فصل دهم

شبکه عصبی مصنوعی

1-8: مقدمه

شبکه‌های عصبی مصنوعی (ANN)[1] یا به زبان ساده‌تر شبکه‌های عصبی، سیستم‌ها و روش‌های محاسباتی نوینی هستند که جهت یادگیری ماشینی، نمایش دانش و در انتها اعمال دانش به دست آمده در جهت پیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده به‌کار می‌روند. ایده اصلی این‌گونه شبکه‌ها (تاحدودی) الهام‌گرفته از شیوه کارکرد سیستم عصبی زیستی، برای پردازش داده‌ها و اطلاعات به منظور یادگیری و ایجاد دانش است. عنصر کلیدی این ایده، ایجاد ساختارهای جدید برای سامانه پردازش اطلاعات است. این سیستم از شمار زیادی عناصر پردازشی فوق العاده به‌هم‌پیوسته، با نام نرون تشکیل شده که برای حل یک مسأله با هم هماهنگ عمل می‌کنند و توسط سیناپس[2]ها (ارتباطات الکترومغناطیسی) اطلاعات را منتقل می‌کنند. در این شبکه‌ها اگر یک سلول آسیب ببیند بقیه سلول‌ها می‌توانند نبود آن‌را جبران کرده و نیز در بازسازی آن سهیم باشند. این شبکه‌ها قادر به یادگیری‌اند. برای مثال با اعمال سوزش به سلول‌های عصبی لامسه، سلول‌ها یاد می‌گیرند که به طرف جسم داغ نروند و با این الگوریتم سیستم می‌آموزد که خطای خود را اصلاح کند. یادگیری در این سیستم‌ها به صورت تطبیقی صورت می‌گیرد، یعنی با استفاده ازمثال‌ها وزن سیناپس‌ها به گونه‌ای تغییر می‌کند که در صورت دادن ورودی‌های جدید، سیستم پاسخ درستی تولید کند.

توافق دقیقی بر تعریف شبکه عصبی در میان محققان وجود ندارد؛ اما اغلب آن‌ها موافقند که شبکه عصبی شامل شبکه‌ای از عناصر پردازش ساده (نرون[3]ها) است که می‌تواند رفتار پیچیده کلی تعیین شده‌ای از ارتباط بین عناصر پردازش و پارامترهای عنصر را نمایش دهد. منبع اصلی و الهام بخش برای این تکنیک، از آزمایش سیستم مرکزی عصبی و نرون‌ها (آکسون‌ها، شاخه‌های متعدد سلول‌های عصبی و محل‌های تماس دو عصب) نشأت گرفته‌است که یکی از قابل توجه‌ترین عناصر پردازش اطلاعات سیستم عصبی را تشکیل می‌دهد. در یک مدل شبکه عصبی، گره‌های ساده (نرون، نئورون، PE[4]ها (عناصر پردازش)

---

[1] Artificial Neural Network

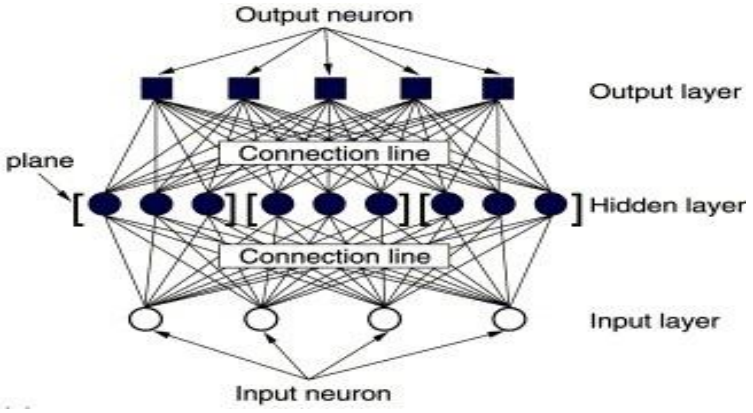
[2] Synapse

[3] Neuron

[4] Process element

یا واحدها) برای تشکیل شبکه‌ای از گره‌ها، به هم متصل شده‌اند و به همین دلیل به آن، اصطلاح "شبکه‌های عصبی" اطلاق می‌شود. در حالی‌که یک شبکه عصبی نباید به خودی خود سازگارپذیر باشد، استفاده عملی از آن به‌واسطه الگوریتم‌هایی امکان‌پذیر است که جهت تغییر وزن ارتباطات در شبکه (به منظور تولید سیگنال موردنظر) طراحی شده باشند. با استفاده از دانش برنامه‌نویسی رایانه می‌توان ساختار داده‌ای طراحی کرد که همانند یک نرون عمل نماید. سپس با ایجاد شبکه‌ای از این نرون‌های مصنوعی به هم پیوسته، ایجاد یک الگوریتم آموزشی برای شبکه و اعمال این الگوریتم به شبکه، آن‌را آموزش داد.

این شبکه‌ها برای تخمین[5] و تقریب[6] کارایی بسیار بالایی از خود نشان داده‌اند. گستره کاربرد این مدل‌های ریاضی برگرفته از عملکرد مغز انسان، بسیار وسیع می‌باشد که به عنوان چند نمونه کوچک می‌توان استفاده از این ابزار ریاضی در پردازش سیگنال‌های بیولوژیکی، مخابراتی و الکترونیکی تا کمک در نجوم و فضانوردی را نام برد.



شکل (8-1): ساختار کلی شبکه‌های عصبی مصنوعی

شبکه‌های عصبی با توانایی قابل‌توجه خود در استنتاج از داده‌های پیچیده می‌توانند در استخراج الگوها و شناسایی گرایش‌های مختلفی که برای انسان‌ها و کامپیوتر، شناسایی آنها بسیار دشوار است استفاده شوند.

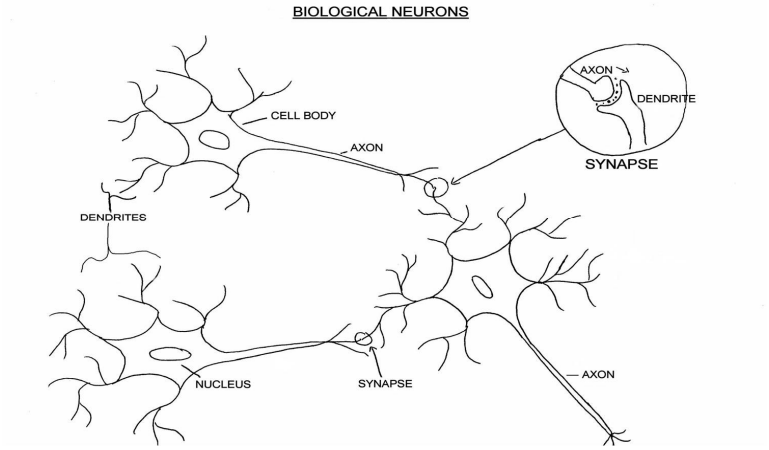
---

[5] Estimation

[6] Approximation

شبکه عصبی روش خام تلاش انسان‌ها جهت شبیه‌سازی الکترونیکی مغز است. بنابراین برای این‌که بفهمیم شبکه عصبی چطور کار می‌کند، ابتدا باید بفهمیم ماده خاکستری چطور کارش را انجام می‌دهد. مغز ما از چیزی حدود 100 بیلیون واحد بسیار ریز به نام نرون ساخته شده است. هر نرون به هزاران نرون دیگر متصل است و از طریق سیگنال‌های الکتروشیمیایی با آنها در ارتباط است. سیگنال‌هایی که به یک نرون می‌رسند از طریق اتصالاتی که سیناپس نام دارند دریافت می‌شود. این اتصالات در انتهای هسته سلول عصبی که منشعب می‌شود دندریت[7] نام دارد. نرون به‌طور پیوسته از این ورودی‌ها سیگنال می‌گیرد و به همراه یک مقدار آن را ارائه می‌کند. آنچه که نرون انجام می‌دهد، جمع کردن ورودی‌ها در خودش است، چنانچه نتیجه نهایی از یک مقدار آستانه[8] بیشتر شود نرون برانگیخته[9] می‌شود و یک ولتاژ ایجاد می‌کند و سیگنالی را در امتداد جسمی‌که آکسون[10] نام دارد می‌فرستد.

نگران نباشید... لازم نیست تمام این لغت‌های جدید را به‌خاطر بسپارید. در ادامه می‌بینید که ما احتیاج زیادی به استفاده از این لغت‌ها نداریم، حالا فقط به تصویر نگاه کنید و سعی کنید نحوه عملکرد این سلول کوچک را مجسم کنید.



[7] Dendrites

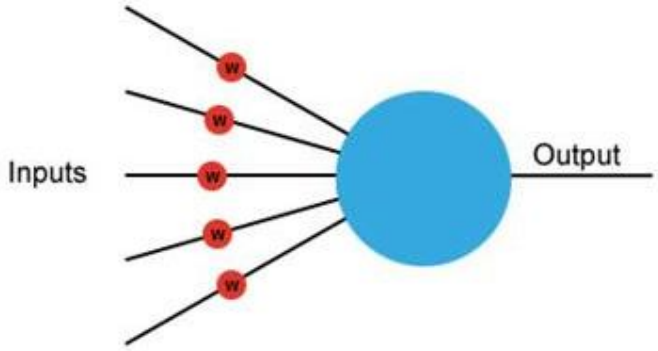
[8] Threshold

[9] Fire

[10] Axon

شکل (8-2): نرون و سیناپس‌ها

شبکه عصبی از تعداد زیادی نرون‌های عصبی ساخته شده است. اما نرون عصبی مصنوعی چیست؟ یک نرون مصنوعی، به ساده‌ترین بیان، نرون بیولوژیکی را بطور الکترونیکی مدل می‌کند. تعداد نرون‌هایی که استفاده می‌شود به وظیفه‌ای که در حال اجراست بستگی دارد. روش‌های متعددی جهت اتصال نرون‌های مصنوعی برای ایجاد یک شبکه عصبی وجود دارد.



