و AND را می‌توان با استفاده از پرسپترون پیاده‌سازی کرد ولی گیت XOR قابل پیاده‌سازی نمی‌باشد و لذا نمی‌توان از شبکه عصبی برای پردازش اطلاعات استفاده کرد. الگوریتم پس انتشار<sup>۳۲</sup> پال ورباس<sup>۳۳</sup> [۲۹] در سال ۱۹۷۴ ارائه شد و بعدها توسط رامل هارت<sup>۳۴</sup> [۱۶] به‌طور مستقل کشف گردید. این الگوریتم از زمان پیدایش به‌طور گسترده به‌عنوان یک الگوریتم آموزش در شبکه‌های عصبی پیش‌خورد<sup>۳۵</sup> مورد استفاده قرار گرفته است. سال ۱۹۸۲ تا ۱۹۸۶ را می‌توان تولد دوباره شبکه عصبی دانست. در این سال‌ها دو اتفاق مهم در این زمینه رخ داد. اول ارائه شبکه عصبی بازگشتی و مفهوم تابع انرژی توسط جان هاپفیلد<sup>۳۶</sup> [۲۱] بود که در سال ۱۹۸۲ منتشر شد. دومین اتفاق، حل مسأله فروشنده دوره‌گرد (TSP) توسط شبکه هاپفیلد بود [۲۲]. هاپفیلد نشان داد که از این شبکه بازگشتی می‌توان برای حل مسائل بهینه‌سازی استفاده کرد و افق جدیدی را در شبکه‌های عصبی مصنوعی گشود.

---

<sup>۲۴</sup>Nathanial Rochester

<sup>۲۵</sup>Perceptron

<sup>۲۶</sup>Bernard Widrow

<sup>۲۷</sup>Marcian Hoff

<sup>۲۸</sup>ADELIN

<sup>۲۹</sup>Adaptive Linear Elements

<sup>۳۰</sup>MADALINE

<sup>۳۱</sup>Papert and Minsky

<sup>۳۲</sup>Back Propagation

<sup>۳۳</sup>Paul Werbos

<sup>۳۴</sup>Rumelhart

<sup>۳۵</sup>Feed Forward

<sup>۳۶</sup>John Hopfield

### ۵.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی

طیف گسترده‌ای از مسائل مهم در رشته‌های علوم پایه و مهندسی از جمله کنترل بهینه، طراحی ساختمان، پردازش تصویر، تقریب توابع، تحلیل رگرسیون و غیره قابل تبدیل به مسائل بهینه‌سازی خطی و غیرخطی هستند. در دهه‌های گذشته الگوریتم‌های بهینه‌سازی عددی فراوانی برای حل مسائل بهینه‌سازی خطی و غیرخطی ارائه شده‌اند. از آن‌جا که زمان مورد نیاز برای حل مسائل بهینه‌سازی و به‌خصوص مسائل بهینه‌سازی غیرخطی بسیار وابسته به بعد و ساختار مسأله است بنابراین الگوریتم‌های عددی کارایی کمی از خود نشان می‌دهند. یک رهیافت امیدوارکننده برای حل مسائل بهینه‌سازی با ابعاد بالا استفاده از شبکه‌های عصبی مصنوعی می‌باشد.

در سال ۱۹۸۶ تانک و هاپفیلد [۱۴] اولین شبکه عصبی را برای حل مسائل برنامه‌ریزی خطی معرفی کردند و نشان دادند که وضعیت شبکه در هر تکرار به‌گونه‌ای تغییر می‌کند که تابع انرژی متناظر با آن به‌طور یکنواخت کاهش می‌یابد تا جایی که به نقطه مینیمم خود می‌رسد و این نقطه مینیمم متناظر با نقطه تعادل شبکه عصبی می‌باشد. آن‌ها این شبکه عصبی را توسط یک مدار الکتریکی پیاده سازی کردند. هم‌چنین از این شبکه برای حل مسأله فروشنده دوره‌گرد با ۳۰ شهر استفاده نمودند. این شبکه دارای نقص‌هایی بود به‌خصوص این‌که نقطه تعادل شبکه در شرایط  $KKT^{۳۷}$  صدق نمی‌کرد و لذا جواب مطلوبی از مسأله حاصل نمی‌شد. با این وجود کارهای هاپفیلد انگیزه بسیار خوبی را برای محققین به‌وجود آورد تا در این زمینه فعالیت کنند. کندی و چا [۲۵] با افزودن یک پارامتر جریمه متناهی به شبکه هاپفیلد آن را توسعه دادند و از آن برای حل مسأله برنامه‌ریزی غیرخطی محدب استفاده نمودند. با پارامترهای جریمه این شبکه در یافتن نقطه بهینه دقیق ناتوان است به‌خصوص اگر پارامتر جریمه بزرگ باشد این شبکه به‌سختی عمل می‌کند.

برای جلوگیری از بکار بردن پارامتر جریمه یک شبکه عصبی سوئیچ-خازن<sup>۳۹</sup> توسط رودریگز-وازکز<sup>۴۰</sup> و همکاران [۷] معرفی شد. هم‌چنین ما و شانبلات<sup>۴۱</sup> [۱۳] یک شبکه عصبی دو فازی ارائه کردند که فاز اول شبکه مشابه شبکه کندی و چا بود و در فاز دوم آن مسیر شبکه به جواب دقیق مسأله همگرا می‌شد بنابراین این روش جواب‌های دقیق‌تری نسبت به شبکه کندی و چا ارائه می‌داد. مشکل این شبکه در این بود که پایداری فاز دوم شبکه بستگی به انتخاب یک مقدار بزرگ از پارامتر جریمه داشت و لذا اگرچه تأثیر پارامتر جریمه در این روش کاهش یافته بود و جواب‌های حاصل دقیق‌تر بودند اما هنوز مستقل از پارامتر نبود.

ژانگ و کنستانتینیدس<sup>۴۲</sup> [۳۹] بر مبنای روش لاگرانژ یک شبکه عصبی ارائه کردند که کاملاً مستقل از پارامتر جریمه و قادر به حل مسائل غیرخطی بود. نقطه تعادل این شبکه عصبی در شرایط بهینگی

<sup>۳۷</sup>Karush-Kuhn-Tucker

<sup>۳۸</sup>Kennedy and Chua

<sup>۳۹</sup>Switched-Capacitor

<sup>۴۰</sup>Rodriguez-Vazquez

<sup>۴۱</sup>Maa and Shanblatt

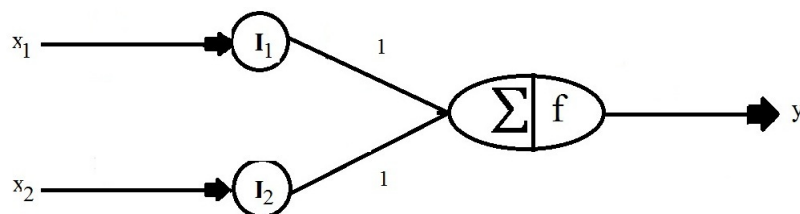
<sup>۴۲</sup>Zhang and Constantinides



مرتبه اول و دوم صدق می‌کرد و همچنین شبکه حاصل همگرا بود. در سال ۱۹۹۳ بوزردوم و پتیسن<sup>۴۳</sup> [۵] شبکه‌ای را بر مبنای گرادیان و روش تصویر و مستقل از پارامتر جریمه ابداع کردند که تنها قادر به حل مسائل درجه دوم با متغیرهای کران‌دار بود. این روش در عمل کارایی بود اما نمی‌توانست مسائل کلی برنامه‌ریزی خطی و درجه دوم را حل کند.

زیا<sup>۴۴</sup> و همکاران [۴۳، ۴۲، ۴۴، ۴۵] چندین مدل را برای حل مسائل بهینه‌سازی غیرخطی با قیود خطی و غیرخطی ارائه کردند. همچنین چندین مدل شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی توسط عفتی و همکاران [۲۷، ۳، ۲۸، ۳۶، ۳۸، ۳۷] ارائه شده است.

### ۶.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND



شکل ۱۲.۱: نرون مک کلاچ برای تابع AND

در این نوع شبکه عصبی هرگاه دومقدار دریافتی درست باشند تابع AND پاسخ درست و هرگاه یکی از دومقدار دریافتی نادرست باشند تابع پاسخ نادرست را برمی‌گرداند.

$$y = f(1 \times x_1 + 1 \times x_2) = f(x_1 + x_2) = \begin{cases} 1 & x_1 + x_2 \geq 2, \\ 0 & x_1 + x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

### ۷.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع OR

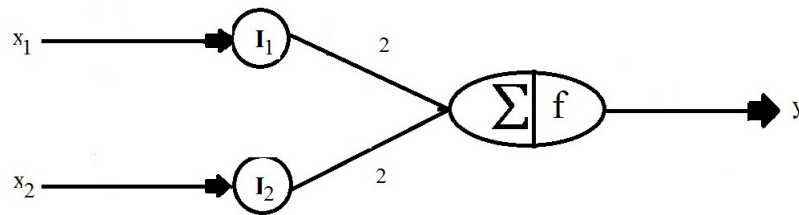
در این نوع شبکه عصبی هرگاه یکی از دومقدار دریافتی درست باشند تابع OR پاسخ درست و هرگاه هر دو دومقدار دریافتی نادرست باشند تابع پاسخ نادرست را برمی‌گرداند.

$$y = f(2 \times x_1 + 2 \times x_2) = f(2x_1 + 2x_2) = \begin{cases} 1 & 2x_1 + 2x_2 \geq 2, \\ 0 & 2x_1 + 2x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

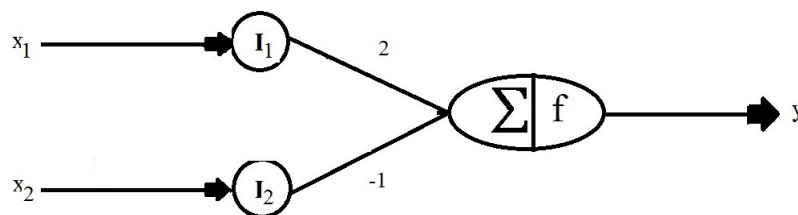
<sup>۴۳</sup>Bouzerdoun and Pattison

<sup>۴۴</sup>xia



شکل ۱۳.۱: نرون مک کلاچ برای تابع OR

### ۸.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND NOT



شکل ۱۴.۱: نرون مک کلاچ برای تابع AND NOT

در این نوع شبکه عصبی هرگاه مولفه‌ی اول درست و مولفه‌ی دوم نادرست باشند تابع AND NOT پاسخ درست و در غیر اینصورت پاسخ نادرست می‌دهد.

$$y = f(2 \times x_1 + (-1) \times x_2) = f(2x_1 - x_2) = \begin{cases} 1 & 2x_1 - x_2 \geq 2, \\ 0 & 2x_1 - x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

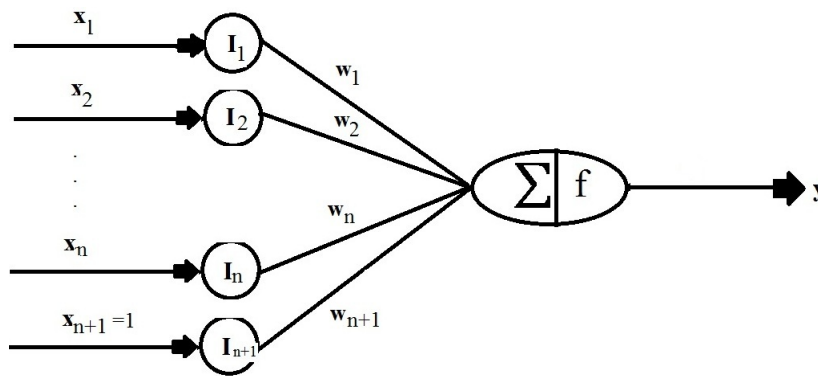
### ۹.۱.۱ شبکه عصبی هب

در ادامه الگوریتم آموزش را برای یک شبکه با  $n$  ورودی و یک خروجی متشکل از دولایه ارائه داده شده است.

#### الگوریتم هب

گام اول: مقدار دهی اولیه به تمام وزن‌ها  $w_i = 0$ ,  $i = 1, 2, \dots, n+1$  همان بایاس است که ورودی ۱ دارد.

گام دوم: مولفه‌های بردار ورودی  $x_i$  ( $i = 1, 2, \dots, n+1$ ) را روی واحدهای ورودی تنظیم <sup>۴۵</sup> می‌کنیم.



شکل ۱۵.۱: شبکه عصبی هب

گام سوم: خروجی مورد انتظار مربوط به بردار ورودی گام قبل را روی واحد خروجی تنظیم می‌کنیم.

گام چهارم: وزن‌های موجود روی یال‌ها<sup>۴۶</sup> را به صورت زیر بروز می‌کنیم.

$$w_i(\text{new}) = w_i(\text{old}) + x_i y, \quad i = 1, 2, \dots, n+1.$$

مثال ۲.۱.۱. الگوریتم هب را برای تابع AND با ورودی‌های دوقطبی و خروجی دوقطبی تکرار می‌کنیم. بنابراین داده‌های آموزشی به صورت زیر خواهند بود:

$$\left\{ \begin{array}{ll} (x_1, x_2, 1) & y \\ (1, 1, 1) & 1 \\ (1, -1, 1) & -1 \\ (-1, 1, 1) & -1 \\ (-1, -1, 1) & -1 \end{array} \right.$$

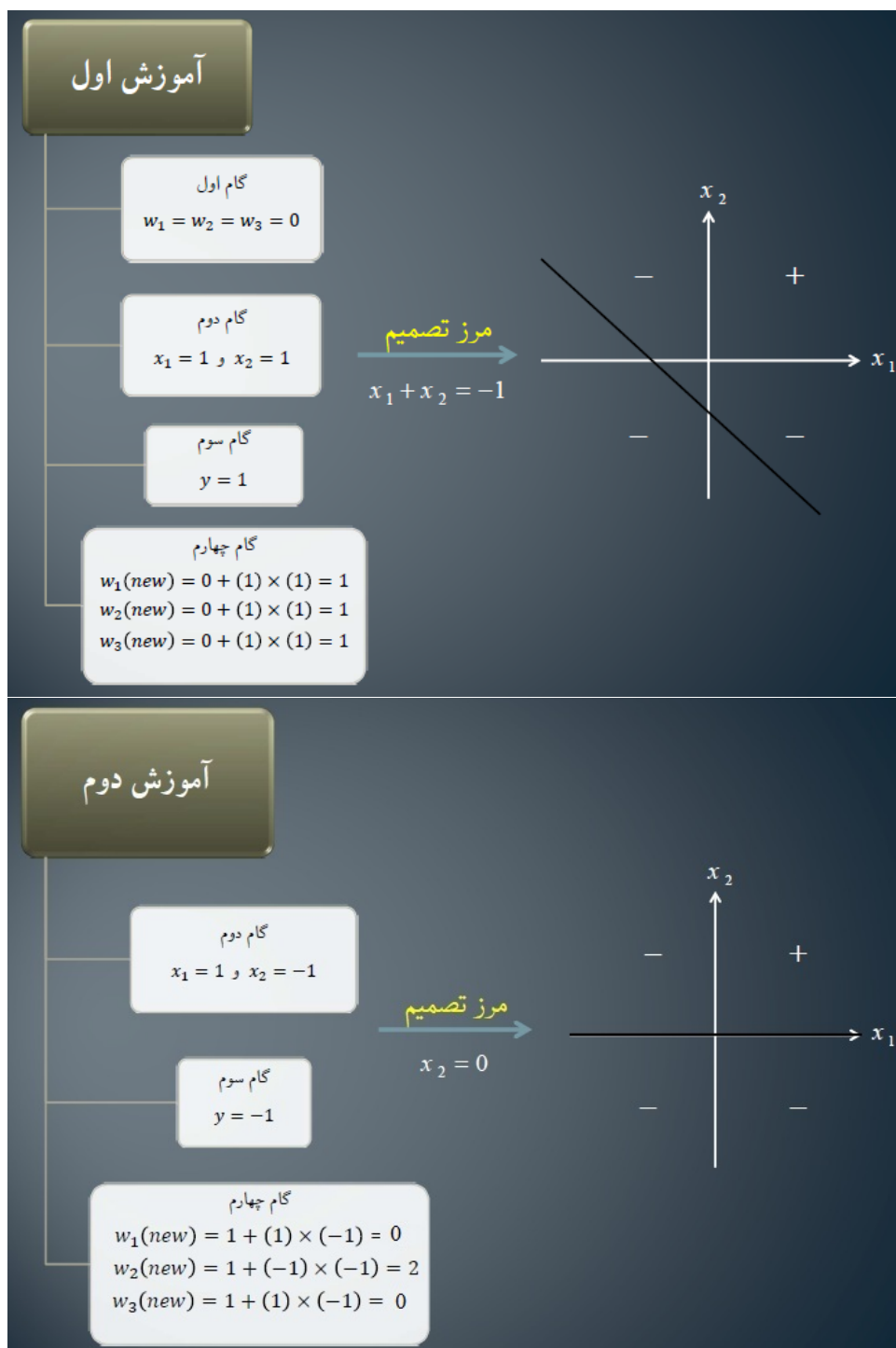
$$w_1 = w_2 = w_3 = 0$$

که در شکل‌های (۱۶.۱) و (۱۷.۱) روند الگوریتم نمایش داده شده است.

### ۱۰.۱.۱ شبکه عصبی آدالین

این شبکه چند ورودی دارد که یکی از ورودی‌ها همواره مقدار یک را به شبکه ارسال می‌کند (بایاس). نرون آدالین سیگنال‌ها را پس از ترکیب با وزن‌های روی یال‌ها جمع می‌کند. تابع فعال سازی روی نرون آدالین همان تابع همانی است، بنابراین همین مجموع به عنوان خروجی به بیرون فرستاده می‌شود و اختلاف بین مقدار خروجی و مقدار مورد انتظار مبنای الگوریتم آموزش محسوب می‌شود.

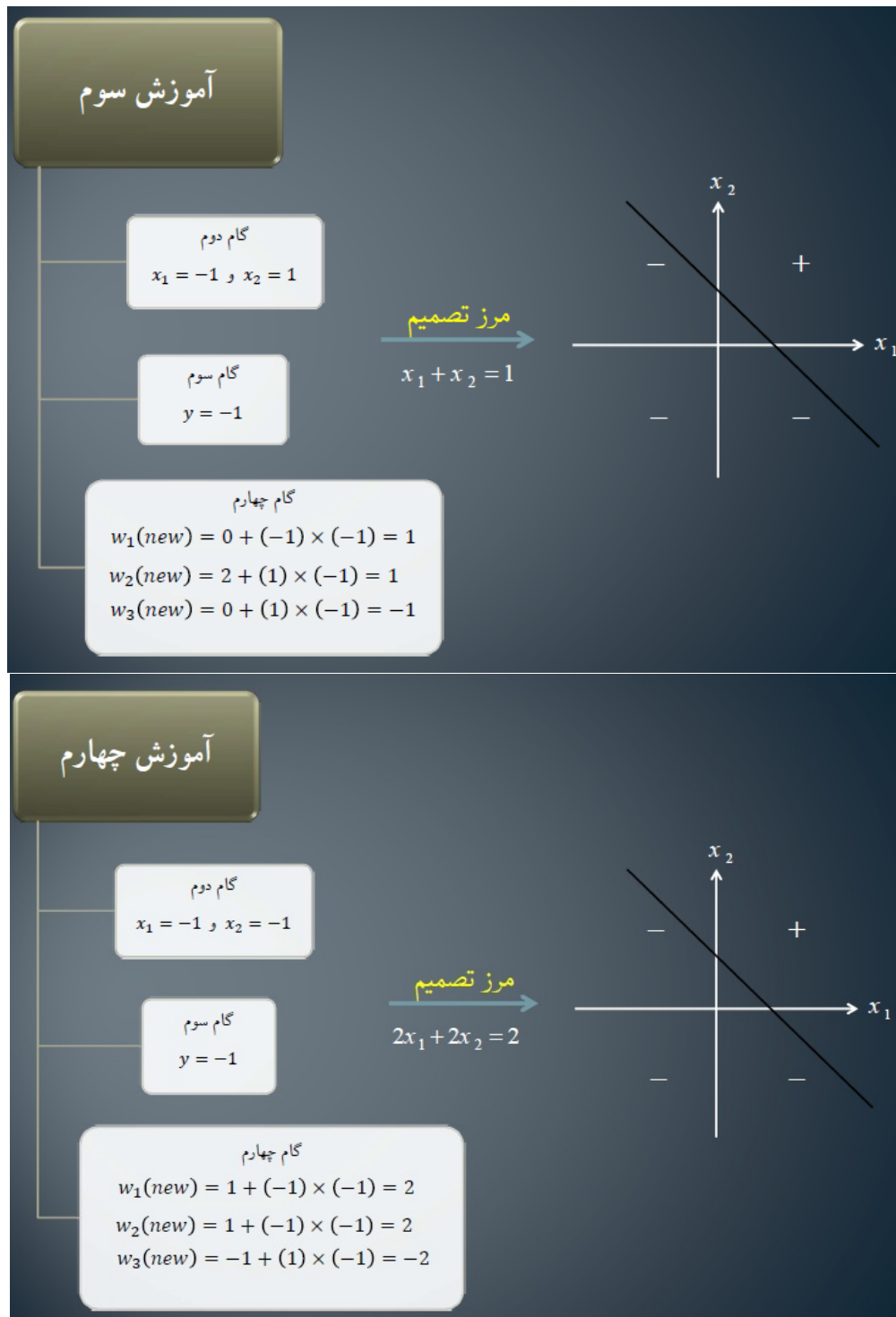
<sup>۴۶</sup>Edge



شکل ۱۶.۱: الگوریتم هب برای مثال (۲۰.۱.۱)

## الگوریتم آدالین

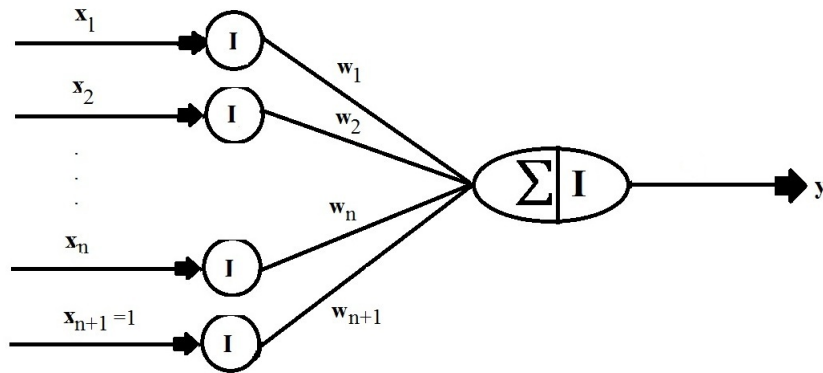
گام اول: به وزن‌ها مقادیر تصادفی بین صفر و یک نسبت دهید و ضریب آموزشی  $\alpha$  را عددی در بازه  $[0, 1]$  در نظر بگیرید.



شکل ۱۷.۱: الگوریتم هب برای مثال (۲.۱.۱)

گام دوم: تا زمانی که شرط توقف نادرست باشد گام‌های سوم تا هفتم را تکرار کنید.

گام سوم: برای هر جفت داده  $((x_1, x_2, \dots, x_n), y)$  گام‌های چهارم تا ششم را اجرا کنید.



شکل ۱۸.۱: شبکه عصبی آدالین

گام چهارم: مولفه‌های بردار ورودی  $x_i$  ( $i = 1, 2, \dots, n+1$ ) را روی واحدهای ورودی تنظیم می‌کنیم.  
گام پنجم:

$$net = \sum_{i=1}^n w_i x_i,$$

$$I(net) = net.$$

گام ششم: تنظیم وزن‌ها

$$w_i(new) = w_i(old) + \alpha(y - net), \quad i = 1, 2, \dots, n+1$$

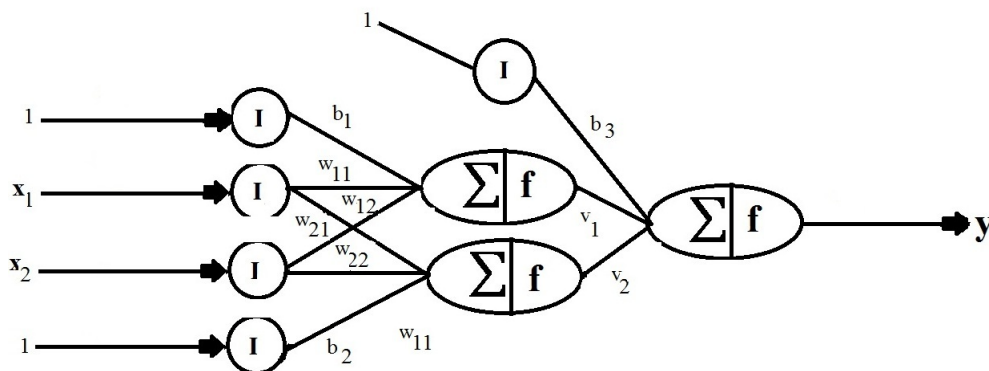
گام هفتم: اگر بیشترین تغییرات انجام گرفته برای وزن‌ها در گام دوم کمتر از یک تلورانس معین باشد، متوقف شوید در غیر اینصورت ادامه دهید.

