

بنام خدا

پیش بینی عددی ماکسیمم دمای فرودگاه
شهید بهشتی اصفهان با استفاده از شبکه
های عصبی مصنوعی

استاد راهنما:
مهندس پیام سنایی

دانشجو:
عباس نیاز عزیزاده مقدم

بنام خدا

مقدمه

از عوامل طبیعی که ارتباط نزدیکی با انسان دارد وضعیت جوی است و از آنجا که نقش به سزایی در زندگی انسان دارد، همیشه بر آن بوده ایم که از آینده آن مطلع باشیم و از آنجا که تغییرات آب و هوا به عوامل زیادی وابسته است پیش‌بینی آن کاری بسیار مشکل و پیچیده است و نیاز به دانش بالا در علوم فیزیک دارد. از سالها پیش دانشمندان به دنبال کشف روشهایی بودند که بتوانند درصد صحت پیش‌بینی را بالا ببرند و همانگونه که امروزه می‌بینیم دقت پیش‌بینی آب و هوا به شدت بالا رفته و در اکثر مواقع پیش‌بینی به واقعیت تبدیل می‌شود ولی همچنان بشر به دنبال کشف راه‌هایی برای از بین بردن خطاست.

با گسترش شبکه‌های عصبی مصنوعی توجه بسیاری از دانشمندان و گروه‌های پژوهشی به آن جلب شد تا شاید بتوانند از این علم جدید استفاده کرده و میزان خطای پیش‌بینی را کاهش دهند. تحقیقات زیادی در این زمینه صورت گرفت و نتایج مطلوبی حاصل شد.

در این تحقیق از نوع خاصی از شبکه‌های عصبی بنام شبکه‌های عصبی مصنوعی تاخیر زمانی (TDNN)¹ برای پیش‌بینی ماکسیمم دمای هوای فرودگاه شهید بهشتی اصفهان استفاده شده است که در ادامه به تفصیل در مورد آن صحبت خواهد شد.

در ابتدا جا دارد از استاد عزیزم جناب آقای مهندس پیام سنایی صمیمانه تشکر و سپاسگزاری نمایم که در این تحقیق بنده را در تمام لحظات حمایت کردند و برای رسیدن به نتایج مطلوب از راهنمایی‌های ایشان استفاده نمودم. همچنین از مدیریت محترم اداره کل هواشناسی استان اصفهان کمال تشکر را دارم که در راه رسیدن به نتایج این تحقیق تمام نیازهای آماری و اطلاعات پیش‌بینی را در اختیار بنده قرار دادند.

و در آخر از تمامی عزیزانی که در این راه بنده را حمایت کردند به خصوص خانواده عزیزم کمال تشکر را دارم و صمیمانه سپاسگزاری می‌کنم.

با تشکر
عباس نیاز علیزاده

1) Time Delay Neural Network

1-1 چکیده :

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 ، یادگیری با تنظیماتی در اتصالات سیناپسی که بین اعصاب قرار دارد همراه است . این روش ANN ها هم می باشد.

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 ی 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 McCulloch Warren و یک منطق دان به نام Walter Pitts ساخته شد . اما محدودیتهای تکنولوژی در آن زمان اجازه کار بیشتر را به آنها نداد. 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 و تحلیل به آن داده شده به حساب آید. از این متخصص می توان برای بر آورد وضعیت های دخواه جدید و جواب سؤال های " چه می شد اگر " استفاده کرد.

شبکه های عصبی نسبت به کامپیوترهای معمولی مسیر متفاوتی را برای حل مسئله طی می کنند. کامپیوترهای معمولی یک مسیر الگوریتمی 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 خصوصی که کامپیوتر نیاز به طی کردن دارد، شناخته شده باشند. کامپیوتر قادر به حل مسئله نیست. این حقیقت، قابلیت حل مسئله کامپیوترهای معمولی را به مسائلی محدود می کند که ما قادر به درک آنها هستیم و می دانیم چگونه حل می شوند. اما اگر کامپیوترها می توانستند کارهایی 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 اند 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 نادرست کار کند. امتیاز شبکه عصبی این است که خودش کشف می کند که چگونه مسئله را حل کند، عملکرد آن غیر قابل پیشگویی است.

از طرف دیگر، کامپیوترهای 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 پیشگویی هستند و اگر چیزی به خطا انجام

شود به یک اشتباه سخت افزاری یا نرم افزاری بر می گردد.

شبکه های عصبی و کامپیوترهای معمولی با هم در حال رقابت نیستند بلکه کامل کننده یکدیگرند. وظایفی وجود دارد که بیشتر مناسب روش های الگوریتمی 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.

2-1 شبکه عصبی چیست؟

شبکه های عصبی را می توان با اغماض زیاد، مدل های الکترونیکی از ساختار عصبی مغز انسان نامید. مکانیسم فراگیری و آموزش مغز اساساً بر تجربه استوار است. مدل های الکترونیکی شبکه های عصبی طبیعی نیز بر اساس همین الگو بنا شده اند و روش برخورد چنین مدل هایی با مسائل، با روش های محاسباتی که به طور معمول توسط سیستم های کامپیوتری در پیش گرفته شده اند، تفاوت دارد. می دانیم که حتی ساده ترین مغزهای جانوری هم قادر به حل مسائلی هستند که اگر نگوییم که کامپیوترهای امروزی از حل آنها عاجز هستند، حداقل در حل آنها دچار مشکل می شوند. به عنوان مثال، مسائل مختلف شناسایی الگو، نمونه ای از مواردی هستند که روش های معمول محاسباتی برای حل آنها به نتیجه مطلوب نمی رسند. درحالی که مغز ساده ترین جانوران به راحتی از عهده چنین مسائلی بر می آید. تصور عموم کارشناسان IT بر آن است که مدل های جدید محاسباتی که بر اساس شبکه های عصبی بنا می شوند، جهش بعدی صنعت IT را شکل می دهند. تحقیقات در این زمینه نشان داده است که مغز، اطلاعات را همانند الگوها (pattern) ذخیره می کند. فرآیند ذخیره سازی اطلاعات به صورت الگو و تجزیه و تحلیل آن الگو، اساس روش نوین محاسباتی را تشکیل می دهند. این حوزه از دانش محاسباتی (computation) به هیچ وجه از روش های برنامه نویسی سنتی استفاده نمی کند و به جای آن از شبکه های بزرگی که به صورت موازی آرایش شده اند و تعلیم یافته اند، بهره می جوید.

قبل از آشنا شدن با شبکه عصبی و نحوه یادگیری آن لازم است ابتدا تا حدودی با ساختار مغز انسان و نحوه

یادگیری آن آشنا شویم لذا بطور مختصر ساختار مغز انسان را بررسی می‌کنیم.

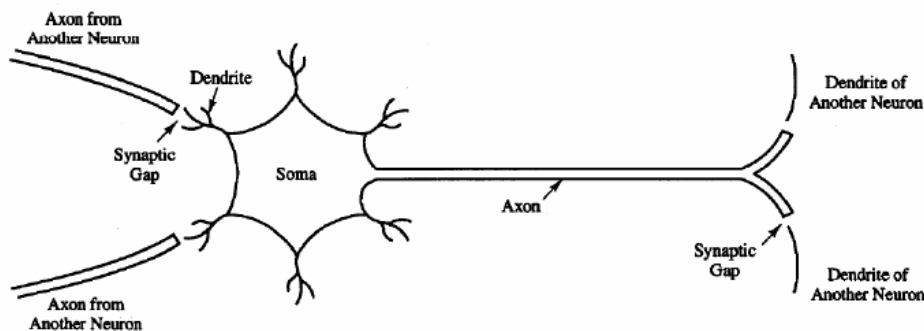
1-2-1 بررسی سلولهای مغزی افراد

مغز بشر متشکل از میلیونها نرون عصبی منحصر بفرد است و این رشته‌های عصبی به اشکال و اندازه‌های مختلف تغییر می‌کنند. هر نرون طبیعی از سه قسمت اصلی تشکیل شده است :

(Soma) بدنه سلول

(Dendrite) دندریت

(Axon) اکسون



شکل ۱ - ساختار نرون طبیعی انسان

دندریته‌ها به عنوان به عنوان مناطق دریافت سیگنالهای الکتریکی، شبکه‌هایی تشکیل یافته از فیبرهای سلولی هستند که دارای سطح نامنظم و شاخه‌های انشعابی بیشمار می‌باشند. دندریته‌ها سیگنالهای الکتریکی را به هسته سلول منتقل می‌کنند. بدنه سلول انرژی لازم را برای فعالیت نرون فراهم کرده و بر روی سیگنالهای دریافتی عمل میکند در اینجا برای حالت برانگیزش باید آشفته‌گی از یک حدی تجاوز کند که آستانه یا سرحد نامیده می‌شود و پس از برانگیزش، نرون‌ها پیامهایی برای نرون‌های دیگر می‌فرستند و خود را برای دریافت و پاسخ به اطلاعات عمومی آماده می‌کنند.

اکسون برخلاف دندریته‌ها از سطحی هموارتر و تعداد شاخه‌های کمتری برخوردار است. اکسون طول بیشتری دارد و سیگنالهای الکتروشیمیایی دریافتی از هسته سلول را به نرونهای دیگر منتقل می‌کند. محل تلاقی یک اکسون از یک سلول به دندریته‌های سلولهای دیگر را سیناپس می‌گویند. توسط سیناپسها ارتباطات مابین نرونها برقرار می‌شود. به فضای مابین اکسون و دندریته‌ها، فضای سیناپسی گویند.

در حقیقت دندریته‌ها به عنوان ورودی نرون و اکسون به عنوان خروجی و فضای سیناپسی محل اتصال این دو

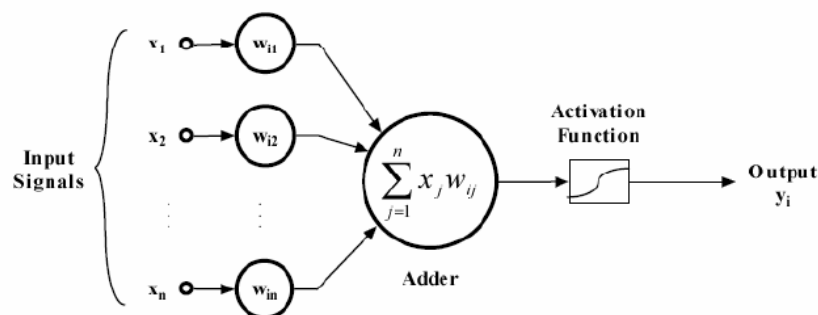
میباشد. زمانیکه سیگنال عصبی از اکسون به نرونها و یا عناصر دیگر بدن مثل ماهیچه ها میرسد ، باعث تحریک آنها میشود.

نرونها از هر یک از اتصالات ورودی خود یک ولتاژ کم دریافت میکند (توسط سیگنال عصبی ورودی) و آنها را جمع میزنند. اگر این حاصل جمع به مقدار آستانه رسید اصطلاحاً نرون آتش میکند و روی اکسون خود یک ولتاژ خروجی ارسال میکند که این ولتاژ به دندریتهایی که به این اکسون متصلند رسیده و باعث یکسری فعل و انفعالات شیمیایی در اتصالات سیناپسی میشود و میتواند باعث آتش کردن نرونهای دیگر شود. تمام فعالیتهای مغزی انسان توسط همین آتش کردنها انجام میشود.

حافظه کوتاه مدت انسان جرقه های لحظه ای الکتریکی میباشند و حافظه بلند مدت به صورت تغییرات الکتروشیمیایی در اتصالات سیناپسی ذخیره میشود که عمدتاً منجر به تغییر یونها میشود.

2-2-1 شبکه های عصبی مصنوعی

شبکه های عصبی شبیه به مغز انسان اطلاعاتی را پردازش میکنند. شبکه از تعداد زیادی سلولهای عصبی (Neuronها) تشکیل شده با پردازشی بسیار بزرگ و به هم پیوسته که در حل موازی مسائل ویژه مشغول به کارند.



شکل ۲ ساختار یک نرون مصنوعی

یادگیری شبکه های عصبی از طریق مثالهاست. آنها برای انجام یک کار خاص برنامه ریزی نشده اند. مثالها باید با دقت بسیار بالایی انتخاب شوند والا زمان مفید هدر خواهد رفت و یا حتی ممکن است شبکه به طور ناقص دایر شود و در اینجا راهی برای فهمیدن اینکه سیستم معیوب است یا خیر وجود ندارد مگر اینکه خطایی رخ دهد.

شبکه های عصبی مصنوعی یک ترکیبی از مجموعه نرونهاست و البته نرونهای مصنوعی که بسیار شبیه به نرونهای زیستی کار میکنند. و بدین گونه است که ورودیهای زیادی

با وزنهای مختلف می‌گیرد و یک خروجی که به ورودی وابسته است تولید می‌کند. نرونها زیستی می‌توانند در حال برانگیزش باشند یا نباشند. (وقتی یک نرون برانگیخته می‌شود ضربه علائم خروجی آن مقداری کمتر از 100 هرتز است) شبکه‌های عصبی استفاده وسیعی در شناسایی الگوها دارند زیرا از خودشان قابلیت آن را دارند که بطور عمومی به ورودی‌های غیر منتظره نیز پاسخ دهند. در طول ساخت نرونها می‌آموزند که چگونه الگوهای ویژه گوناگون را تشخیص دهند. اگر الگویی پذیرفته شود در حالی که در طول اجرا ورودی با خروجی مرتبط نباشد، نرون از مجموعه‌ای از الگوهای که سابقاً آموخته خروجی را که شبیه به الگو می‌باشد و کمترین تفاوت را با ورودی دارد انتخاب می‌کند. این روال عموماً فراخوانی می‌شود.