شکل (8-3) : نرون مصنوعی

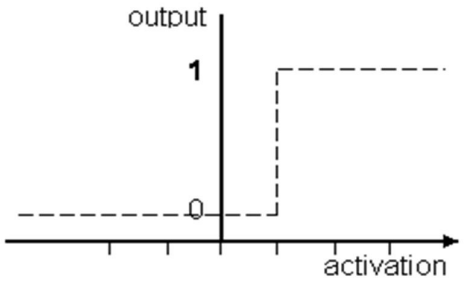
هر ورودی به نرون وزن[11] خودش را دارد که مقدار آن توسط دایره‌های قرمز نشان داده‌شده، داده‌شده، تعیین می‌شود. یک وزن عددی با ممیز شناور است و این‌ها چیزهایی هستند که هنگام آموزش شبکه عصبی تنظیم می‌شوند. وزن‌ها در شبکه عصبی می‌توانند منفی یا مثبت باشند، بنابراین می‌توانند تاثیر بازدارنده یا تحریکی بر روی ورودی داشته باشند. وقتی هر ورودی به هسته (دایره بزرگ شکل) وارد می‌شود در وزنش ضرب می‌شود، سپس هسته تمام ورودی‌های رسیده را جمع می‌کند و در نهایت به ما تحریک را می‌دهد. (عدد با ممیز شناور نیز می‌تواند مثبت یا منفی باشد.)

اگر تحریک بیش از مقدار آستانه باشد، (به عنوان مثال اجازه دهید از عدد 1 استفاده کنیم) نرون یک سیگنال خروجی خواهد داشت. اگر تحریک کمتر از یک باشد خروجی نرون صفر

---

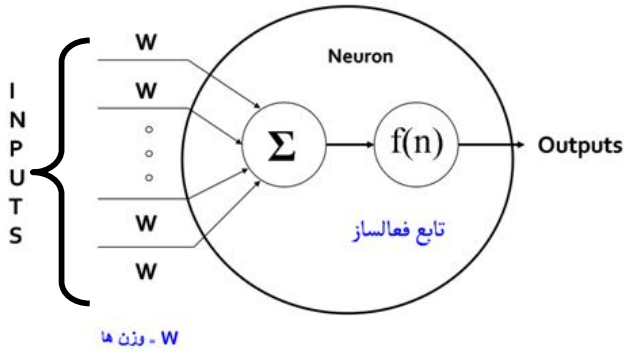
[11] Weight

است. این مورد عموما تابع پله نامیده می‌شود. (یک نگاه زیرچشمی به تصویر زیر بیاندازید.) می‌توانید حدس بزنید چرا...



شکل (8-4) : فعال‌سازی و نمود آن در خروجی

نرون مجموع وزن‌دار ورودی‌ها را محاسبه می‌کند و آن‌را با یک مقدار آستانه مقایسه می‌کند. اگر این مجموع بزرگتر از مقدار آستانه باشد خروجی 1 و اگر کوچکتر از آن باشد خروجی 1- خواهد بود (در صورتی‌که ( ) تابع پله باشد). این امر در شکل نمود دارد.



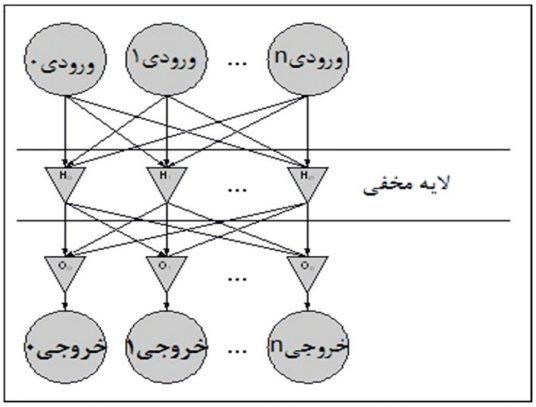
شکل (8-5) : یک عصب مصنوعی پایه (تابع فعال‌ساز)

شبکه عصبی روشی برای محاسبه است که بر پایه اتصال به هم پیوسته چندین واحد پردازشی ساخته می‌شود.

شبکه از تعداد دلخواهی سلول یا گره یا واحد یا نرون تشکیل می‌شود که مجموعه ورودی را به خروجی ربط می‌دهند.

شبکه عصبی مصنوعی روشی عملی برای یادگیری توابع گوناگون نظیر توابع با مقادیر حقیقی، توابع با مقادیر گسسته و توابع با مقادیر برداری می‌باشد.

یادگیری شبکه عصبی در برابر خطاهای داده‌های آموزشی مصون بوده و این‌گونه شبکه‌ها با موفقیت به مسائلی نظیر شناسائی گفتار، شناسائی و تعبیر تصاویر و یادگیری روبات اعمال شده است.



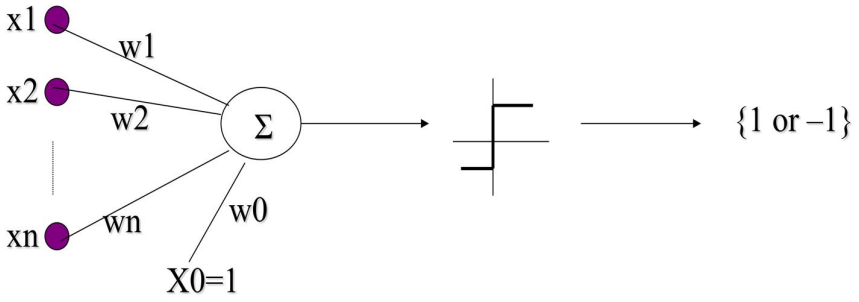
شکل (8-6) : لایه مخفی ورودی را به خروجی می‌برد

شبکه عصبی قابلیت‌هائی زیر را دارد:

- محاسبه یک تابع معلوم
- تقریب یک تابع ناشناخته
- شناسائی الگو
- پردازش سیگنال
- یادگیری

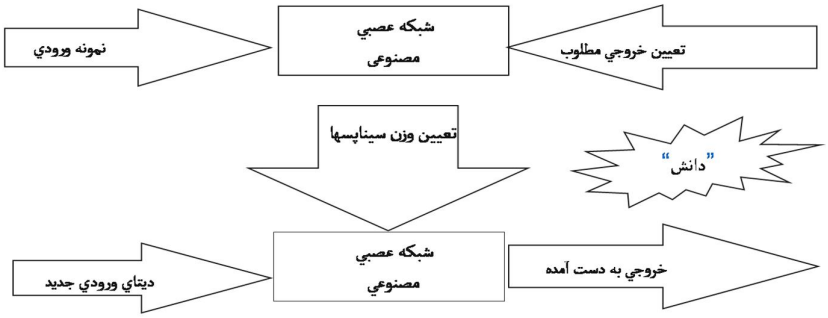
2-8: پرسپترون[12]

نوعی از شبکه عصبی برمبنای یک واحد محاسباتی به نام پرسپترون ساخته می‌شود. پرسپترون برداری از ورودی‌هایی با مقادیر حقیقی را گرفته و ترکیب خطی از این ورودی‌ها را محاسبه می‌کند. اگر حاصل از یک مقدار آستانه بیشتر باشد خروجی پرسپترون برابر با 1 و در غیر این‌صورت معادل 1- خواهد بود.



شکل (8-7) : عملکرد تابع فعال‌سازی

در شکل چگونگی عملکرد شبکه عصبی در حالت کلی نشان داده شده است.



شکل (8-8) : چگونگی عملکرد شبکه عصبی

[12] Perceptron

1-2-8: یادگیری[13]

تعریف: یادگیری پروسه‌ای است که طی آن شبکه عصبی پارامترهای خود را به گونه‌ای تغییر می دهد که بهترین شبیه‌سازی محیط را انجام دهد.
در شبکه‌های عصبی با پروسه تعریف شده تغییر اوزان در زمان باعث یادگیری می‌شود.
در یادگیری چند موضوع اساسی وجود دارد:

- پارادایم[14](مثال یا الگو، روش کلی)
- مکانیزم‌ها و روش‌ها
- معیار

1-1-2-8: پارادایم‌های یادگیری

یادگیری با مربی[15]: که در آن مجموعه‌ای از مثال داریم که به صورت زوج‌های مرتب برداری (ورودی-خروجی مطلوب [X,t]) هستند.
یادگیری بدون مربی[16](خودسازمان‌ده[17]) که در آن تنها ورودی X وجود دارد و توزیع آن در فضا قابل استخراج است.
یادگیری تشدیدی[18] که در آن یک سری وجودی و یک معیار مطلوبیت اسکالر[19] (نه خروجی برداری) داریم.

2-1-2-8: مکانیزم‌های یادگیری

- یادگیری تصحیح خطا[20]

---

[13] Learning
[14] Paradigm
[15] Supervised learning
[16] Unsupervised learning
[17] Self organizing
[18] Reinforcement learning
[19] Scalar
[20] Error correcting learning

- یادگیری هب[21]
- یادگیری رقابتی[22]
- یادگیری بولتزمان[23]

3-1-2-8: معیارهای یادگیری

پیاده‌سازی مکانیزمهای یادگیری مستلزم ارائه الگوریتم‌ها و فرمول‌های ریاضی است.
در همه مکانیزم‌ها نوعی ایده بهینه‌سازی وجود دارد که طی آن یک تابع هدف، هزینه یا انرژی، در فضای اوزان، مورد بررسی قرار گرفته و آرایش اوزان شبکه عصبی بطور دفعه‌ای و یا به‌تدریج از نقطه اولیه به سمت نقطه بهینه منتقل می‌شود. چگونگی محاسبه تغییر لازم در بردار اوزان را روش یادگیری تعیین می‌کند.