### ۱۱.۱.۱ شبکه عصبی مادالین

یک مادالین، ترکیبی از چند آدالین می‌باشد که در یک شبکه چندلایه مرتب شده‌اند. در اینجا ماروی یک مادالین با یک لایه مخفی (مربک از دو آدالین) و یک خروجی متمرکز می‌شویم. در زیر یک مادالین با دو ورودی و یک خروجی نشان داده شده است. این شبکه شامل دو آدالین با ورودی‌های  $x_1, x_2$  و خروجی‌های  $z_1, z_2$  می‌باشد. و توابع بکاربرده شده در آدالین‌ها توابع آستانه‌ای دو مقداری متقارن می‌باشد و خروجی  $y$  تابعی غیر خطی از بردار ورودی  $(x_1, x_2)$  می‌باشد.

### الگوریتم مادالین

در این الگوریتم فقط وزن‌های لایه اول تنظیم می‌شوند و وزن‌های لایه بعدی در مدت آموزش ثابت می‌مانند. وزن‌ها به گونه‌ای هستند که اگر  $z_1$  یا  $z_2$  یا هر دو سیگنال ۱ بفرستند،  $y$  سیگنال ۱ و اگر هر دو



شکل ۱۹.۱: شبکه عصبی مادالین

سیگنال ۱ - بفرستند  $y$  سیگنال ۱- را تولید می‌کند بنابراین داریم:

$$v_1 = \frac{1}{4}, \quad v_2 = \frac{1}{4}, \quad b_3 = \frac{1}{4}$$

گام اول:  $v_1 = v_2 = b_3 = \frac{1}{4}$  و به وزن‌ها مقادیر تصادفی بین صفر و یک نسبت دهید و ضریب آموزشی  $\alpha$  را عددی در بازه  $[0/1, 1]$  در نظر بگیرید.

گام دوم: تا زمانی که شرط توقف نادرست باشد گام‌های سوم تا نهم را تکرار کنید.

گام سوم: برای هر جفت داده  $((x_1, x_2), y)$  گام‌های چهارم تا هشتم را اجرا کنید.

گام چهارم: بردار ورودی  $(x_1, x_2)$  را روی واحدهای ورودی تنظیم می‌کنیم.

گام پنجم: ورودی‌های مربوط به آدالین‌ها را محاسبه کنید.

$$net_1 = b_1 + x_1 w_{11} + x_2 w_{21},$$

$$net_2 = b_2 + x_1 w_{12} + x_2 w_{22}.$$

گام ششم: محاسبه خروجی هر آدالین

$$z_1 = f(net_1),$$

$$z_2 = f(net_2).$$

گام هفتم: تعیین خروجی شبکه

$$net = b_3 + z_1 v_1 + z_2 v_2, \quad f(net)$$

گام هشتم: اگر  $y = f(net)$  وزن‌ها تغییر نخواهند کرد در غیر این صورت:

$$\text{if } y = 1, \quad w_{ij}(new) = w_{ij}(old) + \alpha(1 - net_j)x_j, \quad b_j(new) = b_j(old) + \alpha(1 - net_j),$$

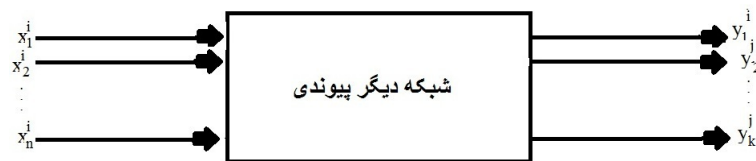
$$\text{if } y = -1, \quad w_{ij}(new) = w_{ij}(old) + \alpha(-1 - net_j)x_j, \quad b_j(new) = b_j(old) + \alpha(-1 - net_j).$$

گام نهم: اگر تغییرات در وزن‌ها متوقف شود یا به یک سطح قابل قبول برسم، متوقف شوید.

### ۱۲.۱.۱ شبکه‌های حافظه پیوندی

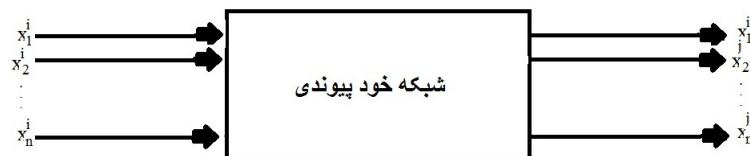
شبکه‌های حافظه پیوندی شبکه‌هایی تک لایه می‌باشند و وزن‌های موجود در شبکه، عمل ذخیره سازی الگوهای پیوندی را بر عهده دارند. این نوع شبکه عصبی بطور کلی به سه دسته تقسیم میشوند:

- شبکه‌های عصبی دیگر پیوندی:
- شبکه‌های عصبی دیگر پیوندی  $p$  بردار  $(x^1, x^2, \dots, x^n)$  در فضای  $n$  بعدی را به  $p$  بردار خروجی  $k$  بعدی  $(y^1, y^2, \dots, y^k)$  می‌برد.



شکل ۲۰.۱: شبکه عصبی دیگر پیوندی

- شبکه‌های عصبی خود پیوندی:
- شبکه‌های عصبی خود پیوندی حالت خاصی از شبکه‌های عصبی دیگر پیوندی است که هر بردار را با خودش مرتبط می‌سازد.



شکل ۲۱.۱: شبکه عصبی خود پیوندی

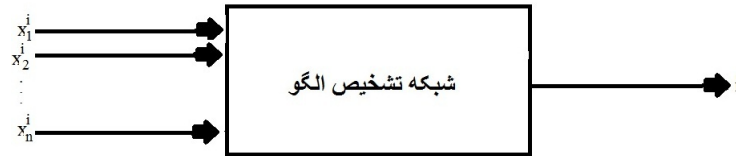
- شبکه‌های تشخیص الگو:
- در شبکه تشخیص الگو هر بردار  $x^i$  از فضای  $n$  با اسکالر  $i$  پیوند داده می‌شود.

### ۱۳.۱.۱ شبکه‌های عصبی هاپفیلد

شبکه عصبی مصنوعی هاپفیلد

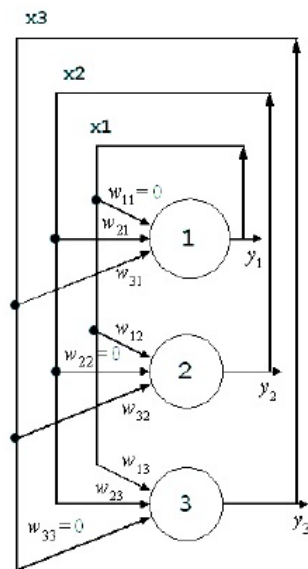
شبکه هاپفیلد یک شبکه بازگشتی است. به این مفهوم که در آن خروجی هر نرون به ورودی تمام نرون‌های دیگر متصل است. پس از اعمال ورودی اولیه به شبکه، خروجی آن محاسبه می‌شود و در قدم بعد به





شکل ۲۲.۱: شبکه عصبی تشخیص الگو

عنوان ورودی به شبکه اعمال می‌شود. این فرآیند آنقدر تکرار می‌شود تا خروجی شبکه به یک نقطه ثابت همگرا شود. در اینجا شبکه به عنوان یک سیستم دینامیکی که وضعیتش در هر تکرار تغییر می‌کند، عمل می‌کند. و همگرایی شبکه یعنی رسیدن پاسخ شبکه به جوابی که بر اثر تکرار تغییر نکند، در این صورت می‌توان گفت که شبکه به حالت پایدار یا تعادل خود رسیده است. شکل (۲۳.۱) یک شبکه هاپفیلد با سه نرون را نشان می‌دهد [۴]. شبکه هاپفیلد به دو نوع گسسته و



شکل ۲۳.۱: ساختار شبکه هاپفیلد با سه نرون

پیوسته تقسیم می‌شود. شبکه هاپفیلد گسسته یک شبکه بازگشتی شامل نرون‌هایی است که تابع فعالیت آن‌ها یک تابع گسسته است. در واقع هر نرون در شبکه گسسته یک مدل مکلوث<sup>۴۷</sup> و پیتس<sup>۴۸</sup> است [۴۱]. اما در شبکه هاپفیلد پیوسته که بحث اصلی این بخش است، هر نرون دارای یک تابع فعالیت به طور پیوسته مشتق پذیر است.

<sup>۴۷</sup>Warren McCulloch

<sup>۴۸</sup>Walter Pitts

## نحوه عملکرد شبکه هاپفیلد پیوسته

در سال ۱۹۸۲ [۲۱] هاپفیلد بر مبنای ایده تابع انرژی یا تابع لیاپانوف برای سیستم‌های دینامیکی شبکه عصبی بازگشتی را ارائه داد که به نام خود او شبکه عصبی هاپفیلد نام گذاری و اولین بار از آن برای شناسایی الگوهای دارای نویز استفاده شد. همانطور که در ادامه بحث، خواهد آمد، هر شبکه هاپفیلد پیوسته یک سیستم دینامیکی است که دارای یک تابع انرژی می‌باشد. هدف از اجرای یک شبکه هاپفیلد، تشخیص الگوهای ناشناس است. با در نظر گرفتن منحنی تابع انرژی شبکه درمی‌یابیم که منحنی دارای نقاط مینیمم است که می‌تواند به عنوان مکان ذخیره سازی الگوها یا جواب‌های مطلوب شبکه مورد استفاده قرار گیرد.

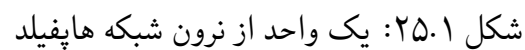
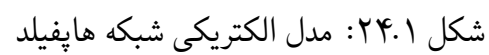
این حفره‌ها یا نقاط مینیمم محل نقاط جذب شبکه نامیده می‌شوند. نقاط جذب به بیان ریاضی همان نقاط ساکن پایدار مجانبی سیستم دینامیکی هستند و در شبکه هاپفیلد، نقاط جذب نقاطی هستند که شبکه به آن‌ها همگرا می‌شود.

توصیف هندسی مساله همگرایی را می‌توان به این صورت نیز بیان کرد که الگوهای ورودی که وضعیت اولیه شبکه را مشخص می‌کند، به سمت کوچکترین ناحیه‌ای که شامل یکی از حفره‌های جذب (نقاط ساکن پایدار مجانبی) می‌باشد، نزدیک می‌شود و هنگامی که به نزدیکی آن رسید از هر طرف که حرکت کند نقطه جذب آن را به سمت خود می‌کشاند و نهایتاً در آن ساکن می‌ماند. بخش بعد ما را در بهتر درک کردن ساختار شبکه عصبی پیوسته هاپفیلد و پایداری آن یاری می‌دهد.

## ساختار شبکه هاپفیلد پیوسته

هاپفیلد این شبکه را بر مبنای یک مدار الکتریکی ارائه می‌دهد. مدار الکتریکی در شکل (۲۴.۱) را در نظر بگیرید [۳۱]. هر آمپلی فایر ولتاژ ورودی  $u_i$  را به ولتاژ  $x_i = f(u_i)$  تبدیل می‌کند. تابع  $f$  را یک تابع مشتق پذیر و به طور یکنواخت صعودی مانند تابع سیگموئید  $\tanh(\beta u_i)$  قرار می‌دهیم. هر مقاومت  $R_{ij}$  ولتاژ خروجی  $x_j$  را به آمپلی فایر  $i$ -ام هدایت می‌کند. جریان  $I_i$  به عنوان اولین ورودی یا بایاس برای آمپلی فایر  $i$  در نظر گرفته می‌شود. مدار فوق به عنوان شبکه هاپفیلد شناخته می‌شود.

در این شبکه ولتاژ اولیه  $x(0)$  که همان ورودی ناشناس است به عنوان ورودی اولیه به شبکه اعمال می‌شود. ولتاژ خروجی  $x = [x_1, \dots, x_n]$  اندازه گیری می‌شود و سپس  $x_1$  مجدداً به عنوان ورودی وارد شبکه می‌شود. این فرآیند آنقدر تکرار می‌شود تا ولتاژ خروجی شبکه تغییر نکند یا در همسایگی نقطه‌ای ثابت قرار گیرد. در این صورت گوئیم شبکه به حالت پایدار خود رسیده است. لذا این فرآیند را می‌توان به صورت یک معادله بازگشتی در نظر گرفت. پس شبکه هاپفیلد یک شکل بازگشتی است و اثبات پایداری شبکه از روی ورودی معادله دینامیکی مدار آن به شکل زیر نتیجه می‌شود. شکل (۲۵.۱) یک واحد از این شبکه را به عنوان یک نرون شبکه نمایش می‌دهد. معادله دینامیکی که تغییر وضعیت  $x_i$  را توضیح


$$\sum_{j=1}^n \frac{1}{R_{ij}} x_j - \sum_{j=1}^n \frac{1}{R_{ij}} u_i - \frac{u_i}{R_i} - C_i \frac{du_i}{dt} + I_i = 0, \quad (1.1)$$

$$C_i \frac{du_i}{dt} = \sum_{j=1}^n \frac{1}{R_{ij}}(x_j - u_i) - \frac{u_i}{R_i} + I_i, \quad (2.1)$$

$$C_i \frac{du_i}{dt} = -\alpha_i \times u_i + \sum_{j=1}^n x_{ij} f(u_j) + I_i. \quad (3.1)$$

که در آن  $w_{ij} = \frac{1}{R_{ij}}$ ،  $\alpha_i = \sum_{j=1}^n \frac{1}{R_{ij}} + \frac{1}{R_i}$  و  $x_i = f(u_i)$  معادله (۳.۱) به صورت برداری زیر نوشته می‌شود.

$$C \frac{du}{dt} = -\alpha \times u + w \times x + \theta, \quad (4.1)$$

که در آن

$$C = (C_1, \dots, C_n), \quad \alpha = (\alpha_1, \dots, \alpha_n), \quad x = F(u) = (f(u_1), \dots, f(u_n))^T,$$

$$\theta = [I_1, \dots, I_n]^T, \quad w = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nn} \end{bmatrix}.$$

نقطه ساکن معادله دینامیکی (۴.۱) با قرار دادن  $\frac{du}{dt} = 0$  حاصل می‌شود. در این صورت:

$$\alpha \times u = w \times x + \theta.$$

معادله (۴.۱) سیستم دینامیکی شبکه عصبی هاپفیلد می‌باشد. هاپفیلد ثابت کرد که شرط لازم برای پایداری این شبکه این است که  $w$  یک ماتریس متقارن باشد. علاوه بر این هاپفیلد نشان داد که نقطه تعادل شبکه، نقطه مینیمم محلی تابع لیاپانوف زیر است.

$$E(x) = -\frac{1}{2} x^T w x - x^T \theta + \sum_{j=1}^n \alpha_j \int_0^1 f^{-1}(z) dz. \quad (5.1)$$

بردار  $x = [x_1, \dots, x_n]$  بردار خروجی شبکه است که هر درایه آن  $x_i = f(u_i)$ . همانطور که قبلاً ذکر شد  $f(u_j) = \tanh(\beta u_j) = \frac{e^{\beta u_j} - e^{-\beta u_j}}{e^{\beta u_j} + e^{-\beta u_j}}$ ، هرگاه  $\beta \rightarrow \pm\infty$  در این صورت  $f(u_j) \rightarrow \pm 1$  و لذا  $E(x)$  تابعی کراندار خواهد بود.

به شبکه هاپفیلد به مشابه سیستم گرادیان نیز می‌توان نگاه کرد.

$$\frac{du}{dt} = -\rho \nabla E(x), \quad (6.1)$$

که در آن  $\rho = \text{diag}(\frac{1}{C_1}, \dots, \frac{1}{C_n})$ . نقطه ساکن سیستم (۶.۱) مینیمم محلی برای  $E(x)$  است زیرا از  $\frac{du}{dt} = 0$  نتیجه می‌شود که  $\nabla E(x) = 0$  است. حال ثابت می‌کنیم که نقطه ساکن سیستم (۶.۱) به طور مجانبی پایدار است. برای این منظور باید در همسایگی نقطه ساکن سیستم، یک تابع لیاپانوف به دست آوریم.

**قضیه ۳.۱.۱.** نقطه ساکن سیستم  $\dot{x}(t) = f(x(t))$  پایدار است اگر تابع لیاپانوف متناظر با این سیستم وجود داشته باشد.

□

برهان. به مرجع [۶] مراجعه کنید.

**قضیه ۴.۱.۱.** نقطه ساکن سیستم دینامیکی (۶.۱) به طور مجانبی پایدار است.

برهان. [۳۱] برای هر نقطه ساکن  $x^*$  همسایگی مانند  $\Sigma$  وجود دارد که تابع  $V(x) = E(x) - E(x^*)$  روی این همسایگی به طور پیوسته مشتق پذیر است بدلیل اینکه  $E(x)$  به طور پیوسته مشتق پذیر است. همچنین  $V(x)$  اکیدا مثبت است و  $V(x^*) = 0$ .  
همچنین

$$\frac{dV}{dt} = \frac{dE}{dt} = \nabla E(x) \dot{x}(t) = \sum_{j=1}^n \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial t} = -C \sum_{j=1}^n \frac{du_j}{dt} \cdot \frac{dx_j}{dt} = -C \sum_{j=1}^n \frac{dx_j}{du_j} \cdot \left(\frac{du_j}{dt}\right)^2 \leq 0.$$

$\frac{dx_j}{du_j}$  همواره مثبت است چون  $x_i = f(u_i)$  است و همانطور که قبلا ذکر شد  $f$  یک تابع غیر نزولی است. لذا نتیجه می‌گیریم که  $V(x)$  یک تابع لیاپانوف است، بنا به قضیه (۳.۱.۱) نقطه ساکن سیستم (۶.۱) به طور مجانبی پایدار است.  $\square$

به طور کلی اگر بخواهیم شبکه هاپفیلد را از دید نرم افزاری بیان کنیم، می‌توان اینطور گفت که معادله دینامیکی (۶.۱) وضعیت شبکه نسبت به زمان را مشخص می‌کند. متناظر با این سیستم دینامیکی تابع انرژی  $E(x)$  به گونه‌ای تعریف می‌شود که یک تابع لیاپانوف باشد لذا این اطمینان حاصل می‌شود که شبکه همواره همگرا باشد. همچنین نقاط ساکن  $\frac{du}{dt} = -\rho \nabla E(x)$  نقاط مینیم برای تابع نزولی  $E(x)$  می‌باشد که همان الگوهای موردنظر مساله است. با وارد کردن یک الگوی ناشناس اولیه، شبکه آنقدر تکرار می‌شود تا در همسایگی یکی از این الگوهای مشخص قرار بگیرد به همین دلیل گفته می‌شود که شبکه هاپفیلد قابلیت ذخیره سازی الگو را دارد.

## ۲.۱ مقدمه‌ای بر نظریه کنترل

در نظریه کنترل نشان داده شده است که رفتار دینامیکی بسیاری از سیستم‌ها را می‌توان به شکل معادلات یا نامعادلات ریاضی و بخصوص به شکل معادلات دیفرانسیل بیان نمود. البته ساختن یک مدل ریاضی برای یک سیستم ممکن است بسیار پیچیده باشد این مبحث در جای خود از اهمیت زیادی برخوردار است. ما فرض خواهیم کرد که کسی معادلات ریاضی سیستم را که خواص دینامیکی سیستم مورد بررسی را مشخص می‌کند تهیه کرده باشد. پس برای کنترل یک سیستم لازم است کنترل‌ها مناسب انتخاب شوند چنانکه معادلات ریاضی مربوط به آن سیستم برآورده شوند. البته محاسبه جواب‌های معادلات بخصوص معادلات دیفرانسیل غیرخطی عموماً پیچیده است. به‌عنوان مثال ما می‌خواهیم مساله مینیم‌کردن مصرف سوخت راکت را مورد بررسی قرار دهیم. معادله حاکم بر حرکت راکت عبارت است از:

$$m \frac{dV}{dt} = mF + F_{ext},$$

که در آن  $m$  جرم راکت،  $V$  سرعت آن،  $F$  نیروی پرتاب که بوسیله راکت تولید می‌شود و  $F_{ext}$  نیروهای خارجی که از محیط به راکت وارد می‌شود. همچنین داریم [۴۶]:

$$|F| = \frac{c}{m} \beta,$$

که در آن  $c$  سرعت نسبی اگزوز و  $\beta = -\frac{dm}{dt}$  نرخ مصرف سوخت است. اگر  $\varphi$  زاویه<sup>۴۹</sup> واقعی پرتاب (زاویه بین محور راکت و افق) باشد، آنگاه معادلات حرکت عبارتند از:

$$\begin{aligned}\frac{dv_1}{dt} &= \frac{c\beta}{m} \cos \varphi, \\ \frac{dv_2}{dt} &= \frac{c\beta}{m} \sin \varphi - mg,\end{aligned}$$

که در آن  $V = [v_1, v_2]^T$  و  $F_{ext} = [0, -mg]^T$  اختیار شده است. مساله مصرف مینیم سوخت عبارت است از انتخاب کنترل های  $\beta$  و  $\varphi$  که راکت را از نقطه اولیه به ارتفاع موردنظر  $\bar{y}$  برساند به قسمی که مقدار سوخت مصرفی مینیم شود. سوخت مصرف شده عبارت است از:

$$I = \int_0^T \beta dt$$

که در آن  $T$  زمان رسیدن به  $\bar{y}$  است.

### ۳.۱ شکل کلی مساله کنترل بهینه

در اینجا شکل کلی از يك مساله کنترل بهینه<sup>۵۰</sup> معرفی می‌کنیم که در فصول بعدی با استفاده از روش شبکه عصبی به حل آن می‌پردازیم.

فرض کنید سیستم دینامیکی توسط  $n$  متغیر حالت  $x_1, x_2, \dots, x_n$  مشخص شود، اکنون بردار حالت  $x$  را به صورت زیر در نظر می‌گیریم:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

مولفه‌های  $x$  را می‌توان به عنوان محورهای مختصات  $n$  بعدی بنام فضای حالت<sup>۵۱</sup> در نظر گرفت. بردار کنترل  $u$  را به عنوان برداری از  $m$  متغیر کنترل  $u_1, u_2, \dots, u_m$  به صورت زیر در نظر می‌گیریم:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}.$$

<sup>۴۹</sup> Angle

<sup>۵۰</sup> Optimal Control Problem

<sup>۵۱</sup> State Space

فرض کنید  $(x, u)^T$  در يك دستگاه معادلات دیفرانسیل مرتبه اول<sup>۵۲</sup> بصورت زیر صدق می‌کند:

$$\begin{cases} \dot{x} = g(t, x, u) & t_a \leq t \leq t_b, \\ x(t_a) = x_a & x(t_b) = x_b, \end{cases} \quad (۷.۱)$$

که در آن  $x(t_a)$  حالت اولیه و  $x(t_b)$  حالت نهایی است. مساله کنترل بهینه عبارت است از پیدا کردن  $(x, u)^T$  که در (۷.۱) صدق کرده و تابعی زیر را ماکزیم یا مینیم کند:

$$I = \int_{t_a}^{t_b} f_0(t, x(t), u(t)) dt.$$

در اینجا  $f_0$  و  $g_i$  ( $i = 1, 2, \dots, n$ )، توابعی پیوسته از  $t$ ،  $x$  و  $u$  هستند.

## ۴.۱ شرط لازم مسائل کنترل بهینه

مساله کلی مینیم‌سازی

$$J = \int_0^T f_0(t, x(t), u(t)) dt.$$

را با شرط معادله دیفرانسیل

$$\begin{cases} \dot{x} = f(t, x, u) \end{cases} \quad (۸.۱)$$

و شرط مرزی  $x(0) = a$  در نظر می‌گیریم. با ضریب لاگرانژ،  $\lambda$ ، تابعی افزوده زیر را تشکیل می‌دهیم،

$$J^* = \int_0^T [f_0(t, x(t), u(t)) + \lambda(f(t, x(t), u(t)) - \dot{x})] dt.$$

تابع انتگرال  $F = f_0 + \lambda(f - \dot{x})$  تابعی از دو متغیر  $x$  و  $u$  است و در نتیجه دو معادله‌ی اوایلر زیر را خواهیم داشت:

$$\frac{\partial F}{\partial x} - \frac{d}{dt} \left( \frac{\partial F}{\partial \dot{x}} \right) = 0 \longrightarrow \frac{\partial f_0}{\partial x} + \lambda \frac{\partial f}{\partial x} + \dot{\lambda} = 0. \quad (۹.۱)$$

$$\frac{\partial F}{\partial u} - \frac{d}{dt} \left( \frac{\partial F}{\partial \dot{u}} \right) = 0 \longrightarrow \frac{\partial f_0}{\partial u} + \lambda \frac{\partial f}{\partial u} = 0. \quad (۱۰.۱)$$

با تعریف هامیلتونی به صورت

$$\mathcal{H} = f_0 + \lambda f \quad (۱۱.۱)$$

<sup>۵۲</sup>First-Order Differential Equations

معادله‌های (۹.۱) و (۱۰.۱) را می‌توان چنین نوشت:

$$\dot{\lambda} = -\frac{\partial \mathcal{H}}{\partial x} \quad (12.1)$$

$$\circ = \frac{\partial \mathcal{H}}{\partial u} \quad (13.1)$$

مسیر بهینه باید در این معادلات و  $\dot{x} + x = u$  صدق کنند. چون دو معادله دیفرانسیل مرتبه اول داریم. برای انتگرال‌گیری و کامل کردن جواب به دو شرط نیاز داریم. یکی  $x(\circ) = a$  و دیگری  $x(T) = b$ ، و اگر  $x$  در  $t = T$  معین نباشد از شرط انتهای آزاد  $\frac{\partial F}{\partial \dot{x}} = \circ$  یعنی  $\lambda = \circ$  در  $t = T$  استفاده می‌کنیم.