مثال:

وقتی که ورودی نرون 1111 باشد چهار ورودی بر حسب برانگیزش مرتب شده‌اند و وقتی ورودی‌های 0000 را داریم نرون برای برانگیزش مرتب نیست. قاعده عمومی این است که نرونها مایلند برانگیخته شوند وقتی که ورودی‌ها 0111 ، 1011 ، 1101 ، 1110 یا 1111 باشند و در صورتی که ورودی آنها 1000 ، 0001 ، 0010 ، 0100 یا 0000 باشند مایل به برانگیخته شدن نیستند.

شناسایی الگوهای پیچیده سطح بالا می‌تواند به وسیله شبکه‌ای از نرونها انجام شود و بدین ترتیب نام آن را شبکه‌های عصبی مصنوعی گذاشتند. اکنون شبکه‌های عصبی کاربردهای زیادی دارند (در منطق و کلام و شناسایی عکسها) البته شناسایی الگوها می‌تواند به‌طور موفقیت آمیز بر روی کامپیوترهای عمومی انجام شود. این شبکه‌های عمومی که برای شناسایی الگوها استفاده می‌شوند Feed-Forward نامیده می‌شوند زیرا آنها یک بازخورد (Feed-Back) داشتند. آنها به‌طور ساده ورودی‌ها را با خروجی‌ها می‌آمیختند. اما شناسایی الگوها به تدریج کاملتر شد به‌طوری‌که بر روی کامپیوترهای عمومی با سیستم خاص خودشان به‌سختی انجام می‌شد پس برای شناسایی الگوها شبکه‌های Feed-Forward کافی نبودند.

در شبکه‌های عصبی خروجی هر نرون به ورودی نرونها مجاورش متصل شده است. همانطور که قبلاً هم گفته شد شبکه‌های عصبی نمی‌توانند معجزه کنند اما اگر به درستی استفاده شوند نتایج شگفت‌انگیزی خواهند داشت.

3-1 کاربردهای شبکه های عصبی مصنوعی

از عمده ترین کاربردهای شبکه های عصبی مصنوعی میتوان به موارد زیر اشاره کرد

- سیستم آنالیز ریسک
- کنترل هواپیما بدون خلبان
- ردیابی انحراف هواپیما
- شبیه سازی مسیر
- سیستم راهنمایی اتوماتیک اتومبیل
- سیستمهای بازرسی کیفیت
- آنالیز کیفیت جوشکاری
- پیش بینی کیفیت
- آنالیز کیفیت کامپیوتر
- آنالیز عملیاتهایی آسیب
- آنالیز طراحی محصول شیمیایی
- آنالیز نگهداری ماشین
- پیشنهاد پروژه
- مدیریت و برنامه ریزی
- کنترل سیستم فرایند شیمیایی و دینامیکی
- طراحی اعضای مصنوعی
- بهینه سازی زمان پیوند اعضا
- کاهش هزینه بیمارستان
- بهبود کیفیت بیمارستان
- آزمایش اتاق اورژانس
- اکتشاف روغن و گاز
- کنترل مسیر در دستگاههای خودکار ، ربات ، جراثقال
- سیستمهای بصری
- تشخیص صدا
- اختصار سخن
- کلاس بندی صوتی
- آنالیز بازار
- سیستمهای مشاوره ای محاسبه هزینه موجودی
- اختصار اطلاعات و تصاویر
- خدمات اطلاعاتی اتوماتیک
- مترجم لحظه ای زبان
- سیستمهای پردازش وجه مشتری
- سیستمهای تشخیص ترمز کامیون
- زمانبندی وسیله نقلیه
- سیستمهای مسیریابی

- کلاسه بندي نمودارهاي مشتري/بازار
- تشخيص دارو
- بازبيني امضا
- تخمين ريسک وام
- شناسايي طيفي
- ارزيابي سرمايه
- کلاسه بندي انواع سلولها ، ميكروبها و نمونه ها
- پيش بيني فروشهاي آينده
- پيش بيني نيازهاي محصول
- پيش بيني وضعيت بازار
- پيش بيني شاخصهاي اقتصادي
- پيش بيني ملزومات انرژي
- پيش بيني واکنشهاي دارويي
- پيش بيني بازتاب محصولات شيميايي
- **پيش بيني وضعيت هوا**
- پيش بيني محصول
- پيش بيني ريسک محيطي
- پيش بيني جداول داوري
- مدل کردن کنترل فرآيند
- آناليز فعاليت گارانتی
- بازرسي اسناد
- تشخيص هدف
- تشخيص چهره
- انواع جديد سنسورها
- دستگاه کاشف زير دريائي بوسيله امواج صوتي ، رادار
- پيش بيني فرآيندهاي توليد
- ارزيابي بکارگيري يک سياست
- بهينه سازي محصول
- تشخيص ماشين و فرآيند
- مدل کردن کنترل سيستمها
- مدل کردن ساختارهاي شيميايي
- مدل کردن سيستمهاي ديناميکي
- مدل کردن سيگنال تراکم
- مدل کردن قالبسازي پلاستيکي
- مديريت قراردادهاي سهام
- مديريت وجوه بيمه
- مديريت سهام

- بازبینی امضا از چکها
- پیش بینی ارزش نسبه
- مدیریت ریسک رهن
- تشخیص حروف و اعداد
- تشخیص بیماری

در این تحقیق در بحث پیش‌بینی وضعیت هوا شبکه عصبی را برای پیش‌بینی حداکثر دما استفاده خواهیم کرد که در ادامه به جزئیات کار خواهیم پرداخت.

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

شبکه های عصبی ، با قابلیت قابل توجه در استنتاج معانی از داده های پیچیده یا مبهم ، 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 می شوند. 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 دخواه جدید و جواب سؤال های " چه می شد اگر " استفاده کرد.

مزایتهای دیگر آن شامل موارد زیر می شود :

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

2. **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.

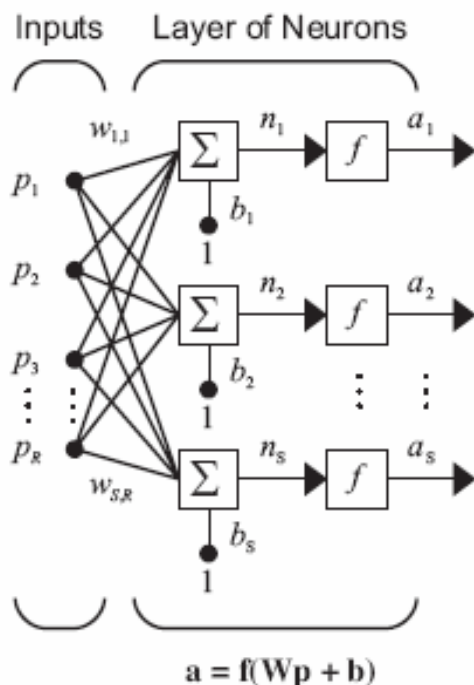
3. **عملکرد بهنگام (Real time) :** محاسبات ANN می تواند بصورت موازی انجام شود، و سخت افزارهای مخصوصی طراحی و ساخته شده است که می تواند از این قابلیت استفاده کند.

4. تحمل 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.

5-1 ساختار شبکه های عصبی مصنوعی

شبکه های عصبی مصنوعی از قسمت های مختلف تشکیل شده اند
که برخی از این قسمت ها عبارتند از:

- 1- ورودی (Input)
- 2- خروجی (Output)
- 3- نتایج مورد انتظار (Target)
- 4- تابع انتقال (Transfer function)
- 5- تعداد نرونهاى هر لایه
- 6- ماتریس وزنها
- 7- تعداد لایه های مخفی



Where

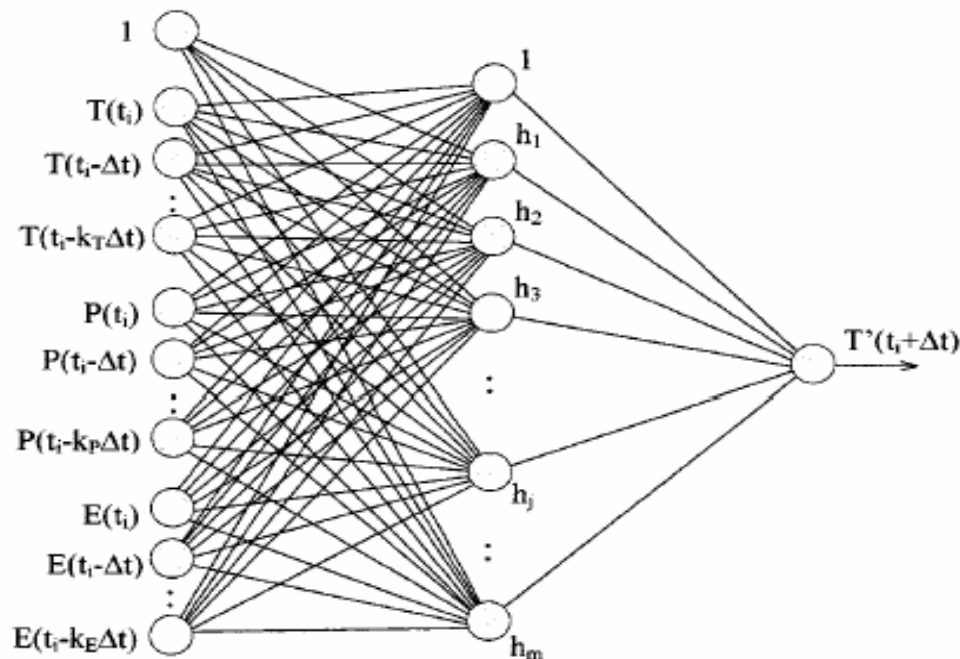
R = number of
elements in
input vector

S = number of
neurons in layer

1-5-1 ورودی (Input)

در هر شبکه عصبی یک لایه به نام لایه ورودی وجود
دارد که معمولاً بصورت یک بردار به شبکه اعمال می شود.
بطور مثال در شکل فوق بردار ورودی، بردار P است که
تعداد عناصر آن R است ($P=[1..R]$). هر پارامتر از
بردار ورودی با تک تک نرونهاى لایه بعدی ارتباط دارد
که معمولاً این ارتباط را توسط خطی از آن پارامتر

ورودی به نرون نشان می‌دهند. که این موضوع در شکل زیر نشان داده شده است.



در اصل هر یک از خطوط ارتباطی یک عدد می‌باشد که در آینده خواهیم گفت که این خطوط همان ماتریس وزن‌ها هستند.

1-5-2 خروجی (Output)

آخرین لایه در شبکه عصبی مصنوعی لایه خروجی است. این لایه شامل نتایجی است که شبکه تولید کرده است. نتایج بدست آمده بستگی به انتخاب تابع فعالیت دارد. و هدف کلی ما از طراحی شبکه بدست آوردن خروجی‌های مطلوب و نزدیک به واقعیت می‌باشد.

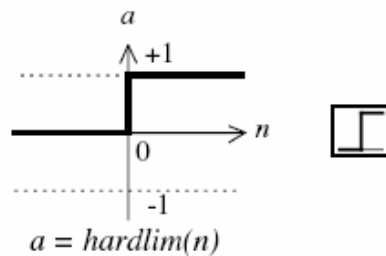
1-5-3 نتایج مورد انتظار (Target)

نتایج مورد انتظار در مرحله آموزش شبکه مورد استفاده قرار می‌گیرند و نتایجی هستند که به ازای ورودیهایی اتفاق افتاده‌اند. از مقایسه خروجی‌های تولید شده توسط شبکه و نتایج مورد انتظار به ازای ورودیهای یکسان، شبکه خود را بروز رسانی می‌کند یا به عبارتی ماتریس وزن‌ها را تولید می‌کند. این بحث در قسمت بعدی تحت عنوان فرآیند یادگیری شبکه های عصبی مصنوعی بررسی می‌شود.

4-5-1 تابع انتقال (Transfer function)

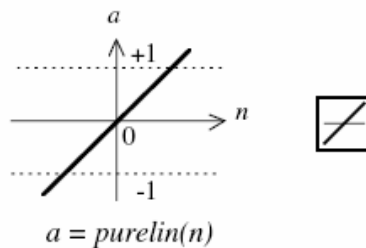
در شبکه‌های عصبی تعداد زیادی تابع انتقالی وجود دارد که لیست کامل آنها در ضمیمه 1 قابل دسترسی است. در این قسمت به توضیح مختصری در مورد برخی از پرکاربردترین این توابع می‌پردازیم.

اولین تابع که مورد بررسی قرار می‌گیرد تابع محدود کننده¹ می‌باشد که در شکل زیر نشان داده شده است.



Hard-Limit Transfer Function

این تابع خروجی را به صفر محدود می‌کند اگر مجموع حاصلضرب ورودی‌ها در ماتریس ضرایب از صفر کمتر باشد و در صورتی که مجموع آنها بزرگتر یا مساوی صفر باشد خروجی یک تولید می‌کند. از این تابع در شبکه‌های پرسپترون (Perceptrons) استفاده می‌شود. تابع بعدی که مورد بحث قرار می‌گیرد تابع خطی² است که در شکل زیر نشان داده شده است.



