

وزارت علوم، تحقیقات و فناوری  
**مرکز آموزش علمی – کاربردی پارس آباد 1**  
تحت نظارت دانشگاه جامع – علمی کاربردی

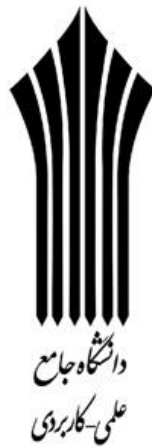
پایان نامه دوره کاردانی ناپیوسته ترمی رشته نرم افزار کامپیوتر

عنوان پروژه :  
شبکه های عصبی

پروژه دات کام  
**www.Prozhe.com**

نگارش :  
مهدی عزیزی

تاریخ تنظیم  
نیمسال اول – پاییز 91-90



**Ministry of Science, Research and Technology of Islamic Republic of IRAN**  
**University of Applied Science and Technology**  
**Parsabad1 College**

Course  
*Associate degrees of Computer Software*

Project Title  
***Neural network***

Supervisor  
***M.babavand***

Authors  
***Mehdi azizi***

*First Semester - Fall 90-91*

## ارزشیابی توسط اساتید

این گزارش در تاریخ ..... مورد ارزیابی قرار گرفته و کار انجام شده از نظر مضمون و کیفیت به عنوان دو/سه واحد پروژه کاردانی (ترمی) نرم افزار پذیرفته شده است.

ردیف	عنوان	نام و نام خانوادگی	امضاء
1	استاد پروژه	مهرداد باباوند	
2	مسئول کمیته تخصیص پروژه		
3	استاد داور		
4	عضو کمیته تخصیص پروژه		

## سپاسگزاری

با سپاس و قدردانی فراوان از استاد بزرگوارم مهرداد باباوند ، ریاست محترم و مسئول

آموزش که همواره از راهنمایی‌های ایشان بهره‌مند شدم

## تقدیم به

پدر بزرگوار و مادر مهربانم  
آن دو فرشته ای که از خواسته  
هایشان گذشتند ، سختی ها را به  
جان خریدند و خود را سپر بلای  
مشکلات و ناملایمات کردند تا من به  
جایگاهی که اکنون در آن ایستاده ام  
برسم.

## فهرست مطالب

عنوان	صفحه
چکیده فارسی.....	1
مقدمه.....	2
<b>فصل اول</b>	
شبکه عصبی.....	4
1-1- مقدمه.....	4
1-2- شبکه عصبی.....	4
1-3- سابقه تاریخی.....	4
1-4- چرا از شبکه های عصبی استفاده می کنیم.....	5
1-5- مزیت های دیگر شبکه های عصبی.....	5
1-6- شبکه های عصبی در مقابل کامپیوتر های معمولی.....	6
1-7- انسان و سلول های عصبی مصنوعی - در جستجوی شباهت ها.....	7
1-8- از سلول های عصبی انسانی تا سلول های عصبی مصنوعی.....	8
1-9- انواع یادگیری برای شبکه های عصبی.....	8
1-10- زمینه ای در مورد perceptron.....	10
1-11- دنباله های Perceptron.....	11
1-12- قضیه بنیادی دنباله ها.....	12
1-13- هوش جمعی.....	13
1-14- Particle Swarm Optimisation (PSO).....	14
<b>فصل دوم</b>	
یک شبکه عصبی جدید و کاربرد آن.....	17
2-1- مقدمه.....	17
2-2- یک شبکه عصبی جدید و کاربرد آن.....	17
2-3- معرفی.....	17
2-4- نورون با خاصیت آشوبگونه.....	18

19	2-5- شکل شبکه
21	2-6- قانون آموزش شبکه
24	2-7- مدلسازی ژنراتور سنکرون دریایی
28	2-8- نتایج مدلسازی
29	2-9- نتیجه فصل
	فصل سوم
31	آنالیز رفتار آشوبگونه مدل شبکه عصبی مکانیسم لرزش عضله
31	3-1- مقدمه
31	3-2- آنالیز رفتار آشوبگونه مدل شبکه عصبی مکانیسم لرزش عضله
31	3-3- معرفی
32	3-4- منحنی طول - کشش
34	3-5- ساختار برگشتی
36	3-6- تغییرات طیف
37	3-7- نتایج فصل
	فصل چهارم
40	هماهنگ سازی نمایی شبکه های عصبی آشوبگونه با اغتشاش تصادفی
40	4-1- مقدمه
40	4-2- هماهنگ سازی نمایی شبکه های عصبی آشوبگونه با اغتشاش تصادفی
40	4-3- معرفی
41	4-4- نمادها و مقدمات
42	4-5- نتیجه فصل
45	فهرست منابع (ابتدا منابع فارسی و سپس منابع غیر فارسی)
45	منابع فارسی
46	منابع لاتین
47	چکیده انگلیسی

## فهرست جداول

جدول	صفحه
شکل 1-2: نوروں آشوب گونه .....	18
شکل 2-2: cmn .....	21
شکل 2-3: سیستم شناسایی ژنراتور سنکرون دریایی به وسیله شبکه عصبی .....	26
شکل 2-4: توان گشتاور ورودی و فرکانس خروجی ژنراتور .....	27
شکل 2-5: جریان تحریک ورودی و ولتاژ خروجی پایانه .....	28
شکل 2-6: منحنی mse هنگام trainin شبکه .....	29
شکل 2-7: فرکانس خروجی ژنراتور ، شبکه و خطای بین آنها .....	29
شکل 3-1: تابع کشش - طول فعال .....	33
شکل 3-2: شبکه برگشتی .....	34
شکل 3-3: نمودار دو شاخه شدن .....	35
شکل 3-4: : نمودار دو شاخه شدن ۹۴۵&۱ نسبت ۹۴۵&۲ .....	36
شکل 3-5: حساسیت به شرط اولیه .....	37
شکل 3-6: ایجاد طیف پیوسته از طریق افزایش فرکانس های گسسته .....	38
شکل 4-1: نرخ هماهنگ سازی نمایی سیستم با خطای دینامیک .....	43
شکل 4-2: دینامیک های سنکرون نشده در فضای حالت .....	44



## چکیده

مسئله هماهنگ سازی شبکه های عصبی آشوبگونه با اغتشاش تصادفی مورد بررسی قرار گرفته است. مسئله کنترل و هماهنگ سازی این سیستم ها به شدت مورد توجه قرار گرفت و روش های مختلفی مثل کنترل حلقه بسته خطی و غیرخطی ، کنترل تطبیقی و نظایر آن جهت رسیدن به این هدف ، ارائه گردید . در مدلسازی سیستم ها ، توانایی تقریب به وسیله نوروں ها ، شکل شبکه و قانون آموزش ، محدود می گردد. چگونگی بهبود خاصیت ارگادیک ANN یک مسئله مهم جهت تحقیق و بررسی می باشد. سیستم های آشوبگونه دارای مشخصاتی تصادفی هستند و الگوریتم آشوبی باعث ایجاد خاصیت قوی ارگادیک در شبکه می گردد.

واژگان کلیدی : شبکه های عصبی ، هوش جمعی ، نوروں ، مدلسازی ژنراتور سنکرون دریایی ، perceptron

## مقدمه

این مقاله مقدمه ای بر شبکه های عصبی مصنوعی است. گونه های مختلف شبکه های عصبی توضیح و شرح داده شده است و کاربرد های شبکه های عصبی، نظیر ANN ها در پزشکی بیان شده و همچنین سابقه ای تاریخی از آن به تفصیل آورده شده است. همچنین رابطه بین چیزهای ساختگی و واقعی مورد بررسی قرار گرفته و در مورد آن توضیح داده شده است و به شرح مدل های ریاضی در رابطه با این موضوع و آنالیز رفتار آشوبگونه مدل شبکه عصبی مکانیسم لرزش عضله و هماهنگ سازی نمایی شبکه های عصبی آشوبگونه با اغتشاش تصادفی و شناسایی شبکه های آشوبگونه آغشته به نویز بر مبنای شبکه های عصبی feedforward رگولاریزاسیون و همچنین شبکه های عصبی و الگوریتم های ژنتیک در تجارت می پردازیم.

## فصل اول

www.Prozhe.com

## شبکه عصبی

### 1-1- مقدمه

یک شبکه عصبی مصنوعی ایده ای است برای پردازش اطلاعات که از سیستم عصبی زیستی الهام گرفته شده و مانند مغز به پردازش اطلاعات می پردازد. عنصر کلیدی این ایده، ساختار جدید سیستم پردازش اطلاعات است اولین سلول عصبی مصنوعی در سال 1943 بوسیله یک neurophysiologist به نام Warren McCulloch و یک منطق دان به نام Walter Pitts ساخته شد. اما محدودیتهای تکنولوژی در آن زمان اجازه کار بیشتر به آنها نداد.

### 1-2- شبکه عصبی

یک شبکه عصبی مصنوعی (Artificial Neural Network (ANN)) ایده ای است برای پردازش اطلاعات که از سیستم عصبی زیستی الهام گرفته شده و مانند مغز به پردازش اطلاعات می پردازد. عنصر کلیدی این ایده، ساختار جدید سیستم پردازش اطلاعات است. این سیستم از شمار زیادی عناصر پردازشی فوق العاده بهم پیوسته تشکیل شده (neurons) که برای حل یک مسأله با هم هماهنگ عمل می کند. ANN ها، نظیر انسانها، با مثال یاد می گیرند. یک ANN برای انجام وظیفه های مشخص، مانند شناسایی الگوها و دسته بندی اطلاعات، در طول یک پروسه یاد گیری، تنظیم می شود. در سیستم های زیستی یاد گیری با تنظیماتی در اتصالات سیناپسی که بین اعصاب قرار دارد همراه است. این روش ANN ها هم می باشد.

### 1-3- سابقه تاریخی

به نظر می آید شبیه سازی های شبکه عصبی یکی از پیشرفت های اخیر باشد. اگرچه این موضوع پیش از ظهور کامپیوترها بنیان گذاری شده و حداقل یک مانع بزرگ تاریخی و چندین دوره مختلف را پشت سر گذاشته است. خیلی از پیشرفت های مهم با تقلیدها و شبیه سازی های ساده و ارزان کامپیوتری بدست آمده است. در پی یک دوره ابتدائی اشتیاق و فعالیت در این زمینه، یک دوره بی میلی و بدنامی راهم پشت سر گذاشته است.

در طول این دوره سرمایه گذاری و پشتیبانی حرفه ای از این موضوع در پایین ترین حد خود بود ، پیشرفت های مهمی به نسبت تحقیقات محدود در این زمینه صورت گرفت . که بدین وسیله پیشگامان قادر شدند تا به گسترش تکنولوژی متقاعد کننده ای بپردازند که خیلی برجسته تر از محدودیت هایی بود که توسط Minsky و Papert شناسانده شد. Minsky و Papert ، کتابی را در سال 1969 منتشر کردند که در آن عقیده عمومی راجع به میزان محرومیت شبکه های عصبی را در میان محققان معین کرده بود و بدین صورت این عقیده بدون تجزیه و تحلیل های بیشتر پذیرفته شد. هم اکنون ، زمینه تحقیق شبکه های عصبی از تجدید حیات علایق و متناظر با آن افزایش سرمایه گذاری لذت می برد .

اولین سلول عصبی مصنوعی در سال 1943 بوسیله یک neurophysiologist به نام Warren McCulloch و یک منطق دان به نام Walter Pitts ساخته شد . اما محدودیتهای تکنولوژی در آن زمان اجازه کار بیشتر به آنها نداد.