## ۵.۱ اصل حداقلیابی پونتریاگین

در دستگاه‌های واقعی اغلب محدودیت‌هایی روی کنترل‌ها و وضعیت‌های قابل قبول وجود دارد. می‌خواهیم ببینیم این محدودیت‌ها چگونه بر شرایط لازم که با فرض غیر مقید بودن مقادیر کنترل‌های قابل قبول به دست آمده‌اند، تاثیر می‌گذارد.

طبق تعریف، اگر برای کنترل‌های قابل قبولی که به اندازه‌ی کافی نزدیک  $u^*$  هستند، داشته باشیم

$$J(u) - J(u^*) = \Delta J \geq \circ$$

آنگاه کنترل  $u^*$  باعث می‌شود تابعی  $J$  کمینه نسبی داشته باشد.

اگر  $u = u^* + \delta u$ ، نمو  $J$  با رابطه‌ی زیر بیان می‌شود:

$$\Delta J(u^*, \delta u) = \delta J(u^*, \delta u) + \text{جملات مراتب بالاتر}$$

$\delta J$  بر حسب  $\delta u$  خطی است و جملات مراتب بالاتر زمانی که  $\delta u$  به سمت صفر میل می‌کند، همگی به سمت صفر میل می‌نمایند.

اگر کنترل نامقید باشد، با استفاده از این که  $\delta J$  بر حسب  $\delta u$  خطی است و  $\delta u$  می‌تواند به صورت اختیاری تغییر کند می‌توان نشان داد شرایط لازم برای اینکه  $u^*$  منحنی کنترل اکسترمم باشد این است که تغییرات  $\delta J(u^*, \delta u)$  برای تمام  $\delta u$  های قابل قبول که نورم کوچکی دارند صفر شود. اما چون فرض شده است که کنترل‌های قابل قبول محدود باشند،  $\delta u$ ، فقط زمانی اختیاری است که کنترل اکسترمم در تمام بازه‌ی زمانی  $[t_0, t_f]$  بین دو حد تعیین شده باشد.

با بررسی تمام تغییرات قابل قبول با  $\|\delta u\|$  به اندازه‌ی کافی کوچک، بطوریکه علامت  $\Delta J$  توسط  $\delta u$  تعیین می‌شود، نتیجه می‌شود شرط لازم برای اینکه  $u^*$  تابعی  $J$  را حداقل کند، این است که:

$$\delta J(u^*, \delta u) \geq \circ$$

حال باید ببینیم این تغییرات چگونه بر شرایط لازمی که با فرض غیر مقید بودن مقادیر کنترل‌های قابل قبول به دست آمده‌اند، تاثیر می‌گذارد.



نمو  $J$  به صورت زیر است:

$$\begin{aligned}\Delta J(u^*, \delta u) = & \left[ \frac{\partial h}{\partial x}(x^*(t_f), t_f) - p^*(t_f) \right]^T \delta x_f \\ & + \left[ \mathcal{H}(x^*(t_f), u^*(t_f), p^*(t_f), t_f) + \frac{\partial h}{\partial t}(x^*(t_f), t_f) \right] \delta t_f \\ & + \int_{t_0}^{t_f} \left\{ \left[ \dot{p}^*(t) + \frac{\partial \mathcal{H}}{\partial x}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta x(t) \right. \\ & + \left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) \\ & \left. + \left[ \frac{\partial \mathcal{H}}{\partial p}(x^*(t), u^*(t), p^*(t), t) - \dot{x}^*(t) \right]^T \delta p(t) \right\} dt + \text{جملات مراتب بالاتر}\end{aligned}$$

اگر معادلات حالت برقرار باشند و  $p^*(t)$  چنان انتخاب شود که ضرایب  $\delta x(t)$ ، در انتگرال صفر شوند و معادلات شرایط حدی برقرار باشند داریم:

$$\Delta J(u^*, \delta u) = \int_{t_0}^{t_f} \left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) dt + \text{جملات مراتب بالاتر}$$

از طرفی می‌دانیم که

$$\begin{aligned}\left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) = & \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \\ & - \mathcal{H}(x^*(t), u^*(t), p^*(t), t)\end{aligned}$$

در نتیجه داریم:

$$\begin{aligned}\Delta J(u^*, \delta u) = & \int_{t_0}^{t_f} \left\{ \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \right. \\ & \left. - \mathcal{H}(x^*(t), u^*(t), p^*(t), t) \right\} dt + \text{جملات مراتب بالاتر}\end{aligned}$$

بنابراین برای اینکه  $u^*$  کنترل حداقل کننده‌ای باشد، باید به ازای هر مقدار قابل قبول  $\delta u$  داشته باشیم

$$\int_{t_0}^{t_f} \left\{ \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) - \mathcal{H}(x^*(t), u^*(t), p^*(t), t) \right\} dt \geq 0$$

که ایجاب می‌کند برای هر  $\delta u$  قابل قبول داشته باشیم

$$\mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \geq \mathcal{H}(x^*(t), u^*(t), p^*(t), t)$$

بنابراین نتیجه می‌گیریم که شرط لازم برای اینکه  $u^*$  تابعی  $J$  را حداقل کند این است که به ازای تمام کنترل‌های قابل قبول داشته باشیم:

$$\mathcal{H}(x^*(t), u^*(t), p^*(t), t) \leq \mathcal{H}(x^*(t), u(t), p^*(t), t)$$

این معادله نمایانگر این مطلب است که کنترل بهینه باید هامیلتونین را حداقل کند و به اصل حداقل یابی پونتریاگین<sup>۵۳</sup> معروف می‌باشد.

در اینجا بطور خلاصه به بیان تعاریف و مباحثی که در این پایان نامه از آن استفاده شده است می‌پردازیم.

<sup>۵۳</sup>Pontryagin's Minimum Principle

**تعریف ۱.۵.۱.** فرض کنید تابع  $f$  برای  $0 \leq t < \infty$  تعریف شده و  $s$  یک متغیر حقیقی<sup>۵۴</sup> باشد. تابع  $F$  را به صورت زیر تعریف می‌کنیم:

$$F(s) = \int_0^{\infty} e^{-st} f(t) dt.$$

حوزه تعریف  $F$  مجموعه  $s$ هایی است که به ازای آن‌ها انتگرال فوق همگراست، یعنی  $s$ هایی که به ازای آن‌ها حد زیر موجود است:

$$\lim_{b \rightarrow \infty} \int_0^b e^{-st} f(t) dt.$$

تابع  $F(s)$  را تبدیل لاپلاس<sup>۵۵</sup> تابع  $f(t)$  می‌نامیم و آن را با  $L\{f(t)\}$  نشان می‌دهیم، پس:

$$L\{f(t)\} = F(s) = \int_0^{\infty} e^{-st} f(t) dt.$$

**تعریف ۲.۵.۱.** اگر در تعریف بالا  $-\infty \leq t < \infty$  باشد آنگاه  $F(s)$  را تبدیل لاپلاس دو طرفه<sup>۵۶</sup> تابع  $f(t)$  می‌نامیم و آن را بصورت زیر نشان می‌دهیم:

$$B\{f(t)\} = F(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt.$$

$$\forall x \in \mathbb{R}^n, \quad x^T A x \geq 0.$$

## ۶.۱ معادلات دیفرانسیل تاخیری

یک نوع خاص از معادله دیفرانسیل، معادله دیفرانسیل تاخیری است که معادله دیفرانسیل با شناسه عقب افتاده نیز نامیده می‌شود و در برخی مسائل عددی مطرح می‌شود. مدل‌های جمعیت و مسائل اختلاط دارای این شکل خاص هستند، که در آن‌ها مقدار  $\dot{x}(t)$  به تابع  $x$  در مقادیر قبلی بستگی دارد. برای مثال فرض کنید:

$$\dot{x}(t) = f(x(t-1)).$$

اگر مقدار  $x$  را در  $t-1$  بشناسیم، با استفاده از معادله دیفرانسیل قادر به محاسبه  $\dot{x}(t)$  خواهیم بود. برای انتگرال‌گیری معادله دیفرانسیل با شروع در  $t=0$ ، به مقدار  $x(t)$  با شروع در  $t=-1$  نیاز خواهیم داشت. بنابراین مقادیر  $x(t)$  بر روی بازه  $[-1, 0]$  باید به عنوان مقادیر اولیه برای ما فراهم باشند. یک مثال از یک مساله خاص و خوش تعریف از این نوع مساله زیر است:

$$\begin{cases} \dot{x}(t) = x(t-1), & t \geq 0, \\ x(t) = t^2, & -1 \leq t \leq 0. \end{cases}$$

<sup>۵۴</sup>Real Variable

<sup>۵۵</sup>Laplace Transformation

<sup>۵۶</sup>Two-sided Laplace

دومین معادله این مساله مقادیر اولیه موردنیاز را برای  $x(t)$  ارائه می‌دهد. اگر  $t$  به بازه  $[0, 1]$  محدود باشد آنگاه  $t - 1$  در  $[-1, 0]$  خواهد بود. بنابراین:

$$\begin{cases} \dot{x}(t) = x(t-1) = (t-1)^2, & 0 \leq t \leq 1, \\ x(0) = 0. \end{cases}$$

جواب این معادله به راحتی با انتگرال‌گیری و فراهم نمودن یک مقدار ثابت مناسب انتگرال‌گیری به صورت زیر به دست می‌آید:

$$x(t) = \frac{1}{3}(t-1)^3 + \frac{1}{3}, \quad 0 \leq t \leq 1,$$

اگر جواب در بازه بعدی  $[1, 2]$  پیوسته باشد، گام دیگری مانند اولین گام می‌تواند اختیار شود. بنابراین به ازای  $t$  های واقع در  $[1, 2]$  داریم:

$$\begin{cases} \dot{x}(t) = x(t-1) = \frac{1}{3}(t-1)^3 + \frac{1}{3}, & 1 \leq t \leq 2, \\ x(1) = \frac{1}{3}. \end{cases}$$

جواب این معادله عبارت است از

$$x(t) = \frac{1}{12}(t-1)^4 + \frac{1}{3}t - \frac{1}{12}, \quad 1 \leq t \leq 2.$$

جواب می‌تواند با محاسبات مشابه به طور نامتناهی به راست ادامه یابد [۴۷].

## ۱.۶.۱ تقریب پاده

تقریب گویا برای تابع  $f(x)$  روی فاصله  $[a, b]$  از خارج قسمت دو چندجمله‌ای  $P_n(x)$ ،  $Q_m(x)$  بترتیب از درجه  $n$  و  $m$  بصورت زیر بدست می‌آید:

$$R_L(x) = R_{m,n}(x) = \frac{P_n(x)}{Q_m(x)} = \frac{a_0 + a_1x + a_2x^2 + \dots + a_nx^n}{b_0 + b_1x + b_2x^2 + \dots + b_mx^m}.$$

درجه تقریب را با مجموع درجه چندجمله‌ای‌های  $Q_m(x)$ ،  $P_n(x)$  یعنی  $L = m + n$  تعریف می‌کنیم. برای محاسبه  $R_{m,n}$  باید  $L + 2$  پارامتر  $a_0, a_1, \dots, a_n$  و  $b_0, b_1, \dots, b_m$  را بدست آورد. روش‌های مختلفی برای محاسبه آن‌ها وجود دارد که یکی از این روش‌ها، روشی است که توسط پاده ارائه شد. در این تقریب برای محاسبه  $L + 2$  پارامتر، ابتدا  $b_0$  را برابر یک فرض می‌کنیم.

برای محاسبه  $a_i$  ( $i = 0, 1, \dots, n$ )،  $b_j$  ( $j = 1, \dots, m$ ) از بسط مک‌لورن تابع  $f(x)$  استفاده می‌شود. یعنی:

$$f(x) = c_0 + c_1x + c_2x^2 + \dots = \sum_{j=0}^{\infty} c_jx^j.$$

بطوریکه رابطه زیر برقرار باشد:

$$f^{(z)}(0) = R_L^{(z)}(0), \quad z = 0, 1, \dots, L. \quad (14.1)$$

و خطای محاسبه  $f(x) - R_L(x)$  را بصورت زیر محاسبه می‌کنیم:

$$\begin{cases} Z(x) = f(x) - R_L(x) = f(x)Q_m(x) - P_n(x), \\ (\sum_{j=0}^{\infty} c_j x^j)(\sum_{j=0}^m b_j x^j) - \sum_{j=0}^n a_j x^j = \sum_{j=L+1}^{\infty} d_j x^j. \end{cases} \quad (15.1)$$

با توجه به رابطه (۱۴.۱) کمترین درجه چندجمله‌ای  $Z(x)$ ،  $L+1$  است. با متحد قرار دادن ضرایب جملات با درجه کمتر از  $L+1$  در طرفین رابطه (۱۵.۱)  $L+1$  معادله زیر بدست می‌آید:

$$\begin{cases} c_0 - a_0 = 0, \\ b_1 c_0 - c_1 - a_1 = 0, \\ b_2 c_0 + b_1 c_1 + c_2 - a_2 = 0, \\ b_3 c_0 + b_2 c_1 + b_1 c_2 + c_3 - a_3 = 0, \\ \vdots \\ b_n c_0 + b_{n-1} c_1 + \dots + c_n - a_n = 0, \end{cases} \quad (16.1)$$

و

$$\begin{cases} b_m c_{n-m+1} + b_{m-1} c_{n-m+2} + \dots + b_1 c_n + c_{n+1} = 0, \\ b_m c_{n-m+2} + b_{m-1} c_{n-m+3} + \dots + b_1 c_{n+1} + c_{n+2} = 0, \\ \vdots \\ b_m c_n + b_{m-1} c_{n+1} + \dots + b_1 c_{n+m-1} + c_{n+m} = 0. \end{cases} \quad (17.1)$$

ابتدا از معادله (۱۷.۱) پارامترهای  $b_1, \dots, b_m$  را محاسبه کرده سپس با جایگذاری در رابطه (۱۶.۱)،  $a_0, \dots, a_n$  را بدست می‌آوریم [۱].

## فصل ۲

# حل عددی مسائل کنترل بهینه تاخیری

### ۱.۲ مقدمه

در این فصل، در ابتدا با استفاده از تقریب پاده مرتبه اول<sup>۱</sup> مساله کنترل بهینه تاخیر زمانی<sup>۲</sup> را به یک مساله بدون تاخیر تبدیل می‌کنیم و با استفاده از روش تقریبی شبکه عصبی جواب عددی مساله را به دست می‌آوریم و با جواب دقیق مقایسه می‌کنیم.

### ۲.۲ معرفی مساله

تاخیر زمانی [۴۸]–[۵۴] اغلب در سیستم‌های مهندسی<sup>۳</sup> مطرح می‌شود. همچنین بسیاری از سیستم‌ها مانند سیستم‌های فیزیکی<sup>۴</sup> و راکتورهای شیمیایی صنعتی<sup>۵</sup> با معادلات دیفرانسیل تاخیری مدل‌بندی می‌شوند. مسائل کنترل بهینه تاخیر زمانی سیستم‌های خطی و غیرخطی علاقه بسیاری از محققان را به خود جذب کرده است.

---

<sup>۱</sup>First-order Páde Approximation

<sup>۲</sup>Time-delayed optimal control problem

<sup>۳</sup>Engineering Systems

<sup>۴</sup>Physical Systems

<sup>۵</sup>Industrial Chemical Reactors

مساله کنترل بهینه تاخیری زیر را در نظر می‌گیریم:

$$\min_{u, \pi} J = \eta(x_f, \pi) + \int_0^{t_f} L(x(t), x(t - \sigma), u(t), \pi, t) dt \quad (1.2)$$

subject to

$$\dot{x}(t) = f(x(t), x(t - \sigma), u(t), \pi, t), \quad (2.2)$$

$$x(t) = \xi(t), \quad -\sigma \leq t \leq 0, \quad (3.2)$$

$$\eta(x_f, \pi) \geq 0, \quad (4.2)$$

که در آن  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$  و  $\pi \in \mathbb{R}^n$  به ترتیب بردار حالت، کنترل و پارامترهای مجهول هستند.  $\eta \in \mathbb{R}^n$  قیود نامساوی نهایی<sup>۶</sup> را نشان می‌دهد و در اینجا  $(\cdot)_f$  و  $\sigma$  به ترتیب مشتق نسبت به متغیر  $t$ ، شرط متغیر وابسته به زمان در زمان نهایی  $t_f$  و تاخیر زمانی مربوط به متغیر حالت است. برای سادگی بحث‌مان را به مسائلی محدود می‌کنیم که دارای تنها یک تاخیر زمانی هستند.

## ۳.۲ تبدیل مساله تاخیری به مساله بدون تاخیر

برای این‌که مسائل کنترل بهینه تاخیری را با استفاده از روش عددی شبکه عصبی حل کنیم، ابتدا باید مساله را از حالت تاخیر خارج کنیم. با استفاده از تقریب پاده و تبدیل لاپلاس که در فصل اول بیان شد، فرض می‌کنیم که  $X(s)$  یک تبدیل لاپلاس دوطرفه از  $x(t)$  باشد:

$$x(t) \Leftrightarrow X(s) \triangleq \int_{-\infty}^{+\infty} e^{-st} x(t) dt,$$

که در آن  $X(s)$  روی  $\alpha < \text{Re}(s) < \beta$  تعریف می‌شود،  $\text{Re}(s)$  قسمت حقیقی  $s$  است و “ $\Leftrightarrow$ ” نشان‌دهنده یک تبدیل لاپلاس دو طرفه است. از تبدیل لاپلاس دو طرفه به این دلیل استفاده می‌شود که برای  $t \leq 0$ ،  $x(t) = \xi(t) \neq 0$  است. داریم:

$$x(t) \Leftrightarrow X(s),$$

بنابراین تبدیل لاپلاس دو طرفه از  $x(t - \sigma)$  و  $\dot{x}(t)$  به صورت زیر است:

$$x(t - \sigma) \Leftrightarrow e^{-\sigma s} X(s), \quad \alpha < \text{Re}(s) < \beta, \quad (5.2)$$

$$\dot{x}(t) \Leftrightarrow sX(s), \quad \alpha_1 < \text{Re}(s) < \beta_1. \quad (6.2)$$

برای حذف متغیرهای حالت دارای تاخیر زمانی در روابط (۱.۲)–(۴.۲) تعریف می‌کنیم:

$$y(t) \triangleq x(t - \sigma), \quad (7.2)$$

$$y(t) \Leftrightarrow Y(s). \quad (8.2)$$

<sup>۶</sup>Terminal Inequality Constraint

از ادغام روابط (۵.۲)، (۷.۲) و (۸.۲) داریم:

$$Y(s) = e^{-\sigma s} X(s). \quad (۹.۲)$$

با استفاده از تقریب پاده مرتبه اول ( $m = n = ۱$ ) و با فرض این که  $f(\sigma s) = e^{-\sigma s}$  داریم:

$$e^{-(\sigma s)} = 1 - (\sigma s) + \frac{(\sigma s)^2}{2!} - \frac{(\sigma s)^3}{3!} + \dots = \frac{a_0 + a_1(\sigma s)}{1 + b_1(\sigma s)}. \quad (۱۰.۲)$$

با استفاده از روابط (۱۶.۱) و (۱۷.۱) داریم:

$$\begin{cases} 1 - a_0 = 0, \\ b_1 - 1 - a_1 = 0, \end{cases} \quad (۱۱.۲)$$

و

$$-b_1 + \frac{1}{\sigma} = 0. \quad (۱۲.۲)$$

با استفاده از روابط (۱۱.۲) و (۱۲.۲) مقادیر  $a_0, a_1, b_1$  به دست می آید. بنابراین داریم:

$$Y(s) = \frac{1 - \frac{\sigma s}{2}}{1 + \frac{\sigma s}{2}} X(s), \quad (۱۳.۲)$$

$$Y(s) = \frac{\frac{2}{\sigma} - s}{\frac{2}{\sigma} + s} X(s), \quad (۱۴.۲)$$

$$\left(\frac{2}{\sigma} + s\right)Y(s) = \left(\frac{2}{\sigma} - s\right)X(s). \quad (۱۵.۲)$$

اگر از طرفین (۱۵.۲) تبدیل لاپلاس معکوس<sup>۷</sup> بگیریم به رابطه زیر می رسیم:

$$\dot{y}(t) = \frac{2}{\sigma}[x(t) - y(t)] - \dot{x}(t), \quad (۱۶.۲)$$

با استفاده از (۲.۲) داریم:

$$\dot{y}(t) = \frac{2}{\sigma}[x(t) - y(t)] - f(x(t), y(t), u(t), \pi, t). \quad (۱۷.۲)$$

بنابراین با استفاده از (۱۶.۲) و (۱۷.۲) مساله کنترل بهینه تاخیری بطور تقریبی به مساله کنترل بهینه زیر تبدیل می شود:

$$\min_{u, \pi} J = \eta(x_f, \pi) + \int_0^{t_f} L(x(t), y(t), u(t), \pi, t) dt \quad (۱۸.۲)$$

subject to

$$\dot{x}(t) = f(x(t), y(t), u(t), \pi, t), \quad (۱۹.۲)$$

$$\dot{y}(t) = \frac{2}{\sigma}[x(t) - y(t)] - f(x(t), y(t), u(t), \pi, t), \quad (۲۰.۲)$$

با شرایط اولیه<sup>۸</sup> و نهایی:

$$x(0) = \xi(0), \quad y(0) = \xi(-\sigma), \quad (۲۱.۲)$$

$$\eta(x_f, \pi) \geq 0. \quad (۲۲.۲)$$

<sup>۷</sup>Inverse Laplace

<sup>۸</sup>Initial condition

برای بهتر شدن تقریب، تاخیر زمانی را به بخش‌های کوچکتر تقسیم می‌کنیم. ما در اینجا زمان تاخیر  $\sigma$  را به  $\frac{\sigma}{4}$  تقسیم می‌کنیم. بنابراین تعریف می‌کنیم:

$$y(t) \triangleq x(t - \frac{\sigma}{4}),$$

$$z(t) \triangleq y(t - \frac{\sigma}{4}) = x(t - \sigma).$$

دوباره با استفاده از تقریب پاده مرتبه اول داریم:

$$Y(s) = \frac{1 - \frac{\sigma s}{4}}{1 + \frac{\sigma s}{4}} X(s), \quad (23.2)$$

$$Z(s) = \frac{1 - \frac{\sigma s}{4}}{1 + \frac{\sigma s}{4}} Y(s). \quad (24.2)$$

از دو رابطه بالا تبدیل لاپلاس معکوس می‌گیریم. بنابراین داریم:

$$\dot{y}(t) = \frac{4}{\sigma} [x(t) - y(t)] - f(x(t), z(t), u(t), \pi, t),$$

$$\dot{z}(t) = \frac{4}{\sigma} [y(t) - z(t)] - \dot{y}(t) = \frac{4}{\sigma} [2y(t) - z(t) - x(t)]$$

$$+ f(x(t), z(t), u(t), \pi, t).$$

در نتیجه مساله کنترل بهینه (۱۸.۲)-(۲۲.۲) به مساله کنترل بهینه زیر تبدیل می‌شود:

$$\min_{u, \pi} J = \eta(x_f, \pi) + \int_0^{t_f} L(x(t), z(t), u(t), \pi, t) dt, \quad (25.2)$$

subject to

$$\dot{x}(t) = f(x(t), z(t), u(t), \pi, t), \quad (26.2)$$

$$\dot{y}(t) = \frac{4}{\sigma} [x(t) - y(t)] - f(x(t), z(t), u(t), \pi, t), \quad (27.2)$$

$$\dot{z}(t) = \frac{4}{\sigma} [2y(t) - x(t) - z(t)] - f(x(t), z(t), u(t), \pi, t), \quad (28.2)$$

با شرایط اولیه و نهایی:

$$x(\circ) = \xi(\circ), \quad y(\circ) = \xi(-\frac{\sigma}{4}), \quad z(\circ) = \xi(-\sigma), \quad (29.2)$$

$$\eta(x_f, \pi) \geq \circ. \quad (30.2)$$

با توجه به مطالب ارائه شده می‌توان گفت که یک مساله کنترل بهینه تاخیری می‌تواند به یک مساله غیرتاخیری تبدیل شود. در ادامه تلاش می‌کنیم تا یک طرح عددی بر اساس شبکه عصبی پیشنهاد کنیم تا شکل کلی مسائل کنترل بهینه غیرخطی بدون تاخیر را حل کنیم.