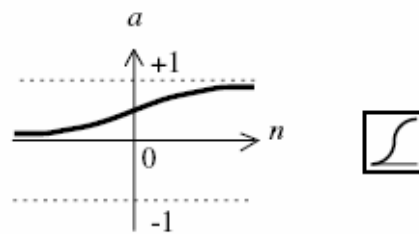
Linear Transfer Function

در این تحقیق از این تابع به عنوان تابع انتقالی استفاده شده است. خروجی این تابع دقیقاً با مقداری که به آن وارد می‌شود یکسان است و همان مقدار ورودی را به خروجی ارسال می‌کند.

$$a = \text{purelin}(n) = \text{purelin}(Wp+b) = Wp+b$$

تابع بعدی که مورد بررسی قرار می‌گیرد تابع سیگموئید³ است که در شکل زیر نشان داده شده است.

1- Hard-Limit Transfer Function
2- Linear Transfer Function
3- Log-Sigmoid Transfer Function



$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function

خروجی این تابع مقادیری بین 0.1 است و یکی از پر کاربرترین توابع انتقالی در شبکه‌های **backpropagation** می‌باشد.

توابع انتقالی دیگر در ضمیمه 1 معرفی شده است. برای آشنایی بیشتر با توابع انتقالی با استفاده از دستور **nnd2n1** در خط فرمان نرم افزار مطلب می‌توان کاربرد هر یک از توابع را با مثال مشاهده نمود.

5-5-1 تعداد نرونهاى هر لایه

یکی از مهمترین مسائل در شبکه‌های عصبی بدست آوردن تعداد نرونهاى هر لایه است چراکه هر چه تعداد نرونها کمتر باشد سرعت پردازش بیشتر می‌شود و رسیدن به جواب سریعتر می‌شود ولی از طرفی ممکن است با کاهش تعداد نرونها دقت محاسبات نیز کاهش یابد لذا باید در انتخاب تعداد نرونها دقت کافی نمود تا بتوان کمترین تعداد نرون که باعث کاهش دقت نمی‌شود را بدست آورد.

هیچ رابطه خاصی برای بدست آوردن تعداد نرونها وجود ندارد و این کار فقط با آزمون و خطا امکان پذیر می‌باشد و نحوه کار به اینصورت است که باید از کمترین تعداد نرون شبکه را آموزش داد و سپس نتایج را بررسی کرد سپس تعداد نرونها را افزایش داد و مجدداً شبکه را آموزش داد اینکار تا رسیدن به نتیجه مطلوب ادامه می‌یابد.

5-5-1 ماتریس وزنها

کل کاری که شبکه عصبی انجام می‌دهد بدست آوردن مقادیر ماتریس وزنهاست. مقادیر این ماتریس باید بگونه‌ای انتخاب شوند که به ازای ورودیهای مختلف خروجی مطلوبی تولید کنند. نحوه بدست آوردن مقادیر ماتریس وزنها (**W**) به ترتیب زیر است:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

در اولین اجرای شبکه ، مقادیر تصادفی برای ماتریس وزنها (W) در نظر گرفته می‌شود. با فرض اینکه مقادیر بردار ورودی را p و مقادیر نتایج مورد انتظار را t و مقدار خطا را $e=t-a$ در نظر بگیریم که a نتایج بدست آمده از شبکه است سه حالت خواهیم داشت:

$a=t$ یا $e=t-a=0$ که در این صورت مقادیر ماتریس وزنها صحیح می‌باشد و نیازی به تغییر ندارند
 $a<t$ یا $e=t-a>0$ که در این حالت مقادیر بردار ورودی با مقادیر ماتریس وزنها جمع می‌شود تا مقادیر ماتریس وزنها به مقادیر ورودی نزدیک شود.
 $a>t$ یا $e=t-a<0$ که در این صورت مقادیر بردار ورودی از مقادیر ماتریس وزنها کسر می‌شود تا مقادیر ماتریس وزنها اختلاف بیشتری نسبت به مقادیر ورودی داشته باشند.

روند اصلاح مقادیر ماتریس وزنها به این صورت می‌باشد (با فرض اینکه Δw تغییرات ماتریس وزنها باشد)

اگر $e=0$ آنگاه $\Delta w=0$

اگر $e>0$ آنگاه $\Delta w=pt$

اگر $e<0$ آنگاه $\Delta w=-pt$

می‌توان سه حالت بالا را در یک رابطه بصورت زیر خلاصه کرد

$$\Delta w = (t-a)pt = ept$$

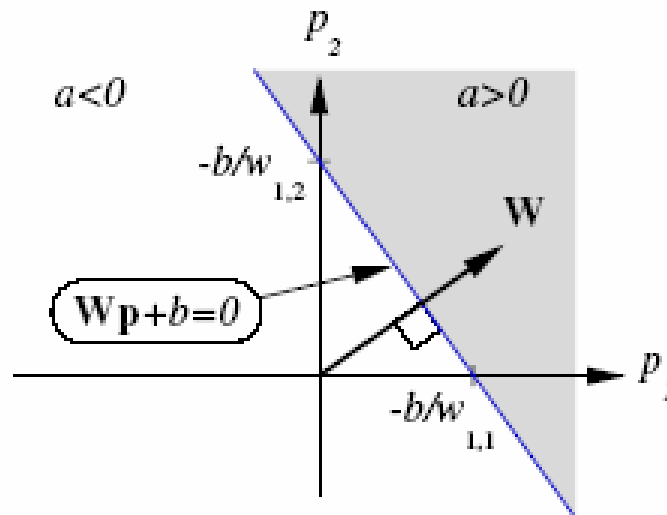
در صورت مطلوب نبودن نتایج تولید شده توسط شبکه مقادیر وزنها باید اصلاح شوند که نحوه اصلاح مقادیر مذکور توسط رابطه زیر صورت می‌گیرد

$$w_{\text{new}} = w_{\text{old}} + ept$$

این روند تا رسیدن به نتیجه مورد انتظار ادامه می‌یابد.

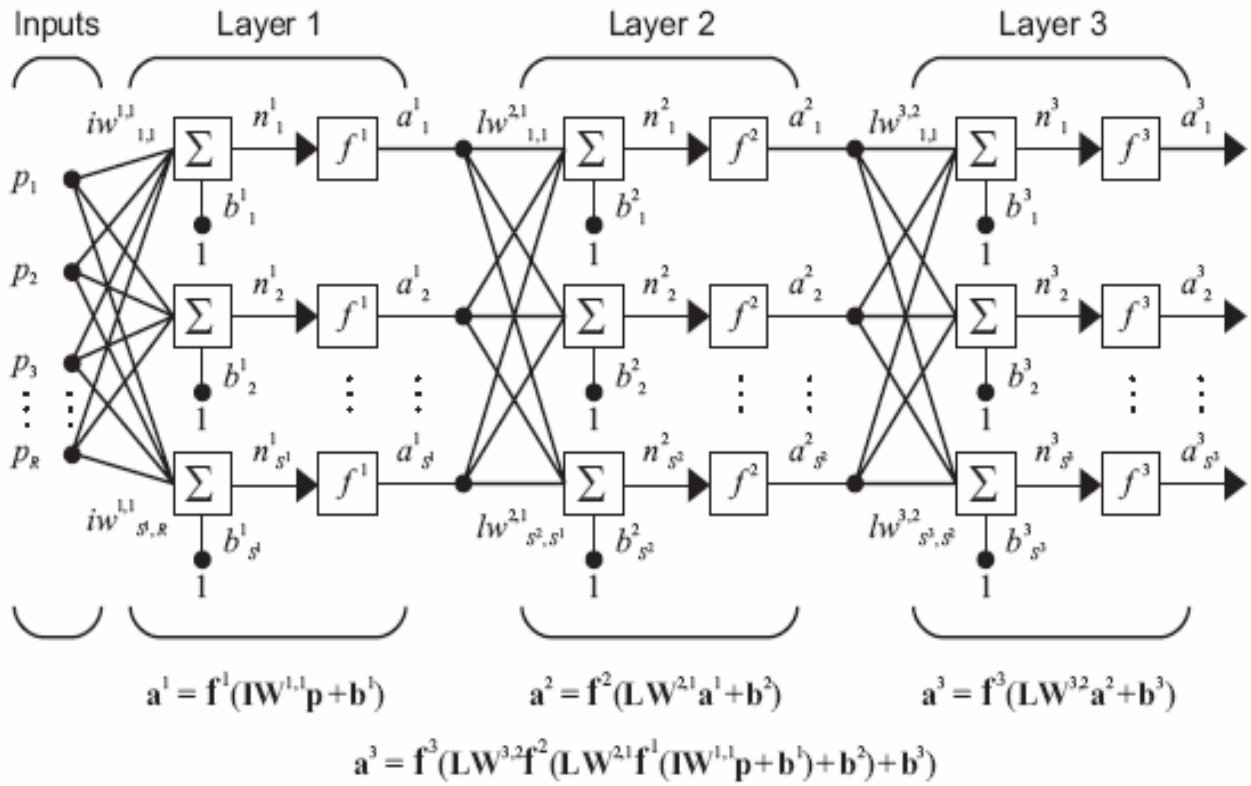
ابعاد ماتریس وزنها به این صورت بدست می‌آید. تعداد سطرهای آن برابر با تعداد نودهای ورودی شبکه و تعداد ستونهای آن برابر با تعداد نرونهای آن لایه است . بطور مثال اگر در شبکه‌ای 10 نود ورودی و 20 نرون

در لایه میانی داشته باشیم آنگاه ابعاد ماتریس وزنها (w) برابر 10×20 خواهد بود. در روند آموزش شبکه مقادیر این ماتریس مدام در حال تغییر می‌باشد تا شبکه به نتیجه مطلوب برسد.

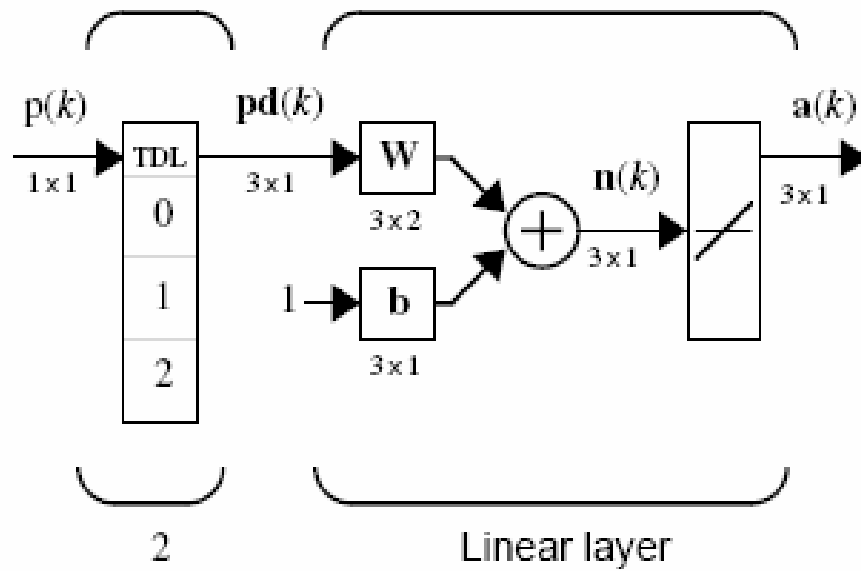


7-5-1 تعداد لایه‌های مخفی

عامل مهم دیگر در طراحی شبکه عصبی مصنوعی تعداد لایه‌های مخفی می‌باشد. حداکثر تعداد لایه‌های مورد نیاز در شبکه عصبی بدون در نظر گرفتن لایه ورودی و خروجی دو لایه می‌باشد و بیشتر از دو لایه مورد نیاز نمی‌باشد. طبق تحقیقات بعمل آمده با دو لایه مخفی هر نوع داده‌ای را می‌توان طبقه‌بندی کرد.

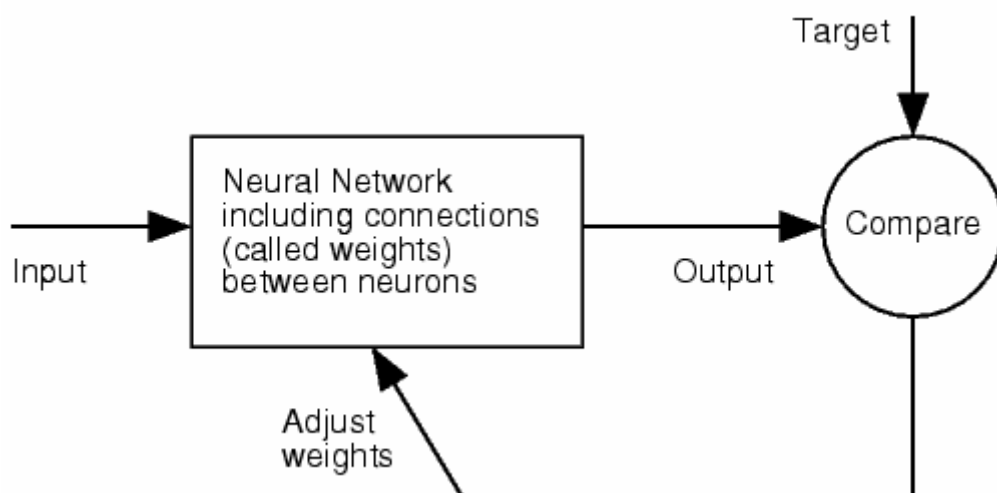


در این تحقیق از یک لایه مخفی استفاده شده است (شکل زیر).



6-1 فرآیند یادگیری شبکه‌های عصبی مصنوعی

همانطور که قبلاً عنوان شد شبکه‌های عصبی مصنوعی قبل از اینکه جهت پیش‌بینی استفاده شوند باید آموزش ببینند. نحوه آموزش به اینصورت است که ابتدا ورودیها و نتایج که به ازای آنها حاصل شده را به عنوان Input و Target به شبکه معرفی می‌کنیم سپس شبکه را به ازای این پارامترها جهت آموزش فعال می‌کنیم تا شبکه خود را سازگار کند. قابل ذکر است که هرچه دوره آموزش بلندتر باشد دقت شبکه افزایش خواهد یافت.