2-2-8: آموزش[24]

در مورد آموزش شبکه عصبی سوالات زیر مطرح می‌شود:
سوال: هنگام آموزش شبکه عصبی چرا سعی داریم به جواب 100% برسیم و آیا این دقت نشان‌دهنده دقت نهایی شبکه است؟
پاسخ: از آنجایی که در طی فرآیند آموزش شبکه داده‌های محدودی در دست داشته و نیز به طور معمول محدودیتی در تعداد تکرار نداریم می‌توانیم از این فرصت حداکثر استفاده را داشته و سعی کنیم برای داده‌های آموزش به پاسخ 100% دست بیابیم. اما از آنجایی که مطمئنا تمامی حالت‌های مورد نظر یا به عبارت بهتر تمامی ورودی و خروجی‌ها را در مرحله آموزش در اختیار نداشته‌ایم، بدیهی می‌باشد که شبکه برای برخی از دادها در مرحله تست به درستی عمل نکند و پاسخ مورد نظر را تولید نکند. به عبارت ساده‌تر برخی از داده‌ها برای شبکه تازگی دارند. یک نتیجه که می‌توان از این بحث داشت این است که برای آموزش بهتر است از داده‌های بیشتر و متفاوت استفاده کنیم.

---

[21] Hebbian learning
[22] Competitive learning
[23] Boltzman learning
[24] Train

سوال: چگونگی ترتیب اعمال داده‌ها چه تاثیری در روند یادگیری دارد؟

پاسخ: می‌دانیم هر داده‌ای که در مرحله آموزش به شبکه اعمال می‌شود بر روی وزن‌های شبکه تاثیر گذاشته و آنها را به نفع خود تصحیح می‌کند. بنابراین ترتیب اعمال این داده‌ها بر روی شبکه بسیار حائز اهمیت است. فرض کنید داده‌هایی داریم که شبکه باید آنها را در دو کلاس دسته‌بندی کند. حال اگر در مرحله آموزش، داده‌های مربوط به کلاس اول و کلاس دوم را به صورت یکی در میان به شبکه اعمال کنیم، شبکه در هر تکرار وزن‌ها را به گونه‌ای تغییر می‌دهد که با تکرار قبلی و بعدی خود کاملا تناقض دارد و این مساله باعث می‌شود آموزش شبکه به تاخیر بیافتد و به تعداد تکرار بیشتری احتیاج داشته باشد.

سوال : چگونه وزن‌های یک پرسپترون واحد را یاد بگیریم به نحوی‌که پرسپترون برای مثال‌های آموزشی مقادیر صحیح را ایجاد نماید؟

پاسخ: این سوال الگوریتم یادگیری و قانون پرسپترون است.

8-2-3: الگوریتم یادگیری

1- مقادیری تصادفی به وزن‌ها نسبت می‌دهیم.
2- پرسپترون را به تک‌تک مثال‌های آموزشی اعمال می‌کنیم. اگر مثال غلط ارزیابی شود، مقادیر وزن‌های پرسپترون را تصحیح می‌کنیم.
3- آیا تمامی مثال‌های آموزشی درست ارزیابی می‌شوند:
بله ← پایان الگوریتم
خیر←به مرحله 2 برمی‌گردیم.

8-2-4: قانون پرسپترون

برای یک مثال آموزشی (,…, ,) = در هر مرحله وزن‌ها بر اساس قانون پرسپترون به‌صورت زیر تغییر می‌کند:

$$w + 1 = wi + \Delta w$$

که در آن

$$\Delta w = \eta (t - o) x_i$$

خروجی هدف[25] :t

خروجی تولید شده به‌وسیله پرسپترون[26] :o

ثابت فراخوانی شده به‌وسیله نرخ یادگیری[27] :η

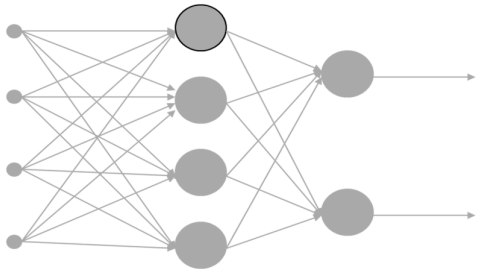
اثبات شده است که برای یک مجموعه مثال جداپذیر خطی، این روش همگرا شده و پرسپترون قادر به جداسازی صحیح مثال‌ها خواهد شد.

3-8: ساختارهای شبکه عصبی

1-3-8: شبکه‌های پیشخور (روبه جلو)[28]

این نوع شبکه‌ها به دو نوع کلی پرسپترون تک لایه (تنها یک لایه مخفی) و پرسپترون چندلایه (مخفی) تقسیم می‌شوند. پرسپترون‌ها به‌وسیله اتصالات بین لایه‌ای منظم شده‌اند و نرون (گره)ها در لایه i به نرون‌های لایه i+1 متصل هستند و اتصال فرالایه‌ای وجود ندارد.

این نوع شبکه مانند زیر دارای یک یا چند لایه (مخفی) بین نرون‌های ورودی و خروجی هستند.



شکل (8-9) : شبکه چند ورودی چند لایه

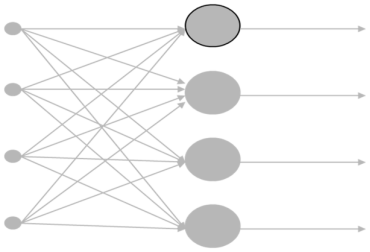
---

[25] Target output

[26] Output generated by the perceptron

[27] Constant called the learning rate

[28] Feed-forward

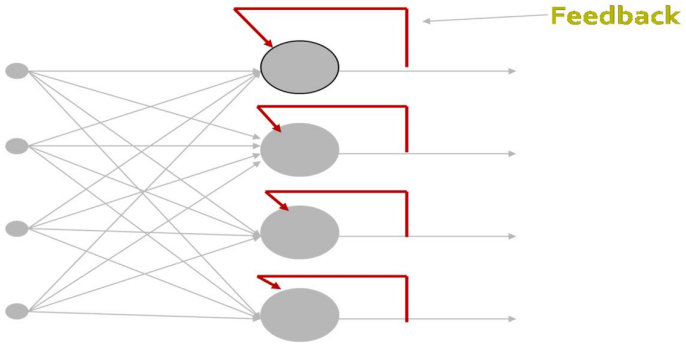


شکل (8-10) : شبکه چند ورودی یک لایه

شکل (8-9) : شبکه چند ورودی **چند لایه** و دو نوع متفاوت از شبکه‌های رو به جلو را نشان می‌دهد. شکل یک پرسپترون با ساختار 2-4-4 می‌باشد.

8-3-2: ساختارهای بازگشتی[29]

این شبکه‌ها می‌توانند بوسیله الگوریتم انتشار رو به عقب[30] آموزش[31] ببینند.



شکل (9-11) : شبکه های بازگشتی

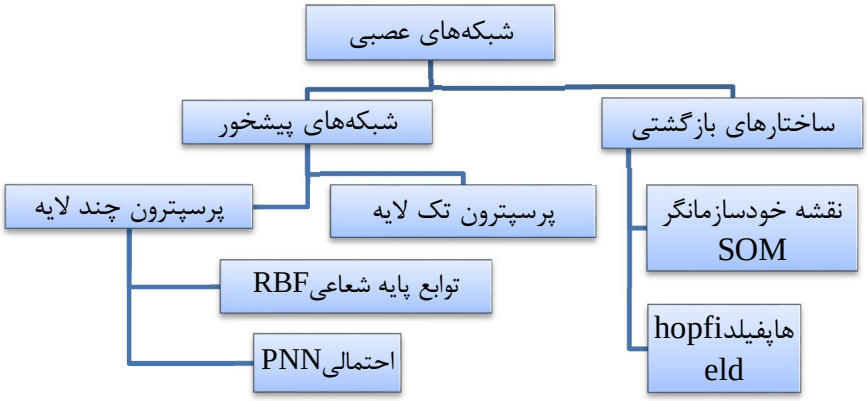
نگاه کلی به شبکه‌های عصبی دسته بندی شکل زیر را نتیجه می‌دهد:

---

[29] Recurrent or feedback

[30] Back propagation

[31] Train



شکل (8-12) : سلسله مراتب شبکه‌های عصبی مصنوعی

8-3-3: معروفترین انواع شبکه های عصبی

شبکه‌های عصبی گاهی از روی اجزاء، گاهی ساختار و گاهی روش یادگیری یا ابداع‌کننده آنها نام‌گذاری می‌شوند.

- شبکه های رو به جلو
- شبکه های (RBF)[32]
- شبکه های Hopfield
- شبکه های (SOFM)[33] یا Kohonen
- شبکه های PNN[34]

حداقل 30 نوع شبکه عصبی دیگر نیز وجود دارد که در این کتاب به آنها پرداخته نمی‌شود.

تعاریف:

شبکه های رو به جلو

---

[32] Radial Basis Function

[33] Self organizing feature map

[34] Probability neural network

زمانی است که تمام جهات اتصال نرون به نرون در حرکت شبکه تنها از ورودی به خروجی است. الگوها وارد شبکه می‌شوند و خروجی از شبکه گرفته می‌شود و تا اتمام الگوهای ورودی همین کار تکرار می‌شود.