بنا به مساله (۲۵.۲)-(۳۰.۲) می‌توانیم هامیلتونین را به صورت زیر بسازیم:

$$\begin{aligned} H(t, x(t), y(t), z(t), u(t), \gamma(t), \lambda(t), \mu(t)) &= L(x(t), z(t), u(t), \pi, t) \\ &+ \gamma(t) \cdot f(x(t), z(t), u(t), \pi, t) + \lambda(t) \cdot \left( \frac{\gamma}{\sigma} [x(t) - y(t)] - f(x(t), z(t), u(t), \pi, t) \right) \\ &+ \mu(t) \cdot \left( \frac{\gamma}{\sigma} [2y(t) - z(t) - x(t)] + f(x(t), z(t), u(t), \pi, t) \right). \end{aligned}$$

با توجه به  $PMP$  (اصل مینیمم پونتریاگین) [۹۵]، اگر  $x(t), y(t)$  و  $z(t)$  متغیرهای حالت،  $\gamma(t), \lambda(t)$  و  $\mu(t)$  متغیرهای الحاقی و  $u(t)$  متغیر کنترل مساله (۲۵.۲)-(۳۰.۲) باشند، باید در شرایط زیر صدق کنند:

$$\left\{ \begin{array}{l} \frac{\partial H}{\partial x} = -\dot{\gamma}(t), \\ \frac{\partial H}{\partial y} = -\dot{\lambda}(t), \\ \frac{\partial H}{\partial z} = -\dot{\mu}(t), \\ \frac{\partial H}{\partial u} = \dot{x}(t), \\ \frac{\partial H}{\partial \gamma} = \dot{y}(t), \\ \frac{\partial H}{\partial \lambda} = \dot{z}(t), \\ \frac{\partial H}{\partial \mu} = 0, \\ \frac{\partial u}{\partial \eta(x_f, \pi)} - \gamma(t_f) = 0, \\ \lambda(t_f) = 0, \\ \mu(t_f) = 0, \\ x(0) = \xi(0), \\ y(0) = \xi(-\frac{\sigma}{\gamma}), \\ z(0) = \xi(-\sigma), \end{array} \right. \quad (۳۱.۲)$$

برای هر  $t \in [0, t_f]$  که  $H := H(t, x(t), y(t), z(t), u(t), \gamma(t), \lambda(t), \mu(t))$  دستگاه (۳۱.۲) یک سیستم  $ODE$  می‌دهد که می‌توان آن را با روش‌های عددی موجود حل کرد. در برخی موارد دستگاه (۳۱.۲) یک سیستم  $ODE$  را فراهم می‌کند که به آسانی می‌توان آن را حل کرد اما در بیشتر موارد (مخصوصاً در مسائل کاربردی)، سیستم به راحتی حل نمی‌شود و باید روش‌های تقریبی به کار گرفته شود. در بخش بعد، سعی می‌کنیم توانایی شبکه عصبی را در تخمین تابع برای حل دستگاه (۳۱.۲) به کار ببریم.

## ۴.۲ شبکه عصبی

در شبکه عصبی پیش-خورد می‌توانیم خروجی شبکه را به وسیله‌ی فرمول زیر محاسبه کنیم:

$$\left\{ \begin{array}{l} \text{output} = \sum_{i=1}^I v_i \sigma(s^i), \\ s^i = w_i x + b_i, \end{array} \right. \quad (۳۲.۲)$$

که  $I$  تعداد سیگموئیدهای واحد است،  $w_i$  پارامتر وزن از ورودی به واحد پنهان  $i$  ام است،  $v_i$  پارامتر وزن  $i$  ام از واحد پنهان  $i$  به لایه خروجی،  $b_i$  بایاس  $I$  ام برای واحد  $i$  ام در لایه پنهان است،  $s^i$  ورودی از واحد پنهان  $i$  ام است و  $\sigma$  یک تابع سیگموئید دلخواه است. تابع انتقال  $\sigma$  به طور کلی یک تابع یک بعدی غیر خطی یکنواخت (تابع فعال) است که می تواند به راحتی مورد بررسی قرار گیرد. در مثال های عددی ارائه شده در این پایان نامه، تابع فعال ساز که یک تابع سیگموئید است به صورت زیر در نظر گرفته می شود:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (33.2)$$

هر تابع به اندازه کافی هموار می تواند به طور دلخواه در یک مجموعه فشرده و بسته با استفاده از یک شبکه عصبی دو لایه با وزن مناسب تقریب زده شود (برای جزئیات بیشتر [۵۵] - [۵۷] ببینید). این نتیجه نشان می دهد که هر تابع پیوسته می توان با استفاده از یک ترکیب خطی از توابع سیگموئید تقریب زده شود. این به عنوان ویژگی تقریب کلی شبکه های عصبی شناخته شده است. با توجه به این حقیقت، از توانایی شبکه های عصبی در تقریب تابع، برای تقریب توابع حالت، الحاقی و کنترل برای مساله کنترل بهینه (۲۵.۲) - (۳۰.۲) استفاده می کنیم. اکنون طرح تخمینی برای حل دستگاه (۳۱.۲) پیشنهاد می کنیم. برای هر تابع، ۷ شبکه عصبی در نظر می گیریم: حالت (شبکه های عصبی متناظر آن  $n_z$ ،  $(n_x, n_y)$ ، الحاقی (شبکه های عصبی متناظر آن  $n_\gamma, n_\lambda, n_\mu$ ) کنترل (شبکه عصبی متناظر آن  $n_u$ )

$$\begin{cases} n_x = \sum_{i=1}^I v_x^i \bar{\sigma}(s_x^i), & s_x^i = w_x^i \tau + b_x^i, \\ n_y = \sum_{i=1}^I v_y^i \bar{\sigma}(s_y^i), & s_y^i = w_y^i \tau + b_y^i, \\ n_z = \sum_{i=1}^I v_z^i \bar{\sigma}(s_z^i), & s_z^i = w_z^i \tau + b_z^i, \\ n_\gamma = \sum_{i=1}^I v_\gamma^i \bar{\sigma}(s_\gamma^i), & s_\gamma^i = w_\gamma^i \tau + b_\gamma^i, \\ n_\lambda = \sum_{i=1}^I v_\lambda^i \bar{\sigma}(s_\lambda^i), & s_\lambda^i = w_\lambda^i \tau + b_\lambda^i, \\ n_\mu = \sum_{i=1}^I v_\mu^i \bar{\sigma}(s_\mu^i), & s_\mu^i = w_\mu^i \tau + b_\mu^i, \\ n_u = \sum_{i=1}^I v_u^i \bar{\sigma}(s_u^i), & s_u^i = w_u^i \tau + b_u^i, \end{cases} \quad (34.2)$$

برای  $i = 1, 2, \dots, n$  که در آن  $I$  تعداد نرون ها است که می تواند برای هر شبکه عصبی متفاوت باشد، و  $\bar{\sigma}(s) = (t_f - t_0)\sigma(s) + t_0$  می توانیم رابطه بین جواب برای شرایط بهینگی (۳۱.۲) و جواب برای یک مساله مینیم سازی بدون محدودیت معادل از طریق جواب های آزمایشی ایجاد کنیم [۹۲]، [۹۳]. این تابع آزمایشی به عنوان مجموع دو بخش نوشته شده و بر اساس دو واقعیت ساخته شده است: عبارت اول شامل هیچ پارامترهای قابل تنظیم نیست و فقط شرایط اولیه و انتهایی را ارضا می کند. عبارت دوم به کارگیری شبکه عصبی فید-فوروارد خروجی منحصر به فرد، با پارامترهای قابل تنظیم و واحدهای ورودی تغذیه شده با بردار ورودی است. لازم به ذکر است عبارت دوم، به ازای زمان اولیه و نهایی صفر می شود. بر اساس این واقعیتهای و با استفاده از شبکه های عصبی ارائه شده در (۳۴.۲)، جواب های

آزمایشی (برای حالت، الحاقی و تابع کنترل) می تواند به صورت زیر تعریف شود:

$$\begin{cases} x_T = x_o + (t - t_o)n_x, \\ y_T = y_o + (t - t_o)n_y, \\ z_T = z_o + (t - t_o)n_z, \\ \gamma_T = \frac{\partial \eta(x_f, \pi)}{\partial x} + (t - t_f)n_\gamma, \\ \lambda_T = (t - t_f)n_\lambda, \\ \mu_T = (t - t_f)n_\mu, \\ u_T = n_u, \end{cases} \quad (35.2)$$

جواب آزمایشی (35.2) تقریب کلی هستند و باید در شرایط (31.2) صدق کند. بنابراین داریم:

$$\begin{cases} \frac{\partial H_T}{\partial x_T} = -\dot{\gamma}_T(t), \\ \frac{\partial H_T}{\partial y_T} = -\dot{\lambda}_T(t), \\ \frac{\partial H_T}{\partial z_T} = -\dot{\mu}_T(t), \\ \frac{\partial H_T}{\partial \gamma_T} = \dot{x}_T(t), \\ \frac{\partial H_T}{\partial \lambda_T} = \dot{y}_T(t), \\ \frac{\partial H_T}{\partial \mu_T} = \dot{z}_T(t), \\ \frac{\partial H_T}{\partial u_T} = 0, \end{cases} \quad (36.2)$$

که در آن  $H_T := H(t, x_T(t), y_T(t), z_T(t), u_T(t), \gamma_T(t), \lambda_T(t), \mu_T(t))$  به منظور فرمولبندی (36.2) به عنوان یک مساله مینیم سازی بدون محدودیت، ابتدا سیستم های بهینگی (36.2) را روی  $m$  نقطه  $t_k, k = 1, \dots, m$  از بازه  $[t_o, t_f]$  مرتب می کنیم و سپس یک مساله بهینه سازی به صورت زیر تعریف می کنیم

$$\begin{aligned} \text{minimize}_y E(y) = & \frac{1}{m} \sum_{k=1}^m \{E_1(t_k, y) + E_2(t_k, y) + E_3(t_k, y) \\ & + E_4(t_k, y) + E_5(t_k, y) + E_6(t_k, y) + E_7(t_k, y)\}, \end{aligned} \quad (37.2)$$

که

$$\begin{aligned} y = & (w_x, w_y, w_z, w_\gamma, w_\lambda, w_\mu, w_u, b_x, b_y, b_z, b_\gamma, b_\lambda, b_\mu, b_u, v_x, v_y, v_z, v_\gamma, v_\lambda, v_\mu, v_u)^T \\ & \in \mathbb{R}^{VI(9n+m)} \end{aligned} \quad (38.2)$$

و

$$\begin{cases} E_1(t_k, y) = \left[ \frac{\partial H_T}{\partial x_T} + \dot{\gamma}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_2(t_k, y) = \left[ \frac{\partial H_T}{\partial y_T} + \dot{\lambda}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_3(t_k, y) = \left[ \frac{\partial H_T}{\partial z_T} + \dot{\mu}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_4(t_k, y) = \left[ \frac{\partial H_T}{\partial \gamma_T} - \dot{x}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_5(t_k, y) = \left[ \frac{\partial H_T}{\partial \lambda_T} - \dot{y}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_6(t_k, y) = \left[ \frac{\partial H_T}{\partial \mu_T} - \dot{z}_T(t) \right]^T, & k = 1, 2, \dots, m, \\ E_7(t_k, y) = \left[ \frac{\partial H_T}{\partial u_T} \right]^T, & k = 1, 2, \dots, m, \end{cases} \quad (39.2)$$

لم ۱.۴.۲. اگر

$$y^* = (w_x^*, w_y^*, w_z^*, w_\gamma^*, w_\lambda^*, w_\mu^*, w_u^*, b_x^*, b_y^*, b_z^*, b_\gamma^*, b_\lambda^*, b_\mu^*, b_u^*, v_x^*, v_y^*, v_z^*, v_\gamma^*, v_\lambda^*, v_\mu^*, v_u^*)^T$$

در معادله زیر صدق کند

$$\eta(y) = \begin{bmatrix} E_1(t_1, y) \\ \vdots \\ E_1(t_m, y) \\ E_2(t_1, y) \\ \vdots \\ E_2(t_m, y) \\ \vdots \\ E_7(t_1, y) \\ \vdots \\ E_7(t_m, y) \end{bmatrix} = 0, \quad (40.2)$$

آنگاه  $y^*$  یک جواب بهینه از (۳۸.۲) است.

برهان. فرض کنید  $\eta(y^*) = 0$  آنگاه  $E_i(t_k, y^*) = 0$  برای  $i = 1, 2, \dots, 7$  و  $k = 1, \dots, m$  چون در (۳۸.۲)  $E(y) \geq 0$  بنابراین  $y^*$  یک جواب بهینه از (۳۸.۲) است.  $\square$

بنابر لم (۱.۴.۲) می‌توان بررسی کرد که مساله بهینه سازی (۳۸.۲) با مساله زیر هم ارز است

$$\text{minimize}_y E(y) = \frac{1}{7} \|\eta(y)\|^2 \quad (41.2)$$

برای حل (۴۱.۲) که یک مساله بهینه سازی نامقید است، می‌توان از الگوریتم‌های بهینه سازی مانند تندترین کاهش، نیوتن، شبه نیوتن، گرادیان مزدوج و غیره استفاده کرد [۵۸]-[۶۰]. همچنین می‌توانیم از الگوریتم‌های ابتکاری مانند الگوریتم ژنتیک، بهینه سازی گروه پرندگان، الگوریتم مورچگان و غیره

استفاده کنیم [۶۱].

## ۵.۲ مثال‌های عددی

کارایی روش شبکه عصبی ارائه شده در فصل اول با بررسی شش مثال نشان داده می‌شود.

مثال ۱۰.۵.۲. این مثال مربوط به مساله کنترل بهینه از یک نوسانگر هارمونیک با میرای تاخیری است که مستلزم تابع هزینه

$$\text{minimize } J = 5x_1^2 + \frac{1}{\tau} \int_0^2 u^2(t) dt \quad (42.2)$$

تحت قيود دستگاه معادلات دیفرانسیل تاخیری

$$\begin{cases} \dot{x}_1(t) = x_2(t), & 0 \leq t \leq 2, \\ \dot{x}_2(t) = -x_1(t) - x_2(t-1) + u(t), & 0 \leq t \leq 2, \end{cases} \quad (43.2)$$

و شرایط اولیه زیر است:

$$x_1(0) = 1, \quad x_2(t) = 0, \quad -1 \leq t \leq 0. \quad (44.2)$$

جواب تحلیلی برای مساله ارائه شده به صورت زیر است [۶۲]:

$$u(t) = \begin{cases} \delta \sin(2-t) + (\frac{\delta}{\tau})(1-t) \sin(t-1), & 0 \leq t \leq 1, \\ \delta \sin(2-t), & 1 \leq t \leq 2, \end{cases}$$

که در آن  $\delta \approx 2/5599$  است. همان‌طور که در (۴۳.۲) می‌بینیم زمان تاخیر مربوط به متغیر حالت  $x_2$  است که در اینجا مقدار آن برابر ۱ است. با استفاده از روش توصیف شده در بخش ۳.۲ داریم:

$$\begin{cases} x_3(t) \triangleq x_2(t - \frac{1}{\tau}), \\ x_4(t) \triangleq x_3(t - \frac{1}{\tau}) = x_2(t - 1). \end{cases}$$

بنابراین معادلات دیفرانسیل تاخیری به معادلات زیر که فاقد تاخیر زمانی هستند تبدیل می‌شوند:

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = -x_1(t) - x_4(t) + u(t), \\ \dot{x}_3(t) = x_1(t) + 4x_2(t) - 4x_3(t) + x_4(t) - u(t), \\ \dot{x}_4(t) = -x_1(t) - 4x_2(t) + 8x_3(t) - 5x_4(t) + u(t). \end{cases} \quad (45.2)$$

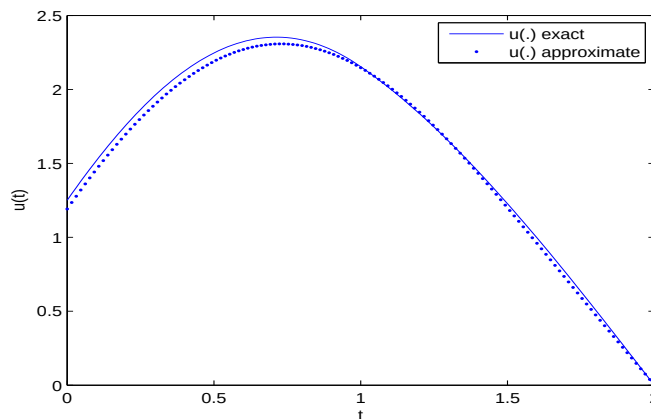
با شرایط اولیه

$$x_1(0) = 1, \quad x_2(0) = 0, \quad x_3(0) = 0, \quad x_4(0) = 0. \quad (46.2)$$

فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_1T = 1^\circ + (t)n_{x1}, \\ x_2T = 0^\circ + (t)n_{x2}, \\ x_3T = 0^\circ + (t)n_{x3}, \\ x_4T = 0^\circ + (t)n_{x4}, \\ \gamma_1T = 1^\circ x_1(2) + (t - 2)n_{\gamma1}, \\ \gamma_2T = (t - 2)n_{\gamma2}, \\ \lambda_T = (t - 2)n_\lambda, \\ \mu_T = (t - 2)n_\mu, \\ u_T = n_u, \end{cases}$$

مساله مینیم سازی با تابع هدف (۴۲.۲) تحت قیود (۴۵.۲) را با استفاده از روش تقریبی شبکه عصبی حل می کنیم مقدار به دست آمده برای  $J$  برابر ۳/۲۵۲۸ است و مقدار داده شده در [۶۵] برابر ۳/۳۹۹۱ است. کنترل بهینه  $u$  و مسیرهای جدید  $x_1, x_2, x_3$  و  $x_4$  به ترتیب در شکل های ۱.۲-۳.۲ رسم شده است.



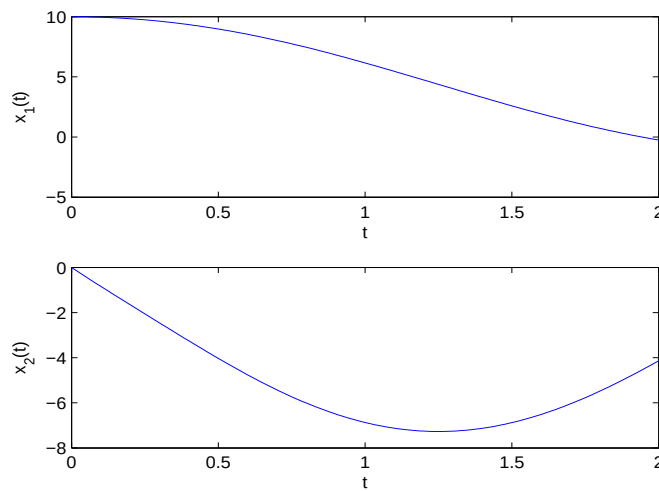
شکل ۱.۲: نمودار کنترل  $u(t)$  برای مثال ۱.۵.۲

مثال ۲.۵.۲. مساله زیر را در نظر بگیرید:

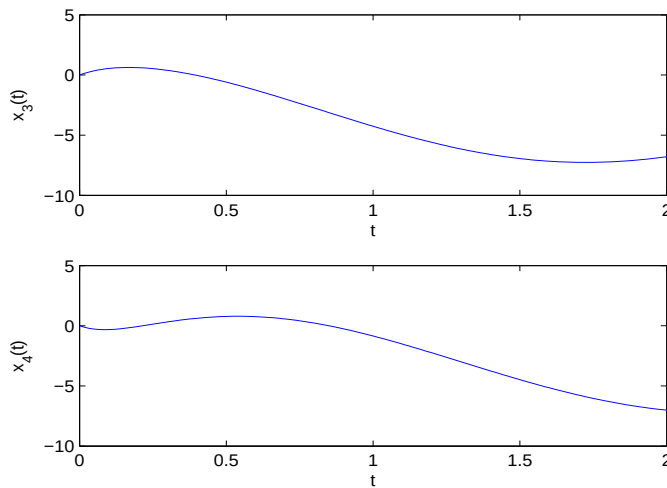
$$\text{minimize } J = \frac{1}{\gamma} x_1^2 + \frac{1}{\gamma} \int_0^2 [x_1^2(t) + u^2(t)] dt$$

subject to

$$\dot{x}_1(t) = x_1(t) \sin(x_1(t)) + x_1(t - 1) + u(t), \quad 0 \leq t \leq 2,$$



شکل ۲.۲: نمودار مسیرهای  $x_1(t)$  و  $x_2(t)$  برای مثال ۱.۵.۲



شکل ۳.۲: نمودار مسیرهای  $x_3(t)$  و  $x_4(t)$  برای مثال ۱.۵.۲

با شرط اولیه

$$x_1(t) = 1, \quad -1 \leq t \leq 0.$$

برای حذف زمان تاخیر  $\sigma = 1$  در این مساله مانند مثال ۱.۵.۲ تعریف می‌کنیم:

$$\begin{cases} x_2(t) \triangleq x_1(t - \frac{1}{\tau}), \\ x_3(t) \triangleq x_2(t - \frac{1}{\tau}) = x_1(t - 1). \end{cases}$$

بنابراین با استفاده از روش گفته شده در بخش ۳.۲ قید مساله به قیود زیر تبدیل می شود:

$$\begin{cases} \dot{x}_1(t) = x_1(t) \sin(x_1(t)) + x_3(t) + u(t), \\ \dot{x}_2(t) = 4x_1(t) - 4x_2(t) - x_3(t) - x_1(t) \sin(x_1(t)) - u(t), \\ \dot{x}_3(t) = -4x_1(t) + 8x_2(t) - 3x_3(t) + x_1(t) \sin(x_1(t)) + u(t), \end{cases} \quad (47.2)$$

با شرایط اولیه

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 1. \quad (48.2)$$

فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_{1T} = 1 + (t)n_{x_1}, \\ x_{2T} = 1 + (t)n_{x_2}, \\ x_{3T} = 1 + (t)n_{x_3}, \\ \gamma_T = x_1(2) + (t-2)n_\gamma, \\ \lambda_T = (t-2)n_\lambda, \\ \mu_T = (t-2)n_\mu, \\ u_T = n_u, \end{cases}$$

ملاحظه می کنید که در این روش دو قید به قیود مساله و دو شرط به شرایط اولیه و بطور کلی دو متغیر به متغیرهای حالت افزوده می شود. با حل عددی مساله به روش شبکه عصبی مقدار بهینه تابع هدف ۱۶۲/۰۲۶۶ به دست می آید و مقدار  $J$  در [۶۳] و [۶۴] به ترتیب ۱۶۲/۰۱۹ و ۱۶۲/۱۰۴ داده شده است. نمودار بهینه  $u(t)$  و  $x_1(t)$  برای  $t \in [0, 2]$  در شکل ۴.۲ رسم شده است.