#### 1-4- چرا از شبکه های عصبی استفاده می کنیم؟

شبکه های عصبی ، با قابلیت قابل توجه آنها در استنتاج معانی از داده های پیچیده یا مبهم ، میتواند برای استخراج الگوها و شناسایی روشهایی که آگاهی از آنها برای انسان و دیگر تکنیک های کامپیوتری بسیار پیچیده و دشوار است به کار گرفته شود. یک شبکه عصبی تربیت یافته می تواند به عنوان یک متخصص در مقوله اطلاعاتی ای که برای تجزیه تحلیل به آن داده شده به حساب آید. از این متخصص می توان برای بر آورد وضعیت های دلخواه جدید و جواب سؤال های " چه می شد اگر " استفاده کرد.

#### 5-1- مزایای دیگر شبکه های عصبی

یادگیری انطباق پذیر: قابلیت یاد گیری نحوه انجام وظایف بر پایه اطلاعات داده شده برای تمرین و تجربه های مقدماتی.

سازماندهی توسط خود: یک ANN می تواند سازماندهی یا ارائه اش را ، برای اطلاعاتی که در طول دوره یادگیری در یافت می کند، خودش ایجاد کند. عملکرد بهنگام (Real time) : محاسبات ANN می تواند

بصورت موازی انجام شود، و سخت افزارهای مخصوصی طراحی و ساخته شده است که می تواند از این قابلیت استفاده کند.

تحمیل اشتباه بدون ایجاد وقفه در هنگام کد گذاری اطلاعات : خرابی جزئی یک شبکه منجر به تنزل کارایی متناظر با آن می شود اگر چه تعدادی از قابلیت های شبکه ممکن است حتی با خسارت بزرگی هم باقی بماند.

## 6-1- شبکه های عصبی در مقابل کامپیوتر های معمولی

شبکه های عصبی نسبت به کامپیوتر های معمولی مسیر متفاوتی را برای حل مسئله طی می کنند . کامپیوتر های معمولی یک مسیر الگو ریتمی را استفاده می کنند به این معنی که کامپیوتر یک مجموعه از دستورالعمل ها را به قصد حل مسئله پی می گیرد. بدون اینکه، قدم های مخصوصی که کامپیوتر نیاز به طی کردن دارد، شناخته شده باشند کامپیوتر قادر به حل مسئله نیست. این حقیقت قابلیت حل مسئله ی کامپیوتر های معمولی را به مسائلی، محدود می کند که ما قادر به درک آنها هستیم و می دانیم چگونه حل میشوند. اما اگر کامپیوتر ها می توانستند کار هایی را انجام دهند که ما دقیقاً نمیدانیم چگونه انجام دهیم ، خیلی پر فایده تر بودند.

شبکه های عصبی اطلاعات را به روشی مشابه با کاری که مغز انسان انجام می دهد پردازش می کنند. آنها از تعداد زیادی از عناصر پردازشی (سلول عصبی) که فوق العاده بهم پیوسته اند تشکیل شده است که این عناصر به صورت موازی باهم برای حل یک مسئله مشخص کار می کنند. شبکه های عصبی با مثال کار می کنند و نمی توان آنها را برای انجام یک وظیفه خاص برنامه ریزی کرد مثال ها می بایست با دقت انتخاب شوند در غیر این صورت زمان سودمند، تلف می شود و یا حتی بدتر از این شبکه ممکن است نا درست کار کند. امتیاز شبکه عصبی این است که خودش کشف می کند که چگونه مسئله را حل کند ، عملکرد آن غیر قابل پیش گویی است.

از طرف دیگر ، کامپیوتر های معمولی از یک مسیر مشخص برای حل یک مسئله استفاده می کنند . راه حلی که مسئله از آن طریق حل می شود باید از قبل شناخته شود و به صورت دستورات کوتاه و غیر مبهمی شرح داده شود. این دستورات سپس به زبان های برنامه نویسی سطح بالا برگردانده می شود و بعد از آن به

کدهایی که کامپیوتر قادر به درک آنها است تبدیل می شود. به طور کلی این ماشین ها قابل پیش گویی هستند و اگر چیزی به خطا انجام شود به یک اشتباه سخت افزاری یا نرم افزاری بر می گردد. شبکه های عصبی و کامپیوتر های معمولی با هم در حال رقابت نیستند بلکه کامل کننده یکدیگرند. وظایفی وجود دارد که بیشتر مناسب روش های الگو ریتمی هستند نظیر عملیات محاسباتی و وظایفی نیز وجود دارد که بیشتر مناسب شبکه های عصبی هستند. حتی فراتر از این، مسائلی وجود دارد که نیازمند به سیستمی است که از ترکیب هر دو روش بدست می آید (بطور معمول کامپیوتر های معمولی برای نظارت بر شبکه های عصبی به کار گرفته می شوند) به این قصد که بیشترین کارایی بدست آید. شبکه های عصبی معجزه نمی کنند اما اگر خردمندانه به کار گرفته شوند نتایج شگفت آوری را خلق میکنند.

## 7-1- انسان و سلول های عصبی مصنوعی - در جستجوی شباهت ها

چگونه مغز انسان می آموزد؟

مسائل زیادی راجع به این که مغز چگونه خود را برای پردازش اطلاعات آموزش می دهد، ناشناخته باقی مانده است بنابر این تئوری های فراوانی وجود دارد. در مغز انسان یک سلول سیگنال ها را از دیگران از طریق یک گروه از ساختار های ریز به نام dendrites جمع آوری می کند سلول عصبی جهش سریع فعالیت الکتریکی را در طول یک پایه بلند و نازک که axon نامیده میشود، می فرستد که به داخل هزاران شاخه گسترش می یابد و کشیده می شود. در انتهای هر شاخه، ساختاری که synapse نامیده می شود این فعالیت را از axon به اثرات الکتریکی تبدیل می کند که فعالیت یک axon به صورت اثرات الکتریکی فعال کننده یا غیر فعال کننده تبدیل می شود که این کار باعث برانگیخته شدن یا آرام شدن سلول های عصبی مرتبط می شود. وقتی یک سلول عصبی پیام های فعال کننده را دریافت می کند، که بطور قانع کننده و وسیعی با پیام های ورودی غیر فعال کننده اش مقایسه شده باشد، در این زمان این سلول نیز یک جهش از فعالیت الکتریکی را به داخل axon خودش می فرستد.

یادگیری با تغییر تاثیر synapses اتفاق می افتد در نتیجه تاثیر یک سلول بر دیگران تغییر میکند.

## 8-1- از سلول های عصبی انسانی تا سلول های عصبی مصنوعی

ما این شبکه های عصبی را با تلاش اولیه در جهت یافتن خصوصیات اساسی سلول های عصبی و اتصالات آنها، هدایت می کنیم. سپس بطور معمول یک کامپیوتر را برای شبیه سازی این خصوصیات برنامه ریزی می کنیم. اگر چه بدلیل اینکه دانش ما از سلول های عصبی ناقص است و قدرت محاسبات ما محدود است، مدل های ما لزوما آرمان های خام و ناقصی از شبکه های واقعی سلول های عصبی است.

## 9-1- انواع یادگیری برای شبکه های عصبی

### 1. یادگیری با ناظر

در یادگیری با ناظر به قانون یاد گیری مجموعه ای از زوجهای داده ها به نام داده های یادگیری  $\{ (P_i, T_i) \}_{i=1}^l$  می دهند که در آن  $P_i$  ورودی به شبکه و  $T_i$  خروجی مطلوب شبکه برای ورودی  $P_i$  است. پس از اعمال ورودی  $P_i$  به شبکه عصبی در خروجی شبکه  $a_i$  با  $T_i$  مقایسه شده و سپس خطای یادگیری محاسبه و از آن در جهت تنظیم پارامترهای شبکه استفاده می شود به گونه ای که اگر دفعه بعد به شبکه همان ورودی  $P_i$  اعمال شود خروجی شبکه به  $T_i$  نزدیکتر می گردد با توجه به این نکته که معلم سیستمی است که بر محیط وقوف دارد (مثلا می داند که برای ورودی  $P_i$  خروجی مطلوب  $T_i$  است). توجه داریم که محیط برای شبکه عصبی مجهول است. در لحظه  $k$  بردار ورودی  $P_i(k)$  با تابع توزیع احتمال معینی که برای شبکه عصبی نا معلوماست انتخاب و بطور همزمان به شبکه عصبی و معلم اعمال می شود. جواب مطلوب  $T_i(k)$  نیز توسط معلم به شبکه عصبی داده می شود. در حقیقت پاسخ مطلوب پاسخ بهینه ای است که شبکه عصبی برای ورودی مفروض باید به آن برسد. پارامترهای شبکه عصبی توسط دو سیگنال ورودی و خطا تنظیم می شود. به این صورت که پس از چند تکرار الگوریتم یادگیری که عموما توسط معادله تفاضلی بیان می شود به پارامترهایی در فضای پارامترهای شبکه همگرا می شوند که برای آنها خطای یادگیری بسیار کوچک است و عملا شبکه عصبی شبکه عصبی معادل معلم می شود. یا به عبارتی دیگر اطلاعات مربوط به محیط (نگاشت بین  $P_i$  و  $T_i$ ) که برای معلم روشن است به شبکه عصبی منتقل می شود و پس از این مرحله عملا می توان بجای معلم از شبکه عصبی استفاده کرد تا یادگیری تکمیل شود.



## 2. یادگیری تشدید:

یک اشکال یادگیری با ناظر این است که شبکه عصبی ممکن است بدون معلم نتواند مواضع جدیدی را که توسط مجموعه داده های جدید تجربی پوشانده نشده است یاد بگیرد. یادگیری از نوع تشدید این محدودیت را برطرف می کند. این نوع یادگیری بطور on-line صورت می گیرد در حالی که یادگیری با ناظر را به دو صورت on-line & off-line می توان انجام داد. در حالت off-line می توان از یک سیستم محاسب با در اختیار داشتن داده های یادگیری استفاده کرد و طراحی شبکه عصبی را به پایان رساند. پس از مرحله طراحی و یادگیری شبکه عصبی به عنوان یک سیستم استاتیکی عمل می کند. اما در یادگیری on-line شبکه عصبی همراه با خود سیستم یادگیر در حال انجام کار است و از این رو مثل یک سیستم دینامیکی عمل می کند.

یادگیری از نوع تشدید یک یادگیری on-line از یک نگاشت ورودی-خروجی است. این کار از طریق یک پروسه سعی و خطا به صورتی انجام می پذیرد که یک شاخص اجرایی موسوم به سیگنال تشدید ماکزیمم شود و بنابر این الگوریتم نوعی از یادگیری با ناظر است که در آن به جای فراهم نمودن جواب واقعی، به شبکه عددی که نشانگر میزان عملکرد شبکه است ارایه می شود. این بدین معنی است که اگر شبکه عصبی پارامترهایش را به گونه ای تغییر داد که منجر به یک حالت مساعد شد آنگاه تمایل سیستم یادگیر جهت تولید آن عمل خاص تقویت یا تشدید می شود. در غیر این صورت تمایل شبکه عصبی جهت تولید آن عمل خاص تضعیف می شود. یادگیری تقویتی مثل یادگیری با ناظر نیست و این الگوریتم بیشتر برای سیستمهای کنترلی کاربرد دارد.

## 3. یادگیری بدون ناظر

در یادگیری بدون ناظر یا یادگیری خود سامانده پارامترهای شبکه عصبی تنها توسط پاسخ سیستم اصلاح و تنظیم می شوند. به عبارتی تنها اطلاعات دریافتی از محیط به شبکه را برداگرهای ورودی تشکیل می دهند. و در مقایسه با مورد بالا (یادگیری با ناظر) بردار جواب مطلوب به شبکه اعمال نمی شود. به عبارتی به شبکه عصبی هیچ نمونه ای از تابعی که قرار است بیاموزد داده نمی شود. در عمل می بینیم که یادگیری با ناظر

در مورد شبکه هایی که از تعداد زیادی لایه های نرونی تشکیل شده باشند بسیار کند عمل می کند و در این گونه موارد تلفیق یادگیری با ناظر و بدون ناظر پیشنهاد می گردد .