Recurrent or feedback networks

جهت‌ها عمل پس خوران[1] به لایه‌های قبلی را انجام می‌دهند. به عبارت دیگر نتیجه پردازش ورودی همزمان به خروجی و ورودی الگوی بعدی وارد می‌شود و در واقع ورودی شبکه الگوی جدید به همراه خروجی شبکه از الگوی قبلی است.

Hidden layer

لایه میانی نرون‌ها است که نه لایه ورودی و نه لایه خروجی است.

Hidden units

واحد(نرون) هایی که بطور مستقیم به نرون‌های ورودی و نرون‌های خروجی متصل نیستند.

(Multi) Perceptron

شبکه‌ای با یک لایه از وزن‌هاست و منظور از پرسپترون چندلایه شبکه‌ای، چند لایه از وزن‌ها است.

مثال: به عنوان مثالی از دسته‌بندی الگو[2]، دسته‌بندی سیگنال‌های الکتریکی را انجام می‌دهیم.

الگوی ورودی: دارای 2 ویژگی[3] است و به مقادیر واقعی بین 1- و 1 نرمال شده است.

الگوهای خرورجی در 3 دسته[4] قرار می‌گیرد.

ساختار شبکه به صورت 3-5-2 می‌باشد بدین معنی که 2 نرون ورودی، 5 نرون برای یک لایه مخفی و 3 نرون خروجی وجود دارد.

---

[1] Feedback

[1]Pattern classification

[3] Feature

[4] Class

با استفاده از 332 الگوی برای آموزش شبکه و 20000 بار تکرار عمل پردازش ورودی برای خروجی، حاصل کار به صورت جدول زیر است:

جدول 8- 1دسته بندی الگوها در 3 کلاس مختلف

دسته هایی که شبکه الگوها را در آن قرار می‌دهد	1	2	3
دسته مطلوب و هدف[1]			
دسته 1	75	5	0
دسته 2	3	88	9
دسته 3	2	19	131

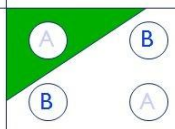
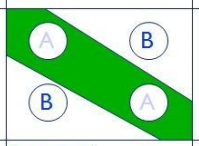
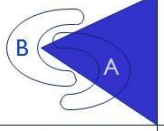
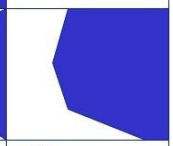
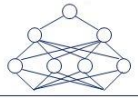
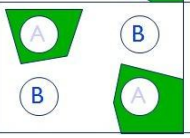
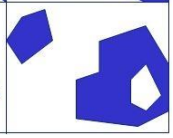
در انتها 38 الگو در هیچ‌کدام از دسته‌ها قرار نگرفتند.

خانه‌های هاشورخورده در جدول تعداد الگوهایی را نشان می‌دهد که در کلاس مورد نظر خود قرار گرفته‌اند. ولی بقیه خانه‌ها تعداد الگوهای به اشتباه دسته‌بندی‌شده را نشان می‌دهد. برای مثال، 19 الگو به جای قرار گرفتن در دسته2 به علت خطای شبکه در دسته3 قرار گرفته اند.

قدرت تحلیل مسائل مختلف به‌وسیله جدول برای پرسپترون‌های یک لایه، دولایه و سه لایه نشان داده شده است.

جدول 8-2مسائل تفکیک پذیر غیر خطی متفاوت

[1] Target

ساختار	انواع زمینه های تصمیم	مسئله Exclusive-OR	دسته ها با قسمتهای درهم	شکلهای با قسمتهای در حالت کلی
تک لایه 	نصف کردن فضای مورد مطالعه			
دو لایه 	Convex Open یا محدوده های بسته			
سه لایه 	دلخواه (پیچیدگی محدود به تعداد نرون(گره)ها ست.			

4-8: شبکه عصبی هاپفیلد[1]

1-4-8: حافظه انجمنی[2]

شبکه هاپفیلد از آنجا که رفتار حافظه‌های انجمنی را از خود نشان می‌دهد می‌تواند در مواردی بسیار کاربردی باشد. برای روشن‌تر شدن موضوع به رفتار حافظه انسان در به خاطر آوردن یک تصویر دقت کنید. انسان با دیدن قسمتی از یک تصویر به سادگی و با سرعت بسیار زیاد می‌تواند یک خاطره مرتبط با آن تصویر را به خاطر آورد. ورودی ذهن انسان در این پروسه قسمتی از محتوای ذخیره شده در حافظه است و نه آدرس یک سلول از حافظه، به این نوع حافظه، حافظه انجمنی اطلاق می‌گردد. واضح است که اگر مغز انسان با استفاده از اطلاعات ورودی اقدام به یک جستجوی ترتیبی در میان انبوه اطلاعات خود می‌نمود به خاطر آوردن یک خاطره امری ناممکن می‌نمود حال آنکه با استفاده از حافظه انجمنی نه تنها این کار ممکن می‌باشد بلکه با سرعتی بسیار بالا نیز انجام پذیر گردیده است. شبکه عصبی هاپفیلد اگر چه ساختار بسیار ساده‌ای دارد ولی تقلیدی از این نوع حافظه می‌باشد.

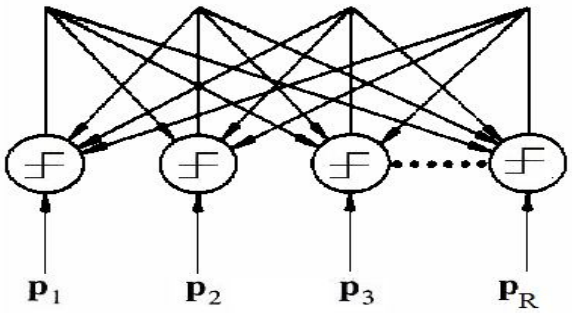
---

[1] Hopfield

[2] Associative memory

2-4-8: معماری شبکه هاپفیلد

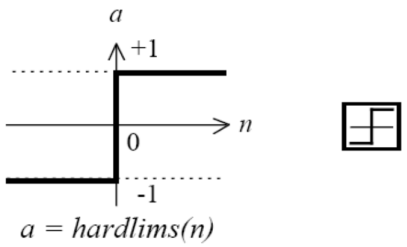
معماری شبکه به صورت شکل می‌باشد:



شکل (8-13) : ساختار شبکه هاپفیلد

این شبکه از نوع بازگشتی تک لایه است.

در این شبکه هر نرون، یک نرون پرسپترون ساده با تابع فعال‌سازی hardlims می‌باشد. این تابع فعال‌سازی در شکل زیر نشان داده شده است:



Symmetric Hard Limit Trans. Funct

شکل (8-14) : تابع فعال‌ساز hardlims

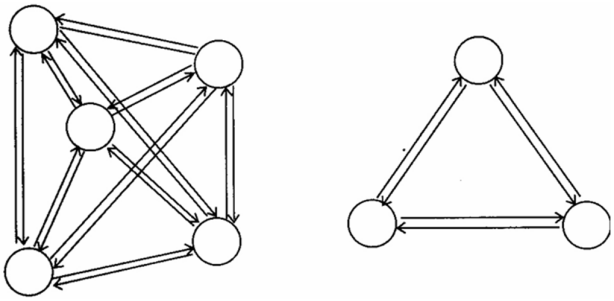
نحوه کار شبکه بدین صورت است که الگوی ورودی در بردار a ذخیره می‌شود:

$$a(0) = P$$

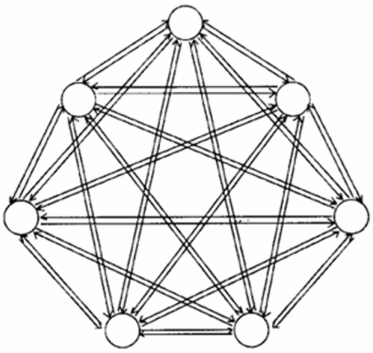
طبق فرمول زیر مراحل بعدی محاسبه بردار a محاسبه می‌شود و این کار تا زمان معینی ادامه می‌یابد.

$$a_i(t+1) = \text{hardlims}(\sum_j w_{ij} a_j(t))$$

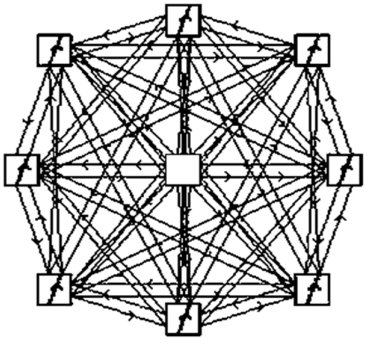
شبکه هاپفیلد شبکه‌ای است که ورودی هر نرون در این لایه در مرحله اول الگوها هستند و در مراحل بعدی خروجی نرون‌های دیگر ورودی نرون را تشکیل می‌دهند. هیچ نرونی در این شبکه خروجی خود را به عنوان ورودی نمی‌گیرد و مانند یگ گراف کامل بدون حلقه است که هر جفت نرون به طور متقابل خروجی خود را به یکدیگر می‌دهند. دید گراف گونه از شبکه در تصاویر زیر نمود دارد:



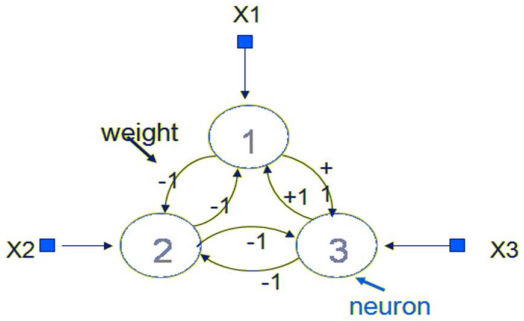
شکل (8-15) : شبکه 3و 5 گره‌ای هاپفیلد



شکل (8-16) : شبکه 7 گره‌ای هاپفیلد

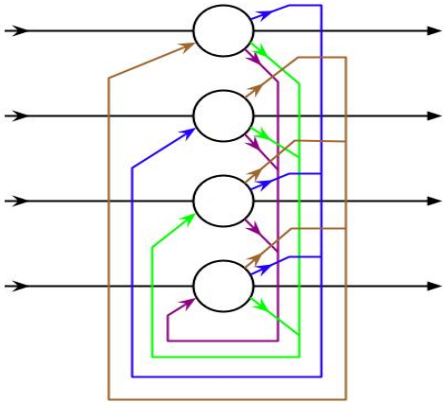


شکل (8-17) : شبکه 9گره‌ای هاپفیلد



شکل (8-18) : شبکه 3گره‌ای هاپفیلد

این چرخه تا رسیدن به ثبات که همان مساوی شدن خروجی مرحله iام با i+1ام است.
شکل نمای خوبی از شبکه هاپفیلد است.