مثال ۳.۵.۲. مساله کنترل بهینه تاخیری مینیم سازی با زمان تاخیر  $\sigma = 1$  را به صورت زیر در نظر می گیریم:

$$\text{minimize } J = \frac{1}{2} \{ [x_1(2)]^2 + [x_2(2)]^2 \} + \frac{1}{2} \int_0^2 [u(t)]^2 dt \quad (49.2)$$

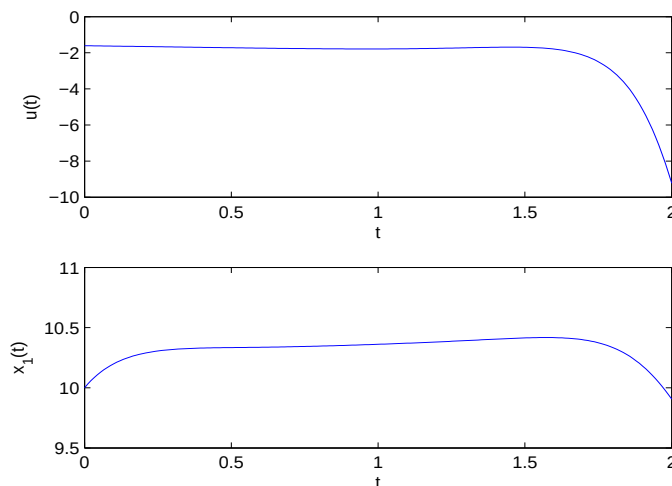
subject to

$$\begin{cases} \dot{x}_1(t) = x_2(t), & 0 \leq t \leq 2, \\ \dot{x}_2(t) = -x_2(t) - x_1(t-1) + u(t), & 0 \leq t \leq 2, \end{cases} \quad (50.2)$$

با شرایط اولیه

$$x_1(t) = 1, \quad x_2(0) = 0, \quad -1 \leq t \leq 0.$$





شکل ۴.۲: نمودار کنترل  $u(t)$  و مسیر  $x_1(t)$  برای مثال ۲.۵.۲

جواب دقیق برای این مساله به صورت زیر است [۶۲]:

$$u(t) = \begin{cases} (\mu - \delta)e^{t-2} + [2\mu - 3\delta - (\mu - \delta)t]e^{t-1} + \delta(t + 2) - \mu, & 0 \leq t \leq 1, \\ (\mu - \delta)e^{t-2} + \delta, & 1 \leq t \leq 2, \end{cases}$$

که در آن  $\delta \approx -0.259256$ ,  $\mu \approx 0.5226194$  است. همچنین مقدار بهینه  $J$  برابر  $0.197478$  است. همان طور که در قیود این مساله ملاحظه می‌کنیم زمان تاخیر مربوط به متغیر حالت  $x_1$  است برای حذف این زمان تاخیر تعریف می‌کنیم:

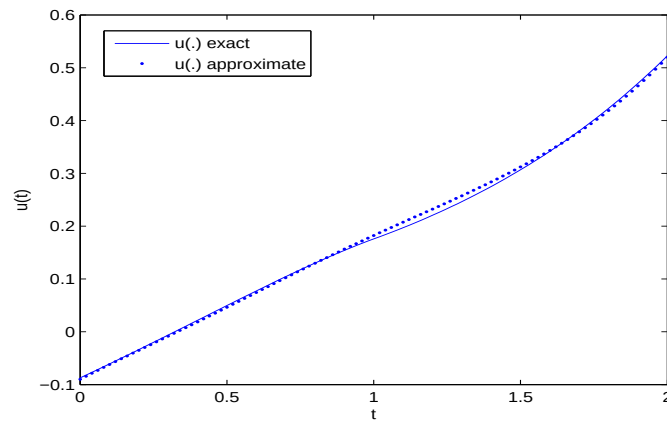
$$\begin{cases} x_3(t) \triangleq x_1(t - \frac{1}{\tau}), \\ x_4(t) \triangleq x_3(t - \frac{1}{\tau}) = x_1(t - 1). \end{cases} \quad (51.2)$$

با استفاده از روش گفته شده در بخش ۳.۲ و روابط (۵۱.۲) دستگاه معادلات دیفرانسیل تاخیری (قیودمساله) (۵۰.۲) به دستگاه معادلات دیفرانسیل بدون تاخیر زیر تبدیل می‌شود:

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = -x_2(t) - x_4(t) + u(t), \\ \dot{x}_3(t) = 4x_1(t) - x_2(t) - 4x_3(t), \\ \dot{x}_4(t) = -4x_1(t) + x_2(t) + 8x_3(t) - 4x_4(t), \end{cases} \quad (52.2)$$

با شرایط اولیه

$$x_1(0) = 1, \quad x_2(0) = 0, \quad x_3(0) = 1, \quad x_4(0) = 1. \quad (53.2)$$



شکل ۵.۲: نمودار کنترل  $u(t)$  برای مثال ۳.۵.۲

فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_{1T} = 1 + (t)n_{x_1}, \\ x_{2T} = 0 + (t)n_{x_2}, \\ x_{3T} = 1 + (t)n_{x_3}, \\ x_{4T} = 1 + (t)n_{x_4}, \\ \gamma_{1T} = x_1(2) + (t-2)n_{\gamma_1}, \\ \gamma_{2T} = x_2(2) + (t-2)n_{\gamma_2}, \\ \lambda_T = (t-2)n_{\lambda}, \\ \mu_T = (t-2)n_{\mu}, \\ u_T = n_u, \end{cases}$$

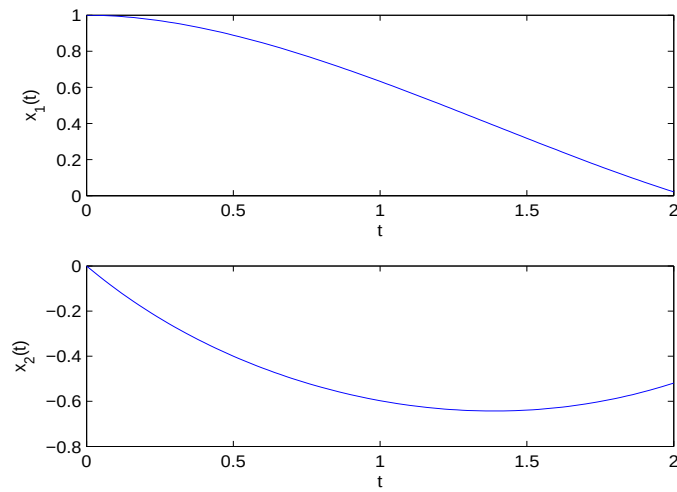
مساله کنترل بهینه بدون تاخیر زمانی با تابع هدف درجه دوم (۴۹.۲) تحت قيود خطی (۵۲.۲) و شرایط اولیه (۵۳.۲) را به روش شبکه عصبی حل می‌کنیم. مقدار بهینه  $J$  برابر ۱۹۷۱/۰ به دست می‌آید. نمودار کنترل بهینه  $u(t)$  و مسیرهای متناظر  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  و  $x_4(t)$  و نمودار دقیق متناظر آنها به ترتیب در شکل‌های ۵.۲-۷.۲ رسم شده است.

مثال ۴.۵.۲. مساله کنترل بهینه مینیمم‌سازی تحت قيودی به صورت دستگاه معادلات دیفرانسیل تاخیری

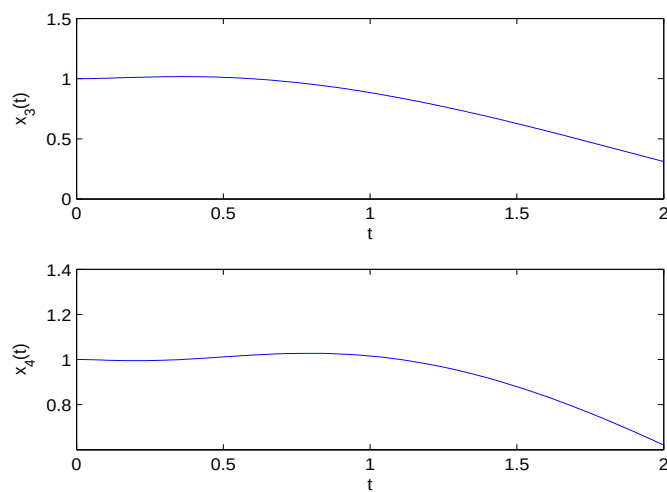
$$\begin{cases} \dot{x}_1(t) = x_1(t-1) + u(t), & 0 \leq t \leq 2, \\ x_1(t) = 1, & -1 \leq t \leq 0, \end{cases} \quad (54.2)$$

و تابع هدف زیر را در نظر می‌گیریم.

$$\text{minimize } J = \frac{1}{2} [10^5 x_1^2(2) + \int_0^2 u^2(t) dt]. \quad (55.2)$$



شکل ۶.۲: نمودار مسیرهای  $x_1(t)$  و  $x_2(t)$  برای مثال ۳.۵.۲



شکل ۷.۲: نمودار مسیرهای  $x_3(t)$  و  $x_4(t)$  برای مثال ۳.۵.۲

که در آن زمان تاخیر  $\sigma = 1$  مربوط به متغیر حالت است برای حذف این زمان تاخیر با استفاده از روش توصیف شده در بخش ۳.۲ به روابط زیر می‌رسیم:

$$\begin{cases} x_2(t) \triangleq x_1(t - \frac{1}{\gamma}), \\ x_3(t) \triangleq x_2(t - \frac{1}{\gamma}) = x_1(t - 1). \end{cases}$$

جواب دقیق این مساله به صورت زیر است [۶۶]:

$$u(t) = \begin{cases} -2/1 + 1/5t, & 0 \leq t \leq 1, \\ -1/5, & 1 \leq t \leq 2. \end{cases}$$

معادلات دینامیکی تاخیری به دستگاه زیر تبدیل می‌شوند:

$$\begin{cases} \dot{x}_1(t) = x_3(t) + u(t), \\ \dot{x}_2(t) = 4x_1(t) - 4x_2(t) - x_3(t) - u(t), \\ \dot{x}_3(t) = -3x_3(t) + 8x_2(t) - 4x_1(t) + u(t), \end{cases} \quad (56.2)$$

شرایط اولیه این مساله به صورت زیر است:

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 1. \quad (57.2)$$

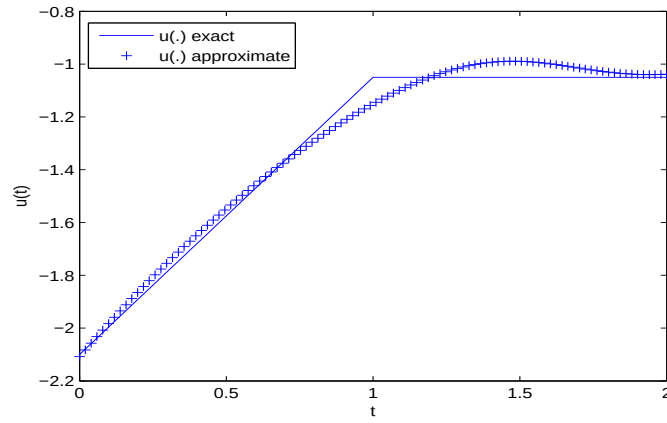
فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_{1T} = 1 + (t)n_{x_1}, \\ x_{2T} = 1 + (t)n_{x_2}, \\ x_{3T} = 1 + (t)n_{x_3}, \\ \gamma(T) = 1/5x_1(2) + (t-2)n_\gamma, \\ \lambda_T = (t-2)n_\lambda, \\ \mu_T = (t-2)n_\mu, \\ u_T = n_u, \end{cases}$$

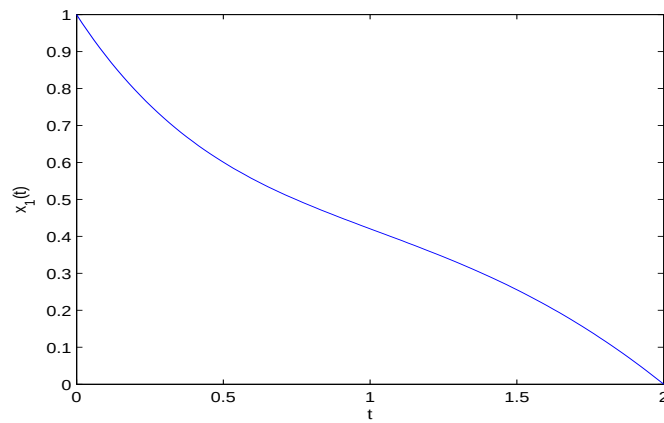
هدف از حل این مساله به دست آوردن کنترل بهینه  $u(t)$  به طوریکه تابع هدف (۵۵.۲) را تحت قیود (۵۶.۲) و شرایط اولیه (۵۷.۲) مینیمم کند. با حل عددی با روش شبکه عصبی مقدار مینیمم  $J$ ، ۱/۸۱۹۶ به دست می‌آید. نمودار کنترل بهینه و مسیرهای به دست آمده با روش شبکه عصبی و نمودار جواب دقیق متناظرشان به ترتیب در شکل‌های ۸.۲-۱۰.۲ رسم شده است.

مثال ۵.۵.۲. سیستم تاخیر زمانی زیر را در نظر بگیرید:

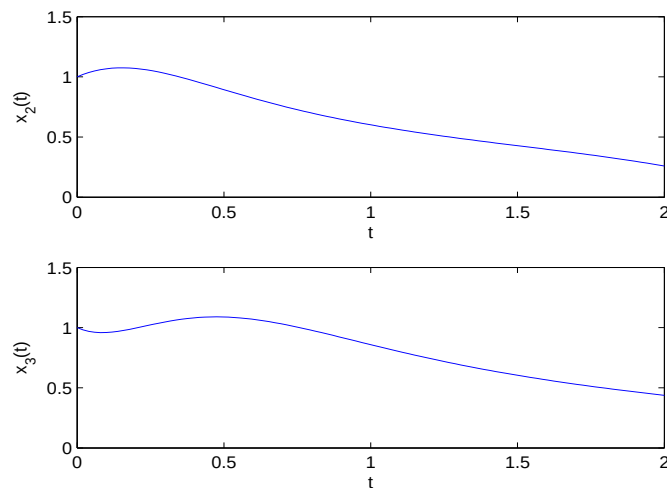
$$\text{minimize } J = 3/2 x_{1f}^2 + \frac{1}{2} \int_0^3 u^2(t) dt \quad (58.2)$$



شکل ۸.۲: نمودار کنترل  $u(t)$  برای مثال ۴.۵.۲



شکل ۹.۲: نمودار مسیر  $x_1(t)$  برای مثال ۴.۵.۲



شکل ۱۰.۲: نمودار مسیرهای  $x_2(t)$  و  $x_3(t)$  برای مثال ۴.۵.۲

تحت قید دستگاه معادلات دیفرانسیل تاخیری

$$\begin{cases} \dot{x}_1(t) = x_1(t-1) + u(t), & 0 \leq t \leq 3, \end{cases} \quad (۵۹.۲)$$

و شرایط اولیه زیر است:

$$x_1(0) = 1, \quad -1 \leq t \leq 0. \quad (۶۰.۲)$$

جواب تحلیلی برای مساله ارائه شده به صورت زیر است [۶۲]:

$$u(t) = \begin{cases} \delta \left\{ \frac{-(t-2)^2}{2} - \frac{3}{2} \right\}, & 0 \leq t \leq 1, \\ \delta(t-3), & 1 \leq t \leq 2, \\ -\delta, & 2 \leq t \leq 3, \end{cases}$$

که در آن  $\delta \approx ۰/۵۶۲۳۱$  است. همان طور که در (۵۹.۲) می بینیم زمان تاخیر مربوط به متغیر حالت  $x_1$  است که در اینجا مقدار آن برابر ۱ است. با استفاده از روش توصیف شده در بخش ۳.۲ داریم:

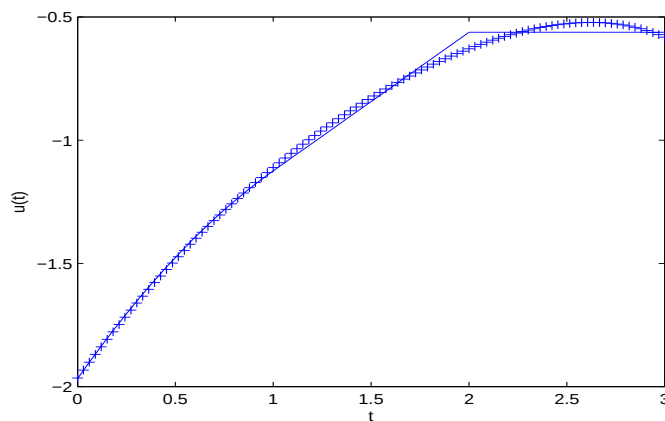
$$\begin{cases} x_2(t) \triangleq x_1(t - \frac{1}{2}), \\ x_3(t) \triangleq x_2(t - \frac{1}{2}) = x_1(t - 1). \end{cases}$$

بنابراین معادلات دیفرانسیل تاخیری به معادلات زیر که فاقد تاخیر زمانی هستند تبدیل می شوند:

$$\begin{cases} \dot{x}_1(t) = x_3(t) + u(t), \\ \dot{x}_2(t) = 4x_1(t) - 4x_2(t) - x_3(t) - u(t), \\ \dot{x}_3(t) = -4x_1(t) + 8x_2(t) - 3x_3(t) + u(t), \end{cases} \quad (۶۱.۲)$$

با شرایط اولیه

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 1 \quad (۶۲.۲)$$



شکل ۱۱.۲: نمودار کنترل  $u(t)$  برای مثال ۵.۵.۲

فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_{1T} = 1 + (t)n_{x1}, \\ x_{2T} = 1 + (t)n_{x2}, \\ x_{3T} = 1 + (t)n_{x3}, \\ \gamma_T = 3x_1(3) + (t-3)n_\gamma, \\ \lambda_T = (t-3)n_\lambda, \\ \mu_T = (t-3)n_\mu, \\ u_T = n_u, \end{cases}$$

مساله مینیم‌سازی با تابع هدف (۵۸.۲) تحت قیود (۶۱.۲) را با استفاده از روش تقریبی شبکه عصبی حل می‌کنیم مقدار به دست آمده برای  $J$  برابر  $۱/۷۴۳۳$  است و مقدار داده شده در [۶۲] برابر  $۱/۷۳۳۸$  است. کنترل بهینه  $u$  و مسیرهای جدید  $x_1, x_2, x_3$  به ترتیب در شکل‌های ۱۱.۲-۱۳.۲ رسم شده است.

مثال ۶.۵.۲. سیستم تاخیر زمانی زیر را در نظر بگیرید:

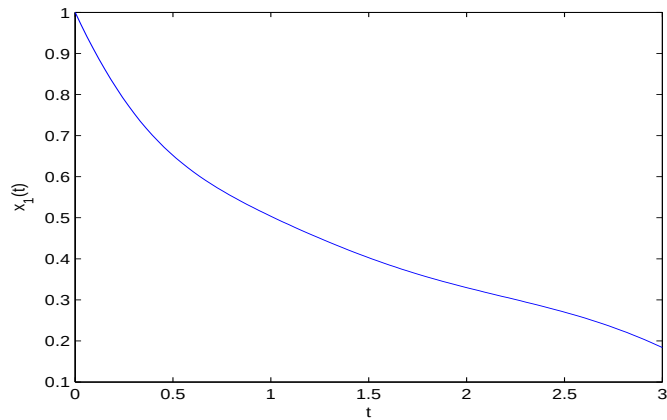
$$\text{minimize } J = 3/2 x_{1f}^2 + \frac{1}{2} \int_0^2 u^2(t) dt \quad (63.2)$$

تحت قید دستگاه معادلات دیفرانسیل تاخیری

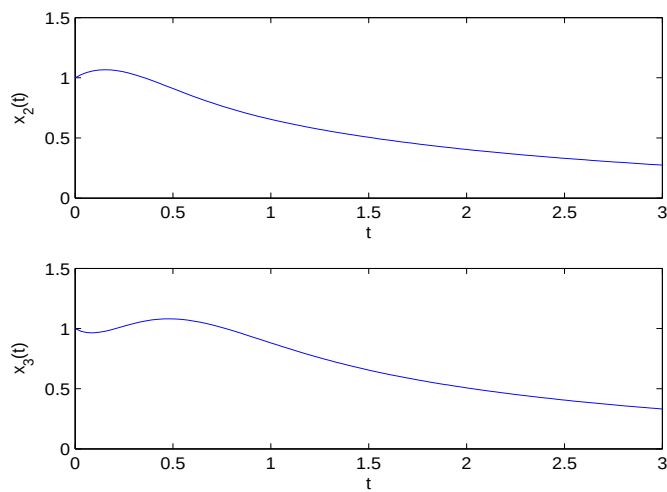
$$\begin{cases} \dot{x}_1(t) = x_1(t) + x_1(t-1) + u(t), & 0 \leq t \leq 2, \end{cases} \quad (64.2)$$

و شرایط اولیه زیر است:

$$x_1(0) = 1, \quad -1 \leq t \leq 0. \quad (65.2)$$



شکل ۱۲.۲: نمودار مسیر  $x_1(t)$  برای مثال ۵.۵.۲



شکل ۱۳.۲: نمودار مسیرهای  $x_2(t)$  و  $x_3(t)$  برای مثال ۵.۵.۲



جواب تحلیلی برای مساله ارائه شده به صورت زیر است [۶۲]:

$$u(t) = \begin{cases} \delta\{e^{(2-t)} + (1-t)e^{(1-t)}\}, & 0 \leq t \leq 1, \\ \delta e^{(2-t)}, & 1 \leq t \leq 2, \end{cases}$$

که در آن  $\delta \approx -0.3932$  است. همان‌طور که در (۶۴.۲) می‌بینیم زمان تاخیر مربوط به متغیر حالت  $x_1$  است که در اینجا مقدار آن برابر ۱ است. با استفاده از روش توصیف شده در بخش ۳.۲ داریم:

$$\begin{cases} x_2(t) \triangleq x_1(t - \frac{1}{\gamma}), \\ x_3(t) \triangleq x_2(t - \frac{1}{\gamma}) = x_1(t - 1). \end{cases}$$

بنابراین معادلات دیفرانسیل تاخیری به معادلات زیر که فاقد تاخیر زمانی هستند تبدیل می‌شوند:

$$\begin{cases} \dot{x}_1(t) = x_1(t) + x_3(t) + u(t), \\ \dot{x}_2(t) = 3x_1(t) - 4x_2(t) - x_3(t) - u(t), \\ \dot{x}_3(t) = -3x_1(t) + 8x_2(t) - 3x_3(t) + u(t), \end{cases} \quad (66.2)$$

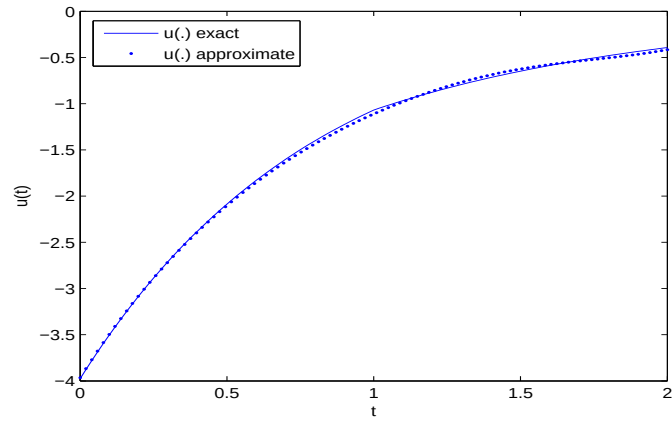
با شرایط اولیه

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 1 \quad (67.2)$$

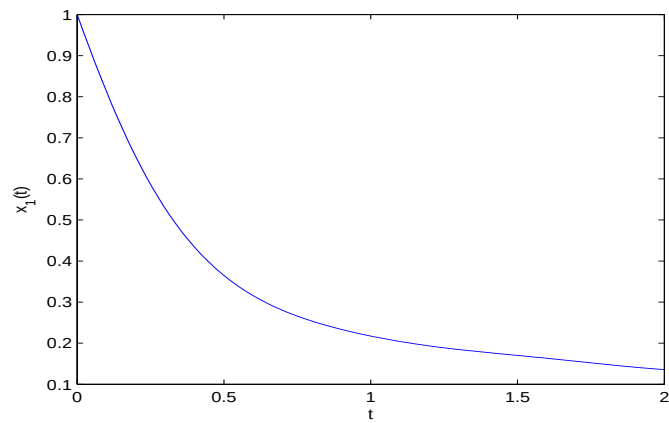
فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} x_{1T} = 1 + (t)n_{x_1}, \\ x_{2T} = 1 + (t)n_{x_2}, \\ x_{3T} = 1 + (t)n_{x_3}, \\ \gamma_T = 3x_1(2) + (t-2)n_\gamma, \\ \lambda_T = (t-2)n_\lambda, \\ \mu_T = (t-2)n_\mu, \\ u_T = n_u, \end{cases}$$

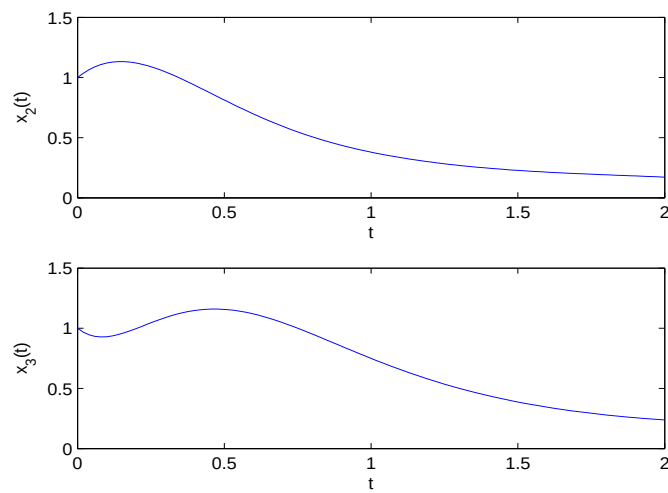
مساله مینیمم‌سازی با تابع هدف (۶۳.۲) تحت قیود (۶۶.۲) را با استفاده از روش تقریبی شبکه عصبی حل می‌کنیم مقدار به دست آمده برای  $J$  برابر  $3/1731$  است و مقدار داده شده در [۶۲] برابر  $3/1017$  است. کنترل بهینه  $u$  و مسیرهای جدید  $x_1, x_2, x_3$  به ترتیب در شکل‌های ۱۴.۲-۱۶.۲ رسم شده است.