## 10-1- زمینه ای در مورد perceptron

Perceptron های ساده:

یک خانواده ساده از شبکه های عصبی مدل perceptron می باشد. در یک دسته بندی تک خروجی، تعداد  $n$  ورودی و یک خروجی دارد. با هر ورودی یک ضریب وزنی  $W_i$  و با هر خروجی یک مقدار آستانه  $q$  مرتبط است.

Perceptron به گونه زیر عمل می کند:

ورودی های Perceptron یک بردار ورودی از  $n$  مقدار حقیقی است.

Perceptron مجموع وزن ها را محاسبه می کند  $a = \sum W_i X_i$ . این مقدار با مقدار آستانه  $q$  مقایسه می شود. اگر این مقدار از مقدار آستانه کوچکتر باشد خروجی 0 است و در غیر این صورت 1 است.

قدرت Perceptron

به وسیله تنظیم اعداد ورودی، وزن آنها و مقدار آستانه می توان یک Perceptron برای انجام نسبتاً خوب محاسبات گوناگون طراحی کرد. برای مثال توابع منطقی بولین مانند AND، OR و NOT را می توان به وسیله Perceptron طراحی کرد و هر مدار منطقی دیگر را به وسیله گیت های AND و NOT یا AND و OR طراحی کرد. دسته های زیادی از Perceptron ها ممکن است خروجی های دسته های دیگر را به عنوان ورودی خود درخواست کنند.

به عنوان مثالی از Perceptron ها می توان یک تشخیص دهنده قالب متن را نام برد. حرف A در آرایه ای  $5 \times 5$  به رمز درمی آید (encode می شود). این متن (حرف) به وسیله یک Perceptron با 25 ورودی تشخیص داده می شود که در آن وزن ها مقادیری برابر با مقادیر عددی داخل آرایه را می گیرند و مقدار آستانه برابر است با:  $q = 25e-1$  که در آن  $0 < 1e$ .

خروجی Perceptron 1 است اگر و فقط اگر ورودی آن از 1 و 1- هایی باشد که عیناً در آرایه آمده است.

## 11-1- دنباله‌های Perceptron

یکی از خصوصیات جالب Perception این است که آنها می‌توانند به وسیله مثالهای مثبت و منفی ( صحیح و اشتباه) برای انجام توابع دسته‌بندی شده مخصوص بارها مرتب شوند.

حال به یک مثال ساده از Perceptron با دو ورودی  $1X$  و  $2X$ ، که تشخیص می‌دهد که کدام یک از دو کلاس، عناصر متعلق به خودش را دارد. ما فرض می‌کنیم که این Perceptron دو طرح از کارکترهای چاپ شده از یک متن را بررسی کند، خروجی 1 است اگر و فقط اگر کاراکتر رقم 8 باشد. فرض کنیم که  $1X$  بیانگر تعداد حفره‌های کاراکتر است و  $2X$  درجه راستی سمت چپ کاراکتر را نشان می‌دهد. ما با 4 ورودی .

اگر ما perceptron را در اول کار با وزنهایی برابر 0 و مقدار آستانه را برابر 10 مقداردهی کنیم یک رده‌بندی از همه مثالهای منفی انجام داده‌ایم. با قرار دادن رده‌بندی‌های نادرست از 8، مقادیر ورودی از مثال 8 با بعضی فاکتورها مثل  $d$  جمع می‌شوند و تولیدات جدید با وزنهای متناظر با ایجاد می‌شوند.

فرض کنیم  $d=1$  پس وزن ورودی‌ها از 0 به 1 و 2 رشد پیدا می‌کند. حال در اینجا  $a = 5$  به دست می‌آید که هنوز از مقدار آستانه 10 کوچکتر است. مثال هنوز به رده‌بندی صحیحی نرسیده است و این قدم دنباله باید تکرار شود. بعد از دو قدم وزنهای برابر 2 و 4 می‌شوند که مقدار  $a = 10$  را نتیجه می‌دهد که برابر مقدار آستانه است و مثال مثبت از 8 به طور صحیح دسته‌بندی شده است. از آنجا که ضرایب وزنی تغییر کرده بودند لازم است که در همه مثالها رده‌بندی‌ها باز نشان (Reset) شوند. این را می‌توان به سادگی دید که مثال B رده‌بندی نادرستی است زیرا با وزنهای 2 و 4 داریم  $a = 24$  ولی این حرف مورد نظر ما نیست، چون این مرحله را پیش رفته‌ایم لازم است که  $1d$  از  $1W$  و  $2d$  از  $2W$  کم شود تا رده‌بندی نادرستی از B ثابت شود. به هر حال یک رده‌بندی از 8 را دوباره بیرون می‌دهد.

بعدها موقع بروز خطا ما وزنهای را برای درست کردن خطاهای رده‌بندی اصلاح می‌کنیم. اگر مثالها دارای خاصیت صحیحی باشند وزنهای در مجموعه‌ای از مقادیری که به درستی روی هر ورودی کار می‌کنند قرار می‌گیرند.

## 12-1- قضیه بنیادی دنباله‌ها

یک خصوصیت قابل توجه perceptron این است که آنها می‌توانند دنباله‌ای از رده‌بندی صحیح مثالهای مثبت و منفی باشند.

فرض کنیم:  $-X + X = X$

$+X$ : مجموعه‌ای از مثالهای مثبت

$-X$ : مجموعه‌ای از مثالهای منفی

گوییم که رشته بی‌کران  $x = X_1, X_2, \dots, X_k, \dots$  یک رشته متوالی (ترتیبی) برای  $X$  است در صورتی که هر  $X_i$  یک مثال در  $X$  است و هر عنصر از  $X$  اغلب به طور نامحدود در  $S_x$  رخ می‌دهد (نمایان می‌شود). فرض کنیم  $W_k$  ضریب وزنی در سطح  $k$  دنباله باشد. وزن اولیه می‌تواند به صورت قراردادی باشد (برای مثال  $W_0 = 1$ ). حال رشته استاندارد حاصله، وزنهای را به صورت زیر ارتقا می‌دهد: بسته به استراژی مورد نظر ممکن است مقادیر  $C_k$  همگی یکسان باشند یا ممکن است با  $k$  تغییر کنند. قضیه 1)

یک بردار حل وزنهای برای  $X$  وجود داشته باشد، در این صورت رویه رشته استاندارد باید بعد از یک تعداد فرض کنیم یک مجموعه از رشته نمونه  $X$  و هر رشته ترتیبی برای آن داریم، اگر  $C_k$  یک ثابت مثبت مراحل مشخص یک راه‌حل پیدا کند به طوری که اگر برای بعضی  $0_k$  ها داشته باشیم:

$$W_k 0_k = W_k 1 + 0 = W_k 2 + 0 = \dots$$

که  $0_k W_k$  یک راه‌حل برای  $X$  است.

بنابراین ما می‌توانیم با استفاده از شبکه‌های عصبی هر چه بیشتر به شبیه‌سازی انسان توسط کامپیوترها نزدیک شویم به منظور واگذاری کارهای تکراری، وقت‌گیر و مسائلی که با توجه به پیشرفت بشری دیگر درخور بشر نیست.

### 13-1- هوش جمعی

فرض کنید شما و گروهی از دوستانتان به دنبال گنج می‌گردید. هر یک از اعضای گروه یک فلزیاب و یک بی‌سیم دارد که می‌تواند مکان و وضعیت کار خود را به همسایگان نزدیک خود اطلاع بدهد. بنابراین شما می‌دانید آیا همسایگانتان از شما به گنج نزدیک‌ترند یا نه؟ پس اگر همسایه‌ای به گنج نزدیک‌تر بود شما می‌توانید به طرف او حرکت کنید. با چنین کاری شانس شما برای رسیدن به گنج بیشتر می‌شود و همچنین گنج زودتر از زمانی که شما تنها باشید، پیدا می‌شود.

این یک مثال ساده از رفتار جمعی یا Swarm behavior است که افراد برای رسیدن به یک هدف نهایی همکاری می‌کنند. این روش مؤثرتر از زمانی است که افراد جداگانه عمل کنند. Swarm را می‌توان به صورت مجموعه‌ای سازمان یافته از موجوداتی تعریف کرد که با یکدیگر همکاری می‌کنند. در کاربردهای محاسباتی Swarm intelligence از موجوداتی مانند مورچه‌ها، زنبورها، موریانه‌ها، دسته‌های ماهیان و دسته‌ی پرندگان الگو برداری می‌شود. در این نوع اجتماعات هر یک از موجودات ساختار نسبتاً ساده‌ای دارند ولی رفتار جمعی آنها بی‌نهایت پیچیده است. برای مثال در کولونی مورچه‌ها هر یک از مورچه‌ها یک کار ساده‌ی مخصوص را انجام می‌دهد ولی به طور جمعی عمل و رفتار مورچه‌ها، ساختن بهینه‌ی لایه، محافظت از ملکه و نوزادان، تمیز کردن لانه، یافتن بهترین منابع غذایی و بهینه‌سازی استراتژی حمله را تضمین می‌کند. رفتار کلی، یک Swarm به صورت غیر خطی از آمیزش رفتارهای تک‌تک اجتماع بدست می‌آید. یا به عبارتی یک رابطه‌ی بسیار پیچیده بین رفتار جمعی و رفتار فردی یک اجتماع وجود دارد. رفتار جمعی فقط وابسته به رفتار فردی افراد اجتماع نیست بلکه به چگونگی تعامل میان افراد نیز وابسته است. تعامل بین افراد، تجربه‌ی افراد درباره‌ی محیط را افزایش می‌دهد و موجب پیشرفت اجتماع می‌شود. ساختار اجتماعی Swarm بین افراد مجموعه کانالهای ارتباطی ایجاد می‌کند که طی آن افراد می‌توانند به تبادل تجربه‌های شخصی بپردازند، مدل‌سازی محاسباتی Swarmها کاربردهای موفق و بسیاری را در پی داشته است مانند:

Function optimization, Finding optimal roots, scheduling, structural optimization, Image and data analysis

کاربردهای زیادی از مطالعه‌ی Swarmهای مختلف وجود دارد. از این دسته می‌توان به کولونی مورچه‌ها (Ant Colony) و دسته‌ی پرندگان (Bird Flocks) اشاره نمود.

#### 14-1 Particle Swarm Optimization(PSO)

الگوریتم PSO یک الگوریتم جستجوی اجتماعی است که از روی رفتار اجتماعی دسته‌های پرندگان مدل شده است. در ابتدا این الگوریتم به منظور کشف الگوهای حاکم بر پرواز همزمان پرندگان و تغییر ناگهانی مسیر آنها و تغییر شکل بهینه‌ی دسته به کار گرفته شد.

در PSO، particle ها در فضای جستجو جاری می‌شوند. تغییر مکان particle ها در فضای جستجو تحت تأثیر تجربه و دانش خودشان و همسایگانشان است. بنابراین موقعیت دیگر particle های Swarm روی چگونگی جستجوی یک particle اثر می‌گذارد.

نتیجه‌ی مدل‌سازی این رفتار اجتماعی فرایند جستجویی است که particle ها به سمت نواحی موفق میل می‌کنند. Particle ها در Swarm از یکدیگر می‌آموزند و بر مبنای دانش بدست آمده به سمت بهترین همسایگان خود می‌روند.

#### Particle swarm Optimatation Algorithm

اساس کار PSO بر این اصل استوار است که در هر لحظه هر particle مکان خود را در فضای جستجو با توجه به بهترین مکانی که تاکنون در آن قرار گرفته است و بهترین مکانی که در کل همسایگی‌اش وجود دارد، تنظیم می‌کند.