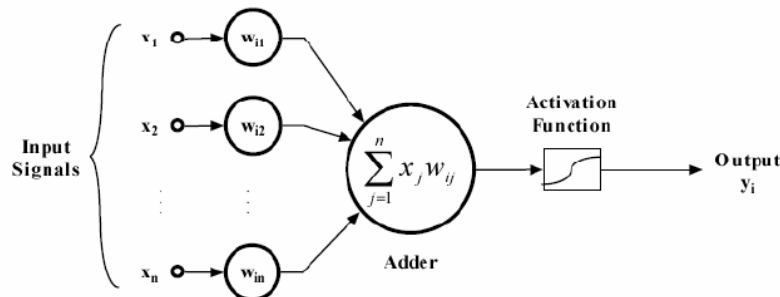


همانطور که در شکل فوق مشخص شده است شبکه در ابتدا ورودیها را دریافت کرده سپس خروجی تولید می‌کند و خروجی تولید شده را با نتایج (Target) مقایسه می‌کند. اگر خروجی بدست آمده با نتایج مورد نظر ما تفاوت زیادی داشت، این‌بار خروجی بدست آمده را به عنوان ورودی به شبکه ارسال می‌کند و این روند تا رسیدن به نتیجه مطلوب ادامه خواهد یافت.

از آنجا که ممکن است هیچگاه خروجی با نتیجه مورد نظر یکسان نشود این احتمال وجود دارد که شبکه در یک حلقه بینهایت واقع شود، برای جلوگیری از این احتمال باید یک درصد خطا برای شبکه در نظر گرفته شود یا اینکه تعداد تکرار حلقه محدود شود.

1-6-1 نحوه یادگیری شبکه

مبنای آموزش در شبکه‌های عصبی مصنوعی بدست آوردن وزن‌ها یا ضرایب است که قبلاً عنوان شد.



شکل ۲ ساختار یک نرون مصنوعی

همانطور که در شکل فوق مشخص است هر ورودی در ماتریسی (w_{ij} ای) ضرب می‌شود و با سایر پارامترهای ورودی جمع می‌شود سپس این حاصل جمع در تابع فعالیت قرار می‌گیرد و اگر به حد آستانه رسید تابع آستانه فعال شده و بسته به نوع تابع فعالیت، خروجی تولید می‌شود.

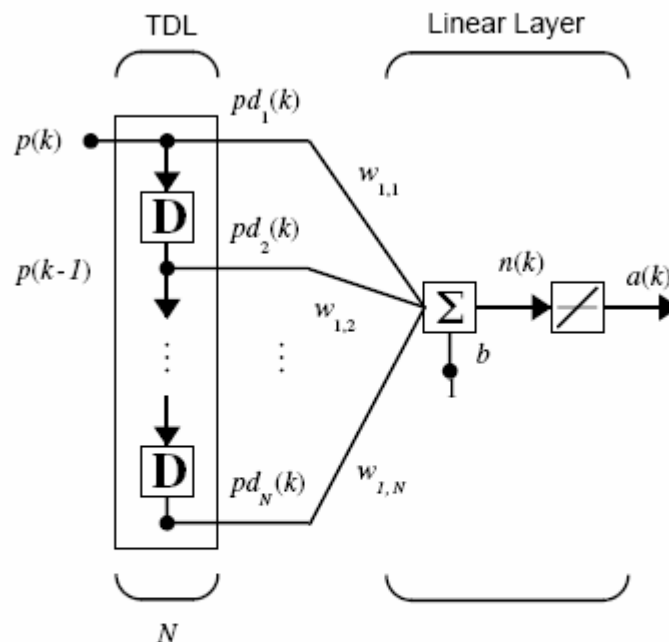
1-7 انواع شبکه‌های عصبی مصنوعی

شبکه‌های مختلفی برای اهداف خاص طراحی شده‌اند که در این قسمت فقط به ذکر نام آنها بسنده می‌کنیم.

competitive
Cascade-forward backprop
Elman backprop
Feed-forward backprop
Time-delay backprop
Generalized regression
Hopfield
Perceptron

...

در این تحقیق از شبکه‌های عصبی تاخیر زمانی (TDNN) استفاده شده است.



The output of the filter is given by

$$a(k) = \text{purelin}(\mathbf{W}\mathbf{p} + b) = \sum_{i=1}^n w_{1,i} a(k-i+1) + b$$

نرم افزارى كه در اين تحقيق استفاده شده است، نرم افزار مطلب ورژن 7.1 است كه مختصرى درباره آن توضيح مى دهيم.

8-1 درباره نرم افزار مطلب:

مسئله يكى از قويترين وسيله هاى محاسباتى در رشته هاى مهندسى كامپيوترها مى باشند، و در انجام محاسبات مهمترين نقش را دارا مى باشند. حال كه بر حسب نيازمان كامپيوتر را به عنوان ماشين حسابى قدرتمند تعريف كرديم نياز به برنامه اى داريم تا تواناييهايش را بكار ببريم.

MATLAB®

The Language of Technical Computing

Version 7.1.0.246 (R14) Service Pack 3

August 02, 2005

License Number: 161051

abbas

adl



Copyright 1984-2005, The MathWorks, Inc.
Protected by U.S. patents. See www.mathworks.com/patents.

مطلب (MATLAB) یکی از زبانهای برنامه نویسی سطح بالا با تمرکز بر روی تکنیکهای محاسباتی است. این نرم افزار محیطی مناسب برای انجام عملیتهای ریاضی، ایجاد محیطهای ویژوال و برنامه نویسی آسان را همزمان فراهم کرده است. در این نرم افزار تلاش بر آن است که مسائل ریاضی و راه حلهای آنها به همان صورتیکه در ریاضیات رایج دانشگاهی وجود دارد ارائه شوند.

MATLAB یک نرم افزار قوی جهت دانشمندان و محققین رشته های مهندسی و ریاضی است که اولین بار نگارشهای آن در دانشگاه نیومکزیکو و استنفورد در سال 1970 در جهت حل مسائل تئوری ماتریسها، جبر خطی و آنالیز عددی بوجود آمد.

این نرم افزار به صورت اختصاصی در موارد زیر کاربرد دارد:

- 1- ریاضیات و محاسبات
- 2- ساخت و پیاده سازی الگوریتم های ریاضی
- 3- جمع آوری داده ها
- 4- مدلسازی، شبیه سازی و تحلیل مدل

5- آنالیز، استخراج مشخصه های آماری، رسم و نمایش دیتا

6- رسم گرافهای مهندسی و علوم (منحنی های دو بعدی و سه بعدی، منحنی های آماری و . . .)

7- تولید نرم افزارهای کاربردی دارای واسط گرافیکی
نرم افزار مطلب دارای سیستمی اندرکنشی بوده که در آن تمامی داده ها به صورت آرایه های بدون تعیین بعد معین و مشخص ذخیره می شوند. این خاصیت این امکان را به شما می دهد که مسائل محاسباتی بسیاری را با استفاده از فرمولهای برداری و ماتریسی برای طیف وسیعی از داده ها بنویسید. این نحوه برنامه نویسی در حقیقت کسری از زمانی است که در یک زبان سطح متوسط غیر دینامیک چون C و FORTRAN صرف می شود.

نام MATLAB از حروف ابتدایی MATRIX LABORATORY آمده است. این نرم افزار در ابتدا به عنوان نرم افزاری جهت انجام سریعتر عملیتهای ماتریس و به عنوان تجمیع دو پروژه LINPACK و ETSPACK طراحی شد. امروزه نرم افزار مطلب از کتابخانه های تابعی BLAS و LAPACK در محاسبات ماتریسی خود استفاده می کند.

طیف متنوعی از کاربران را تحت پوشش قرار داده است. در محیطهای دانشگاهی از مطلب به عنوان ابزاری برای آموزش دوره های مقدماتی تا پیشرفته ریاضیات، علوم مهندسی و علوم پایه استفاده می شود. در صنایع نیز مطلب به عنوان ابزاری برای تحقیقات افزایش تولید و نیز آنالیز ریاضی مسائل درگیر در آن صنایع استفاده می شود.

شرکت MathWorks به همراه نرم افزار MATLAB ، راه حلهای کاربردی معینی در زمینه های علوم نو را با عنوان Toolbox عرضه کرده است. آنچه در این جعبه ابزارهای کاربردی ارائه شده چنان است که کاربران را قادر می سازد ضمن استفاده از تکنولوژی محاسباتی مربوطه به یادگیری و توسعه آن نیز بپردازد، این جعبه ابزارها ، مجموعه ای از توابع مطلب (m-file) را شامل شده که ضمن اضافه شدن به نرم افزار مطلب محیط آن را برای حل مسائل ویژه ای از علوم جدید آماده می کند.

از حوزه های علوم جدیدی که در این جعبه ابزارها به آنها پرداخت شده می توان به جعبه ابزارهای پردازش

تصویر، پردازش سیگنال، سیستم های کنترلی، شبکه های عصبی، منطق فازی، فوریه دو بعدی Wavelett، شبیه سازی و . . . اشاره کرد.

1-8-1 ساختار نرم افزار مطلب:

نرم افزار MATLAB بر 5 رکن اصلی استوار است، این ارکان عبارتند از:

1- میز و محیط کاری نرم افزار:

این مجموعه کاربر را قادر می سازد تا با نرم افزار ارتباط موثری برقرار کند. بدین منظور نرم افزار از واسطه های گرافیکی قدرتمندی بهره می برد که از آن جمله می توان به صفحه کار و پنجره فرمان و پنجره تاریخچه فرامین (Command History)، پنجره ویرایش و رفع عیب برنامه های مطلب، آنالایزر کدهای برنامه نویسی، جستجوگر، help نرم افزار، فضای کاری متغیرهای مطلب و نیز فایلها و مسیرهای جستجو و . . . اشاره کرد.

2- کتابخانه توابع ریاضی:

در این نرم افزار مجموعه وسیعی از الگوریتم های محاسباتی از توابع ابتدایی چون (جمع سینوسی و کسینوسی و ریاضیات مختلط تا توابع پیچیده تر چون معکوس ماتریس، مقادیر ویژه ماتریس، توابع بسل و FFT (تبدیل فوریه سریع) را شامل می شود.

3- زبان برنامه نویسی:

این نرم افزار دارای زبان برنامه نویسی سطح بالای آرایه ای/ماتریسی بافرامین کنترل برنامه، ساخت تابع، پذیرش ساختارهای مختلف داده ای، قابلیت دریافت ورودی/ارسال خروجی و قابلیت برنامه نویسی شی گرا می باشد. برنامه نویسی در این محیط به هر دو صورت نوشتن برنامه ای کوچک

(Small Scale) تا برنامه های بزرگ (Large Scale) امکان پذیر است.

4- خواص گرافیکی نرم افزار:

مطلب قابلیت نمایش بسیار آسان ماتریسها و بردارها، دستکاری و چاپ دیتای مربوطه را دارد. این موارد مشتمل بر نمایش گرافهای دو و سه بعدی، پردازش تصویر، انیمیشن سازی و ارائه تصاویر با فرمت دلخواه می باشد. همچنین در این نرم افزار امکان ساخت واسطهای گرافیکی وجود دارد.

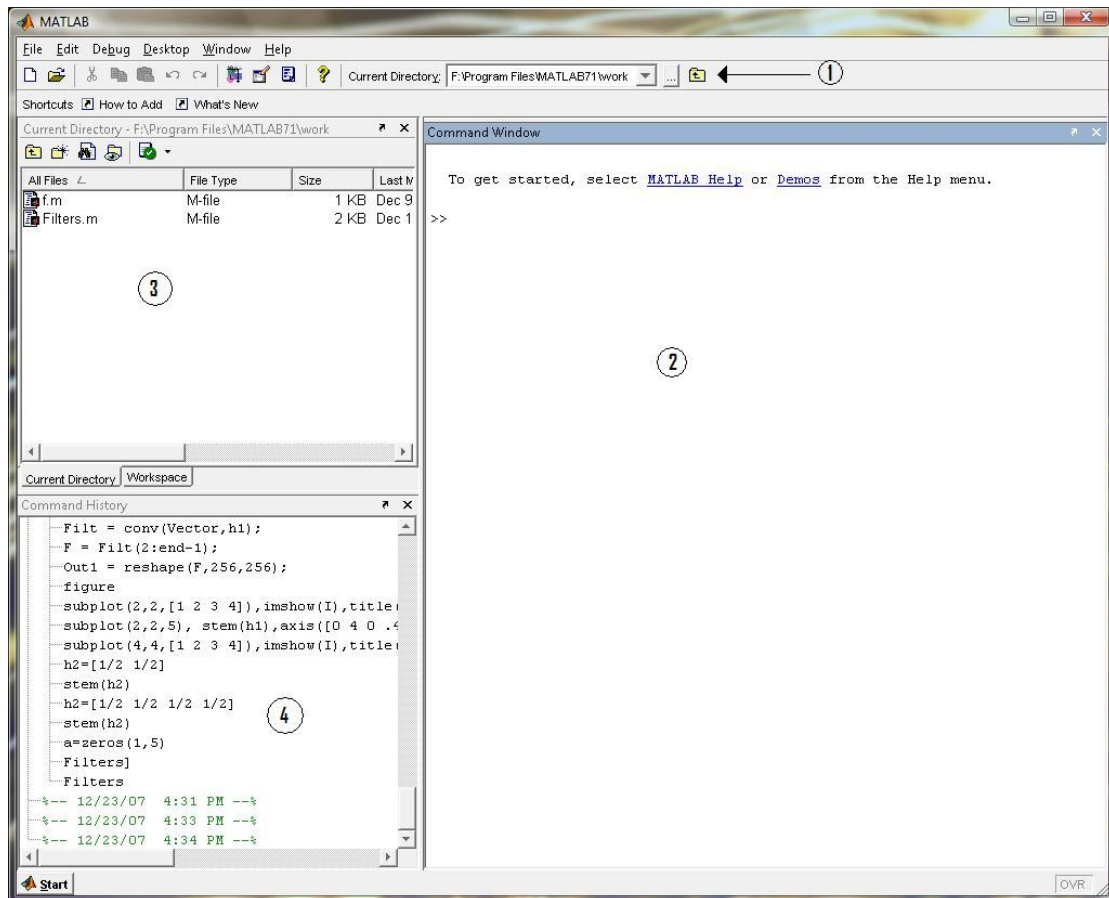
5- واسطهای با محیط بیرونی مطلب/API :

مطلب امکان ارتباط دوطرفه با نرم افزارهای Fortran و C را دارد. این امکان به سه صورت در مطلب قابل استفاده است:

- 1- صداکردن برنامه مطلب در برنامه های C (به صورت dll)
- 2- استفاده از مطلب به عنوان یک موتور محاسباتی
- 3- خواندن و نوشتن mat فایل.