شکل (8-19) : ساختار کلی شبکه هاپفیلد

اگر خروجی نرون‌های مرحله قبلی با اندیس j نشان داده شود و محاسبه مقدار نرون با اندیس i در مرحله جدید مطلوب باشد، می‌توان به یکی از دو حالت زیر محاسبه کرد.

$$\leftarrow \begin{matrix} 1 & > \\ -1 & h \end{matrix}$$

$$\leftarrow \begin{matrix} 1 & > \\ 0 & h \end{matrix}$$

3-4-8: آموزش شبکه

آموزش به این شبکه ساده است ولی مراحل تست آن طولانی است. آموزش از نوع بدون ناظر است و در یک مرحله صورت می‌گیرد. نحوه محاسبه وزنها نیز از طریق دو فرمول زیر ممکن است:

$$= \sum \quad (2 \quad -1)\ 2 \quad -1 \qquad (1)$$

$$= \ - \ \sum \qquad (2)$$

که V^sبردار مقادیر الگوی ورودی است و اندیس i و j شمارنده نرون‌ها است و s اندیس شمارنده الگوها می‌باشد. فرمول شماره 2 همان روش هب است.

4-4-8: بروزرسانی نرون ها

می‌توان از فرمول زیر برای بروزرسانی هر نرون در هر مرحله استفاده کرد:

=

در این شبکه ابتدا یک موج سینوسی به عنوان ورودی به شبکه داده می‌شود و به تکه‌های مساوی تقسیم شده و به عنوان patternهای ورودی در نظر گرفته می‌شود. سپس با استفاده از دستور rand و randn به نمودار سینوسی نویز داده می‌شود. سپس با استفاده از فرمول محاسبه بالا مقادیر نرون‌ها در هر مرحله محاسبه می‌شود و با مقدار مورد نظر که همان الگوی اولیه بدون نویز است مقایسه می‌شود و تا زمانی که به مقدار یکسان نرسد ادامه می‌یابد.

5-8: شبکه SOM

SOM مخفف عبارت Self Organizing Map به معنای "نقشه خودسازمان‌گر" می‌باشد. در واقع SOM نیز همانند شبکه‌های Hopfield, Back propagation , Feed forward یک روش یادگیری است. SOM شبکه‌ای با ساختار بسیار ساده ولی با قابلیت‌های بسیار زیاد

می‌باشد. شبکه SOM دولایه است و اتصالات در شبکه به صورت کاملا متصل[1] می‌باشد. یادگیری در شبکه SOM به صورت بدون سرپرست و رقابتی می‌باشد.

تا به حال با شبکه‌هایی روبه‌رو بودیم که اطلاعات دقیقی در مورد نرون‌های ورودی و خروجی آن در اختیار داشتیم. اما مسائل دیگری نیز وجود دارد که خروجی آنها در دسترس ما نیست و تنها اطلاعات ما به وسیله مجموعه‌ای از الگوهای ورودی فراهم می‌شود. در این گونه مسائل اطلاعات مورد نیاز دیگر خود را از میان الگوهای آموزش پیدا می‌کنیم. روش‌های زیادی برای حل این‌گونه مسائل وجود دارد. چیزی که در تمام این روش‌ها مشترک می‌باشد این است که آموزش باید بدون حضور سرپرست انجام پذیرد.

انواع مختلفی از شبکه‌های عصبی یادگیری بدون سرپرست[2] قادر به حل این‌گونه مسائل هستند. شبکه‌های خود سازمان‌ده (SOM) یا kohonen یکی از انواع شبکه‌های عصبی است که بدون نیاز به سرپرست قادر به آموزش بر روی داده‌ها است.

این شبکه عصبی برای نمایش و سازماندهی داده‌های پیچیده به‌کار می‌رود و هدف آن یادگیری برای ایجاد نقشه صحیح جانمایی از فضای ورودی یا همان کلاس‌بندی ورودی است و برخلاف روش‌های یادگیری قبلی از روش یادگیری رقابتی[3] بدون ناظر[4] استفاده می‌کند.

یادگیری بدون ناظر: به یادگیری الگوهای ورودی بدون تامین مقادیر خروجی ویژه و صحیح گفته می‌شود.

یادگیری رقابتی: هر نرون خروجی برای جذب نمونه ورودی به سمت خود یا دیگر نرون‌ها رقابت می‌کند.

از این روش برای دسته‌بندی داده‌ها استفاده می‌شود. شکل زیر مثالی ساده برای نمایش دسته‌بندی داده هاست که داده‌های شکل را در شکل 8-21 به دو دسته مجزا کرده است.

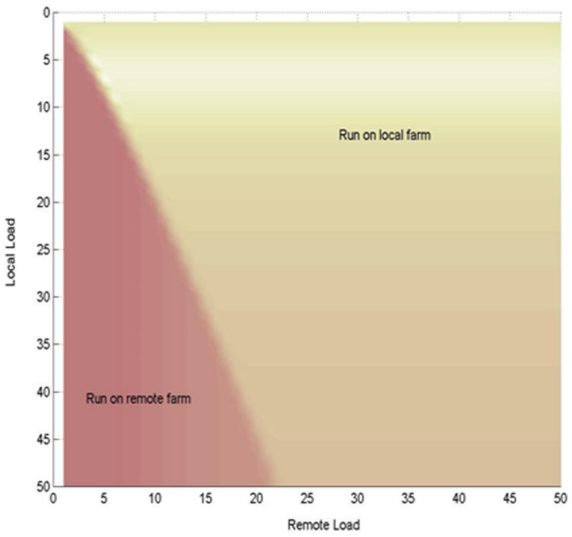
---

[1] Fully connected

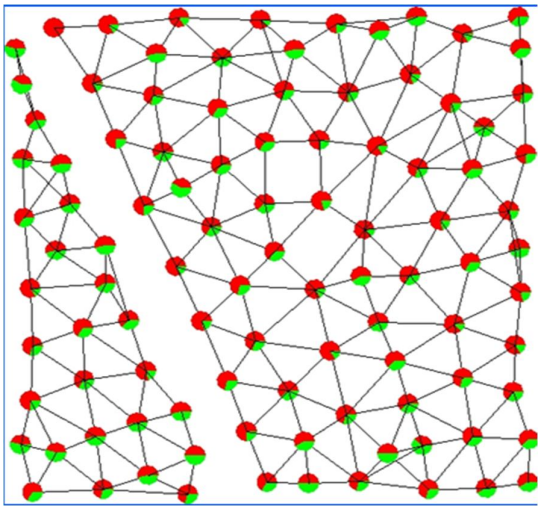
[2] Unsupervised learning network

[3] Competitive

[4] Unsupervised learning



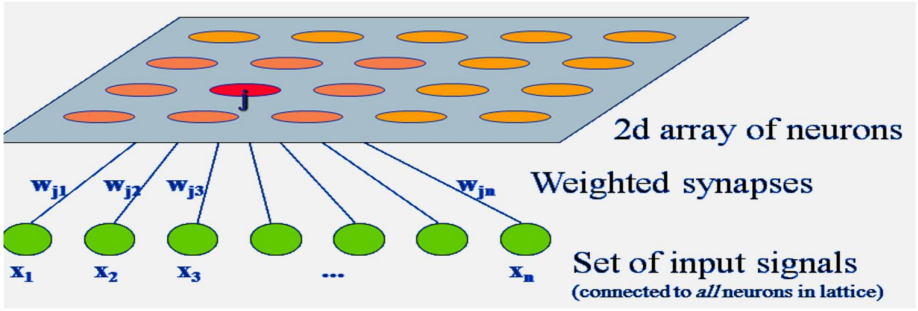
شکل (8-20) : تصویر ورودی برای شبکه SOM



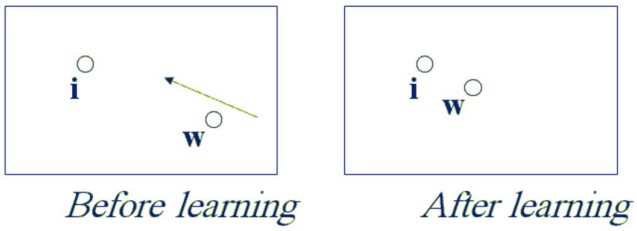
شکل (8-21) : خروجی SOM برای تصویر شکل قبل

1-5-8: معماری شبکه SOM

این شبکه لاتیس[5] نرون‌هاست که مجموعه‌ای از ورودی‌ها را می‌پذیرد و به آن پاسخ می‌دهد و با مقایسه پاسخ‌ها نرونی به عنوان نرون برنده از لاتیس انتخاب می‌شود. نرون منتخب همراه با نرون‌های همسایه‌اش فعال می‌شود. فرایند تطبیقی[6] (آموزش) وزن‌ها را برای شباهت بیشتر به ورودی تغییر می‌دهد(شکل).