فرض کنید می‌خواهیم زوج مرتب  $(x,y)$  را طوری بدست آوریم که تابع  $2+y2F(x,y)=x$ ، مینیمم شود. ابتدا نقاطی را به صورت تصادفی در فضای جستجو، روی صفحه‌ی  $x-y$  انتخاب می‌کنیم. فرض کنید این Swarm را به 3 همسایگی تقسیم کنیم که در هر همسایگی نقاط موجود با یکدیگر تعامل دارند. در هر همسایگی هر یک از نقاط به سمت بهترین نقطه در آن همسایگی و بهترین مکانی که آن نقطه تاکنون در آن قرار داشته است، حرکت می‌کند. برای حل یک مسئله چند متغیر بهینه‌سازی می‌توان از چند Swarm استفاده کرد که هر یک از Swarm ها کار مخصوصی را انجام می‌دهند. این همان ایده‌ای است که Ant colony از آن ریشه می‌گیرد. از آنجا که دانش Swarm intelligence بسیار جدید است در حال حاضر کاربردهای کمی از آن

شناخته شده است. ولی صاحب‌نظران معتقدند با این رشد روزافزون، Swarm intelligence می‌تواند نقش مهمی را در علوم مختلف از جمله مهندسی نفت ایفا کند.

www.Prozhe.com

## فصل دوم

www.Prozhe.com



## یک شبکه عصبی جدید و کاربرد آن

### 2-1- مقدمه

ایجاد شبکه های عصبی آشوبگونه، یک نورون آشوبی معرفی شده است. توانایی عمل بر روی اطلاعات در شبکه های برگشتی به دلیل حضور حلقه های فیدبک، بیشتر از شبکه های معمولی است چگونگی بهبود خاصیت ارگادیک ANN یک مسئله مهم جهت تحقیق و بررسی می باشد. سیستم های آشوبگونه دارای مشخصاتی تصادفی هستند و الگوریتم آشوبی باعث ایجاد خاصیت قوی ارگادیک در شبکه می گردد. تعدادی نورون خطی در لایه خروجی و تعدادی نورون آشوبگونه با تابع لجستیک در لایه برگشتی وجود دارند.

### 2-2- یک شبکه عصبی جدید و کاربرد آن

به منظور بهبود توانایی مدل سازی و شناسایی سیستم های غیرخطی با شبکه های عصبی، خواص نورون ها، قوانین آموزش و شکل شبکه ها، مورد بررسی قرار گرفته است. برای ایجاد شبکه های عصبی آشوبگونه، یک نورون آشوبی معرفی شده است. توانایی عمل بر روی اطلاعات در شبکه های برگشتی به دلیل حضور حلقه های فیدبک، بیشتر از شبکه های معمولی است. در آموزش شبکه برای مدل سازی، از الگوریتم پس انتشار خطا استفاده شده است. در مقایسه با شبکه های عصبی دیگر تعداد نورون لایه پنهان شبکه های آشوبگونه کمتر و توانایی تعمیم دهی آن ها بیشتر است. از شبکه آشوبگونه برگشتی ارائه شده در مدل سازی ژنراتور سنکرون دریایی، استفاده شده است.

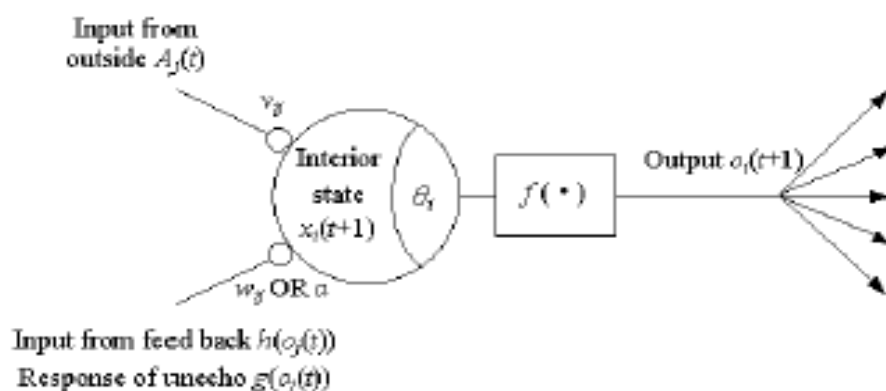
### 3-2- معرفی

در این فصل ANN، ای با توانایی مطلوب جهت بررسی سیستم های غیرخطی ارائه شده است. در مدل سازی سیستم ها، توانایی تقریب به وسیله نورون ها، شکل شبکه و قانون آموزش، محدود می گردد. چگونگی بهبود خاصیت ارگادیک ANN یک مسئله مهم جهت تحقیق و بررسی می باشد. سیستم های آشوبگونه دارای مشخصاتی تصادفی هستند و الگوریتم آشوبی باعث ایجاد خاصیت قوی ارگادیک در شبکه می گردد.

مدلسازی ژنراتور سنکرون دریایی در زمینه شبیه سازی و کنترل real time بسیار مفید می باشد. مدل های مرسوم برای این ژنراتور اغلب با حل معادلات دیفرانسیل مرتبه بالایی سروکار دارند که حل آن ها با محاسبات DSP به صورت real time ، میسر نمی باشد. به همین دلیل در اینجا ، مدلسازی و شناسایی این سیستم توسط ANN مورد توجه قرار گرفته است.

#### 4-2- نورون با خاصیت آشوبگونه

شکل نورون در شکل (1) نشان داده شده است.  $f(\cdot)$  تابع فعالسازی نورون است که یک نگاشت لجستیک با پارامتر شیب 0.04 می باشد.



شکل 1-2: نورون آشوب گونه

رابطه (1) الگوریتم محاسبات مربوط به این نورون را نشان می دهد.

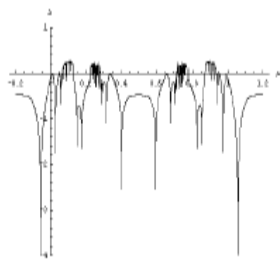
$$\begin{cases} o_i(t+1) = f\left\{\left[\sum_{j=1}^M w_{ij} \sum_{r=0}^t k^r h(o_j(t-r))\right] + \sum_{j=1}^N v_{ij} \sum_{r=0}^t k^r A_j(t-r) \right. \\ \quad \left. - \alpha \sum_{r=0}^t k^r g(o_i(t-r)) - \theta\right\} \\ f(x_i(t+1)) = 1/(1 + \exp(-x_i(t+1)/\varepsilon)) \end{cases} \quad (1)$$

تابع لجستیک برای ایجاد خاصیت آشوبی در شبکه، قرار داده شده است. شکل 2-a دیاگرام این تابع را نشان می دهد. دو شاخه شدن تابع هنگامی رخ می دهد که پارامتر شیب آن با 0.04 برابر شود (شکل 2-b). نمای Lyapunov هم در شکل 2-c نشان داده شده است. وقتی که نمای Lyapunov بزرگتر از صفر شود، شبکه به صورت آشوبگونه رفتار می کند. شکل 2-d و 2-e هم به ترتیب دوشاخه شدن تابع را هنگامی که پارامتر شیب برابر با 0.01 و 0.06 شود، نشان می دهند. پیوند دوبرگه شدن هم به وضوح در شکل 2-f دیده می شود.

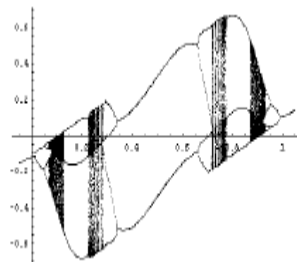
## 5-2- شکل شبکه

به دلیل وجود توانایی بیشتر عمل بر روی اطلاعات توسط شبکه های برگشتی، از یک شبکه برگشتی به فرم آشوبگونه استفاده شده است. این شبکه معمولاً سه لایه با یک لایه پنهان است. فیدبک از نورون های لایه پنهان خروجی به نورون های لایه پنهان ورودی صورت می گیرد. اتصالات برگشتی، امکان آشکارسازی و تولید الگوهای متغیر با زمان را برای شبکه فراهم می سازند. ساختار این شبکه در شکل (3) نشان داده شده است.

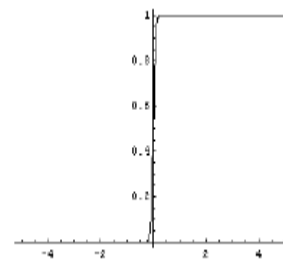
تعدادی نورون خطی در لایه خروجی و تعدادی نورون آشوبگونه با تابع لجستیک در لایه برگشتی وجود دارند. به خاطر وجود خاصیت آشوبی در نورون ها، توانایی تقریب شبکه افزایش می یابد. ترکیب ارائه شده از این نظر خاص می باشد که قادر است هر تابعی را با دقت دلخواه تقریب زند.



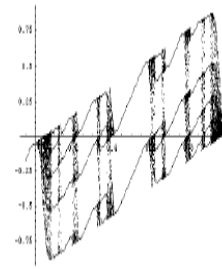
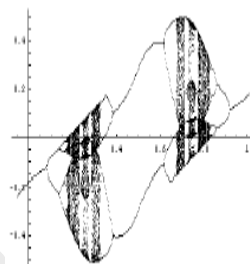
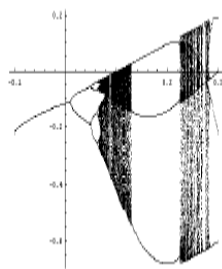
(c) دیاگرام نمای Lyapunov



(b) دیاگرام دو شاخه شدن ( $\varepsilon = 0.04$ )



(a) خروجی تابع ( $\varepsilon = 0.04$ )

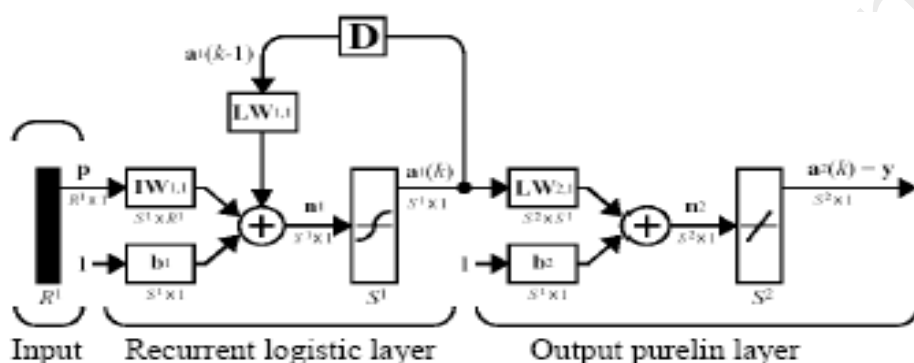


(d) دیاگرام دوشاخه شدن ( $\varepsilon = 0.01$ )      (e) دیاگرام دوشاخه شدن ( $\varepsilon = 0.06$ )      (f) دیاگرام دوشاخه شدن ( $\varepsilon = 0.06$ )

نمودار مشخصه آشوبی نورو

البته باید تعداد نوروهای شبکه کافی باشد. فرق این شبکه با یک شبکه سه لایه معمولی در اتصال برگشتی لایه دوم است. تاخیر در این اتصال مقادیری از پله زمانی گذشته را ذخیره کرده که می توانند در زمان فعلی مورد استفاده قرار بگیرند. بنابراین حتی اگر دو شبکه با وزن ها و بایاس های یکسان تحت یک ورودی قرار بگیرند ، به علت وجد فیدبک های حالت مختلف ممکن است خروجی های متفاوتی تولید کنند. خروجی هر لایه توسط معادلات زیر محاسبه می گردد :

$$\begin{cases} \mathbf{a}^1(k) = \text{logistic}(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{LW}^{1,1}\mathbf{a}^1(k-1) + \mathbf{b}^1) \\ \mathbf{a}^2(k) = \text{purelin}(\mathbf{LW}^{2,1}\mathbf{a}^1(k) + \mathbf{b}^2) \end{cases} \quad (2)$$



شکل 2-2 : CNN

## 2-6- قانون آموزش شبکه

از الگوریتم پس انتشار خطا جهت آموزش استفاده می شود. اگر ورودی  $p$  و خروجی  $y$  باشد، الگوریتم BP به صورت زیر قابل بیان است.