2-8-1 آشنایی با پنجره های مطلب:

هنگامی که اولین بار مطلب را اجرا می کنیم پنجره ای مطابق شکل زیر ظاهر می شود:

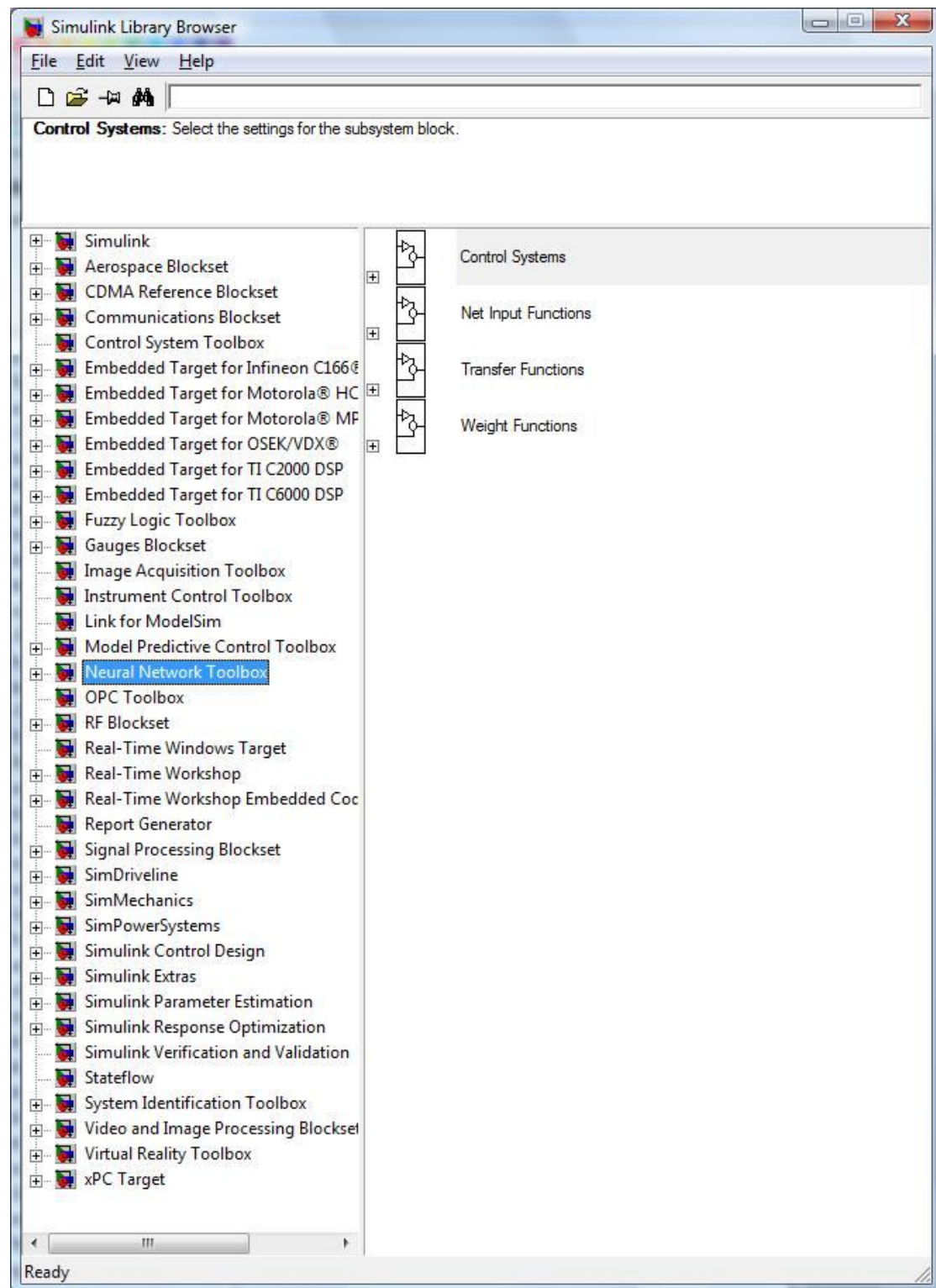


البته می توان بر اساس نیاز خود پنجره های اضافه را بسته و یا پنجره هایی را به آن اضافه کرد.

- 1) نشان دهنده جعبه ابزار مطلب می باشد.
- 2) محیط کار مطلب می باشد: در این قسمت دستورات مطلب را در مقابل علامت `>>` می نویسیم و با زدن کلید `Enter` دستورات را اجرا می کنیم.
- 3) `M-file` ها و سایر فایل های ذخیره شده در این قسمت نمایش داده می شوند
- 4) تاریخچه استفاده از دستورات در این قسمت ذخیره می شود.

مطلب شامل قسمتهای مختلفی است که در بالا به یکی از این محیطهای کاری به نام `command` اشاره شد که در این قسمت می توان دستورات مطلب را در مقابل علامت `>>` نوشت و با زدن کلید `Enter` آنها را اجرا کرد.

از دیگر محیط‌های کاری مطلب میتوان به محیط Simulink اشاره کرد که یکی از پرکاربردترین و پر قدرت ترین ابزارهای کار با استفاده از شبیه سازی است.



همانطور که در شکل فوق که نشان دهنده محیط Simulink مطلب است مشخص است، یکی از ابزارهای این محیط Neural Network Toolbox است که برای شبیه سازی شبکه های عصبی مصنوعی استفاده می شود.

از دیگر ابزارهای این محیط می توان به محیطهای شبیه سازی منطق فازی ، پردازش تصویر و ... اشاره کرد.

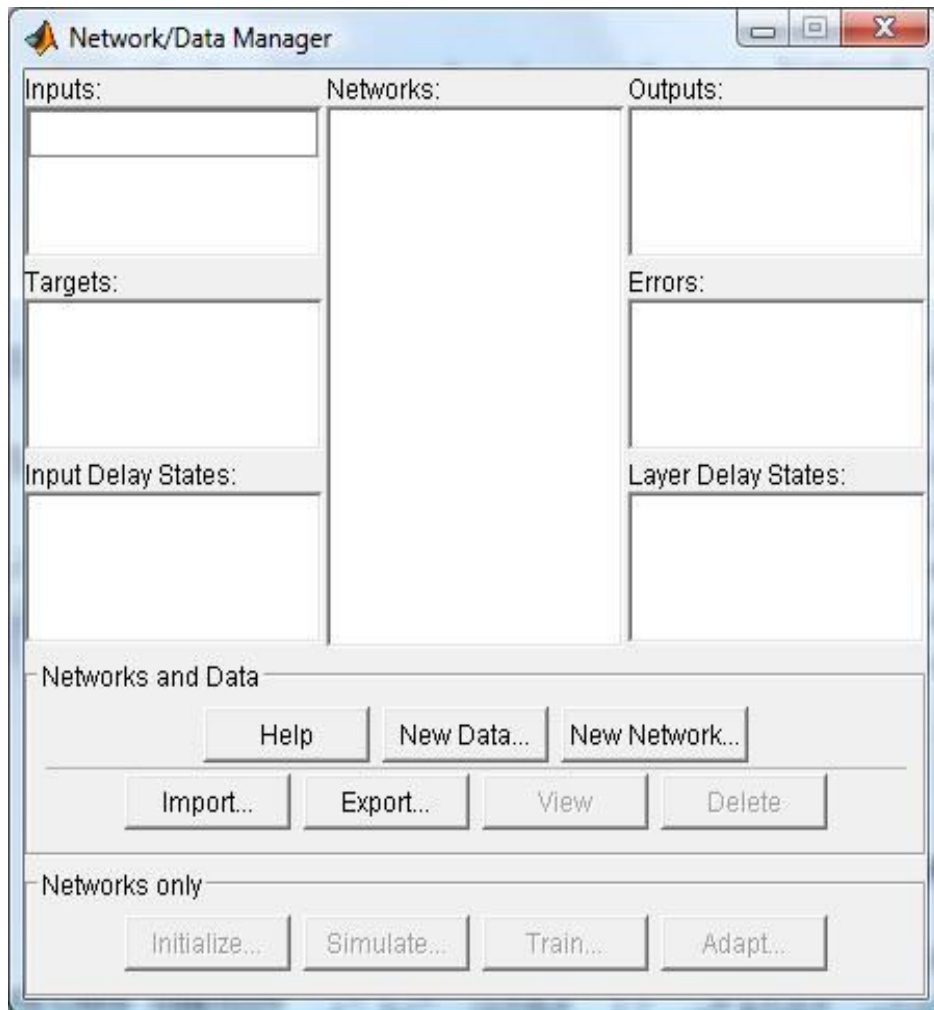
از دیگر ابزارهای کار با نرم افزار مطلب Toolbox های مطلب است که تا حد ممکن کار را برای کاربران مطلب ساده کرده است ، در این تحقیق از یکی از Toolbox های مطلب تحت عنوان Network/Data Manager برای کار با شبکه های عصبی مصنوعی استفاده شده است که در ادامه توضیح مختصری در مورد این ابزار خواهیم داد و در فصل دوم کلیه بخشهای آن را در حین اجرای کار و بدست آوردن نتایج بررسی خواهیم کرد.

3-8-1 بررسی ابزار Network/Data Manager:

همانطور که اشاره شد یکی از ابزارهای کار با شبکه های عصبی مصنوعی در مطلب ابزار Network/Data Manager است. برای اجرای این ابزار در خط فرمان مطلب دستور زیر را تایپ کرده و کلید Enter را می فشاریم:

```
>>nntool
```

با اجرای این دستور ابزار Network/Data Manager اجرا می شود که در شکل زیر ظاهر این ابزار نشان داده شده است.



همانطور که در شکل فوق مشخص است این ابزار از قسمت‌های مختلفی تشکیل شده است که عبارتند از:

1. **Inputs :** ورودی‌های شبکه عصبی مصنوعی در این قسمت مشخص میشوند.
2. **Targets :** نتایج اتفاق افتاده در اثر ورودیها در این قسمت مشخص میشود.
3. **Networks :** در این قسمت نوع شبکه عصبی مشخص میشود، که در این تحقیق از شبکه عصبی مصنوعی تاخیر زمانی استفاده شده است.
4. **Outputs :** نتایجی که شبکه عصبی تولید میکند در این قسمت قابل دسترسی است.
5. **Error :** میزان اختلاف **Target** و **Output** به عنوان خطا در نظر گرفته میشود و در این قسمت قابل دسترسی است.

6. Input Delay States / Layer Delay States : در شبکه های
عصبی مصنوعی تاخیر زمانی این دو قسمت نشان
دهنده میزان تاخیر می باشد.

سایر قسمتهای این ابزار در بخش بعد بطور کامل مورد
بررسی قرار می گیرد ، لذا در این قسمت بحث را کوتاه
می کنیم و در بخش بعد تک تک مراحل شامل تعیین
ورودیها به شبکه عصبی ، اطلاعات هدف (Target) و سایر
قسمتها را مورد بررسی قرار خواهیم داد.

فصل دوم

(روش کار)

1-2 مقدمه :

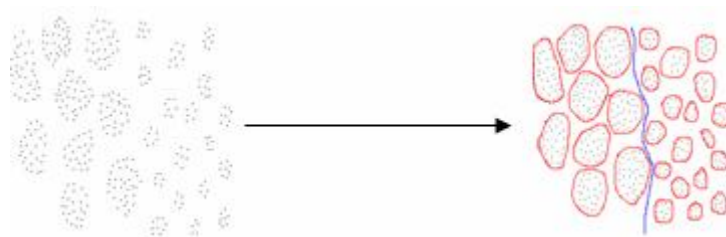
یکی از عوامل مهم در طراحی شبکه های عصبی تعیین ورودیهای شبکه و نتیجه مورد انتظار است که معمولا نتیجه (Target) مشخص است، ولی تعیین ورودیهایی که تاثیر محسوسی بر روی نتیجه داشته باشند کار ساده ای نیست و نیاز به تجربه بالا و تحقیق زیادی در زمینه مورد نظر دارد. بطور مثال در این تحقیق که هدف بدست آوردن دمای حداکثر ایستگاه شرق اصفهان (فرودگاه شهید بهشتی) است نتیجه مشخص است که همان دمای حداکثر است، ولی تعیین پارامترهای ورودی (پارامترهایی که بر حداکثر دما تاثیر گذار هستند) کار ساده ای نیست، هر چند بوسیله تجربه می توان تعدادی پارامتر را که تاثیر بیشتری بر حداکثر دما دارند را کشف کرد ولی مسلما تمام پارامترها نیستند.