شکل ۱۴.۲: نمودار کنترل  $u(t)$  برای مثال ۶.۵.۲



شکل ۱۵.۲: نمودار مسیر  $x_1(t)$  برای مثال ۶.۵.۲



شکل ۱۶.۲: نمودار مسیرهای  $x_2(t)$  و  $x_3(t)$  برای مثال ۶.۵.۲



## فصل ۳

# حل عددی مسائل کنترل بهینه تاخیری افق نامتناهی

### ۱.۳ مقدمه

در این فصل مسائل کنترل بهینه افق نامتناهی<sup>۱</sup> با قیودی به شکل معادلات دیفرانسیل تاخیری<sup>۲</sup> را مورد بررسی قرار می‌دهیم. ابتدا مساله را با استفاده از تقریب پاده مرتبه اول به یک مساله بدون تاخیر تبدیل می‌کنیم و سپس با استفاده از یک تغییر متغیر مناسب<sup>۳</sup> و تبدیل بازه  $[0, \infty)$  به  $[0, 1)$  مساله را به یک مساله افق متناهی<sup>۴</sup> تبدیل می‌کنیم و در انتها با استفاده از روش تقریبی شبکه عصبی جواب عددی مساله را به دست می‌آوریم.

### ۲.۳ معرفی مساله

اخیرا وجود و ساختار جواب مسائل کنترل بهینه روی بازه‌های نامتناهی و بازه‌های خیلی بزرگ مورد مطالعه قرار گرفته است [۶۷]–[۸۱]. این مسائل در مهندسی [۷۹، ۸۰]، مدل‌های رشد اقتصادی<sup>۵</sup> [۸۱] و [۸۲]–[۸۵]، مدل‌های فیزیک حالت جامد<sup>۶</sup> [۸۶، ۸۷] و نظریه ترمودینامیکی<sup>۷</sup> مواد [۸۸]–[۹۰]

---

<sup>۱</sup>Infinite-Horizon Optimal Control Problems

<sup>۲</sup>Time-delayed Differential Equations

<sup>۳</sup>Variation of Variable

<sup>۴</sup>Finite-Horizon Problem

<sup>۵</sup>Models of Economic Growth

<sup>۶</sup>Models of Solid-state Physics

<sup>۷</sup>Theory of Thermodynamical

استفاده می‌شوند. مساله کنترل بهینه افق نامتناهی تاخیری به صورت زیر را در نظر می‌گیریم:

$$\text{minimize } J = \int_0^\infty g(t, x(t), x(t - \sigma), u(t)) dt \quad (1.3)$$

subject to

$$\dot{x}(t) = f(t, x(t), x(t - \sigma), u(t)), \quad (2.3)$$

$$x(t) = \xi(t), \quad -\sigma \leq t \leq 0, \quad (3.3)$$

$$\lim_{t \rightarrow \infty} x(t) = x^1, \quad (4.3)$$

که در آن به ازای هر  $t$ ,  $x(t) = (x_1(t), \dots, x_n(t)) \in Q \subseteq \mathbb{R}^n$  و  $u(t) = (u_1(t), \dots, u_m(t)) \in U \subseteq \mathbb{R}^m$  بترتیب متغیرهای حالت و کنترل هستند و  $Q$  و  $U$  یک مجموعه ناتهی و فشرده در  $\mathbb{R}^n$  و  $\mathbb{R}^m$  هستند. ( ) نشان دهنده مشتق نسبت به متغیر  $t$  و  $\sigma$  تاخیر زمانی مربوط به متغیر حالت  $x$  است.

### ۳.۳ تبدیل مساله تاخیری به یک مساله بدون تاخیر

قبل از این که مساله (۱.۳)-(۴.۳) را به مساله افق متناهی تبدیل کنیم مساله را به یک مساله بدون تاخیر تبدیل می‌کنیم. مانند فصل قبل برای حذف متغیر حالت دارای تاخیر زمانی رابطه زیر را تعریف می‌کنیم:

$$y(t) \triangleq x(t - \sigma),$$

که در آن  $y \in B \subseteq \mathbb{R}^n$  و  $B$  مجموعه‌ای ناتهی و فشرده است. با توجه به روابط (۷.۲)-(۱۶.۲) و (۲.۳) داریم:

$$\dot{y}(t) = \frac{1}{\sigma}[x(t) - y(t)] - \dot{x}(t) = \frac{1}{\sigma}[x(t) - y(t)] - f(t, x(t), y(t), u(t)).$$

در نتیجه مساله کنترل بهینه تاخیری افق نامتناهی (۱.۳)-(۴.۳) به مساله زیر تبدیل می‌شود:

$$\text{minimize } J = \int_0^\infty g(t, x(t), y(t), u(t)) dt$$

subject to

$$\dot{x}(t) = f(t, x(t), y(t), u(t)), \quad (5.3)$$

$$\dot{y}(t) = \frac{1}{\sigma}[x(t) - y(t)] - f(t, x(t), y(t), u(t)), \quad (6.3)$$

با شرایط اولیه

$$x(0) = \xi(0), \quad y(0) = \xi(-\sigma), \quad (7.3)$$

و شرایط نهایی

$$\lim_{t \rightarrow \infty} x(t) = x^1, \quad \lim_{t \rightarrow \infty} y(t) = y^1. \quad (8.3)$$

برای بهتر شدن تقریب همان‌طور که در فصل دوم بیان شد تاخیر زمانی  $\sigma$  را به  $\frac{\sigma}{4}$  تقسیم می‌کنیم و تعریف می‌کنیم:

$$y(t) \triangleq x(t - \frac{\sigma}{4}), \quad (9.3)$$

$$z(t) \triangleq y(t - \frac{\sigma}{4}) = x(t - \sigma). \quad (10.3)$$

که در آن  $z \in D \subseteq \mathbb{R}^n$  و  $D$  مجموعه‌ای ناتهی و فشرده است. با استفاده از روابط (۲۳.۲) و (۲۴.۲) داریم:

$$\dot{y}(t) = \frac{4}{\sigma}[x(t) - y(t)] - f(t, x(t), z(t), u(t)), \quad (11.3)$$

$$\dot{z}(t) = \frac{4}{\sigma}[2y(t) - z(t) - x(t)] + f(t, x(t), z(t), u(t)).$$

بنابراین مساله کنترل بهینه (۱۰.۳)-(۴.۳) به مساله زیر تبدیل می‌شود:

$$\text{minimize } J = \int_0^\infty g(t, x(t), z(t), u(t)) dt \quad (12.3)$$

subject to

$$\dot{x}(t) = f(t, x(t), z(t), u(t)), \quad (13.3)$$

$$\dot{y}(t) = \frac{4}{\sigma}[x(t) - y(t)] - f(t, x(t), z(t), u(t)), \quad (14.3)$$

$$\dot{z}(t) = \frac{4}{\sigma}[2y(t) - z(t) - x(t)] + f(t, x(t), z(t), u(t)), \quad (15.3)$$

با شرایط اولیه

$$x(0) = \xi(0), \quad y(0) = \xi(-\frac{\sigma}{4}), \quad z(0) = \xi(-\sigma), \quad (16.3)$$

و شرایط نهایی

$$\lim_{t \rightarrow \infty} x(t) = x^1, \quad \lim_{t \rightarrow \infty} y(t) = y^1, \quad \lim_{t \rightarrow \infty} z(t) = z^1. \quad (17.3)$$

فرض کنیم تابع برداری  $f: [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  و تابع اسکالر  $g: [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^n \times U \rightarrow \mathbb{R}$  توابع انتگرال پذیر لبگ باشند که بطور پیوسته در  $t$  مشتق پذیرند. رده‌ای از کنترل‌های شدنی برای سیستم (۱۳.۳)-(۱۵.۳) مجموعه توابع اندازه پذیر لبگ  $u: [0, \infty) \rightarrow U$  هستند. مسیر شدنی سیستم (۱۳.۳)-(۱۵.۳) که متناظر با کنترل شدنی  $u$  است، جوابی بصورت  $(x, y, z)$  از معادلات دیفرانسیل (۱۳.۳)-(۱۵.۳) است که روی بازه  $[0, \infty)$  تعریف می‌شود و در شرایط اولیه و نهایی (۱۶.۳) و (۱۷.۳) صدق می‌کند و مقادیر خود را از مجموعه  $Q \times B \times D$  می‌گیرد. فرض می‌کنیم متناظر با هر کنترل شدنی  $u$  مسیر شدنی  $(x, y, z)$  وجود دارد و انتگرال در رابطه (۱۲.۳) بطور مطلق همگراست. یک چهارتایی شدنی  $(\bar{x}(\cdot), \bar{y}(\cdot), \bar{z}(\cdot), \bar{u}(\cdot))$  یک جواب بهینه از مساله (۱۲.۳)-(۱۷.۳) است اگر برای هر چهارتایی شدنی  $(x(\cdot), y(\cdot), z(\cdot), u(\cdot))$  در نامساوی زیر صدق کند:

$$\int_0^\infty g(t, x(t), z(t), u(t)) dt \geq \int_0^\infty g(t, \bar{x}(t), \bar{z}(t), \bar{u}(t)) dt.$$

### ۴.۳ تبدیل مساله افق نامتناهی به مساله افق متناهی

برای حل مساله باید بازه نامتناهی را به متناهی تبدیل کنیم. بنابراین برای تبدیل بازه  $[0, \infty)$  به  $[0, 1)$  از تغییر متغیر زیر استفاده می‌کنیم [۹۴]:

$$t = \tan\left(\frac{\pi}{2}\tau\right), \quad t \in [0, \infty), \quad \tau \in [0, 1). \quad (18.3)$$

بنابراین مساله کنترل بهینه افق نامتناهی (۱۲.۳)-(۱۷.۳) به مساله افق متناهی زیر تبدیل می‌شود:

$$\begin{aligned} \text{minimize } J &= \int_{[0,1)} \frac{\pi}{2} g\left(\tan\left(\frac{\pi}{2}\tau\right), x\left(\tan\left(\frac{\pi}{2}\tau\right)\right), z\left(\tan\left(\frac{\pi}{2}\tau\right)\right), u\left(\tan\left(\frac{\pi}{2}\tau\right)\right)\right) \\ &\quad \sec^2\left(\frac{\pi}{2}\tau\right) d\tau \\ \text{subject to} \end{aligned} \quad (19.3)$$

$$\dot{x}\left(\tan\left(\frac{\pi}{2}\tau\right)\right) = f\left(\tan\left(\frac{\pi}{2}\tau\right), x\left(\tan\left(\frac{\pi}{2}\tau\right)\right), z\left(\tan\left(\frac{\pi}{2}\tau\right)\right), u\left(\tan\left(\frac{\pi}{2}\tau\right)\right)\right), \quad \tau \in [0, 1), \quad (20.3)$$

$$\begin{aligned} \dot{y}\left(\tan\left(\frac{\pi}{2}\tau\right)\right) &= \frac{\varphi}{\sigma} [x\left(\tan\left(\frac{\pi}{2}\tau\right)\right) - y\left(\tan\left(\frac{\pi}{2}\tau\right)\right)] - \\ &\quad f\left(\tan\left(\frac{\pi}{2}\tau\right), x\left(\tan\left(\frac{\pi}{2}\tau\right)\right), z\left(\tan\left(\frac{\pi}{2}\tau\right)\right), u\left(\tan\left(\frac{\pi}{2}\tau\right)\right)\right), \quad \tau \in [0, 1), \end{aligned} \quad (21.3)$$

$$\begin{aligned} \dot{z}\left(\tan\left(\frac{\pi}{2}\tau\right)\right) &= \frac{\varphi}{\sigma} [\varphi y\left(\tan\left(\frac{\pi}{2}\tau\right)\right) - z\left(\tan\left(\frac{\pi}{2}\tau\right)\right) - x\left(\tan\left(\frac{\pi}{2}\tau\right)\right)] + \\ &\quad f\left(\tan\left(\frac{\pi}{2}\tau\right), x\left(\tan\left(\frac{\pi}{2}\tau\right)\right), z\left(\tan\left(\frac{\pi}{2}\tau\right)\right), u\left(\tan\left(\frac{\pi}{2}\tau\right)\right)\right), \quad \tau \in [0, 1), \end{aligned} \quad (22.3)$$

$$x(0) = x^0 = \xi(0), \quad y(0) = y^0 = \xi\left(-\frac{\sigma}{2}\right), \quad z(0) = z^0 = \xi(-\sigma), \quad (23.3)$$

$$\lim_{\tau \rightarrow 1^-} x\left(\tan\left(\frac{\pi}{2}\tau\right)\right) = x^1, \quad \tau \in [0, 1), \quad (24.3)$$

$$\lim_{\tau \rightarrow 1^-} y\left(\tan\left(\frac{\pi}{2}\tau\right)\right) = y^1, \quad \tau \in [0, 1), \quad (25.3)$$

$$\lim_{\tau \rightarrow 1^-} z\left(\tan\left(\frac{\pi}{2}\tau\right)\right) = z^1, \quad \tau \in [0, 1), \quad (26.3)$$

که در آن

$$\begin{aligned} &\left(x\left(\tan\left(\frac{\pi}{2}\tau\right)\right), y\left(\tan\left(\frac{\pi}{2}\tau\right)\right), z\left(\tan\left(\frac{\pi}{2}\tau\right)\right), u\left(\tan\left(\frac{\pi}{2}\tau\right)\right)\right) \\ &\in Q \times B \times D \times U, \quad \tau \in [0, 1). \end{aligned}$$



فرض می‌کنیم:

$$\begin{cases} \mathcal{X}(\tau) = x(\tan(\frac{\pi}{\gamma}\tau)), \\ \mathcal{Y}(\tau) = y(\tan(\frac{\pi}{\gamma}\tau)), \\ \mathcal{Z}(\tau) = z(\tan(\frac{\pi}{\gamma}\tau)), \\ \mathcal{U}(\tau) = u(\tan(\frac{\pi}{\gamma}\tau)). \end{cases} \quad (27.3)$$

بنابراین با استفاده از روابط (27.3) مساله کنترل بهینه (19.3)-(26.3) به مساله زیر تبدیل می‌شود [91]:

$$\text{minimize } J = \int_{[0,1)} \frac{\pi}{\gamma} g(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \sec^{\gamma}(\frac{\pi}{\gamma}\tau) d\tau \quad (28.3)$$

subject to

$$\dot{\mathcal{X}}(\tau) = \frac{\pi}{\gamma} \left( f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau), \quad \tau \in [0, 1), \quad (29.3)$$

$$\begin{aligned} \dot{\mathcal{Y}}(\tau) = & \frac{\pi}{\gamma} \left( \frac{\gamma}{\sigma} [\mathcal{X}(\tau) - \mathcal{Y}(\tau)] - \right. \\ & \left. f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau), \quad \tau \in [0, 1), \end{aligned} \quad (30.3)$$

$$\begin{aligned} \dot{\mathcal{Z}}(\tau) = & \frac{\pi}{\gamma} \left( \frac{\gamma}{\sigma} [2\mathcal{Y}(\tau) - \mathcal{Z}(\tau) - \mathcal{X}(\tau)] + \right. \\ & \left. f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau), \quad \tau \in [0, 1), \end{aligned} \quad (31.3)$$

$$\mathcal{X}(0) = \mathcal{X}^0 = x^0, \quad \mathcal{Y}(0) = \mathcal{Y}^0 = y^0, \quad \mathcal{Z}(0) = \mathcal{Z}^0 = z^0, \quad (32.3)$$

$$\lim_{\tau \rightarrow 1^-} \mathcal{X}(\tau) = \mathcal{X}^1 = x^1, \quad \tau \in [0, 1), \quad (33.3)$$

$$\lim_{\tau \rightarrow 1^-} \mathcal{Y}(\tau) = \mathcal{Y}^1 = y^1, \quad \tau \in [0, 1), \quad (34.3)$$

$$\lim_{\tau \rightarrow 1^-} \mathcal{Z}(\tau) = \mathcal{Z}^1 = z^1, \quad \tau \in [0, 1), \quad (35.3)$$

که در آن

$$(\mathcal{X}(\tau), \mathcal{Y}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \in Q \times B \times D \times U, \quad \tau \in [0, 1).$$

بنا به مساله (28.3)-(35.3) می‌توانیم هامیلتونین را به صورت زیر بسازیم:

$$\begin{aligned} & H(\tau, \mathcal{X}(\tau), \mathcal{Y}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau), \gamma(\tau), \lambda(\tau), \mu(\tau)) \\ &= \frac{\pi}{\gamma} g(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \sec^{\gamma}(\frac{\pi}{\gamma}\tau) \\ &+ \gamma(\tau) \cdot \frac{\pi}{\gamma} \left( f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau) \\ &+ \lambda(\tau) \cdot \frac{\pi}{\gamma} \left( \frac{\gamma}{\sigma} [\mathcal{X}(\tau) - \mathcal{Y}(\tau)] - f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau) \\ &+ \mu(\tau) \cdot \frac{\pi}{\gamma} \left( \frac{\gamma}{\sigma} [2\mathcal{Y}(\tau) - \mathcal{Z}(\tau) - \mathcal{X}(\tau)] + f(\tan(\frac{\pi}{\gamma}\tau), \mathcal{X}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau)) \right) \sec^{\gamma}(\frac{\pi}{\gamma}\tau). \end{aligned}$$

با توجه به  $PMP$  (اصل مینیم پونتریاگین) [۹۵]، اگر  $\mathcal{X}(\tau), \mathcal{Y}(\tau), \mathcal{Z}(\tau)$  متغیرهای حالت،  $\mu(\tau), \gamma(\tau), \lambda(\tau)$  متغیرهای الحاقی و  $\mathcal{U}(t)$  کنترل مساله (۲۸.۳)–(۳۵.۳) باشند، باید در شرایط زیر صدق کنند:

$$\begin{cases} \frac{\partial H}{\partial \mathcal{X}} = -\dot{\gamma}(\tau), \\ \frac{\partial H}{\partial \mathcal{Y}} = -\dot{\lambda}(\tau), \\ \frac{\partial H}{\partial \mathcal{Z}} = -\dot{\mu}(\tau), \\ \frac{\partial H}{\partial \mathcal{X}} = \dot{\mathcal{X}}(\tau), \\ \frac{\partial H}{\partial \mathcal{Y}} = \dot{\mathcal{Y}}(\tau), \\ \frac{\partial H}{\partial \mathcal{Z}} = \dot{\mathcal{Z}}(\tau), \\ \frac{\partial H}{\partial \mathcal{U}} = 0, \end{cases} \quad (۳۶.۳)$$

برای هر  $\tau \in [0, 1]$  که  $H := H(\tau, \mathcal{X}(\tau), \mathcal{Y}(\tau), \mathcal{Z}(\tau), \mathcal{U}(\tau), \gamma(\tau), \lambda(\tau), \mu(\tau))$  دستگاه (۳۶.۳) یک سیستم  $ODE$  می‌دهد که می‌توان آن را با روش‌های عددی حل کرد. در برخی موارد دستگاه (۳۶.۳) یک سیستم  $ODE$  را فراهم می‌کند که به آسانی می‌توان آن را حل کرد اما در بیشتر موارد (مخصوصاً در مسائل کاربردی)، سیستم به راحتی حل نمی‌شود و باید روش‌های تقریبی به کار گرفته شود. در بخش بعد، سعی می‌کنیم توانایی شبکه عصبی را در تخمین تابع برای حل دستگاه (۳۶.۳) به کار ببریم.

## ۵.۳ شبکه عصبی

در شبکه عصبی پیش‌خورد می‌توانیم خروجی شبکه بازگشتی دولایه را به وسیله فرمول زیر محاسبه کنیم:

$$\begin{cases} \text{output} = \sum_{i=1}^I v_i \sigma(z^i), \\ z^i = w_i x + b_i, \end{cases} \quad (۳۷.۳)$$

که  $I$  تعداد سیگموئیدهای واحد است،  $w_i$  پارامتر وزن از ورودی به واحد پنهان  $i$  ام است،  $v_i$  پارامتر وزن  $i$  ام از واحد پنهان  $i$  به لایه خروجی،  $b_i$  بایاس  $i$  ام برای واحد  $i$  ام در لایه پنهان است،  $z^i$  ورودی از واحد پنهان  $i$  ام است و  $\sigma$  یک تابع سیگموئید دلخواه است. تابع انتقال  $\sigma$  به طور کلی یک تابع یک بعدی غیر خطی یکنواخت (تابع فعال) است که می‌تواند به راحتی مورد بررسی قرار گیرد. در مثال‌های عددی ارائه شده در این پایان‌نامه، تابع فعال‌ساز که یک تابع سیگموئید است به صورت زیر در نظر گرفته می‌شود:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (۳۸.۳)$$

هر تابع به اندازه کافی هموار می‌تواند به طور دلخواه در یک مجموعه فشرده و بسته با استفاده از یک شبکه عصبی دو لایه با وزن مناسب تقریب زده شود (برای جزئیات بیشتر [۵۵]–[۵۷] ببینید). این نتیجه نشان می‌دهد که هر تابع پیوسته می‌تواند با استفاده از یک ترکیب خطی از توابع سیگموئید

تقریب زده شود. این به عنوان ویژگی تقریب کلی شبکه‌های عصبی شناخته شده است. با توجه به این حقیقت، از توانایی شبکه‌های عصبی در تقریب تابع، برای تقریب توابع حالت، الحاقی و کنترل برای مساله کنترل بهینه (۲۸.۳)-(۳۵.۳) استفاده می‌کنیم. اکنون طرح تخمینی برای حل دستگاه (۳۶.۳) پیشنهاد می‌کنیم. برای هر تابع، ۷ شبکه عصبی در نظر می‌گیریم: حالت (شبکه‌های عصبی متاظر آن،  $n_x, n_y, n_z$  الحاقی (شبکه‌های عصبی متاظر آن،  $n_\gamma, n_\lambda, n_\mu$  کنترل (شبکه عصبی متاظر آن،  $n_u$ )

$$\begin{cases} n_x = \sum_{i=1}^I v_x^i \bar{\sigma}(z_x^i), & z_x^i = w_x^i \tau + b_x^i, \\ n_y = \sum_{i=1}^I v_y^i \bar{\sigma}(z_y^i), & z_y^i = w_y^i \tau + b_y^i, \\ n_z = \sum_{i=1}^I v_z^i \bar{\sigma}(z_z^i), & z_z^i = w_z^i \tau + b_z^i, \\ n_\gamma = \sum_{i=1}^I v_\gamma^i \bar{\sigma}(z_\gamma^i), & z_\gamma^i = w_\gamma^i \tau + b_\gamma^i, \\ n_\lambda = \sum_{i=1}^I v_\lambda^i \bar{\sigma}(z_\lambda^i), & z_\lambda^i = w_\lambda^i \tau + b_\lambda^i, \\ n_\mu = \sum_{i=1}^I v_\mu^i \bar{\sigma}(z_\mu^i), & z_\mu^i = w_\mu^i \tau + b_\mu^i, \\ n_u = \sum_{i=1}^I v_u^i \bar{\sigma}(z_u^i), & z_u^i = w_u^i \tau + b_u^i, \end{cases} \quad (39.3)$$

برای  $i = 1, 2, \dots, I$  که در آن  $I$  تعداد نرون‌ها است که می‌تواند برای هر شبکه عصبی متفاوت باشد، و  $\bar{\sigma}(z) = (\tau_f - \tau_o)\sigma(z) + \tau_o$  می‌توانیم رابطه بین جواب برای شرایط بهینگی (۳۶.۳) و جواب برای یک مساله مینیم‌سازی بدون محدودیت معادل از طریق جواب‌های آزمایشی ایجاد کنیم [۹۲]، [۹۳]. این تابع آزمایشی به عنوان مجموع دو بخش نوشته شده و بر اساس دو واقعیت ساخته شده است: عبارت اول شامل هیچ پارامترهای قابل تنظیم نیست و فقط شرایط اولیه و انتهایی را ارضا می‌کند. عبارت دوم به کارگیری شبکه عصبی فید-فوروارد خروجی منحصر به فرد با پارامترهای قابل تنظیم و واحدهای ورودی تغذیه شده با بردار ورودی است. لازم به ذکر است عبارت دوم، به ازای زمان اولیه و نهایی صفر می‌شود. بر اساس این واقعیتها و با استفاده از شبکه‌های عصبی ارائه شده در (۳۹.۳)، جواب‌های آزمایشی (برای حالت، الحاقی و تابع کنترل) می‌تواند در ساختارهای زیر تعریف شود:

$$\begin{cases} \mathcal{X}_T = \mathcal{X}_o + (\mathcal{X}_1 - \mathcal{X}_o)(\tau) + (\tau)(\tau - 1)n_x, \\ \mathcal{Y}_T = \mathcal{Y}_o + (\mathcal{Y}_1 - \mathcal{Y}_o)(\tau) + (\tau)(\tau - 1)n_y, \\ \mathcal{Z}_T = \mathcal{Z}_o + (\mathcal{Z}_1 - \mathcal{Z}_o)(\tau) + (\tau)(\tau - 1)n_z, \\ \gamma_T = n_\gamma, \\ \lambda_T = n_\lambda, \\ \mu_T = n_\mu, \\ \mathcal{U}_T = n_u, \end{cases} \quad (40.3)$$