1- ابتدا تابع هدف را تعریف می کنیم :

$$\begin{aligned} J(t) &= \sum_{\tau=t_0+1}^t E(\tau) = \frac{1}{2} \sum_{\tau=t_0+1}^t (p_d(\tau) - a^2(\tau))^2 \\ &= \frac{1}{2} \sum_{\tau=t_0+1}^t e^2(\tau) \end{aligned} \quad (3)$$

که در آن  $p_d(\tau)$  و  $a^2(\tau)$  به ترتیب خروجی داده های نمونه و خروجی شبکه می باشند.

2- معادلات تنظیم به صورت زیر تعریف می شوند :

$$\begin{aligned}\frac{\partial E(\tau)}{\partial Lw_i^{21}} &= \frac{\partial E(\tau)}{\partial a^2(\tau)} \cdot \frac{\partial a^2(\tau)}{\partial a^1(\tau)} \cdot \frac{\partial a^1(\tau)}{\partial Lw_i^{21}} = -e(\tau)f_2'(p(\tau))a_i^1(\tau) \\ &= \delta(\tau)a_i^1(\tau)\end{aligned}\quad (4)$$

که در آن

$$\delta(\tau) = \frac{\partial E(\tau)}{\partial a^1(\tau)} = -e(\tau)f_2'(p(\tau))$$

$LW^{1,1}$  وزن لایه پنهان است ،

$$\frac{\partial E(\tau)}{\partial Lw_y^{11}} = \frac{\partial E(\tau)}{\partial a_i^1(\tau)} \cdot \frac{\partial a_i^1(\tau)}{\partial p_{a^1,i}(\tau)} \cdot \frac{\partial p_{a^1,i}(\tau)}{\partial Lw_y^{11}}$$

$$\begin{aligned}\frac{\partial E(\tau)}{\partial a_i^1(\tau)} &= \frac{\partial E(\tau)}{\partial p(\tau)} \cdot \frac{\partial p(\tau)}{\partial a_i^1(\tau)} = \delta(\tau) \cdot \frac{\partial \left[ \sum_i Lw_i^{21} a_i^1(\tau) \right]}{\partial a_i^1(\tau)} \\ &= \delta(\tau)Lw_i^{21}(\tau)\end{aligned}$$

$$\frac{\partial a_i^1(\tau)}{\partial p_{a^1,i}(\tau)} = f_1'(p_{a^1,i}(\tau))$$

$$\frac{\partial a_i^1(\tau)}{\partial Lw_i^{11}} = \frac{\partial a_i^1(\tau)}{\partial p_{a^1,i}(\tau)} \cdot \frac{\partial p_{a^1,i}(\tau)}{\partial Lw_i^{11}}$$

$$\begin{aligned}
&= f_1'(p_{a^1,i}(\tau)) \frac{\partial \left[ \sum_j Lw_i^{1,1} a_j^1(\tau-1) + \sum_q Iw_{iq}^{1,1} u_q(\tau-1) \right]}{\partial Lw_{ij}^{1,1}} \\
&= f_1'(p_{a^1,i}(\tau)) \left[ a_j^1(\tau-1) + \sum_j Lw_i^{2,1} \frac{\partial a_j^1(\tau-1)}{\partial Lw_i^{2,1}} \right] \quad (6)
\end{aligned}$$

$IW^{1,1}$  وزن بین لایه ورودی و لایه پنهان است ،

$$\frac{\partial E(\tau)}{\partial Iw_{iq}^{1,1}} = \frac{\partial E(\tau)}{\partial p_{a^1,i}(\tau)} \cdot \frac{\partial p_{a^1,i}(\tau)}{\partial Iw_{iq}^{1,1}} = \delta_{a^1,i}(\tau) u_q(\tau-1)$$

همانطور که از معادله (6) پیداست الگوریتم پس انتشار خطا یک الگوریتم دینامیک مرحله ای و تکرار شونده است. این الگوریتم را برای سه لایه می توان به فرم زیر خلاصه نمود :

$$\Delta Lw_i^{2,1} = -\eta e(\tau) f_2'(p(\tau)) a_i^1(\tau) = \eta \delta(\tau) a_i^1(\tau) \quad (7)$$

$$\begin{aligned}
\Delta Lw_{ij}^{1,1} &= -\eta \delta(\tau) Lw_i^{2,1}(\tau) f_1'(p_{a^1,i}(\tau)) \frac{\partial p_{a^1,i}(\tau)}{\partial Lw_{ij}^{1,1}} \\
&= -\eta \delta_{a^1,i}(\tau) \frac{\partial p_{a^1,i}(\tau)}{\partial Lw_{ij}^{1,1}} \quad (8)
\end{aligned}$$

$$\begin{aligned}
\Delta Iw_{iq}^{1,1} &= -\eta \delta(\tau) Lw_i^{2,1}(\tau) f_1'(p_{a^1,i}(\tau)) u_q(\tau-1) \\
&= -\eta \delta_{a^1,i}(\tau) u_q(\tau-1) \quad (9)
\end{aligned}$$

$$\delta_{a^1,i}(\tau) = \frac{\partial E(\tau)}{\partial p_{a^1,i}(\tau)} = \delta(\tau) L w_i^{2,1}(\tau) f'_1(p_{a^1,i}(\tau)) \quad (10)$$

برای افزایش سرعت و کاهش نوسان از الگوریتم gradient descent با ممتم و نرخ آموزش تطبیقی جهت training استفاده شده است. مقدار وزن و بایاس توسط معادله (11) محاسبه می گردد.

$$\begin{cases} \Delta W_k^m = \gamma \Delta W_{k-1}^m - (1-\gamma) \alpha S^m (\mathbf{a}^{1(m-1)})^T \\ \Delta b_k^m = \gamma \Delta b_{k-1}^m - (1-\gamma) \alpha S^m \end{cases} \quad (11)$$

که در آن  $\Delta W_{k-1}^m$  و  $\Delta b_k^m$  به ترتیب مقدار های وزن و بایاس m امین لایه و k امین تکرار هستند.  $\gamma$  ضریب ممتم ،  $\alpha$  نرخ آموزش و  $S^m$  بردار درجه حساسیت است.

اساس تغییرات نرخ آموزش به این ترتیب است که اگر MSE از یک درصد خاصی (برای مثال 1 یا 5 درصد) بیشتر شد ، update وزن متوقف شده ، ضریب ممتم مساوی صفر قرار داده شده و نرخ آموزش در یک فاکتور کوچکتر از یک ضرب می شود. اگر MSE افزایش پیدا کند ولی از درص مورد نظر بیشتر نشود ریال نرخ آموزش و ضریب ممتم بدون تغییر باقی می ماند. اما اگر MSE کاهش پیدا کند ، آن وقت نرخ آموزش در یک فاکتور بزرگتر از یک ضرب شده و اگر ضریب ممتم از قبل صفر شده بود به مقدار اولیه اش ریست می گردد.

## 7-2- مدلسازی ژنراتور سنکرون دریایی

### روش مدلسازی دینامیک

از شبکه سه لایه مذکور در مدلسازی ژنراتور سنکرون استفاده می شود. آموزش به صورت باسرپرست و training به صورت off line صورت می پذیرد. شبکه جهت بهتر نشان دادن مد دینامیک با خط تاخیر TDL

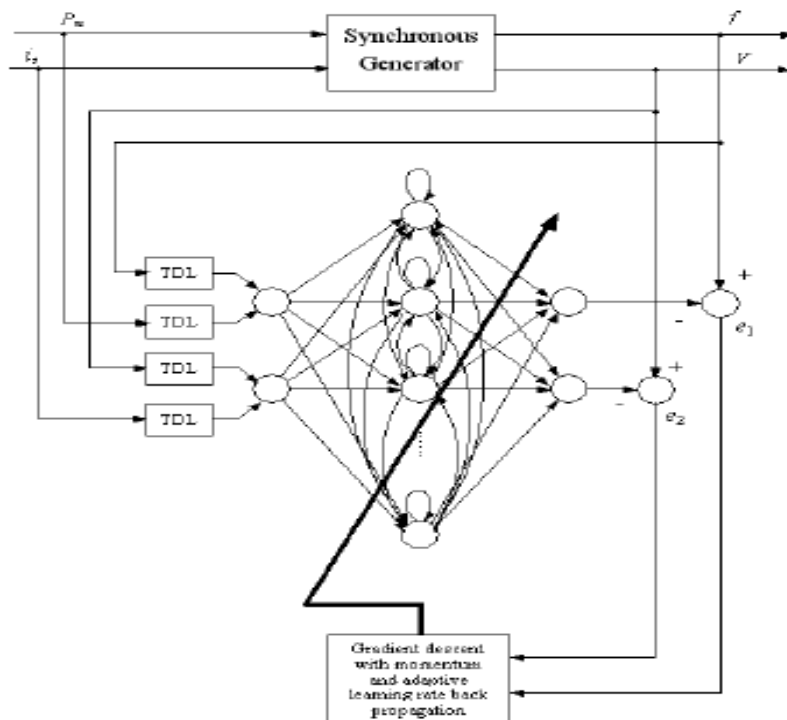


ترکیب شده است. از حافظه های TDL جهت فیدبک به لایه ورودی استفاده می شود. توان گشتاور خروجی موتور دیزل دریایی به عنوان پارامترهای ورودی شبکه و ولتاژ ترمینال ژنراتور و خروجی فرکانس به عنوان خروجی شبکه انتخاب می شوند. این سیستم در شکل (4) نشان داده شده است.

پاسخ مشخصه بین توان گشتاور ورودی و فرکانس خروجی و همچنین بین جریان تحریک ورودی و ولتاژ ترمینال خروجی ژنراتور اندازه گیری شده است. شکل (5) داده های نمونه توان گشتاور ورودی و منحنی های فرکانس خروجی را نشان می دهد.