در این تحقیق برای تعیین ورودیها از علمی از علوم هوش مصنوعی به نام داده کاوی (Data mining) استفاده کردیم که در ادامه توضیح مختصری در مورد این علم و نحوه بدست آوردن ورودیهای شبکه خواهیم داد.

2-2 Data Mining چیست؟

اصطلاح Data Mining همانطور که از ترجمه آن به داده کاوی مشخص می شود ، به مفهوم استخراج اطلاعات نهان و یا الگوها و روابط مشخص در حجم زیادی از داده های یک یا چند بانک اطلاعاتی بزرگ است . اطلاعات استخراج شده در تعریف Data Mining به طور ضمنی به معنی اطلاعاتی است که براساس آن بتوان به نتایجی دست یافت که به طور معمول ملموس نیستند . در این تعریف بر بزرگ بودن بانکهای اطلاعاتی و یا حجم زیاد داده های مورد پردازش تاکید می شود . علت این است که از نظر آماری و تئوری اطلاعات ، تجزیه و تحلیل داده ها و یا آن طور که در این اصطلاح تعبیر می شود ، کاوش در حجم کم داده های یک بانک به نتایج قابل قبولی منجر نمی شود . به کمک ابزارهای Data Mining می توان مقادیر متغیر هایی را پیش بینی و توصیف نمود . این ابزارها در فرایندهای تصمیم گیری مبتنی بر اطلاعات و دانش (Data and Knowledge driven) کاربرد فراوان دارند و فعالیت های تجاری نوین و مدرن امروزه به شدت بر آن متکی است . فرایند Data Mining را نباید با روش های متداول آنالیز داده و اطلاعات و سیستم های تصمیم گیری معمولی یکی دانست . به کمک روش های Data Mining می

توان به پرسش هایی (عمدتاً تجاری) پاسخ گفت که به طور سنتی عملاً امکان پاسخ به آنها وجود ندارد. داده کاوی جستجوی روابط والگوهای سراسری است که در پایگاه داده های بزرگ در پشت انبوهی از داده ها پنهان شده اند.



داده کاوی عبارت است از فرایند استخراج اطلاعات معتبر ، از پیش ناشناخته ، قابل فهم و قابل اعتماد از پایگاه داده های بزرگ ، استفاده از آن در تصمیم گیری در فعالیت های تجاری مهم ، اصطلاح داده کاوی به فرایند نیمه خودکار تجزیه و تحلیل پایگاه داده های بزرگ به منظور یافتن الگوهای مفید اطلاق می شود. داده کاوی یعنی جستجو در یک پایگاه داده ها برای یافتن الگوهایی میان داده ها. داده کاوی یعنی تجزیه و تحلیل مجموعه داده های قابل مشاهده برای یافتن روابط مطمئن بین داده ها.

1-2-2 فرایند داده کاوی :

فرآیند داده کاوی شامل شش مرحله است که به شرح مختصری در مورد آنها می پردازیم :

1. جمع آوری داده های خام (Data gathering) :

داده هایی که دانش از میان آن استخراج می گردد که این داده ها ممکن است در پایگاه داده یا انبار داده ها (Data Warehouse) قرار داشته باشد. انبار داده مکانیزمی است برای بالا بردن سطح دسترسی و سهولت استخراج گزارشات مدیریتی. و یا به عبارتی دیگر انبار داده ها پردازشی است که در آن داده ها جدا از هم از سازمان های مختلف برای اهداف تصمیم گیری گرد هم جمع آوری می شوند.

2. انتخاب (Selection) :

در این مرحله ، داده های جمع آوری شده براساس شرایط مورد نظر مسئله انتخاب و یا تقسیم بندی می شوند .

مثلا اگر اطلاعات افراد بالای 65 سال مورد نظر باشد، اطلاعات افراد مربوط به سایر سنین حذف می گردد.

3. پاکسازی- پیش پردازش - آماده سازی (Data leaning) :

یکی از مهمترین مراحل است که در این مرحله داده های کثیف از پایگاه داده حذف می شوند . داده های کثیف داده هایی هستند که یکی از شرایط زیر را داشته باشند :

- داده هایی که جا افتاده اند. (فیلدهایی که مقادیر آن وارد نشده است)
- داده های اشتباه. (مثل فیلدی که مربوط به سن است و به اشتباه مقدار 723 در آن ذخیره شده است)
- داده هایی که به صورت استاندارد تعیین شده در پایگاه داده ذخیره نشده اند (مثل تاریخ که ممکن است با استانداردهای متفاوتی ذخیره گردند)

یکی دیگر از کارهای این مرحله حذف دادههایی است که که در روند داده کاوی بی اهمیت هستند . (مثلا اطلاع در مورد صاحب اتومبیل بودن شخصی ، که این اطلاعات در روند بررسی تحصیلات وی اهمیتی ندارد)

4. انتقال و تغییر شکل (Transformation) :

در این مرحله ، داده های سودمند به مدلی انتقال می یابند (یا تبدیل می شوند) که الگوریتم های داده کاوی بتوانند آنها را تحلیل کنند.

مراحل این تبدیل عبارتند از :

- صاف کردن : Noise داده ها حذف می شوند
- تجمع : عملیات مربوط به تجمع روی داده ها اعمال می شود . (مثلا فهرست فروش روزانه به فروش ماهانه تبدیل می گردد)
- عمومی سازی : داده ها از سطوح پایین ادراکی به سطوح بالاتر ادراکی آورده می شوند . (مثلا

اگر نام خیابان که خریدار در آن زندگی می کنند برای تمام مشتریان که در کشور X زندگی می کنند ذخیره می گردد ، نام خیابان به نام شهر تبدیل می گردد)

- نرمال سازی : بوسیله نرمال سازی مقادیر یک فیلد که ممکن است دامنه وسیعی داشته باشند ، به مقادیر بین 0 تا 1 تبدیل می گردند .
- ایجاد یک فیلد جدید : در این مرحله یک فیلد جدید برای کمک به اجرای الگوریتم داده کاوی به پایگاه داده اضافه می شود .

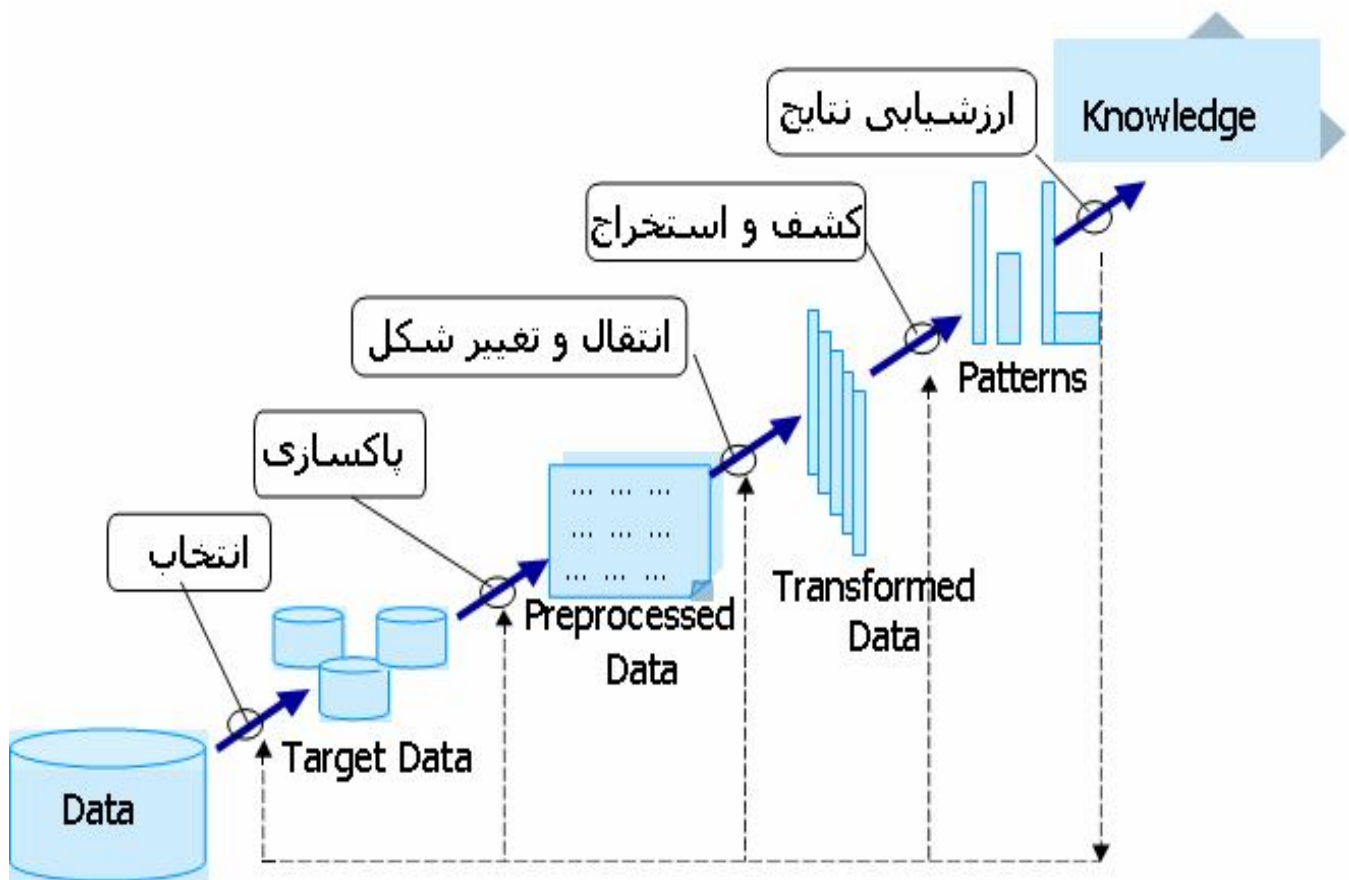
5. کشف و استخراج الگوها (Pattern Discovery) :

اغلب این مرحله را به عنوان داده کاوی می شناسند. در این مرحله الگوریتم های متعددی معرفی شده و هر یک با توجه به نحوه عملکرد برای استخراج الگوهای خاصی بکار می رود .

6. ارزشیابی نتایج (Knowledge) :

در این مرحله الگوهایی که توسط نرم افزار استخراج شده اند مورد تفسیر قرار می گیرند .

در شکل زیر مراحل فرآیند داده کاوی مشخص شده است ، همانطور که عموماً شد مهمترین مرحله در فرآیند داده کاوی مرحله کشف و استخراج الگوهاست که از الگوهای بدست آمده استفاده های متعددی می شود ، در این تحقیق برای بدست آوردن ورودیهای شبکه عصبی مصنوعی استفاده شده است که در ادامه مورد بررسی قرار می شگیرد.

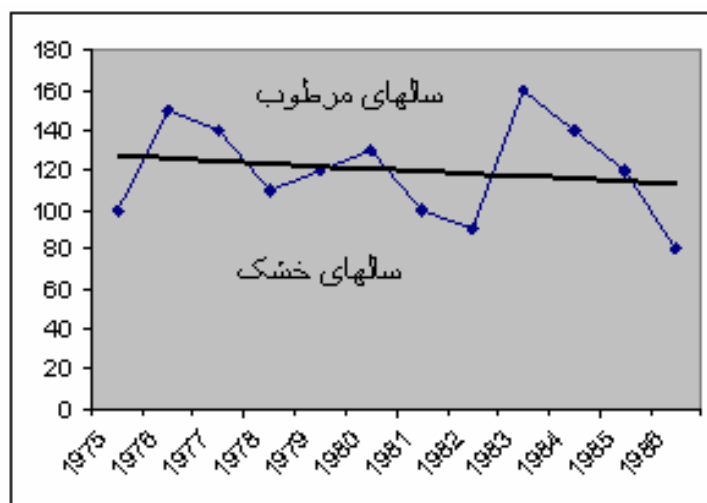


2-2-2 کاربردهای داده کاوی:

در مواردی که نیاز به یافتن نوعی الگو هستیم از داده کاوی استفاده میکنیم. عموماً موارد استفاده داده کاوی به دو دسته اصلی تقسیم می شوند :

1. کاربرد های پیش بینی کننده :

به این ترتیب عمل می کنند که بوسیله استنتاجاتی که از داده های ذخیره شده در پایگاه داده بدست می آورند ، پیش بینی های مورد انتظار را انجام می دهند (دسته بندی داده ها از بارزترین کاربردهای پیش بینی کننده در داده کاوی است) . به عنوان مثال از روی نمودار و مقادیر پارامترهای بارندگی و رطوبت و درجه حرارت سالهای گذشته می توانند پیش بینی کنند که سال های آتی سال خشک خواهد بود یا مرطوب.



2. کاربردهای توصیفی :

توصیف کننده خصوصیات داده هایی است که در پایگاه داده ذخیره شده است (خوشه بندی داده ها (Clustering)) و یافتن قوانین ارتباطی مثالهایی از کاربردهای توصیفی داده کاوی هستند . در این تحقیق از کاربرد توصیف کننده استفاده کرده ایم، با اینکه از خود داده کاوی می توان برای کاربردهای پیش بینی استفاده کرد ، ولی توصیه می شود برای پیش بینیهای سری زمانی مثل پیش بینی وضعیت هوا از سایر متدها مثل منطق فازی ، شبکه عصبی ، الگوریتم ژنتیک و ... استفاده شود.