شکل (8-22) بردار x با نرون j کمترین فاصله را دارد پس نرون j برنده است



شکل (8-23) : نزدیک شدن نرون برنده به ورودی

ساختار شبکه شامل قسمت‌های زیر است:

- دو لایه شامل واحدهای
  - ورودی: n واحد (طول بردارهای آموزش)
  - خروجی: تعداد دسته ها
- وزن‌ها که به صورت کامل ورودی‌ها را به خروجی‌ها متصل کرده‌اند (تمام ورودی‌ها به تمام خروجی‌ها متصل‌اند).

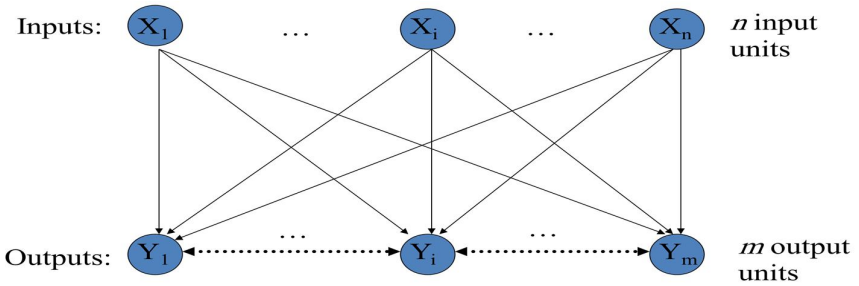
---

[5] Lattice

[6] Adaptive

- اتصالات (عرضی[7]) فرا لایه[8](همسایگی بین نرون‌های خروجی)

معماری شبکه SOM در شکل 8-24 مشاهده می‌شود.

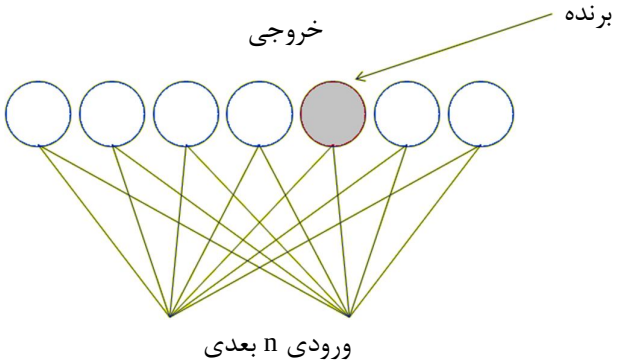


شکل (8-24) : ساختار شبکه SOM

8-5-1-1: نرون برنده

به نرونی گفته می‌شود که به نمونه[9] ورودی نزدیک‌تر باشد.(شکل)

نرون برنده و همسایه‌اش در هر مرحله باید به روز[10] شوند.



---

[7] Lateral

[8] Intralayer

[9] Sample

[10] Update

شکل (8-25) : موقعیت نرون برنده

نرون برنده نرونی است که بهترین تطبیق با ورودی را دارد، به عبارت دیگر معمولا نرونی است که بردار وزنش کوچکترین فاصله از بردار ورودی x یا بیشترین شباهت را دارد و به عبارت ساده به بردار ورودی نزدیک‌ترین باشد.

8-5-2: مقداردهی و بروزرسانی وزن‌ها

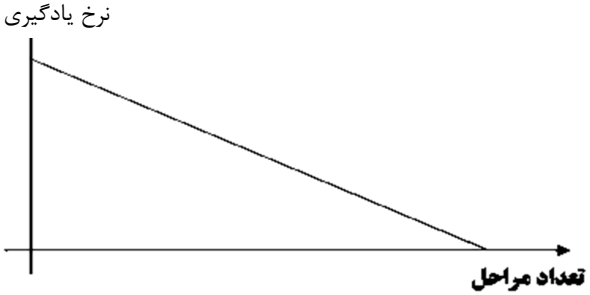
همانند روشهای قبلی به وزن‌ها مقداردهی اولیه تصادفی داده می‌شود.
بر طبق شکل فوق، به ازاء هر واحد خروجی یک بردار وزن بطول n وجود دارد .

برای بروز رسانی وزن هر گره در هر مرحله با اضافه کردن "نرخ یادگیری فعلی x درجه همسایگی به نسبت برنده x تفاوت بین وزن‌های فعلی و بردار ورودی" به "وزن‌های فعلی" بدست می‌آید:

$$w_j(t+1) = w_j(t) + \mu(t)\,\lambda_{\omega(x)}(j,t)\,[x - w_j(t)]$$

8-5-2-1: ضریب یادگیری

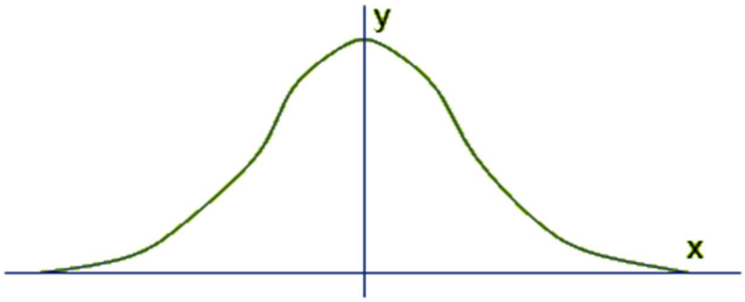
ضریب یادگیری یعنی $\mu(t)$? بر طبق شکل 8-26 با مقادیر بین 0 و 1 محاسبه می‌شود:



شکل (8-26) : مقایسه تعداد مراحل با نرخ یادگیری

8-5-2-2: درجه همسایگی

درجه همسایگی به نسبت برنده یعنی $\lambda_{\omega(x)}(j,t)$ برطبق شکل 8 محاسبه می‌شود:



شکل 8-27 محور x فاصله از نرون برنده را مشخص می‌کند،
محور y درجه همسایگی(حداکثر1) را مشخص می‌کند.

بطوری‌که محور افقی فاصله از نرون برنده را نشان می‌دهد و محور عمودی درجه همسایگی یعنی (j,t) با حداکثر مقدار یک می‌باشد. نرونی که با نرون برنده همسایگی ندارد درجه همسایگی صفر و خود نرون برنده درجه همسایگی یک و دیگر نرون‌ها به نسبت همسایگی مقادیری بین صفر و یک اختیار می‌کنند. در نتیجه نرون برنده بیشترین مقدار بروز شدن و نرون غریبه بدون بروز شده وزن هاست.

8-5-3: میزان شباهت[11] و اندازه‌گیری فاصله[12]

اندازه‌گیری فاصله یکی از معیارهای شباهت است. در این نوع معیار، داده‌ها یک به یک با یکدیگر مقایسه می‌شوند. اندازه فاصله با فرمول‌های مختلفی سنجیده می‌شود که هر کدام بسته به داده‌ها کارایی خاص خود را دارند.

در زیر انواع این نوع اندازه‌گیری مشاهده می‌شود:

1- **فاصله اقلیدسی**[13]: فاصله خط مستقیم بین نقاط داده است درصورتی‌که بر روی گراف چند بعدی باشد.

فاصله اقلیدسی بین دو بردار a و b بصورت زیر محاسبه می‌شود.

$$= (\quad , \quad , \ldots , \quad) , \quad = (\quad , \quad , \ldots , \quad)$$

---

[11] Similarity measure

[12] Distance measure

Euclidian distance [13]

$$, = \overline{\quad\quad\quad} \quad (\quad - \quad)$$

Chebychev -2

این نوع فاصله بین دو بردار x , y به صورت زیر محاسبه می‌شود.

$$h \quad h \quad (\ , \ ) = \max| \quad - \quad |$$

Block (Hamming) -3

$$(\ , \ ) = \sum | \quad - \quad |$$

Minkowski -4

$$(\ , \ ) = \sum | \quad - \quad |$$

اطلاع و انتخاب اندازه فاصله بسیار مهم است. این‌که اندازه فاصله تا چه حدی کم یا زیاد شود به داده‌های ما بستگی دارد.

8-5-4: اندازه مجاورت[14]

این معیار شباهت برخلاف قبلی از دید کلی به داده‌ها نگاه می‌کند و شامل انواع زیر است.

Cosine -1

$$(\ , \ ) = \frac{\sum (\quad , \quad)}{\overline{(\sum \quad)(\sum \quad)}}$$

Correlation -2

$$(\ , \ ) = \frac{\sum \quad (\quad - \bar{}) (\quad - \quad)}{\overline{\sum \quad (\quad - \bar{})} \quad \overline{\sum \quad (\quad - \quad)}}$$

$$= \frac{1}{\quad} \qquad\qquad \bar{} = \frac{1}{\quad}$$

[14] Proximity measures

Chi-Square -3

$$(,) = \frac{ - ( )}{( )} + \frac{ - ( )}{( )}$$

Jensen -4

حاصل این فرمول برای دوبردار ورودی p,q مقداری بین صفر و یک است که اگر دو بردار p,q یکسان باشند حاصل این فرمول صفر می‌شود.