جواب آزمایشی (۴۰.۳) تقریب کلی هستند و باید در شرایط (۳۶.۳) صدق کند. بنابراین داریم:

$$\begin{cases} \frac{\partial H_T}{\partial \mathcal{X}_T} = -\dot{\gamma}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{Y}_T} = -\dot{\lambda}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{Z}_T} = -\dot{\mu}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{X}_T} = \dot{\mathcal{X}}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{Y}_T} = \dot{\mathcal{Y}}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{Z}_T} = \dot{\mathcal{Z}}_T(\tau), \\ \frac{\partial H_T}{\partial \mathcal{U}_T} = 0, \end{cases} \quad (41.3)$$

که در آن  $H_T := H(\tau, \mathcal{X}_T(\tau), \mathcal{Y}_T(\tau), \mathcal{Z}_T(\tau), \mathcal{U}_T(\tau), \gamma_T(\tau), \lambda_T(\tau), \mu_T(\tau))$ . به منظور فرمولبندی (۴۱.۳) به عنوان یک مساله مینیم سازی بدون محدودیت، ما ابتدا سیستم های بهینگی (۴۱.۳) روی  $m$  نقطه  $\tau_k, k = 1, \dots, m$  از بازه  $[t_0, t_f]$  مرتب می کنیم و سپس یک مساله بهینه سازی به صورت زیر تعریف می کنیم

$$\begin{aligned} \text{minimize}_y E(y) = & \frac{1}{m} \sum_{k=1}^m \{E_1(\tau_k, y) + E_2(\tau_k, y) + E_3(\tau_k, y) \\ & + E_4(\tau_k, y) + E_5(\tau_k, y) + E_6(\tau_k, y) + E_7(\tau_k, y)\}, \end{aligned} \quad (42.3)$$

که

$$y = (w_{\mathcal{X}}, w_{\mathcal{Y}}, w_{\mathcal{Z}}, w_{\gamma}, w_{\lambda}, w_{\mu}, w_{\mathcal{U}}, b_{\mathcal{X}}, b_{\mathcal{Y}}, b_{\mathcal{Z}}, b_{\gamma}, b_{\lambda}, b_{\mu}, b_{\mathcal{U}}, v_{\mathcal{X}}, v_{\mathcal{Y}}, v_{\mathcal{Z}}, v_{\gamma}, v_{\lambda}, v_{\mu}, v_{\mathcal{U}})^T \in \mathbb{R}^{VI(\mathcal{E}n+m)}$$

و

$$\begin{cases} E_1(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \mathcal{X}_T} + \dot{\gamma}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_2(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \mathcal{Y}_T} + \dot{\lambda}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_3(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \mathcal{Z}_T} + \dot{\mu}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_4(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \gamma_T} - \dot{\mathcal{X}}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_5(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \lambda_T} - \dot{\mathcal{Y}}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_6(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \mu_T} - \dot{\mathcal{Z}}_T(\tau) \right]^T, & k = 1, 2, \dots, m, \\ E_7(\tau_k, y) = \left[ \frac{\partial H_T}{\partial \mathcal{U}_T} \right]^T, & k = 1, 2, \dots, m, \end{cases} \quad (43.3)$$

لم ۱.۵.۳. اگر

$$y^* = (w_{\mathcal{X}}^*, w_{\mathcal{Y}}^*, w_{\mathcal{Z}}^*, w_{\gamma}^*, w_{\lambda}^*, w_{\mu}^*, w_{\mathcal{U}}^*, b_{\mathcal{X}}^*, b_{\mathcal{Y}}^*, b_{\mathcal{Z}}^*, b_{\gamma}^*, b_{\lambda}^*, b_{\mu}^*, b_{\mathcal{U}}^*, v_{\mathcal{X}}^*, v_{\mathcal{Y}}^*, v_{\mathcal{Z}}^*, v_{\gamma}^*, v_{\lambda}^*, v_{\mu}^*, v_{\mathcal{U}}^*)^T$$

در معادله زیر صدق کند

$$\eta(y) = \begin{bmatrix} E_{\lambda}(\tau_{\lambda}, y) \\ \vdots \\ E_{\lambda}(\tau_m, y) \\ E_{\gamma}(\tau_{\lambda}, y) \\ \vdots \\ E_{\gamma}(\tau_m, y) \\ \vdots \\ E_{\gamma}(\tau_{\lambda}, y) \\ \vdots \\ E_{\gamma}(\tau_m, y) \end{bmatrix} = 0, \quad (44.3)$$

آن‌گاه  $y^*$  یک جواب بهینه از (۴۲.۳) است.

برهان. فرض کنید  $\eta(y^*) = 0$ . آن‌گاه  $E_i(\tau_k, y^*) = 0$  برای  $i = 1, 2, \dots, \gamma$  و  $k = 1, \dots, m$ . چون در (۴۲.۳)  $E(y) \geq 0$  بنابراین  $y^*$  یک جواب بهینه از (۴۲.۳) است.  $\square$

بنابر لم (۱.۵.۳) می‌توان بررسی کرد که مساله بهینه سازی (۴۲.۳) با مساله زیر هم ارز است

$$\text{minimize}_y E(y) = \frac{1}{\gamma} \|\eta(y)\|^2 \quad (45.3)$$

برای حل (۴۵.۳) که یک مساله بهینه سازی نامقید است، می‌توان از الگوریتم‌های بهینه سازی مانند تندترین کاهش، نیوتن، شبه نیوتن، گرادیان مزدوج و غیره استفاده کرد [۵۸]–[۶۰]. همچنین می‌توانیم از الگوریتم‌های ابتکاری مانند الگوریتم ژنتیک، بهینه سازی گروه پرندگان، الگوریتم مورچگان و غیره استفاده کنیم [۶۱].

## ۶.۳ مثال های عددی

در این‌جا کارایی شبکه عصبی برای حل مسائل کنترل بهینه تاخیری افق نامتناهی با دو مثال عددی نشان داده شده است.

مثال ۱۰.۶.۳. مساله کنترل بهینه تاخیری افق نامتناهی زیر را در نظر می‌گیریم:

$$\text{minimize } J = \int_0^\infty \{x_1(t)^2 + x_2(t)^2 + u(t)^2\} dt \quad (46.3)$$

subject to

$$\begin{cases} \dot{x}_1(t) = x_1(t) - u(t), \\ \dot{x}_2(t) = -\frac{1}{2}x_2(t) + \frac{1}{2}x_1(t-1) + 2u(t), \end{cases} \quad (47.3)$$

به‌طوریکه:

$$x_1(t) = e^{(t+1)}, \quad x_2(t) = 0, \quad t \in [-1, 0],$$

$$\lim_{t \rightarrow \infty} x_1(t) = x_1^* = \lim_{t \rightarrow \infty} x_2(t) = x_2^* = 0.$$

همان‌طور که در قیود این مساله ملاحظه می‌کنیم زمان تاخیر مربوط به متغیر حالت  $x_1$  است. برای حذف این تاخیر تعریف می‌کنیم:

$$\begin{cases} x_3(t) \triangleq x_1(t - \frac{1}{2}), \\ x_4(t) \triangleq x_2(t - \frac{1}{2}) = x_1(t - 1). \end{cases} \quad (48.3)$$

با استفاده از روش گفته شده در بخش ۳.۳ و روابط بالا دستگاه معادلات دیفرانسیل تاخیری (۴۷.۳) به دستگاه زیر تبدیل می‌شود:

$$\begin{cases} \dot{x}_1(t) = x_1(t) - u(t), \\ \dot{x}_2(t) = -\frac{1}{2}x_2(t) + \frac{1}{2}x_4(t) + 2u(t), \\ \dot{x}_3(t) = 3x_1(t) - 4x_3(t) + u(t), \\ \dot{x}_4(t) = -3x_1(t) + 8x_3(t) - 4x_4(t) - u(t). \end{cases} \quad (49.3)$$

با شرایط اولیه

$$x_1(0) = e, \quad x_2(0) = 0, \quad x_3(0) = e^{\frac{1}{2}}, \quad x_4(0) = 1, \quad (50.3)$$

و شرایط نهایی

$$\lim_{t \rightarrow \infty} x_1(t) = x_1^* = \lim_{t \rightarrow \infty} x_2(t) = x_2^* = \lim_{t \rightarrow \infty} x_3(t) = x_3^* = \lim_{t \rightarrow \infty} x_4(t) = x_4^* = 0. \quad (51.3)$$

مساله کنترل بهینه افق نامتناهی (۴۹.۳)-(۵۱.۳) با تابع هدف (۴۶.۳) با استفاده از تغییر متغیر (۱۸.۳) به مساله زیر تبدیل می‌شود:

$$\text{minimize } J = \int_{[0,1)} \frac{\pi}{2} (y_1^2(\tau) + y_2^2(\tau) + v^2(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right) d\tau \quad (52.3)$$

subject to

$$\begin{cases} \dot{y}_1(\tau) = \frac{\pi}{\gamma} (y_1(\tau) - v(\tau)) \sec^2\left(\frac{\pi}{\gamma}\tau\right), \\ \dot{y}_2(\tau) = \frac{\pi}{\gamma} (-\mathbb{1}/\mathfrak{L} y_2(\tau) + \circ/\mathfrak{L} y_4(\tau) + \mathfrak{V} v(\tau)) \sec^2\left(\frac{\pi}{\gamma}\tau\right), \\ \dot{y}_3(\tau) = \frac{\pi}{\gamma} (\mathfrak{V} y_1(\tau) - \mathfrak{F} y_3(\tau) + v(\tau)) \sec^2\left(\frac{\pi}{\gamma}\tau\right), \\ \dot{y}_4(\tau) = \frac{\pi}{\gamma} (-\mathfrak{V} y_1(\tau) + \mathfrak{L} y_3(\tau) - \mathfrak{F} y_4(\tau) - v(\tau)) \sec^2\left(\frac{\pi}{\gamma}\tau\right), \end{cases} \quad (53.3)$$

با شرایط اولیه

$$y_1(\circ) = e, \quad y_2(\circ) = \circ, \quad y_3(\circ) = e^{\frac{1}{\gamma}}, \quad y_4(\circ) = \mathbb{1}, \quad (54.3)$$

و شرایط نهایی

$$\begin{aligned} \lim_{\tau \rightarrow \mathbb{1}} y_1(\tau) = y_1^{\mathbb{1}} = \lim_{\tau \rightarrow \mathbb{1}} y_2(\tau) = y_2^{\mathbb{1}} = \\ \lim_{\tau \rightarrow \mathbb{1}} y_3(\tau) = y_3^{\mathbb{1}} = \lim_{\tau \rightarrow \mathbb{1}} y_4(\tau) = y_4^{\mathbb{1}} = \circ. \end{aligned} \quad (55.3)$$

فرم جواب آزمایشی به صورت زیر است:

$$\begin{cases} y_{1T} = e + (\tau)(\mathbb{1} - \tau)n_{y_1} - e(\tau), \\ y_{2T} = \circ + (\tau)(\mathbb{1} - \tau)n_{y_2}, \\ y_{3T} = e^{\frac{1}{\gamma}} + (\tau)(\mathbb{1} - \tau)n_{y_3} - e^{\frac{1}{\gamma}}(\tau), \\ y_{4T} = \mathbb{1} + (\tau)(\mathbb{1} - \tau)n_{y_4} - (\tau), \\ \gamma_{1T} = n_{\gamma_1}, \\ \gamma_{2T} = n_{\gamma_2}, \\ \lambda_T = n_{\lambda}, \\ \mu_T = n_{\mu}, \\ \mathcal{U}_T = n_{\mathcal{U}}, \end{cases}$$

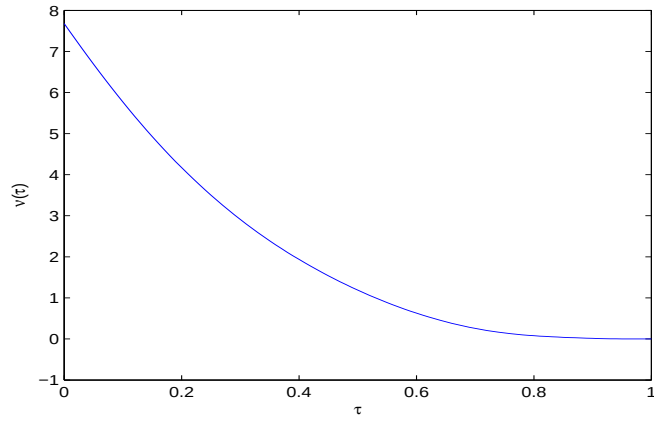
مساله کنترل بهینه افق متناهی (52.3)-(55.3) را با روش شبکه عصبی حل می‌کنیم. مقدار بهینه  $J$  ۳۴/۷۱۵۱ به دست می‌آید. مقدار  $J$  در [۹۴] برابر ۳۵/۲۲۱۴۷ است. نمودار کنترل بهینه  $v(\tau)$  و مسیرهای متناظر  $y_1(\tau)$ ,  $y_2(\tau)$ ,  $y_3(\tau)$  و  $y_4(\tau)$  در شکل‌های ۱.۳-۲.۳ رسم شده است.

مثال ۲.۶.۳. مساله کنترل بهینه مینیم سازی افق نامتناهی تحت قيودی به صورت دستگاه معادلات دیفرانسیل تاخیری

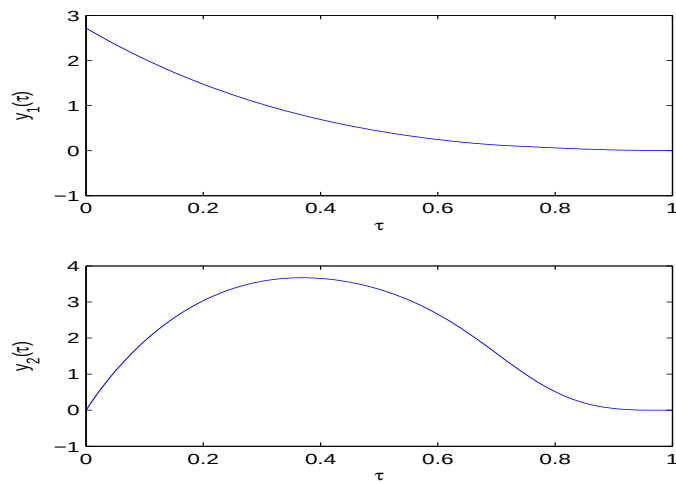
$$\dot{x}(t) = \begin{bmatrix} \mathbb{1} & \circ \\ \circ & -\mathbb{1}/\mathfrak{L} \end{bmatrix} x(t) + \begin{bmatrix} \circ & \circ \\ \mathbb{1} & \circ/\mathfrak{L} \end{bmatrix} x(t - \mathbb{1}) + \begin{bmatrix} -\mathbb{1} \\ \mathfrak{V}/\mathfrak{L} \end{bmatrix} u(t), \quad (56.3)$$

و تابع هدف زیر را در نظر می‌گیریم:

$$\text{minimize } J = \int_0^\infty \{x_1(t)^2 + x_2(t)^2 + u(t)^2\} dt \quad (57.3)$$

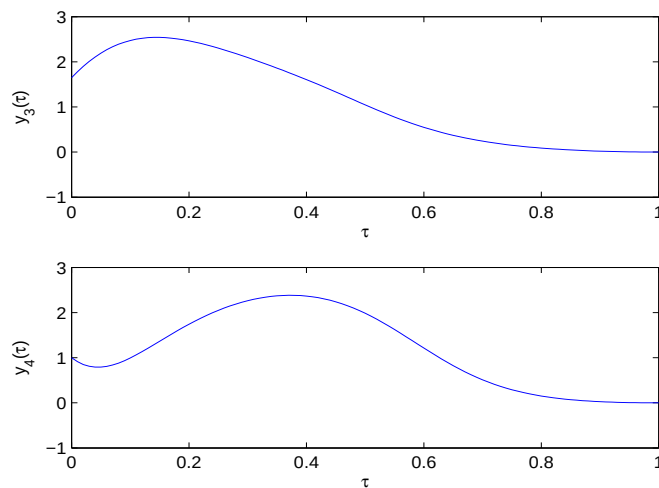


شکل ۱.۳: نمودار کنترل  $v(\tau)$  برای مثال ۱.۶.۳



شکل ۲.۳: نمودار مسیرهای  $y_1(\tau)$  و  $y_2(\tau)$  برای مثال ۱.۶.۳





شکل ۳.۳: نمودار مسیرهای  $y_3(\tau)$  و  $y_4(\tau)$  برای مثال ۱.۶.۳

به طوریکه

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} e^{t+1} \\ 0 \end{bmatrix}, \quad -1 \leq t \leq 0, \\ \lim_{t \rightarrow \infty} x_1(t) = x_1^* = \lim_{t \rightarrow \infty} x_2(t) = x_2^* = 0.$$

زمان تاخیر  $\sigma = 1$  مربوط به متغیر حالت  $x_1$  و  $x_2$  است. با استفاده از روش گفته شده در بخش ۳.۳ داریم:

$$\begin{cases} x_3(t) \triangleq x_1(t - \frac{1}{\tau}), \\ x_4(t) \triangleq x_3(t - \frac{1}{\tau}) = x_1(t - 1), \\ x_5(t) \triangleq x_2(t - \frac{1}{\tau}), \\ x_6(t) \triangleq x_5(t - \frac{1}{\tau}) = x_2(t - 1). \end{cases}$$

بنابراین دستگاه معادلات دیفرانسیل تاخیری (۵۶.۳) به دستگاه زیر تبدیل می شود:

$$\begin{cases} \dot{x}_1(t) = x_1(t) - u(t), \\ \dot{x}_2(t) = -1/5 x_2(t) + x_4(t) + 0/5 x_6(t) + 2/25 u(t), \\ \dot{x}_3(t) = 3x_1(t) - 4x_3(t) + u(t), \\ \dot{x}_4(t) = -3x_1(t) + 8x_3(t) - 4x_4(t) - u(t), \\ \dot{x}_5(t) = 5/5 x_2(t) - x_4(t) - 4x_5(t) - 0/5 x_6(t) - 2/25 u(t), \\ \dot{x}_6(t) = 8x_5(t) - 3/5 x_6(t) - 5/5 x_2(t) + x_4(t) + 2/25 u(t), \end{cases} \quad (58.3)$$

با شرایط اولیه

$$x_1(\circ) = e, \quad x_2(\circ) = \circ, \quad x_3(\circ) = e^{\frac{1}{2}}, \quad x_4(\circ) = 1, \quad x_5(\circ) = \circ, \quad x_6(\circ) = \circ, \quad (59.3)$$

و شرایط نهایی

$$\lim_{t \rightarrow \infty} x_1(t) = x_1^1 = \lim_{t \rightarrow \infty} x_2(t) = x_2^1 = \lim_{t \rightarrow \infty} x_3(t) = x_3^1 = \lim_{t \rightarrow \infty} x_4(t) = x_4^1 = \lim_{t \rightarrow \infty} x_5(t) = x_5^1 = \lim_{t \rightarrow \infty} x_6(t) = x_6^1 = \circ. \quad (60.3)$$

با استفاده از تغییر متغیر (۱۸.۳) مساله کنترل بهینه افق نامتناهی (۵۷.۳)-(۶۰.۳) به مساله زیر تبدیل می‌شود:

$$\begin{aligned} \text{minimize} \quad & J = \int_{[\circ, 1)} \frac{\pi}{2} (y_1^2(\tau) + y_2^2(\tau) + v^2(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right) d\tau \quad (61.3) \\ \text{subject to} \quad & \end{aligned}$$

$$\begin{cases} \dot{y}_1(\tau) = \frac{\pi}{2} (y_1(\tau) - v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \\ \dot{y}_2(\tau) = \frac{\pi}{2} (-1/5 y_2(\tau) + y_4(\tau) + \circ/5 y_6(\tau) + 2/25 v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \\ \dot{y}_3(\tau) = \frac{\pi}{2} (3y_1(\tau) - 4y_3(\tau) + v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \\ \dot{y}_4(\tau) = \frac{\pi}{2} (8y_3(\tau) - 4y_4(\tau) - 3y_1(\tau) - v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \\ \dot{y}_5(\tau) = \frac{\pi}{2} (5/5 y_2(\tau) - y_4(\tau) - 4y_5(\tau) - \circ/5 y_6(\tau) - 2/25 v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \\ \dot{y}_6(\tau) = \frac{\pi}{2} (8y_5(\tau) - 3/5 y_6(\tau) - 5/5 y_2(\tau) + y_4(\tau) + 2/25 v(\tau)) \sec^2\left(\frac{\pi}{2}\tau\right), \end{cases} \quad (62.3)$$

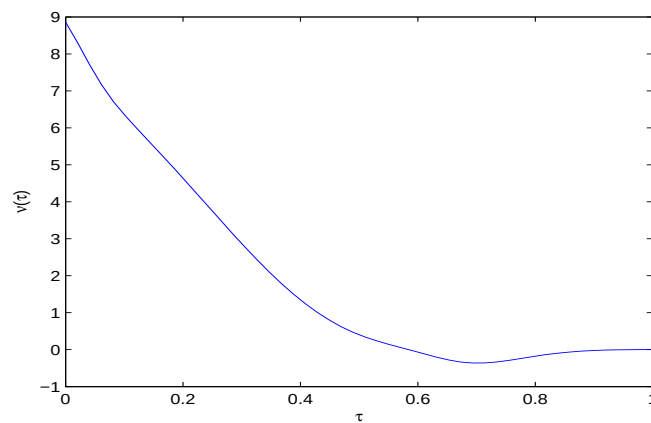
با شرایط اولیه

$$y_1(\circ) = e, \quad y_2(\circ) = \circ, \quad y_3(\circ) = e^{\frac{1}{2}}, \quad y_4(\circ) = 1, \quad y_5(\circ) = \circ, \quad y_6(\circ) = \circ, \quad (63.3)$$

و شرایط نهایی

$$\begin{aligned} \lim_{\tau \rightarrow 1^-} y_1(\tau) = y_1^1 = \lim_{\tau \rightarrow 1^-} y_2(\tau) = y_2^1 = \lim_{\tau \rightarrow 1^-} y_3(\tau) = y_3^1 = \lim_{\tau \rightarrow 1^-} y_4(\tau) = y_4^1 = \lim_{\tau \rightarrow 1^-} y_5(\tau) = y_5^1 = \lim_{\tau \rightarrow 1^-} y_6(\tau) = y_6^1 = \circ. \end{aligned} \quad (64.3)$$

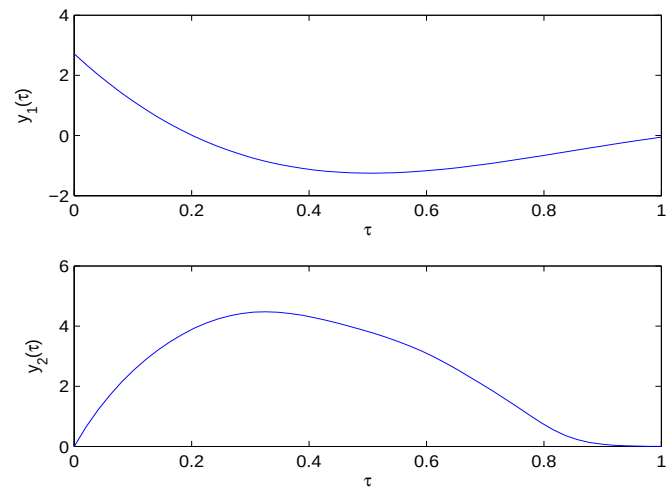
فرم جواب آزمایشی به صورت زیر است:



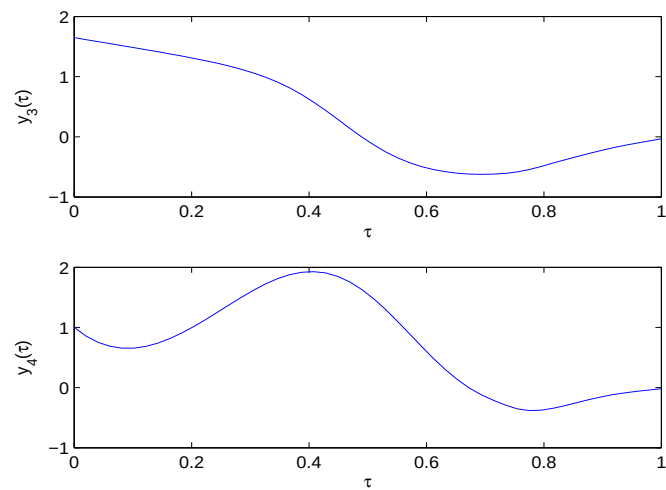
شکل ۴.۳: نمودار کنترل  $v(\tau)$  برای مثال ۲.۶.۳

$$\left\{ \begin{array}{l} y_{1T} = e + (\tau)(1 - \tau)n_{y1} - e(\tau), \\ y_{2T} = (\tau)(1 - \tau)n_{y2}, \\ y_{3T} = e^{\frac{1}{2}} + (\tau)(1 - \tau)n_{y3} - e^{\frac{1}{2}}(\tau), \\ y_{4T} = 1 + (\tau)(1 - \tau)n_{y4} - (\tau), \\ y_{5T} = (\tau)(1 - \tau)n_{y5}, \\ y_{6T} = (\tau)(1 - \tau)n_{y6}, \\ \gamma_{1T} = n_{\gamma1}, \\ \gamma_{2T} = n_{\gamma2}, \\ \lambda_{1T} = n_{\lambda1}, \\ \lambda_{2T} = n_{\lambda2}, \\ \mu_{1T} = n_{\mu1}, \\ \mu_{2T} = n_{\mu2}, \\ \mathcal{U}_T = n_{\mathcal{U}}, \end{array} \right.$$

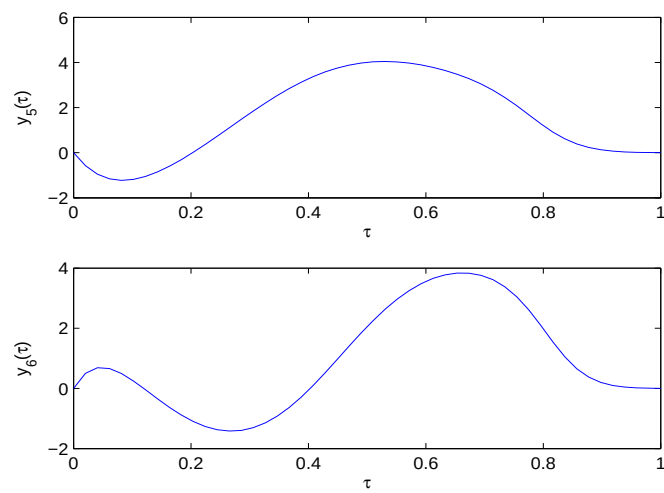
مساله کنترل بهینه افق متناهی (۶۱.۳)-(۶۴.۳) را با روش شبکه عصبی حل می‌کنیم. مقدار بهینه  $J$  ۴۶/۷۶۶۳ به دست می‌آید. مقدار  $J$  در [۹۴] برابر ۴۹/۶۰۰۳۳ است. نمودار کنترل بهینه  $v(\tau)$  و مسیرهای متناظر  $y_1(\tau)$ ,  $y_2(\tau)$ ,  $y_3(\tau)$ ,  $y_4(\tau)$ ,  $y_5(\tau)$  و  $y_6(\tau)$  در شکل‌های ۴.۳-۷.۳ رسم شده است.