2-2-3 الگوریتم های استخراج قوانین ارتباطی :

قوانین ارتباطی بیان کننده ارتباط بین فیلدها و مقادیر داده ها در پایگاه داده می باشند . در حقیقت ما دنبال مجموعه ای از مقادیر هستیم که به کررات با هم اتفاق می افتند.

برای استخراج این قوانین دو نوع الگوریتم وجود دارد :

- 1- الگوریتم های تولید کاندید
- 2- الگوریتم های عدم تولید کاندید

در این تحقیق از الگوریتمهای عدم تولید کاندید برای یافتن ورودیها استفاده شده است. لذا از توضیح دادن در مورد الگوریتمهای تولید کاندید خودداری می کنیم.
الگوریتمهای روش عدم تولید کاندید :

در این روش از طریق جستجو که با روش اول عمق می باشد استفاده می کنیم ، که به شرح یکی از جدیدترین این الگوریتمها به نام رشد الگوی تکرار شونده (Frequent Pattern Growth) می پردازیم . این روش به صورت تقسیم و حل عمل می کند . با اعمال این الگوریتم پایگاه داده ما به یک درخت تبدیل می شود . با یک مثال نحوه عملکرد الگوریتم را شرح خواهیم داد .

مثال:

این الگوریتم را بر روی پایگاه داده زیر اعمال می کنیم . (فرض می کنیم درصد حمایت 50% باشد)
 منظور از درصد حمایت 50% این است که به احتمال 50% یکسری از واقعیتهای اتفاق می افتند.

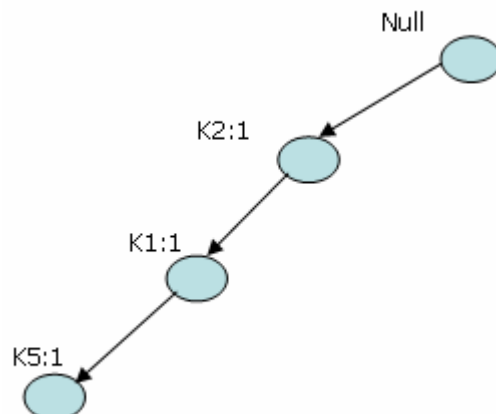
Transaction ID	Items
T1	{ K1,K2,K5 }
T2	{ K2,K4 }
T3	{ K2,K3 }
T4	{ K1,K2,K4 }
T5	{ K1,K3 }
T6	{ K2,K3 }
T7	{ K1,K3 }
T8	{ K1,K2,K3,K5 }
T9	{ K1,K2,K3 }

در مرحله اول با یک بار مرور پایگاه داده ، تعداد تکرار تک تک عناصر موجود در پایگاه داده استخراج می گردد . پس از استخراج مجموعه های تک عنصری آنها را براساس تعداد تکرارشان مرتب می کنیم .

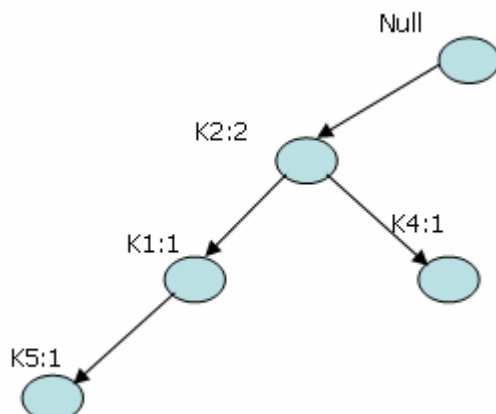
$$L = \{ K2=7 , K1=6 , K3=6 , K4=2 , K5=2 \}$$

حال شروع به ساخت درخت می کنیم . در ابتدا ریشه را NULL قرار می دهیم . به ترتیب رکورد ها را بررسی می کنیم و برای هر رکورد در پایگاه ، یک شاخه ایجاد می کنیم . به این ترتیب که آیتمی که مقدارش بیشتر است

، را ریشه قرار داده و سپس به ترتیب آیتم های دیگر رکورد را با توجه به مقدارشان در آن شاخه ایجاد می کنیم . به عنوان مثال برای رکورد T1 بعد از Null یک شاخه با سه گره ایجاد می کنیم که گره اول K2:1 را قرار داده و بعد گره K1:1 را فرزند گره اول و در آخر K5:1 را فرزند گره دوم قرار می دهیم .

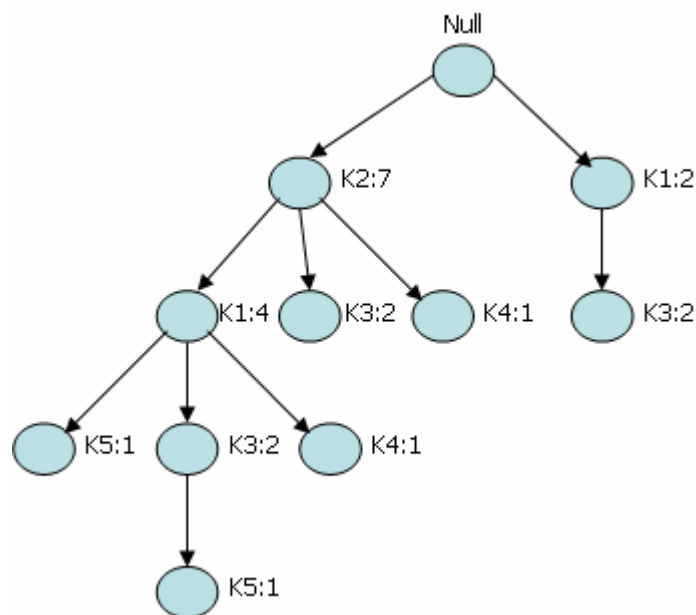


حال به سراغ رکورد T2 می رویم . برای این رکورد که از دو عنصر K2 و K4 تشکیل شده باید یک شاخه ایجاد کنیم . این شاخه از ریشه (NULL) منشعب می شود . چون K2 یک بار ایجاد شده ، به جای ایجاد شاخه جدید یک واحد به تعداد آن اضافه می کنیم . برای گره K4 ، چون این گره از K2 منشعب نشده ، آنرا به عنوان فرزند K2 ایجاد می کنیم .



و به همین ترتیب برای دیگر رکورد ها ، شاخه ایجاد می کنیم و درخت را کامل می کنیم . شکل زیر درختی است

که در پایان و بعد از مرور همه رکوردهای پایگاه ایجاد شده است :



درخت الگوی تکرار (Frequent Pattern)

پس از ساخت درخت ، پایگاه شرطی الگوها ساخته می شود.

برای ایجاد آن برگهای درخت ساخته شده را در نظر می گیریم ، سپس تمام اجداد آن را به ترتیب از بالا به پایین می نویسیم ، سپس آخرین عضو در مجموعه L را یعنی K5 را در نظر می گیریم . K5 در دو شاخه ایجاد شده است .

دو مسیر عبارتند از :

$\langle K2\ K1\ K3\ K5 : 1 \rangle$ و $\langle K2\ K1\ K5 : 1 \rangle$.

در نتیجه اگر K5 را پسوند این مسیر در نظر بگیریم ، $\langle K2\ K1 : 1 \rangle$ و $\langle K2\ K1\ K3 : 1 \rangle$ پیشوندهای K5 خواهد بود که پایگاه الگوی شرطی نام دارد .

در ستون مربوط به FP-tree شرطی ، با توجه به ستون پایگاه الگوهای شرطی و تعداد تکرار عناصر ، عدد حمایتشان را بدست می آوریم . که به عنوان مثال ، $\langle K2:2, K1:2 \rangle$ را خواهیم داشت و چون K3 عدد حمایتش کمتر از 2 است پس نخواهیم آورد .

در آخرین مرحله ، عناصر را با آیتمهای ستون FP-tree شرطی الحاق می کنیم .
(جدول 1-2 مشاهده شود).

الگوهای تکراری ایجاد شده	FP-tree شرطی	پایگاه الگوهای شرطی	عنصر
K2K5:2 , K1K5:2 , K2K1K5:2	< K2:2 , K1:2 >	{(K2K1:1) , (K2K1K3:1)}	K5
K2K4:2	< K2:2 >	{(K2K1:1) , (K2:1)}	K4
K2K3:4 , K1K3:4 , K2K1K3:2	< K2:4 , K1:2 > , < K1:2 >	{(K2K1:2) , (K2:2) , (K1:2)}	K3
K2K1:4	< K2:4 >	{(K2:4)}	K1

جدول 1-2

نتیجه مورد نظر ما ، ستون الگوهای تکراری ایجاد شده می باشد . همانطور که دیده می شود به خاطر اینکه هر الگوریتم طبق قوانین خاصی کار می کند، نتایج حاصله از هر الگوریتم با الگوریتم دیگر متفاوت می باشد .

3-2 تعیین ورودیها :

با توجه به اینکه یکی از کاربردهای داده کاوی، کاربرد پیش بینی کننده است ولی از این خاصیت داده کاوی در این تحقیق استفاده نشده و فقط از کاربردهای توصیف کننده داده کاوی برای تعیین ورودیها استفاده شده است.

برای تعیین ورودیها درصد حمایت 50% در نظر گرفته شده است، به آن معنی که پارمترهایی را به عنوان ورودی تعیین کردیم که تغییرات آنها به احتمال 50% به بالا بر روی خروجی (حداکثر دما) تاثیرگذار بودند.

1-3-2 نحوه تعیین ورودیها :

در ابتدا آمار جمع آوری شده از ایستگاه شرق اصفهان را به پایگاه داده SQL Server انتقال داده و به ازای هر ساعت نمونه برداری شده جدولی بوجود آورده و داده های مورد نظر را در آن قرار دادیم .
از آنجا که اطلاعات هر سه ساعت یکبار نمونه برداری شده بود (00-03-06-09-12-15-18-21) تعداد هشت جدول بوجود آمد که هر جدول شامل فیلدهای زیر می باشد:

1. تاریخ وساعت
2. دید افقی
3. میزان ابناکی هوا
4. سمت باد
5. سرعت باد
6. دمای خشک
7. دمای تر
8. Dew Point
9. رطوبت نسبی
10. فشار بخار اشباع
11. فشار QFE
12. فشار QFF
13. وضعیت هوای حاضر
14. وضعیت هوای سه ساعت گذشته
15. وضعیت هوای شش ساعت گذشته
16. نوع ابر پایین
17. نوع ابر متوسط
18. نوع ابر بالا

یک جدول هم به ازای تمام ساعات در نظر گرفتیم (Total) ، که شامل فیلدهایی مشابه هر یک از هشت جدول قبل می باشد.

همچنین برای پارامترهایی که در روز فقط یکبار نمونه برداری شده بودند هم یک جدول در نظر گرفتیم که شامل فیلدهای زیر می باشد:

1. تاریخ
2. دمای حداکثر
3. دمای حداقل
4. میزان تبخیر روزانه
5. میزان ساعت تابش خورشید

بعد از ساخت پایگاد داده ، اطلاعات را پاکسازی کرده (فیلدهای خالی را با استفاده از روشهای میان یابی پر کردیم) و الگوریتم ¹FPG را بر روی جداول پایگاه داده اعمال کردیم و نتایج بدست آمده را به عنوان ورودی در نظر گرفتیم که این پارامترها عبارتند از:

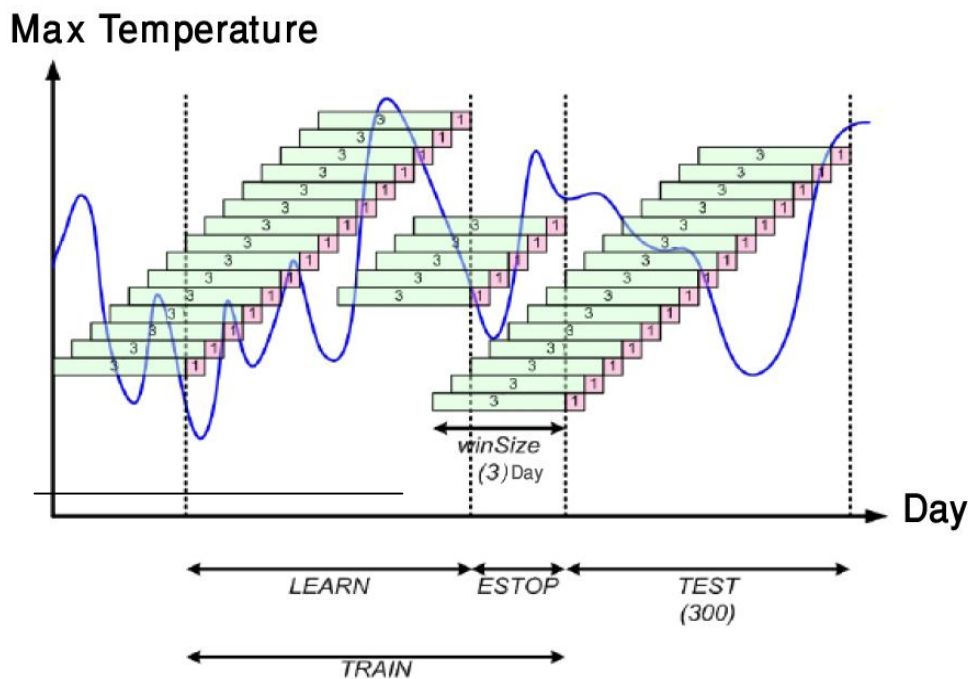
1. سرعت باد
2. دمای خشک
3. میزان ابرناکی هوا
4. فشار QFE
5. رطوبت نسبی

لازم به ذکر است که برای استفاده از الگوریتم FPG از یک SP¹ که در محیط SQL Query Analyzer نوشته شد استفاده کردیم و آن SP را بر روی جداول پایگاه داده اعمال کرده و در نهایت پنج پارامتر فوق را به عنوان ورودی تایین کردیم.