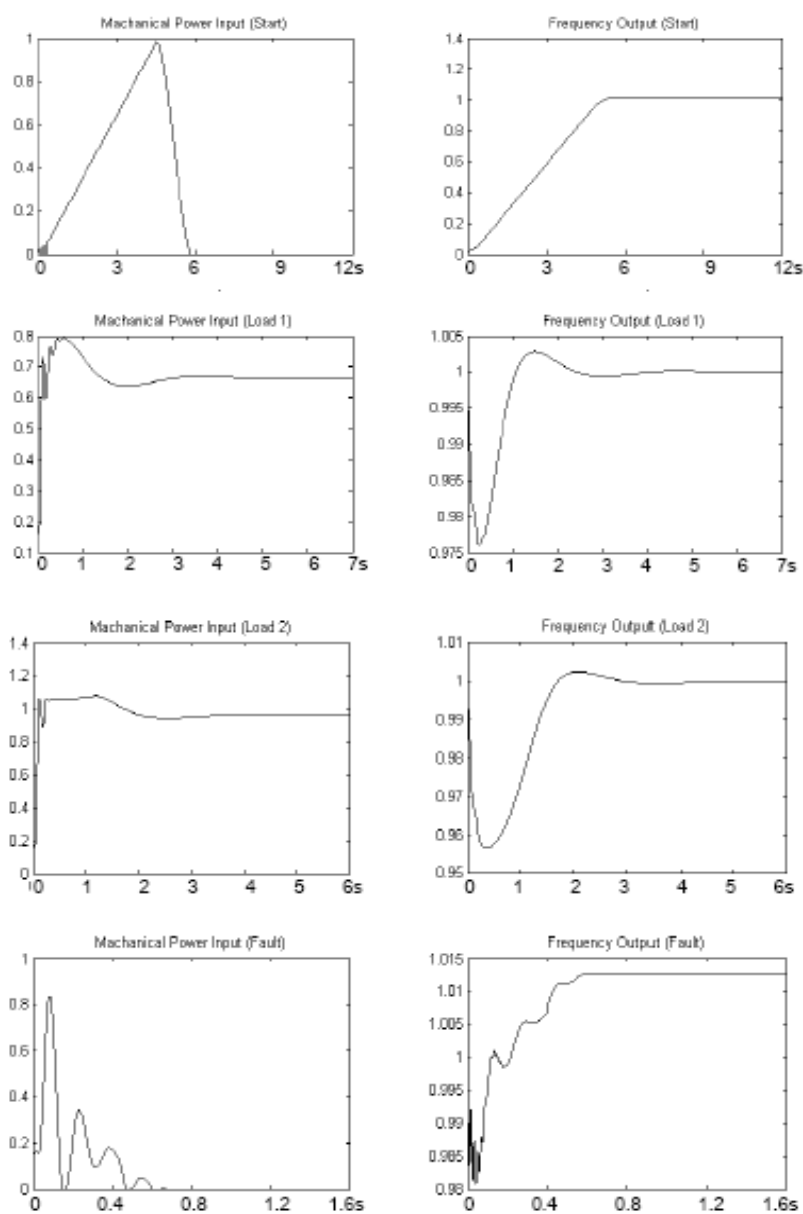
در شبیه سازی انجام شده  $\gamma = 0.9$  ،  $\alpha = 0.01$  ، افزایش نرخ آموزش 1.05 و کاهش آن 0.7 در نظر گرفته شده است. فرآیند آموزش به صورت زیر می باشد.

- 1- به همه وزن ها و بایاس ها مقدار اولیه می دهیم.
- 2- داده های نمونه ورودی و خروجی را برای training به شبکه نشان می دهیم.
- 3- خروجی شبکه بر طبق ورودی ، وزن ها و بایاس ها محاسبه می گردد. وقتی MSE بین داده های نمونه و خروجی شبکه از یک حدی کمتر شد training متوقف می شود ، در غیر این صورت به مرحله (4) می رویم.
- 4- وزن ها بر طبق اختلاف بین مقدار مطلوب و خروجی شبکه ، تنظیم می شوند.
- 5- برو به مرحله (2).



شکل 2-3: سیستم شناسایی ژنراتور سنکرون دریایی به وسیله شبکه عصبی

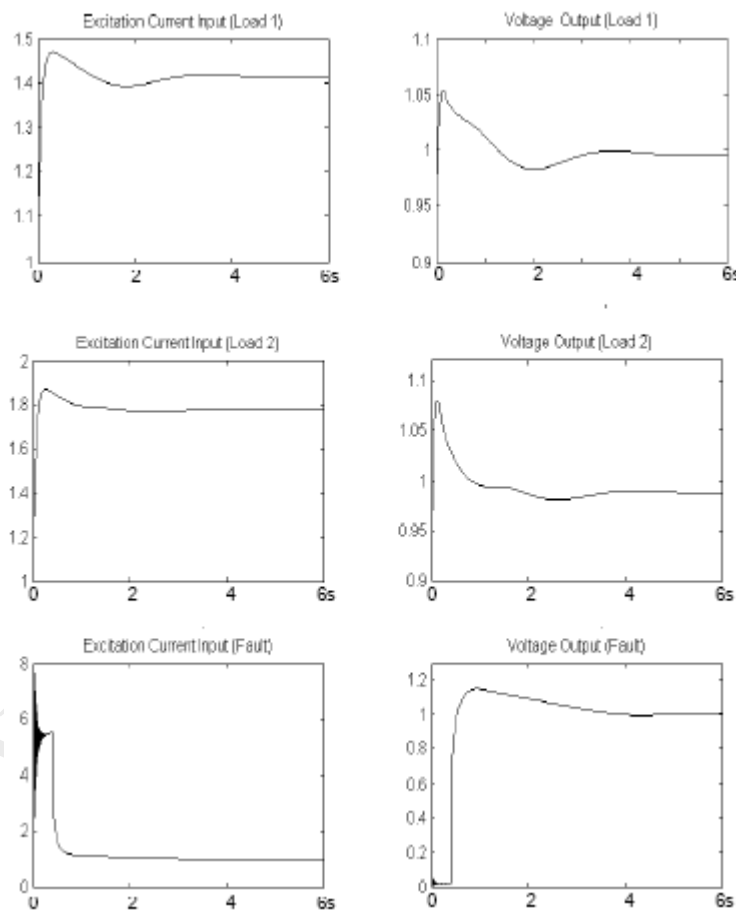
برای تنظیم وزن های شبکه از الگوریتم gradient descent با ممنتوم و BP با نرخ آموزش تطبیقی استفاده شده است. شکل (7) تغییرات MSE را نشان می دهد.



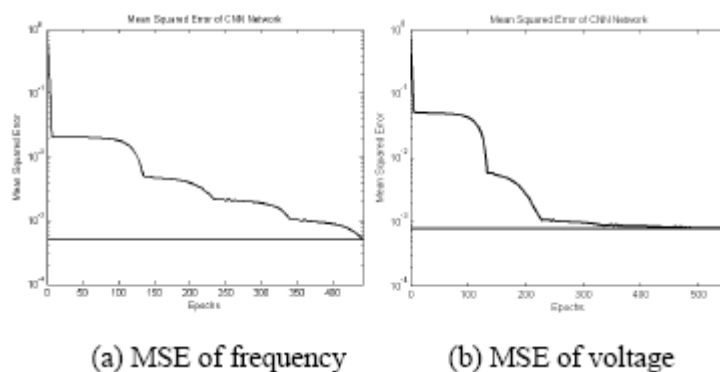
شکل 2-4: توان گشتاور ورودی و فرکانس خروجی ژنراتور

## 8-2- نتایج مدل سازی

فرکانس خروجی مدل با فرکانس خروجی ژنراتور در شکل (8) با هم مقایسه شده اند. خطوط منفصل مربوط به خروجی ژنراتور و خطوط پیوسته مربوط به خروجی شبکه هستند. ژنراتور در سه حالت بدون بار، با بار متوسط و بزرگ راه اندازی شده و نتیجه آن با خروجی مدل مقایسه می گردد. در شکل (9) هم خروجی مدل و ژنراتور از لحاظ ولتاژ پایانه با هم مقایسه شده اند.



شکل 2-5: جریان تحریک ورودی و ولتاژ خروجی پایانه



شکل 2-6: منحنی MSE هنگام trainin شبکه  
 شکل 2-7: فرکانس خروجی ژنراتور، شبکه و خطای بین آن ها

## 9-2- نتیجه فصل

نتایج به دست آمده را می توان به شکل زیر خلاصه کرد :

- 1- داده های نمونه باید به درستی انتخابی شوند به نحوی که حالت های مختلف راه اندازی ژنراتور را مجسم نمایند.
- 2- داده های نمونه باید نرمالیزه شده و اگر نویز با آن ها جمع شده توسط فیلتر حذف گردد.
- 3- درجه مدل باید طبق مطالعات قبلی و یا بررسی داده های نمونه، تعیین گردد.
- 4- اگر سیستم متغیر با زمان است باید از الگوریتم on line تکرار شونده جهت تنظیم وزن ها استفاده نمود.

## فصل سوم

www.Prozhe.com

## آنالیز رفتار آشوبگونه مدل شبکه عصبی مکانیسم لرزش عضله

### 1-3- مقدمه

Armstrong منحنی تغییرات کشش ایزومتریک کزازی عضله را نسبت به طول سارکومر در یک فیبر عضله مخطط قورباغه به دست آورده است. منحنی به دست آمده نگاشتی را نشان می دهد که در نوع خاصی از لرزش های عضلانی به صورت آشوبگونه رفتار می کند. طیف توان در نشان دادن دینامیک های آشوبگونه ، بسیار مفید می باشد. اگر سیگنال پریودیک و یا نیمه پریودیک باشد ، طیف توان آن شامل یک سری ضربه در فرکانس های پایه و یا چند فرکانس هارمونیک ، می باشد. اگر سیگنال آشوبگونه باشد ، طیف توان آن پیوسته خواهد بود. بنابراین ایجاد طیف پیوسته از طیف گسسته حضور دینامیک آشوبگونه را در سیگنال نشان می دهد.

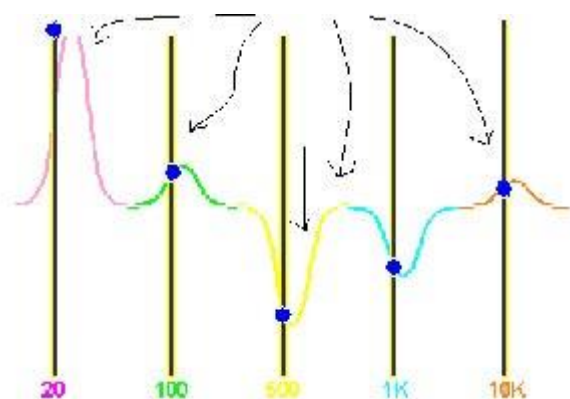
### 2-3- آنالیز رفتار آشوبگونه مدل شبکه عصبی مکانیسم لرزش عضله

این فصل نتایج تجربی مدلسازی فعالیت آشوبگونه نوع خاصی از لرزش های عضلانی را توسط شبکه های عصبی ، ارائه می دهد. وجود آشوب در خروجی شبکه به وسیله ایجاد طیف پیوسته فرکانسی از طریق افزایش فرکانس های گسسته ، قابل تایید می باشد. شبکه مورد استفاده یک شبکه ساده برگشتی از نوع Elman است که نیازی به تغییر در روش آموزش ندارد.

### 3-3- معرفی

Armstrong منحنی تغییرات کشش ایزومتریک کزازی عضله را نسبت به طول سارکومر در یک فیبر عضله مخطط قورباغه به دست آورده است. منحنی به دست آمده نگاشتی را نشان می دهد که در نوع خاصی از لرزش های عضلانی به صورت آشوبگونه رفتار می کند. در این نگاشت خروجی در هر لحظه به عنوان ورودی در لحظه بعد محسوب می شود. کمی تغییر در پارامترهای تابع منجر به تغییرات اساسی در رفتار آن و مشاهده پدیده هایی مثل دو شاخه شدن ، نوسان و رفتار آشوبگونه تصادفی می گردد. در این فصل از شبکه عصبی جهت مدلسازی لرزش عضله ، استفاده شده است. به عنوان مثال برای لرزش عضله میتوان نوسانات

خود به خودی نیروی عضله را تحت انقباض ثابت ، نام برد. پدیده آشوب در خروجی شبکه پس از افزایش فرکانس های گسسته و ضمن پیوسته شدن طیف ، اتفاق می افتد.



#### 4-3- منحنی طول - کشش

رابطه بین کشش و طول در یک عضله ایزوله توسط Hatz به دست آمده و به صورت زیر قابل بیان است :

که در آن  $x$  طول سارکومر و پارامتر مدولاسیون طول منحنی می باشد.