اگر بردارهای p,q هر دو ضرایب مختلفی از بردار ones (درایه‌های مقدار یک) باشد، حاصل فرمول صفر می‌شود به عبارت دیگر:

$$= = \quad (1,) \quad h \qquad h \quad 1, \ 2 \in \mathbb{N}$$

$$(1* \ , \ 2* \ ) = 0$$

$$(,) =$$

$$\frac{1}{2} \quad \{ \log \quad + \quad \log \quad - (\quad + \quad) \log ((\quad + \quad)?2) \} \quad h$$

$$= \qquad , \quad =$$

8-5-5: نویزهای نمونه

در جدول مقایسه انواع معیارهای شباهت موجود و توانایی رفع نویزهای مختلف مشاهده می‌شود.

جدول 8- 3 بررسی انواع معیارهای شباهت

Measures	Properties				
	Time-shift	Amplitude-shift	Time-scale	Amplitude-scale	Phase-delay
Distance measures	×	×	×	×	×
Normalized Euclidian	×	✓	×	✓	×
Chi-Square	×	×	×	×	✓
Jensen	×	×	×	✓	×
Cosine	×	×	×	✓	×
Correlation	×	✓	×	×	×

جابجایی زمانی

h : 1: $\bar{}$ = + , $\bar{}$ =

جابجایی دامنه

h : 2: $\bar{}$ = , $\bar{}$ = +

مقیاس زمانی

: 3: $\bar{}$ = (1 +) , $\bar{}$ =

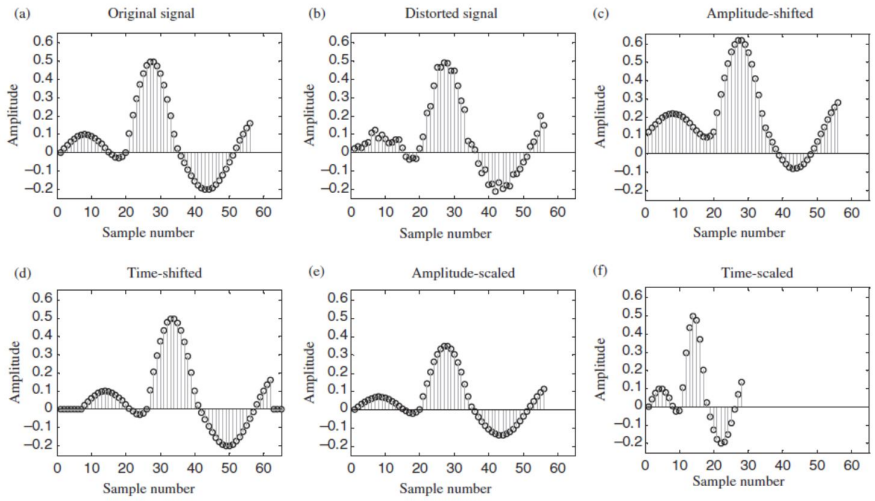
مقیاس دامنه

: 4: $\bar{}$ = $\bar{}$ = (1 +)

تأخیر فاز

h : 5: h _ (, θ)

شکل 8-28 انواع نویزهای مختلف را نشان می دهد.



شکل (8-28) : انواع نویزهای مختلف

8-5-6: ورودی شبکه SOM

ورودی شبکه که به منظور آموزش بکار می‌رود، شامل p بردار متمایز هر کدام بطول n می‌باشد و هر کدام از داده‌های بردارها اعداد حقیقی هستند.

$(x_{1,1}, x_{1,2}, \ldots, x_{1,i}, \ldots, x_{1,n})$
$(x_{2,1}, x_{2,2}, \ldots, x_{2,i}, \ldots, x_{2,n})$
…
$(x_{j,1}, x_{j,2}, \ldots, x_{j,i}, \ldots, x_{j,n})$
…
$(x_{p,1}, x_{p,2}, \ldots, x_{p,i}, \ldots, x_{p,n})$

8-5-7: خروجی شبکه SOM

یک بردار به‌طول m به صورت زیر است:

$(y_1, y_2, \ldots, y_i, \ldots, y_m)$

با در نظر گرفتن n از ورودی و m از خروجی، رابطه بین m,n در سه حالت m>n و m=n و m<n ممکن است. هرکدام از p بردار در داده‌های آموزش در یکی از m گروه[15] یا همان m دسته[16] قرار می‌گیرد.

سوالی که شبکه SOM به آن پاسخ می‌دهد این است که بردار ورودی در کدام دسته قرار می‌گیرد. به عبارت دیگر کدام یک از m دسته، پذیرنده بردار jام ورودی یعنی بردار $(x_{j,1},x_{j,2},...,x_{j,i},...,x_{j,n})$ می‌باشد.

8-5-8: الگوریتم کلی SOM

1- در ابتدا
   1) انتخاب تعداد ورودی‌ها و خروجی ها
   2) انتخاب جانمایی[17] لایه خروجی
   3) مقداردهی اولیه وزن‌ها بطور تصادفی
2- آموزش
   1) آموزش وزن‌های متصل بین ورودی و خروجی.
   2) جانمایی استفاده‌شده در ارتباط با نگاشت فعلی ورودی‌ها به خروجی‌ها برای تشخیص اینکه کدام وزن‌ها باید به‌روز شوند استفاده می‌شود.
   3) اندازه فاصله با استفاده از جانمایی در طول زمان کاهش می‌یابد و همچنین تعداد وزن‌هایی که در هر تکرار باید به‌روز شوند به‌دلیل کاهش مقدار همسایگی کاهش می‌یابد. نرخ یادگیری نیز در طول زمان کاهش می‌یابد.
3- تست

   استفاده از وزن‌های مرحله آموزش

بطور خلاصه الگوریتم SOM به صورت زیر است:

1- مقداردهی اولیه تصادفی به وزن ها

---

[15] Cluster

[16] Category

[17] Topology

2- انتخاب بردار ورودی $x=[x_1, x_2, ..., x_i,..., x_n]$

3- مقایسه x با وزن‌های w_j برای هر نرون j برای تعیین نرون برنده.

4- به‌روز کردن نرون برنده و همسایگانش برای این‌که به x (ورودی) شبیه‌تر شود.

5- تنظیم پارامترها: نرخ یادگیری و تابع همسایگی (بسته به زمان کاهش می‌یابند).

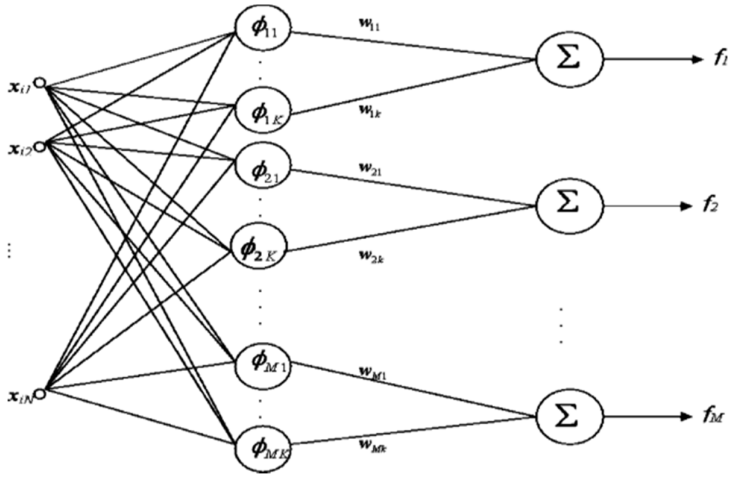
6- برو به 2 تا زمانی که نقشه همگرا شود (تغییر قابل توجهی در وزن‌ها نباشد یا تعداد از پیش تعیین شده مراحل آموزش گذشته باشد).

8-5-8-1: جانمایی[18] لایه خروجی

اغلب مشاهده خروجی بصورت فضایی[19] است مانند چینش یک بعدی و دو بعدی.

8-6: شبکه RBF[20]

به طور معمول شبکه‌های عصبی RBF از سه لایه تشکیل می‌شوند: لایه ورودی، یک لایه پنهان که نرون‌های آن دارای تابع RBF هستند و لایه خروجی.



شکل (8-29) : ساختار کلی شبکه RBF

---

[18] Topology

[19] Spatial

[20] RBF (Radial Basis Function)

شبکه RBF شبیه پرسپترون چندلایه است و یکی از تفاوت‌های موجود با آن پیش‌پردازش ورودی قبل از اعمال در تابع فعال‌ساز است.

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x})$$

مراکز توابع Basis را می‌توان به طور تصادفی از مقادیر ممکن برای ورودی نمونه‌برداری کرد . همچنین می‌توان پس از خوشه‌بندی[21] نمونه‌ها میانگین هر نمونه را به عنوان مرکز تابع انتخاب کرد.

پارامترهای تابع شعاعی[22] به صورت زیر است:

- مرکز[23] $\mathbf{x}_i$
- اندازه فاصله[24] $r = ||\mathbf{x} - \mathbf{x}_i||$
- تشکیل[25] ϕ

روش‌های معمول محاسبه تابع شعاعی در زیر آمده است:

Gaussian

$$\phi(\ ) = \qquad\qquad > 0$$

Hardy Multiquadratic

$$\phi(\ ) = \qquad + \ ? \qquad\qquad > 0$$

---

[21] Clustering

[22] Radial function

[23] Center

[24] Distance measure

Shape [25]

Inverse Multiquadratic

$$\emptyset(\ ) = \ ? \ \overline{\quad + \quad} \qquad > 0$$

اساس کار بدین صورت است که به جای آنکه برای محاسبه نتیجه شبکه از فرمول $f(x)=\sum_{i=1}^{M} w_i x$ استفاده کنیم، از فرمول $f(x)=\sum_{i=1}^{M} w_i \phi_i(x)$ استفاده می‌کنیم.
همانطور که می‌دانیم ϕ_i تابع شعاعی، x داده‌های مسئله و w_i بردار وزن شبکه می‌باشند. به عبارت دیگر، به جای آن‌که خود داده x را به صورت مستقیم به شبکه بدهیم، با اعمال تابع شعاعی ویژگی خاصی را از داده استخراج کرده ($\phi_i(x)$) و آن ویژگی را به شبکه می‌دهیم.