شکل ۵.۳: نمودار مسیرهای  $y_1(\tau)$  و  $y_2(\tau)$  برای مثال ۲.۶.۳



شکل ۶.۳: نمودار مسیرهای  $y_3(\tau)$  و  $y_4(\tau)$  برای مثال ۲.۶.۳



شکل ۷.۳: نمودار مسیرهای  $y_5(\tau)$  و  $y_6(\tau)$  برای مثال ۲.۶.۳



# نتیجه‌گیری و پیشنهادات برای کارهای آتی

## نتیجه‌گیری

مسائل کنترل بهینه تاخیر زمانی می‌توانند با استفاده از یک مدل تقریبی شبکه عصبی حل شوند. در این پایان نامه به حل مسائل کنترل بهینه تاخیر زمانی افق متناهی و افق نامتناهی پرداختیم. در مسائل افق متناهی ابتدا با استفاده از تقریب پاده مساله تاخیر زمانی را به یک مساله بدون تاخیر تبدیل کردیم، سپس با استفاده از مدل شبکه عصبی به حل عددی مساله بدون تاخیر پرداختیم. در مسائل افق نامتناهی نیز ابتدا با استفاده از تقریب پاده مساله تاخیر زمانی نامتناهی را به مساله بدون تاخیر نامتناهی تبدیل کردیم، سپس با استفاده از تغییر متغیر مناسب مساله بدون تاخیر نامتناهی را به مساله بدون تاخیر متناهی تبدیل کردیم و در نهایت با استفاده از شبکه عصبی به حل عددی مساله پرداختیم.

## پیشنهادات برای کارهای آتی

- ۱- استفاده از شبکه عصبی برای حل مسائل کنترل Feed back
- ۲- استفاده از شبکه عصبی در بررسی پایداری سیستم‌های کسری





## مراجع

- [۱] حمیرا امیر محمدی، سیمین مسروری، مریم امیر محمدی، حل معادله موج با استفاده از تقریب پاده، مجله ریاضیات کاربردی واحد لاهیجان، (۱۳۸۶) ۱-۱۲
- [۲] ضمیری، آذر. شبکه‌های عصبی مصنوعی و بهینه‌سازی. پایان‌نامه کارشناسی ارشد، دانشکده علوم ریاضی، دانشگاه فردوسی مشهد، آبان ۱۳۸۷.
- [۳] مقدس، محمد، عفتی، سهراب، و اسحاق‌نژاد، محمد. یک شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی نامحدب. در ششمین کنفرانس بین‌المللی
- [۴] مهرجو، احمد، شبکه‌های عصبی مصنوعی، دانشگاه آزاد اسلامی واحد اسلامشهر، اسلامشهر، ۱۳۸۶.
- [5] A. Bouzerdoum and T. R. Pattison. Neural network for quadratic optimization with bound constraints. IEEE Transactions on Neural Networks, (1993), 4(2):293–304.
- [6] A.Cichocki,R.Unbehauen,K.Weizierl and R.Holzl, A new neural network for solving linear programming problems, European Journal of Operational Research,(1996), 93,244- 245.
- [7] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J.L. Huertas and E. Sanchez Sinencio. Nonlinear switched capacitor ‘neural’ networks for optimization problems. ,IEEE Transactions on Circuits and Systems, (1990), 37(3):384–398.
- [8] A. Lapedes, R. Farber, How neural nets work? in: D.Z. Anderson (Ed.), Neural Information Processing Systems, (1988), pp. 442–456.
- [9] B. Widrow. Generalization and information storage in networks of adaline ‘neurons’. in Self Organizing Systems, pp. 435–461. Spartan Books, 1959.
- [10] B. Widrow and M. E. Hoff. Adapting switching circuits. in 1960 ire wescon Convention Record, no. 4, pp. 96–104, New York, (1960).
- [11] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: perceptron, mada-line, and backpropagation, Proceedings of the IEEE, (1990), 78(9):1415–1442.
- [12] B. Widrow and R. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. IEEE Computer Magazine, (1988), 21(3):25–39.

- [13] C.Y. Maa and M. A. Schanblatt, A two-phase optimization neural network. IEEE Transaction on Neural Networks, (1992), 3(6):1003–1009.
- [14] D.Tank and J. J. Hopfield. Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit.IEEE Transactions on Circuits and Systems, (1986), 33(5):533–541.
- [15] D. O. Hebb. The Organization of Behavior: A Neuropsychological Theory. John Wiley & Sons, Inc., New York, (1949).
- [16] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning representations by backpropagating errors. Nature, (1986), 323:533–536.
- [17] F. Rosenblatt. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Report (Cornell Aeronautical Laboratory). Spartan Books, (1962).
- [18] F. Rosenblatt. Neurocomputing: foundations of research. chap. The perception: a probabilistic model for information storage and organization in the brain, pp. 89–114. MIT Press, Cambridge, MA, USA, (1988).
- [19] J. Stanley, Introduction to Neural Networks, third edition, Sierra Mardre, (1990).
- [20] J.J. Mckeown, F. Stella, G. Hall, Some numerical aspects of the training problem for feed-forward neural nets, Neural Networks, (1997), 10(8) 1455–1463.
- [21] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences of the United states of America, (1982), 79(8):2554–2558.
- [22] J. J. Hopfield and D. W.Tank. "neural" computation of decisions in optimization problems. Biological Cybernetics, (1985), 52(3):141–152.
- [23] K. Hornick, M. Stinchcombe, H. white, Multilayer feedforward networks are universal approximators, Neural Networks, (1989), 2 (5) 359–366.
- [24] M. Minsky, S. Papert, Perceptrons, MIT Press, (1969).
- [25] M. P. Kennedy and L. O. Chua. Neural networks for nonlinear programming, IEEE Transactions on Circuits and Systems, (1988), 35(5):554–562.
- [26] M. L. Minsky and S. Papert. Perceptrons: An Introduction to Computational Geometry. Mit Press, (1972).
- [27] M. Moghaddas and S. Effati. A novel recurrent neural network based on ncp function for solving convex nonlinear optimization problems. in The 5th Iranian Conference on Applied Mathematics, Hamadan, Iran, September 2013. Bu-Ali Sina University.

- [28] M. Moghaddas and M. Eshaghnejad. A recurrent neural network for solving non-convex non linear optimization problem. in The 44th Annual Iranian Mathematics Conference, Mashhad, Iran, August 2013. Ferdowsi University of Mashhad.
- [29] P. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University, Cambridge, MA, (1974).
- [30] P. Picton, Neural Networks, second edition., Palgrave, Great Britain, (2000).
- [31] P. Lawrence, Differential Equations and Dynamical Systems, Springer-Verlag, (2001).
- [32] R.J. Schalkoff, Artificial Neural Networks, McGraw-Hill, New York, (1997).
- [33] R.P. Lippmann, An introduction to computing with neural nets, IEEE ASSP Magazine, (1987), 4–22.
- [34] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey, (1999).
- [35] S. Effati, A. R. Nazemi, A new approach for asymptotic stability of the nonlinear ordinary differential equations, Journal of Applied Mathematics and Computing, (2007), 25, 231-244.
- [36] S. Effati and M. A. Baymani. new nonlinear neural network for solving convex nonlinear programming problems, Applied Mathematics and Computation, ( 2005), 168(2):1370 – 1379.
- [37] S. Effati, A. Ghomashi and M. Abbasi. A novel recurrent neural network for solving mlcps and its application to linear and quadratic programming. Asia-Pacific Journal of Operational Research, 28(4):523–541, November (2011).
- [38] S. Effati, A. Ghomashi, and A. R. Nazemi. Application of projection neural network in solving convex programming problems. Applied Mathematics and Computation, (2007), 188(2):1103 – 1114.
- [39] S. Zhang and A.G. Constantinides. Lagrange programming neural networks. IEEE Transactions on Circuits and Systems, II, Analog Digit. Signal Process., (1992), 39(7):441–452.
- [40] T. Khanna, Foundations of Neural Networks, Addison-Wesley, Reading, MA, (1990).
- [41] W. McCulloch, and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, (1943), 5(4):115–133.

- [42] Y. Xia, G. Feng and J. Wang. A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints, *IEEE Transaction on Neural Networks*, (2008), 19(8):1340–1353.
- [43] Y. Xia and J. Wang. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints, *IEEE Transactions on Circuits and Systems*, (2004), 51(7):1385–1394.
- [44] Y. Xia, and J. Wang. A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Transaction on Neural Networks*, (2005), 16(2):379–386.
- [45] Y. Xia, H. Leung and J. Wang. A projection neural network and its application to constrained optimization problems. *IEEE Transactions on Circuits and Systems*. (2002), 49(4):447–458.
- [46] D. N. Burghes, A. Graham, *Introduction to Control Theory*, E. Horward, Halsted Press: New Yourk. (1980),
- [47] D. R. Kincaid, E. Ward Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, Cengage Learning. (2001),
- [48] I. R. Horng, J. H. Chou, Analysis, parameter estimation and optimal control of time-delay systems via Chebyshev series, *International Journal of Control*, (1985), 41, 1221-1234.
- [49] H. R. Karimi, P. Jabejdar Maralani, B. Lohmann, B. Moshiri, H1 control of linear parameterdependent state-delayed systems using polynomial parameter-dependent quadratic functions, *International Journal of Control*, (2005), 78, 254-263.
- [50] H. R. Karimi, M. J. Yazdanpanah, Robust control for a class of uncertain state-delayed singularly perturbed systems, *Asian Journal of Control*, (2005), 7, 202- 208.
- [51] F. C. Kung, H. Lee, Solution and parameter estimation of in linear timeinvariant delay systems using Laguerre polynomial expansion, *Transactions of the ASME Journal of Dynamic Systems and Measurement Control*, (1983), 105, 297-301.
- [52] H. Lee, F. C. Kung, Shifted Legendre series solution and parameter estimation of linear delayed systems, *International Journal of Systems Science*, (1985), 16, 1249-1256.
- [53] H. R. Marzban, M. Razzaghi, Solution of time-varying delay systems by hybrid functions, *Mathematics and Computers in Simulation*, (2004), 64, 597-607.
- [54] G. P. Rao, L. Sivakumar, Analysis and synthesis of dynamic systems containing time-delays via block-pulse functions, *Proceedings of the IEEE*, (2014), 125, 1064-1068.

- [55] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics Of Control, Signals and Systems*. (1989), 2, 303-314.
- [56] K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks*, (1989), 2, 183-192.
- [57] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, (1989), 2, 359-366.
- [58] Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear Programming-Theory and Algorithms*. Third edition, John Wiley & Sons, Hoboken, New Jersey.
- [59] X-S. Zhang, *Neural Networks in Optimization*, Kluwer Academic Publishers, Dordrecht, Boston, London, (2000).
- [60] J. Nocedal, S. Wright, *Numerical Optimization* (second ed.) Berlin, New York: Springer-Verlag, (2006).
- [61] K. Y. Lee, M. A. El-Sharkawi, *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*, (2008).
- [62] H. T. Banks, J. A. Burns, Hereditary control problem: numerical methods based on averaging approximations, *SIAM Journal on Control and Optimization*, (1978), 16, 169-208.
- [63] H. T. Banks, Approximation of nonlinear functional differential equation control systems, *Journal of Optimization Theory and Applications*, (1979), 29, 383-408.
- [64] K. H. Wong, D. J. Clements, K. L. Teo, Optimal control computation for nonlinear time-lag systems, *Journal of Optimization Theory and Applications*, (1985), 47, 91-107.
- [65] K. L. Teo, K. H. Wong, D. J. Clements, Optimal control computation for linear time-lag systems with linear terminal constraints, *Journal of Optimization Theory and Applications*, (1984), 44, 509-526.
- [66] H. R. Marzban, M. Razzaghi, Optimal control of linear delay systems via hybrid of block-pulse and Legendre polynomials, *Journal of the Franklin Institute*, (2004), 341, 279-293.
- [67] J. Blot, Infinite-horizon Pontryagin principles without invertibility, *Journal of Nonlinear and Convex Analysis*, (2009), 10, 177-189.
- [68] J. Blot, P. Cartigny, Optimality in infinite-horizon variational problems under sign conditions, *Journal of Optimization Theory and Applications*, (2000), 106, 411-419.

- [69] H. Jasso-Fuentes, O. Hernandez-Lerma, Characterizations of overtaking optimality for controlled diffusion processes, *Applied Mathematics and Optimization*, (2008), 57, 349-369.
- [70] A. Leizarowitz, Infinite horizon autonomous systems with unbounded cost, *Applied Mathematics and Optimization*, (1985), 13, 19-43.
- [71] B. Mordukhovich, Minimax design for a class of distributed parameter systems, *Automatic Remote Control*, (1990), 50, 1333-1340.
- [72] B. Mordukhovich, I. Shvartsman, Optimization and feedback control of constrained parabolic systems under uncertain perturbations. In: *Optimal Control, Stabilization and Nonsmooth Analysis, Lecture Notes in Control and Information Sciences*, 121-132. Springer, Berlin .(2004),
- [73] S. Pickenhain, V. Lykina, Sufficiency conditions for infinite horizon optimal control problems. In: *Recent Advances in Optimization, Proceedings of the 12th French-German-Spanish Conference on Optimization, Avignon*, pp. 217-232. Springer, Berlin .(2006),
- [74] S. Pickenhain, V. Lykina, M. Wagner, On the lower semicontinuity of functionals involving Lebesgue or improper Riemann integrals in infinite horizon optimal control problems, *Control and Cybernetics*, (2008), 37, 451-468.
- [75] T. Prieto-Rumeau, O. Hernandez-Lerma, "Bias and overtaking equilibria for zero-sum continuous time Markov games", *Mathematical Methods of Operations Research*, (2005), 61, 437-454.
- [76] A. Rapaport, P. Cartigny, Turnpike theorems by a value function approach, *ESIAM: Control, Optimisation and Calculus of Variations*, (2004), 10, 123-141.
- [77] A. Rapaport, P. Cartigny, Nonturnpike optimal solutions and their approximations in infinite horizon, *Journal of Optimization Theory and Applications*, (2007), 134, 1-14.
- [78] A. J. Zaslavski, *Turnpike Properties in the Calculus of Variations and Optimal Control*, Springer, New York. (2006),
- [79] B. D. O. Anderson, J. B. Moore, *Linear Optimal Control*. Prentice- Hall, Englewood Cliffs. (1971),
- [80] A. Leizarowitz, Tracking nonperiodic trajectories with the overtaking criterion, *Applied Mathematics and Optimization*, (1985), 14, 155-171.
- [81] A. J. Zaslavski, *Turnpike Properties in the Calculus of Variations and Optimal Control*, Springer, New York. (2006),

- [82] D. Gale, On optimal development in a multi-sector economy, *The Review of Economic Studies*,(1967), 34, 1-18.
- [83] V. L. Makarov, A. M. Rubinov, *Mathematical Theory of Economic Dynamics and Equilibria*, Springer, Berlin.(1977),
- [84] A. J. Zaslavski, Turnpike results for a discrete-time optimal control system arising in economic dynamics, *Nonlinear Analysis*,(2007), 67, 2024-2049.
- [85] A. J. Zaslavski, Two turnpike results for a discrete-time optimal control systems, *Nonlinear Analysis*,(2009), 71, 902-909.
- [86] S. Aubry, P. Y. Le Daeron, The discrete Frenkel-Kontorova model and its extensions: I. Exact results for the ground-states, *Physica D*,(1983), 8, 381-422.
- [87] A. J. Zaslavski, Ground states in Frenkel-Kontorova model, *Mathematics of the USSR, Izvestiya*,(1987), 29, 323-354.
- [88] B. D. Coleman, M. Marcus, V. J. Mizel, On the thermodynamics of periodic phases, *Archive for Rational Mechanics and Analysis*,(1992), 117, 321-347.
- [89] A. Leizarowitz, V. J. Mizel, One dimensional infinite horizon variational problems arising in continuum mechanics, *Archive for Rational Mechanics and Analysis*,(1989), 106, 161-194.
- [90] M. Marcus, A. J. Zaslavski, The structure of extremals of a class of second order variational problems, *Annales de Institut Henri Poincaré, Analysis Non Linéaire*,(1999), 16, 593-629.
- [91] M. Marcus, A. J. Zaslavski, The structure of extremals of a class of second order variational problems, *Annales de Institut Henri Poincaré, Analysis Non Linéaire*,(1999), 16, 593-629.
- [92] A. Malek, R. Shekari Beidokhti, Numerical solution for high order differential equations using a hybrid neural network Optimization method , *Applied Mathematics and Computation*,(2006), 183 260-271
- [93] R. Shekari Beidokhti, A. Malek, Solving initial-boundaryvalueproblemsforsystems of partial differential equations using neural networks and optimization techniques, *Jour- nal of the Franklin Institute*, (2009), 346 898-913
- [94] M.Mansoori, A.Nazemi, Solving infinite-horizon optimal control problemsn of the time-delayed systems by Haar wavelet collocation method, *Sociedade Brasileira de Matemática Aplicada e Computacional* , (2014), DOI 10.1007/s40314-014-0184-1

- [95] D.Burghes, A. Graham. Introduction to control theory including optimal control.  
E.Horwood, (1980).



# واژه‌نامه فارسی به انگلیسی

|                                      |                         |
|--------------------------------------|-------------------------|
| Pontryagin Minimum Principle .....   | اصل مینیمم پونتریاگین   |
| Optimization Algorithm .....         | الگوریتم‌های بهینه سازی |
| Genetic Algorithm .....              | الگوریتم ژنتیک          |
| Steepest Descent Algorithm .....     | الگوریتم تندترین کاهش   |
| Optimization .....                   | بهینه سازی              |
| Non-delay .....                      | بدون تاخیر              |
| Control Function .....               | تابع کنترل              |
| Errore Function .....                | تابع خطا                |
| Activation function .....            | تابع فعال               |
| First-order Páde Approximation ..... | تقریب پاده مرتبه اول    |
| Trial Solution .....                 | جواب‌های آزمایشی        |
| Optimal Solution .....               | جواب بهینه              |
| exact Solution .....                 | جواب دقیق               |
| Neuron System .....                  | دستگاه عصبی             |
| Newton Methods .....                 | روش نیوتن               |
| Quasi Newton Methods .....           | روش شبه نیوتن           |
| Neural Network .....                 | شبکه عصبی               |
| Artificial Neural Network .....      | شبکه عصبی مصنوعی        |
| Biological Neural Network .....      | شبکه عصبی بیولوژیکی     |
| Optimality Conddition .....          | شرایط بهینگی            |
| Initial and Boundary Condition ..... | شرایط اولیه و مرزی      |
| Lagrangian Multipliers .....         | ضرایب لاگرانژ           |
| Optimal Control .....                | کنترل بهینه             |
| Ordinary Differential Equation ..... | معادله دیفرانسیل معمولی |
| State Variable .....                 | متغیر وضعیت (حالت)      |

---

|                                 |                    |
|---------------------------------|--------------------|
| Control Variable.....           | متغیر کنترل        |
| Infinite-Horizon Problems ..... | مساله افق نامتناهی |
| Finite-Horizon Problem.....     | مساله افق متناهی   |

# واژه‌نامه انگلیسی به فارسی

|                                    |                             |
|------------------------------------|-----------------------------|
| Activation function                | تابع فعال                   |
| Artificial Neural Network          | شبکه عصبی مصنوعی            |
| Biological Neural Network          | شبکه عصبی بیولوژیکی         |
| Control Function                   | تابع کنترل                  |
| Constraint                         | محدودیت                     |
| Control Variable                   | متغیر کنترل                 |
| Closely Set                        | مجموعه بسته                 |
| Compact Set                        | مجموعه فشرده                |
| Controllability                    | کنترل‌پذیری                 |
| Differential Equations             | معادلات دیفرانسیل           |
| Errore Function                    | تابع خطا                    |
| exact solution                     | جواب دقیق                   |
| Finite-Horizon Problem             | مساله افق متناهی            |
| First-Order Differential Equations | معادلات دیفرانسیل مرتبه اول |
| First-order Páde Approximation     | تقریب پاده مرتبه اول        |
| Genetic Algorithm                  | الگوریتم ژنتیک              |
| Infinite-Horizon Problems          | مساله افق نامتناهی          |
| Initial and Boundary Condition     | شرایط اولیه و مرزی          |
| Inverse Laplace                    | لاپلاس معکوس                |
| Lagrangian Multipliers             | ضرایب لاگرانژ               |
| Laplace Transformation             | تبدیل لاپلاس                |
| neural network                     | شبکه عصبی                   |
| Neuron System                      | دستگاه عصبی                 |
| Newton Methods                     | روش نیوتن                   |
| Non-delay                          | بدون تاخیر                  |

|                                      |                         |
|--------------------------------------|-------------------------|
| Optimization .....                   | بهینه سازی              |
| Optimal Solution .....               | جواب بهینه              |
| Optimality Conddition .....          | شرایط بهینگی            |
| Optimal Control .....                | کنترل بهینه             |
| Ordinary Differential Equation ..... | معادله دیفرانسیل معمولی |
| Pontryagin Minimum Principle .....   | اصل مینیم پونتریاگین    |
| Quasi Newton Methods .....           | روش شبه نیوتن           |
| Steepest Descent Algorithm .....     | الگوریتم تندترین کاهش   |
| State Variable .....                 | متغیر وضعیت (حالت)      |
| Trial Solution .....                 | جواب‌های آزمایشی        |

## **Aabstract**

In this thesis, we solve a class of time-delayed optimal control problems using neural network. In the first chapter, we introduce to introduction and neural network and the needed definitions. In the second chapter, we study Bolza optimal control problems with time-delayed. The problem is first transformed, using a Pade approximation, to one without a time-delayed argument. Then we approximate it using neural network. Numerical results are also given for six test examples. The third chapter consists of infinite-horizon time-delayed optimal control problem. The problem is first transformed, using a Pade approximation, to one without a time-delayed argument. Then by a suitable change of variable, we transformed the obtained non-delay optimal control problem to a finite-horizon optimal control problem. We solve it numerically using neural network. Numerical results are also given for two test examples.

**keywords:** time-delayed optimal control problems , neural network, Pade approximation, finite-horizon optimal control problem



**University of Shahrood**

**Faculty of Mathematical Sciences**

**Solving a class of time delayed optimal  
control problems using a neural network  
model**

**Ensieh Fayyazi**

**Supervisor**

**Alireza Nazemi**

**October 2015**