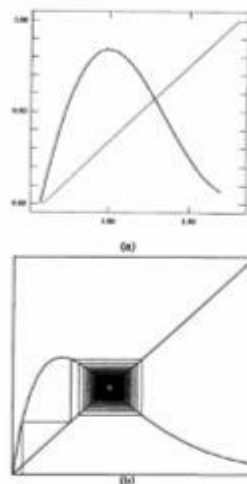


Fig 1 Length-tension function of active tension.

شکل 3-1: تابع کشش-طول فعال

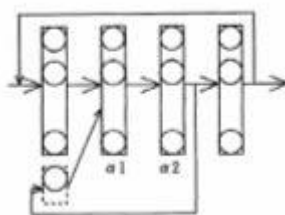
در یک عضله واقعی رابطه بین طول و کشش دو قسمت دارد: یکی رابطه کشش طول فعال و دیگری رابطه الاستیک پسویو. منحنی فعال، طول را به کشش و منحنی پسویو کشش را به طول مرتبط می سازد. به همین دلیل نتیجه یک نگاشت از طول به طول می باشد. شکل (a-1) رابطه کشش طول فعال Hatz را به همراه خط  $y = x$  که رابطه طول کشش پسویو را بیان می کند، نشان می دهد. حالت اولیه از خط  $y = x$  آغاز می شود. یک سطح ثابت تحریک منحنی فعال را به یک سطح جدید کشش می برد. این کشش جدید یک طول جدید از منحنی پسویو را ایجاد می کند و این فرآیند تکرار می شود. در شکل (b-1) مشاهده می شود که فعالیت از یک حالت اولیه شروع شده و در نهایت در یک سیکل محدود شروع به نوسان می کند.

## - شبکه های عصبی 5-3- ساختار برگشتی

ساختار برگشتی

شکل (۲) شبکه برگشتی مورد نظر را نشان می دهد. از توابع سیگموئید به فرم زیر به عنوان تابع فعالیت لایه های پنهان و خروجی شبکه ، استفاده شده است.

۱ و ۲ به ترتیب پارامترهای کنترلی توابع لایه اول و دوم پنهان هستند که البته در طول آموزش شبکه یک در نظر گرفته می شوند. اتصال داخلی هر دو حلقه فیدبک بدو وزن می باشد. فعالیت واحد ها در زمان  $t$  در لایه پنهان دوم ذخیره شده و به عنوان ورودی شبکه در زمان  $t+1$  استفاده می شود. برای راحتی کار از یک شبکه Elman توسعه یافته ، استفاده شده است. تکنیک تعیین وزن های شبکه هم همان الگوریتم پس انتشار خطا می باشد.



شکل 2-3: شبکه برگشتی.

مقایسه با مدل های دیگر

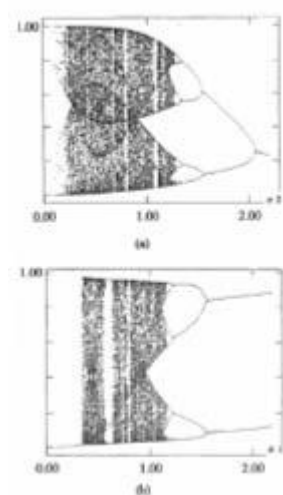
مدل مورد نظر به صورت زمان پیوسته تعریف شده و به همین دلیل توانایی زیادی جهت یادگیری رفتار دینامیک داشته و در مقایسه با مدل های دیگر موجود ، ساده تر می باشد.

نتایج تجربی

نمودار دوشاخه شدن

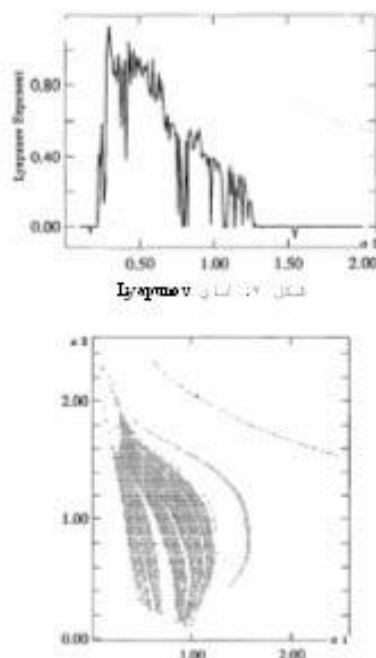
شبکه از توابع فعالسازی توصیف شده با معادله (۲) تشکیل شده است. می توان تغییرات خروجی آن را نسبت به تغییر پارامتر ۲ مشاهده نمود. شکل (۳) دو شاخه شدن خروجی را نسبت به تغییرات این پارامتر نشان می دهد. نمای Lyapunov هم در شکل (۴) مشاهده می شود. تا وقتی که  $2 < 2.09$  ، فقط یک نقطه ثابت برای سیستم وجود دارد. در حوالی  $2.09 = 2$  پاسخ در یک سیکل محدود دوشاخه می گردد. عمل

دو برابر شدن تاوقتی که  $1,241=2$  برسد در این موقع پریود به  $\infty$  برسد. شکل (۵) نمودار دو شاخه شدن را نسبت به تغییرات پارامترهای ۱ و ۲ نشان می دهد. حساسیت به شرط اولیه



شکل 3-3: نمودار دو شاخه شدن.

حساسیت به شرط اولیه یکی از ویژگی مهم آشوب می باشد به همین دلیل رفتار شبکه نسبت به تغییرات شرط اولیه مرد بررسی قرار گرفته است. دو شرط اولیه  $1x = 0,2001$  و  $2x = 0,2000$  انتخاب شده و تراژکتوری های آن ها مقایسه شده اند. همان طور که شکل (۶) نشان می دهد دو تراژکتوری به سرعت از هم دور شده و رشد کاملاً متفاوتی دارند. این قضیه برای دو شرط اولیه دیگر  $1x = 0,60001$  و  $2x = 0,60000$  که اختلاف کمتری نسبت به هم دارند هم آزمایش شده و همان نتیجه قبلی به دست آمده است.



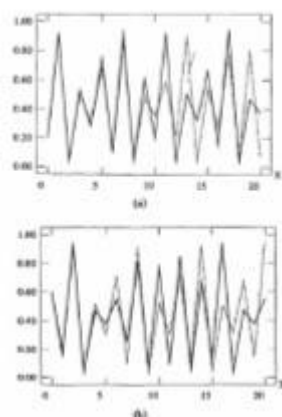
شکل 3-4: نمودار دو شاخه شدن ۹۴۵&۱ نسبت ۹۴۵&۲

### 6-3- تغییرات طیف

طیف توان در نشان دادن دینامیک های آشوبگونه ، بسیار مفید می باشد. اگر سیگنال پریودیک و یا نیمه پریودیک باشد ، طیف توان آن شامل یک سری ضربه در فرکانس های پایه و یا چند فرکانس هارمونیک ، می باشد. اگر سیگنال آشوبگونه باشد ، طیف توان آن پیوسته خواهد بود. بنابراین ایجاد طیف پیوسته از طیف گسسته حضور دینامیک آشوبگونه را در سیگنال نشان می دهد.

در اینجا تغییرات طیف خروجی شبکه نسبت به کاهش ۲ مورد بررسی قرار گرفته است. خروجی شبکه از یک پنجره Hamming عبور داده شده و از آن FFT با ۲۰۴۸ نقطه گرفته شده است. در ابتدای ضربه در فرکانس  $f$  و نوسانات پریودیک دیده می شود. ۲ را کاهش می دهیم تا این که در  $2 = 121,2$  دو ضربه دیگر هم در فرکانس های  $2f/3$  و  $4f/3$  مشاهده می شود مه البته دامنه آن ها نسبت به ضربه قبلی کوچکتر است (شکل ۷-۲b). را باز هم کاهش می دهیم. وقتی  $2 = 1,341$ ، طیف پیچیده تر شده و ضربه های بیشتری را شامل می

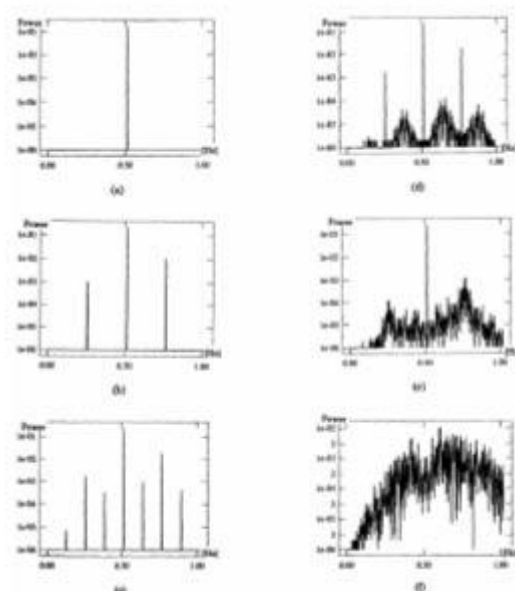
شود اما همچنان گسسته است. دو شاخه شدن تا وقتی ادامه پیدا می کند که دیگر فرکانس ها قابل تشخیص نیستند و در نهایت در  $2=1,248$  طیف کاملاً پیوسته شده و حضور دینامیک آشوبگونه را در سیستم نشان می دهد. بنابراین شبکه با یادگیری نگاشت Hénon رفتار دو شاخه شدن آشوبگونه را به خوبی نمایش می دهد.



شکل 3-5: حساسیت به شرط اولیه. (a: 0.2, نسبت به 0.2001)

### 7-3- نتایج فصل

در این فصل نتایج تجربی و ویژگی های آشوبگونه شبکه عصبی که رفتار آشوبگونه مدل ماهیچه را یادگرفته ، مورد بررسی قرار گرفته است. ویژگی اساسی شبکه مورد نظر این است که ضمن سادگی می تواند دینامیک های غیرخطی را یادگرفته و تقریب بزند.



شکل 3-6: ایجاد طیف پیوسته از طریق افزایش فرکانس های گسسته

در شبکه آموزش دیده ضمن تغییر پارامتر ویژگی های آشوبگونه ای مثل پریود دابل دو شاخه شدن و پنجره پریودیک ، مشاهده شده است.

## فصل چہارم

## هماهنگ سازی نمایی شبکه های عصبی آشوبگونه با اغتشاش تصادفی

### 1-4- مقدمه

در این فصل مسئله هماهنگ سازی شبکه های عصبی آشوبگونه با اغتشاش تصادفی مورد بررسی قرار گرفته است. مسئله کنترل و هماهنگ سازی این سیستم ها به شدت مورد توجه قرار گرفت و روش های مختلفی مثل کنترل حلقه بسته خطی و غیرخطی ، کنترل تطبیقی و نظایر آن جهت رسیدن به این هدف ، ارائه گردید . سیستم آشوبگونه یک سیستم غیرخطی معین است که رفتارهای پیچیده و غیرقابل پیش بینی از خود نشان می دهد.

### 2-4- هماهنگ سازی نمایی شبکه های عصبی آشوبگونه با اغتشاش تصادفی

در این فصل مسئله هماهنگ سازی شبکه های عصبی آشوبگونه با اغتشاش تصادفی مورد بررسی قرار گرفته است. بر اساس تئوری پایداری Lyapunov و با استفاده نامساوی Halanay در مورد معادلات دیفرانسیل تصادفی ، مفهوم پاسخ - درایو و تکنیک های کنترل حلقه بسته تاخیر زمانی ، ضوابطی به دست آمده که هماهنگ سازی نمایی بین دو شبکه آشوبگونه و تاخیر یافته مشابه را به همراه اغتشاش تصادفی ، میسر می سازد. این شرط ها به صورت یک سری نامساوی ماتریسی خطی بیان می شوند. نهایتا از یک مثال برای نشان دادن اثر طرح پیشنهادی ، استفاده شده است.