8-6-1: کاربردهای شبکه RBF

از شبکه عصبی RBF برای کلاس‌بندی[26] و همچنین درون‌یابی[27] استفاده می‌شود. از شبکه عصبیRBF می‌توان جهت پیش بینی سری‌های زمانی (درون‌یابی) نیز استفاده کرد.

برخی از کاربردهای شبکه RBF عبارتند از:

1- استخراج ویژگی[28](با توجه به ویژگی که مدنظر ماست، نوع تابع ϕ را انتخاب نماییم.)
2- درون‌یابی توابع (البته در این مورد به صورت گسترده از RBFها استفاده نمی‌شود)
3- پیش‌بینی سری‌های زمانی[29]، از شبکه عصبیRBF نیز می‌توان جهت پیش‌بینی سری‌های زمانی استفاده کرد. (در این مورد از شبکه‌های MLP هم استفاده می‌شود اما RBFها نتیجه بهتری دارند)
4- حذف نویز[30]
5- فشرده‌سازی و بازیابی داده‌های فشرده شده

---

[1] lustering
[27] Interpolation
[28] Feature extraction
[29] Time series
[30] Noise

8-6-2: انواع توابع شعاعی ϕ

با توجه به ماهیت مسئله و کاربرد شبکه تعیین می‌شود که از آن‌جمله می‌توان موارد زیر را نام برد:

توزیع گوسین : $\phi(x) = e^{-\frac{x^2}{2\sigma^2}}$

سری فوریه : $\phi_k(x) = e^{(j2k\omega_0 x)}$

Hardy Multiquadratic : $\phi(x) = \frac{\sqrt{x^2 + c^2}}{c}$

8-6-3: الگوریتم شبکه

این شبکه دو لایه است و عملکرد آن بدین صورت است که ابتدا ورودی‌ها به شبکه داده شده و تابع یا توابع شعاعی بر روی آنها عمل می‌کنند، سپس با ضرب هریک از آنها در بردارهای وزن، خروجی شبکه را محاسبه می‌کنیم. نوع توابع شعاعی که در مسئله استفاده می‌شود با توجه به داده‌ها و کاربرد مسئله و تعداد توابع شعاعی با سعی و خطا تعیین می‌شود. در لایه اول آموزش به صورت بدون سرپرست[31] انجام می‌گیرد. این مرحله برای تثبیت مراکز تابع‌ها به وسیله فقط آگاهی از داده‌های ورودی است. در لایه دوم آموزش به صورت با سرپرست[32] انجام می‌گیرد. این برای محاسبه وزن‌های شبکه در لایه دوم به‌کار برده می‌شود.

8-7: شبکه عصبی احتمالی PNN[33]

یک نوع شبکه عصبی است که احتمال عضویت یک نمونه به یک کلاس را تخمین می زند. در حقیقت نوعی شبکه است که تابع چگالی احتمال[34] را تخمین می‌زند . از این رو به آن PNN می گویند. PNN زیر مجموعه شبکه RBF و یکی از انواع RBF است.

---

[31] Unsupervised

[32] Supervised

[33] Probabilistic neural network

این نوع خاص از شبکه عصبی مصنوعی، یک دسته راه حل کلی برای مسائل دسته‌بندی الگوها با استفاده از احتمالات بدست آمده براساس تئوری تصمیم گیری بیز[35] فراهم می‌کند.
شبکه عصبی احتمالی در اصل یک دسته‌کننده[36] است.

- نگاشت هر الگوی[37] ورودی به تعدادی از دسته‌ها.
- در حالت عمومی‌تر می‌تواند به یک تقریب‌زننده[38] تابع تبدیل شود.

یک PNN پیاده سازی یک الگوریتم آماری به‌نام kernel discriminant analysis است که عملیات آن در یک شبکه چندلایه feedforward با چهار لایه سازمان داده شده است:

- لایه ورودی
- لایه الگو
- لایه تجمیع
- لایه خروجی

1-7-8: مزایا و معایب PNN

مزایای PNN:

1- فرآیند آموزش سریع ، سریع‌تر از ساختار بازگشتی است.
2- ساختار موازی ذاتی
3- تضمین به همگراشدن به یک دسته‌کننده بهینه با وجود افزایش اندازه مجموعه آموزش نمونه (مشکل مینیمم محلی[39] رخ نمی‌دهد)
4- نمونه‌های آموزشی می‌توانند بدون نیاز به آموزش مجدد گسترده کم یا زیاد شوند.

---

[34] Probability density function
[35] Bayes decision rule
[36] Classifier
[37] Pattern
[38] Approximator
[39] Local minima

معایب PNN :

1- عمومی[40] نبودن به اندازه ساختار بازگشتی
2- نیاز به حافظه بزرگ
3- اجرای کند شبکه
4- نیاز به یک مجموعه آموزش نمونه حتی بیش از دیگر انواع شبکه‌های عصبی

2-7-8: نظریه دسته‌بندی

اگر تابع چگالی احتمال(pdf[41]) از هر جمعیت (داده) معلوم باشد، آنگاه مجهول x متعلق به کلاس i است اگر:

$$( ) > ( ), \quad \neq$$

f_k یک pdf برای کلاس k می‌باشد.

دیگر پارامترهای ممکن:

- احتمال قبلی[42](h)

احتمال یک نمونه نامعلوم که از جمعیت (کلاس) خاصی باشد.

- هزینه دسته‌بندی اشتباه(c)

هزینه دسته‌بندی نادرست یک مجهول

- تصمیم دسته‌بندی به صورت زیر است:

$h_{i\ i\ i}( ) > h_{j\ j\ j}(X)$, all $j \neq i$

(قاعده تصمیم بهینه بیز[43])

1-2-7-8: تخمین PDF

تخمین PDF با استفاده از نمونه‌های جمعیت (مجموعه آموزش) صورت می‌گیرد بطوری‌که برای یک نمونه ورودی، فاصله گاوسی با هرکدام از دسته‌های موجود را اندازه می‌گیرد. نمونه ورودی با هر دسته که فاصله کمتری داشته باشد در آن دسته قرار می‌گیرد.

---

[40] General

[41] Probability density function

[42] Prior probability

[43] Bayes optimal decision rule

اگر ورودی یک نمونه باشد از فرمول (1) و اگر ورودی n نمونه باشد از فرمول (2) استفاده می‌شود. تفاوت فرمول 1 و 2 در میانگین گرفتن از مقادیر ورودی است.

(1) $\frac{1}{\sigma}\varphi\left(\frac{x-x_i}{\sigma}\right)$

(2) $\frac{1}{n\sigma}\sum_{i=1}^{n}\varphi\left(\frac{x-x_i}{\sigma}\right)$

x = مجهول(ورودی)

x_k = نمونه k ام

φ = تابع وزن دار(تابع گاوسین)

σ = پارامتر هموارکننده

که به این فرمول Parzen's pdf estimator گفته می‌شود.

حاصل فرمول 1و2 مقادیر بزرگ برای فاصله‌های کوچک بین نمونه‌های مجهول و نمونه‌های آموزشی است و با افزایش فاصله حاصل به سرعت به صفر میل می‌کند. دلیل استفاده از تابع گاوسین محاسبه راحت و عدم ارتباط با هر فرضی درباره توزیع نرمال است.

Pdf تخمین زده شده به صورت زیر است:

$$( ) = \frac{1}{\quad} \qquad \underline{(\quad)}$$

PNN زمانی باید استفاده شود که

1- یک دسته‌بندی کننده[44] با یک زمان آموزش کوتاه مطلوب باشد.

2- سرعت اجرای کند و نیاز حافظه زیاد قابل تحمل باشد.

[44] Classifier

مراجع

1- روشی جدید برای پردازش اطلاعات در سیتسمهای شنود راداری. گودرز سعادتی مقدم ، علی ناصری . دانشگاه امام حسین(ع)

2- به کار گیری مدلهای ترکیبی میانگین متحرک خودرگرسیون انباشته فازی احتمالی به منظور پیش بینی نرخ ارز. مهدی خاشعی، فریماه مخاطب رفیعی و مهدی بیجاری . دانشکده مهندسی صنایع و سیستمها، دانشگاه صنعتی اصفهان

[1] Mehotra, K., Mohan, C. K., & Ranka, S. (1997). Elements of Artificial Neural Networks. MIT Press

[1] Fausett, L. (1994). Fundamentals of Neural Networks. Prentice Hall.

[2] Bibliography of SOM papers
   i. http://citeseer.ist.psu.edu/104693.html
   ii. http://www.cis.hut.fi/research/som-bibl/

[3] Java applet & tutorial information
   i. http://davis.wpi.edu/~matt/courses/soms/

[4] WEBSOM - Self-Organizing Maps for Internet Exploration
   i. http://websom.hut.fi/websom/

[5] H. Hassanpour and A.Darvishi. A new similarity measure for signal processing applictions. Department of Computer and Electrical Engineering Noushirvani Institute of Technology, University of Mazandaran
[6] Vincent Cheung ,Kevin Cannons . An Introduction to Probabilistic Neural Networks .Signal & Data Compression Laboratory, Electrical & Computer Engineering ,University of Manitoba ,Winnipeg, Manitoba, Canada