### 3-4- معرفی

از زمانی که مفهوم پاسخ- درایو برای هماهنگی سیستم های آشوبگونه کوپل شده مطرح شد ، مسئله کنترل و هماهنگ سازی این سیستم ها به شدت مورد توجه قرار گرفت و روش های مختلفی مثل کنترل حلقه بسته خطی و غیرخطی ، کنترل تطبیقی و نظایر آن جهت رسیدن به این هدف ، ارائه گردید. اخیرا مشخص شده که اگر پارامترهای شبکه های عصبی تاخیر یافته (DNNs) به طور خاصی انتخاب شوند ، این شبکه ها می توانند دینامیک های پیچیده و حتی آشوبگونه ای را از خود نشان دهند. به همین دلیل مسئله هماهنگ سازی DNN های آشوبگونه ، به طور وسیعی مورد ملاحظه قرار گرفته است.



سیستم آشوبگونه یک سیستم غیرخطی معین است که رفتارهای پیچیده و غیرقابل پیش بینی از خود نشان می دهد. در مرجع [16] نشان داده شده که اگر به دو مدار آشوبگونه جفت نشده، نویز اضافه کنیم، با هم سنکرون می شوند. در مرجع [28] نویزی که منجر به هماهنگ سازی می شود یک سیگنال با میانگین غیرصفر بوده و از خواص تصادفی آن استفاده نشده است. در مرجع [31] نشان داده شده که یک سطح بهینه برای نویز وجود دارد که در آن بهترین هماهنگی بین دو سیستم به دست می آید. از آنجایی که نویز می تواند منجر به نوسانات پایدار و ناپایدار در سیستم شود، مسئله هماهنگ سازی به وسیله اغتشاش تصادفی هم در بعضی مراجع مورد بررسی قرار گرفته است. به طور مثال در مرجع [29] بیان شده که اگر به دو شبکه آشوبگونه یکسان به اندازه کافی نویز اضافه شود، می توانند با هم سنکرون شوند. البته مسئله هماهنگ سازی نمایی DNN های آشوبگونه به وسیله اغتشاش تصادفی خیلی مورد توجه قرار نگرفته و هنوز جای بحث دارد.

هدف این فصل بررسی هماهنگ سازی نمایی DNN های آشوبگونه به وسیله اغتشاش تصادفی و بر اساس معادلات Lyapunov، می باشد. با استفاده از نامساوی Halanay برای معادلات دیفرانسیل تصادفی و مفهوم پاسخ-درايو، یک کنترلر فیدبک تاخیری بدین منظور طراحی شده و ضوابط به دست آمده به صورت یک نامساوی ماتریسی خطی (LMIs) بیان شده است.

#### 4-4- نمادها و مقدمات

در ابتدا لازم است نمادهای لازم را معرفی کنیم.  $A$ ، یک ماتریس معین مثبت متقارن،  $\|\cdot\|$ ، ماتریس و یا بردار نرم،  $E$ ، اپراتور امید ریاضی،  $T$  ترانهاده ماتریس و  $I$  به مفهوم ماتریس همانی می باشد. بردار نرم  $\|x\|_2$  هم به صورت زیر تعریف می شود:

$$\|x\|^2 = \sum_{i=1}^n |x_i|^2$$

همچنین  $t(\omega)$ ، نویز سفید و  $D+y(t)$  مشتق را نشان می دهد.

در این فصل مدل DNN آشوبگونه به صورت زیر:

$$\dot{x}_i(t) = -c_i x_i(t) + \sum_{j=1}^n a_{ij} f_j(x_j(t)) + \sum_{j=1}^n b_{ij} f_j(x_j(t - \tau)),$$

$$i = 1, 2, \dots, n, \quad (1)$$

و یا به فرم فشرده زیر :

$$\dot{x}(t) = -Cx(t) + Af(x(t)) + Bf(x_\tau(t)), \quad (2)$$

در نظر گرفته می شود که در آن  $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T \in \mathbb{R}^n$  بردار حالت مربوط به نوروها می باشد.  $C = \text{diag}(c_1, c_2, \dots, c_n) > 0$  ، نشان می دهد که واحد  $i$  ، هنگامی که از شبکه و ورودی های خارجی جدا می شود ، با چه سرعتی پتانسیل خود را به حالت استراحت در ایزولاسیون ، ریست می کند.  $A = (a_{ij})_{n \times n}$  و  $B = (b_{ij})_{n \times n}$  به ترتیب ماتریس وزن اتصال و ماتریس وزن اتصالات تاخیر یافته را نشان می دهند.  $f_j$  و  $g_j$  هم توابع فعالسازی می باشند ، که در آن ها  $\tau_j > 0$  تاخیر انتقال را بیان می کند.

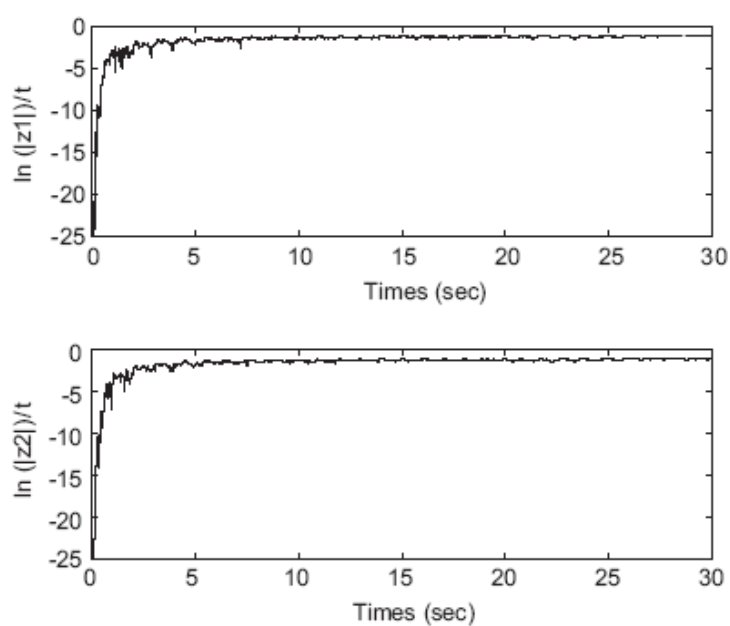
$$f(x(t)) = (f_1(x_1(t)), f_2(x_2(t)), \dots, f_n(x_n(t)))^T \in \mathbb{R}^n \text{ و } g(x_\tau(t)) = (g_1(x_1(t - \tau_1(t))), g_2(x_2(t - \tau_2(t))), \dots, g_n(x_n(t - \tau_n(t))))^T \in \mathbb{R}^n$$

$$(g_n(x_n(t - \tau_n(t))))^T \in \mathbb{R}^n$$

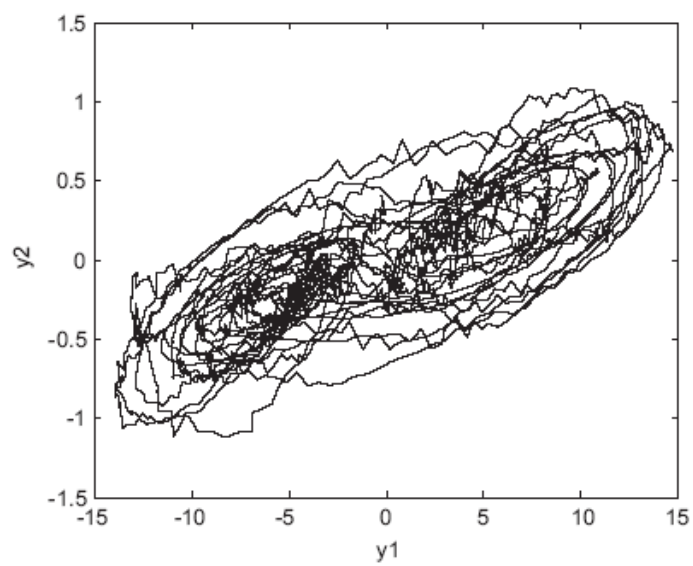
## 5-4- نتیجه فصل

در این فصل مسئله هماهنگ سازی DNNs آشوبگونه مورد بررسی قرار گرفته است. با استفاده از کنترل فیدبک دار تاخیر زمانی و تکنیک های LMI ، ضوابطی به دست آمده که مسئله هماهنگ سازی را تضمین

می کنند. با طراحی مناسب ماتریس های بهره فیدبک ، سیستم درایو و سیستم پاسخ به همراه اغتشاش تصادفی ، می توانند به صورت نمایی سنکرون شوند. تئوری های زیادی در زمینه سیستم های تصادفی وجود دارد که می توان از آن ها در حل مسائل دیگر مربوط به هماهنگ سازی مثل هماهنگ سازی تاخیری ، هماهنگ سازی فاز و نظایر آن استفاده کرد.



شکل 4-1 : نرخ هماهنگ سازی نمایی سیستم با خطای دینامیک.



شکل 4-2: دینامیک های سنکرون نشده در فضای حالت.

## فهرست منابع

### منابع فارسی

- 1- جکسون . تی و بیل . آر . آشنایی با شبکه‌های عصبی ، ترجمه دکتر محمود البرزی - تهران : مؤسسه انتشارات علمی دانشگاه صنعتی شریف ، چاپ دوم ، 1383
- 2- کاتلر ، فیلیپ . مدیریت بازاریابی ، ترجمه بهمن فروزنده - تهران : آتروپات ، 1382
- 3- قمی ، علیرضا " شبکه های عصبی مصنوعی "، نشریه دنیای کامپیوتر و ارتباطات - شماره 12 ، صفحات 66 تا 69
- 4- سعیدی ، مسعود " شبکه های عصبی (2) " ، نشریه شبکه \_ شماره 52 ، اسفند 1383 ، صفحه 210 تا 211
- 5- ممانی ، حامد ، نرگس پور اصغری حقی و ساعد علی ضمیر ، " شبکه های عصبی و کاربرد آن در بهینه سازی " ، نشریه صنایع \_ شماره 30
- 6- نورزاد ، غلامرضا " بیولوژی سلولی مولکولی " ، انتشارات نوردانش ، تهران ، 1376 ، چاپ اول

## منابع انگلیسی

- 1- Metaxiotis , Kostas & John Psarras (2004) "The Contribution of Neural networks and genetic algorithms to business decision support "Management decision , vol 42 ,no .2, Emerald group publishing limited , pp. 229.242
- 2- Curry , B & L. Moutinho (1993) "Neural Network in marketing : Modelling consumer Responses to Advertising Stimuli "European Journal of marketing , vol 27 , no . 7 , MCB university press , pp 5. 20
- 3- Wray , B , A. palmer & D. Bejou (1994) " Using Neural Network Analysis to evaluate 4- Buyer Seller Relationships " European Journal of Marketing , vol 28 , no. 10 , MCB university press , pp 32.48
- 5- Venugopal .V & W. Beats ( 1994 ) " Neural networks and Statistical Techniques in marketing research " Marketing intelligence & planning , vol 12 , no. 7 , MCB university press , pp 30 . 38
- 6- Davies , F , L . Moutinho & B . Curry (1996 ) " ATM user attitudes : a neural network analysis " , marketing intelligence & planning , vol 14 , no . 2 , MCB university press , pp 26 . 32

### **Abstract**

Basic instructions for the preparation of a paper for the Amirkabir Journal are presented. This document is itself an example of the desired layout and can be used as a template. The document contains information for all formats, type sizes, and typefaces in required styles, without need to define any. Style rules are provided to explain how to handle equations, units, figures, tables, abbreviations, and acronyms. Sections are also devoted to the preparation of acknowledgments, references, and authors' biographies

words in one or two paragraphs, and cannot contain equations, 180The English abstract is limited to figures, tables, or references. It should concisely state what was done, how it was done, principal results, and their significance

Key words : preparation – presented- template – acronyms.

[www.Prozhe.com](http://www.Prozhe.com)