بعد از تعیین ورودیها باید این اطلاعات را به فرمی تبدیل کنیم که قابل فهم برای شبکه عصبی مصنوعی باشند.

از آنجا که تنها خروجی ما دمای حداکثر فرودگاه شرق اصفهان می باشد مشخص است که در خروجی با برداری مواجه هستیم که تعداد n سطر و 1 ستون می باشد (یعنی با یک بردار $n*1$ روبرو هستیم) که در این بردار n تعداد روزهایی است که برای آن پیش بینی انجام می دهیم. در این تحقیق با استفاده از اطلاعات سه روز قبل دمای حداکثر فردا را پیش بینی می کنیم که فقط شامل یک پارامتر است به این معنی که بردار خروجی ما $1*1$ است.

هدف از این تفاسیر تعیین بعد ماتریس ورودی می باشد ، از آنجا که در خروجی 1 ستون داریم اطلاعات ورودی هم باید بصورت $m*1$ به شبکه وارد شوند لذا باید اطلاعات ورودی را به فرم $m*1$ تبدیل کنیم ، از طرفی هم اطلاعات هر روز باید از روز دیگر مجزا باشد چرا که ما پنجرهای در نظر میگیریم که با گذشتن هر روز باید تغییر مکان بدهد .



بطور مثال اگر اکنون در روز دوشنبه قرار داشته باشیم با استفاده از اطلاعات روزهای شنبه ، یکشنبه و دوشنبه حداکثر دمای سه شنبه را پیش بینی می کنیم . حال با ورود به روز سه شنبه باید با استفاده از اطلاعات یکشنبه ، دوشنبه و سه شنبه برای روز چهارشنبه پیش بینی را انجام دهیم . لذا باید اطلاعات هر روز را در یک بردار جزا قرار دهیم . که در ادامه نحوه این تبدیل را توضیح می دهیم .

همانطور که عنوان شد اطلاعات هر روز باید در یک ماتریس جزا قرار گیرد ،لذا اطلاعات ورودی ما باید به فرم زیر تبدیل شود

$\{[روز سوم] , [روز دوم] , [روز اول]\} \longrightarrow \{[target]\}$

از آنجا که اطلاعات ما به صورت ستونی در دسترس می باشند باید عملیاتی انجام پذیرد تا به فرم دخواه بالا تبدیل شود که این کار به سادگی با نرم افزار مطلب ممکن می باشد که در ادامه در مورد نحوه تبدیل اطلاعات به فرم بالا توضیح می دهیم .

برای تبدیل ماتریس ستونی به فرم دخواه ابتدا با استفاده از دستور **Reshape** در مطلب بردار سطری را به یک ماتریس تبدیل می کنیم که اطلاعات هر روز در یک

سطر ماتریس واقع شده باشد. با یک مثال نحوه تبدیل را توضیح می دهیم:

مثال:

فرض کنیم اطلاعات یک ماه را می خواهیم به فرم مورد نظر تبدیل کنیم:

حل: از آنجا که در هر ساعت با 5 دسته از اطلاعات روبرو هستیم (سرعت باد ، دمای خشک ، میزان ابرناکی هوا ، فشار QFE ، دمای خشک) و قبلا هم عنوان شد که برای هر روز 8 نمونه برداری موجود است پس بردار ورودی هر روز باید به ابعاد 40×1 تبدیل شود. در ابتدا با یک بردار 1200×1 روبرو هستیم (با فرض اینکه ماه 30 روزه باشد و اطلاعات در بردار input ذخیره شده باشند)

```
In1=Reshape(input,40,30)';
```

با اجرای این دستور ماتریسی به نام In1 با ابعاد 30×40 از روی بردار input تولید میشود که هر سطر در برگزیده اطلاعات 1 روز می باشد. سپس باید اطلاعات هر روز را در یک بردار بصورت سطری ذخیره کنیم که اینکار را با یک m-file به صورت زیرانجام می دهیم. (m-file زیر برنامه ای است که در مطلب نوشته می شود).

```
For i=1:30
```

```
Input_1(i)={In1(i ; :)}'
```

```
End
```

با اجرای این دستور اطلاعات به فرم زیر تبدیل شده و اطلاعات هر روز در یک بردار ذخیره می شود.

	{	0]
		3
اطلاعات ساعت 00 روز اول		851
		4.2-
		41
		2
		5
اطلاعات ساعت 03 روز اول		851.4
		2.4-
		38

اطلاعات ساعت 06 روز اول	{	i5
		i5
		i852.7
		i1.2
اطلاعات ساعت 09 روز اول	{	i31
		i4
		i5
		i851.3
اطلاعات ساعت 12 روز اول	{	i10.4
		i27
		i2
		i6
اطلاعات ساعت 15 روز اول	{	i849.5
		i12.8
		i25
		i4
اطلاعات ساعت 18 روز اول	{	i5
		i849.7
		i6.6
		i37
اطلاعات ساعت 21 روز اول	{	i2
		i7
		i850.4
		i5.4
اطلاعات ساعت 00 روز دوم	{	i32
		i0
		i6
		i850.1
اطلاعات ساعت 00 روز دوم	{	i2
		i35
		i2]
		i5
اطلاعات ساعت 00 روز دوم	{	i850.1
		i0.4
		i44
		i4
اطلاعات ساعت 00 روز دوم	{	i7
		i4
		i7
		i4

•
•
•

همانطور که در مثال بالا مشخص است برای هر روز یک بردار 40×1 تولید می شود که از این بردار به عنوان ورودی شبکه عصبی استفاده می شود.

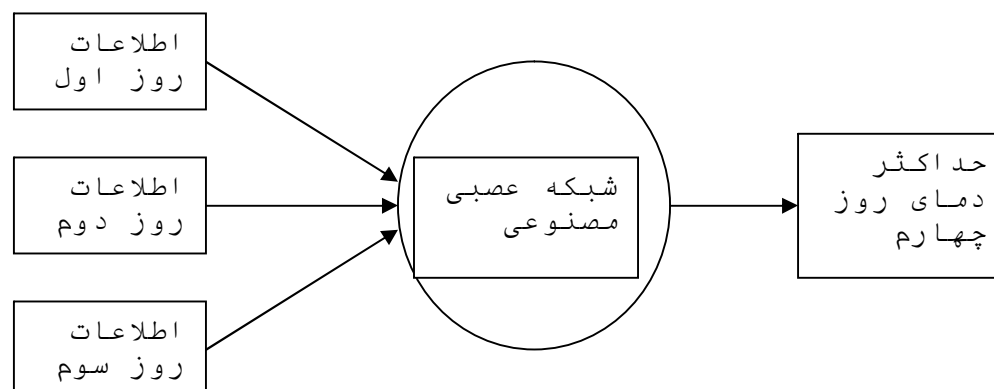
بعد از تبدیل ورودیها باید اطلاعات هدف (target) را تولید کنیم که در ادامه به بحث در این مورد می پردازیم.

2-4 تعیین خروجی های مورد استفاده شبکه عصبی:

روند تهیه خروجیها مشابه به ورودیهاست و تنها نکته ای که باید مد نظر قرار گیرد این است که تعداد بردارهای خروجی با تعداد بردارهای ورودی شبکه باید یکسان باشد.

از آنجا که خروجیهای اتفاق افتاده (خروجیهای مورد استفاده شبکه (target)) برای آموزش شبکه استفاده می شوند لذا نحوه انتخاب خروجیها از اهمیت ویژه ای برخوردار است ، بطور مثال در این تحقیق از اطلاعات سه روز گذشته برای پیش بینی حداکثر دمای روز چهارم استفاده می کنیم لذا تاخیر فاز اطلاعات ورودی با اطلاعات هدف باید 3 روز در نظر گرفته شود.

مثلا اگر اطلاعات روز 1،2،3 یک ماه به عنوان ورودی به شبکه در نظر گرفته شود حداکثر دمای روز 4 آن ماه به عنوان اطلاعات هدف در نظر گرفته می شود.



حال به توضیح در مورد نحوه کار در نرم افزار مطلب می پردازیم.

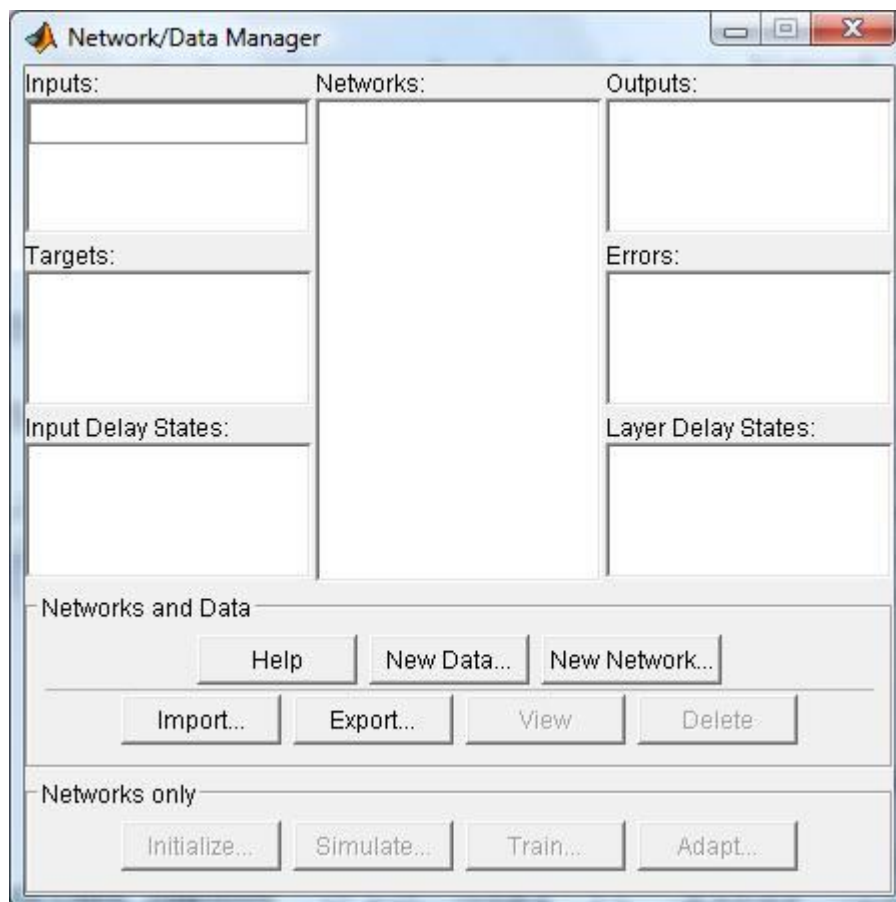
2-5 روند کار در نرم افزار مطلب:

بعد از تولید بردارهای ورودی و خروجی باید شبکه را آموزش دهیم تا از آن بتوانیم برای پیش بینی استفاده کنیم.

همانطور که در فصل قبل عنوان شد در نرم افزار مطلب ابزاری تحت عنوان **Network/Data Manager** برای کار با شبکه های عصبی مصنوعیتعبیه شده است ، برای اجرای این ابزار از دستور زیر استفاده می کنیم:

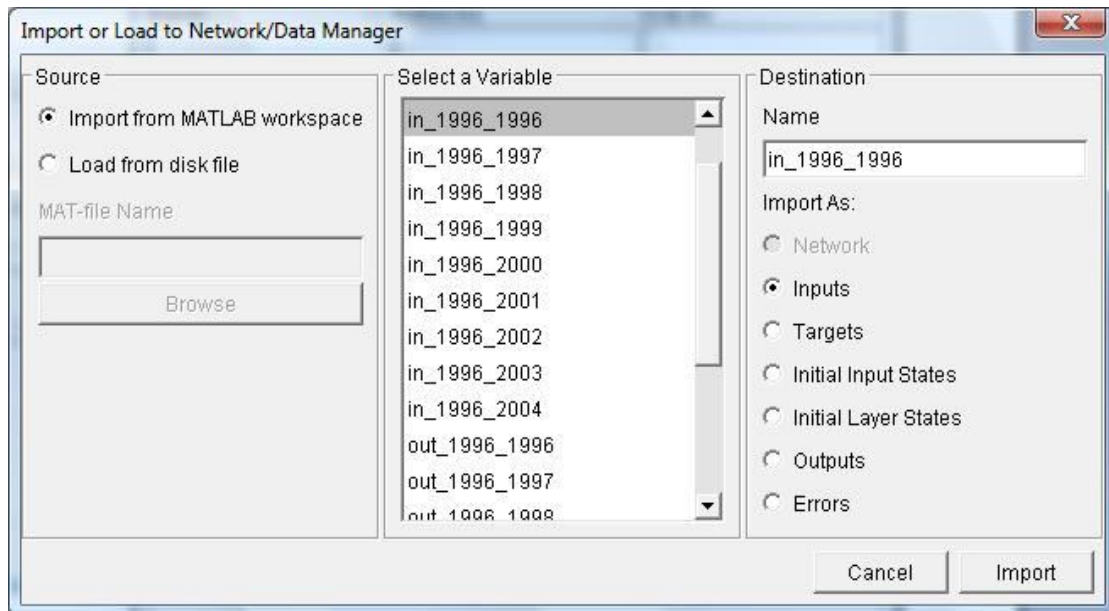
>>nntool

با اجرای این دستور صفحه مشابه شکل زیر ظاهر می شود که در ادامه مورد استفاده قرار می گیرد.

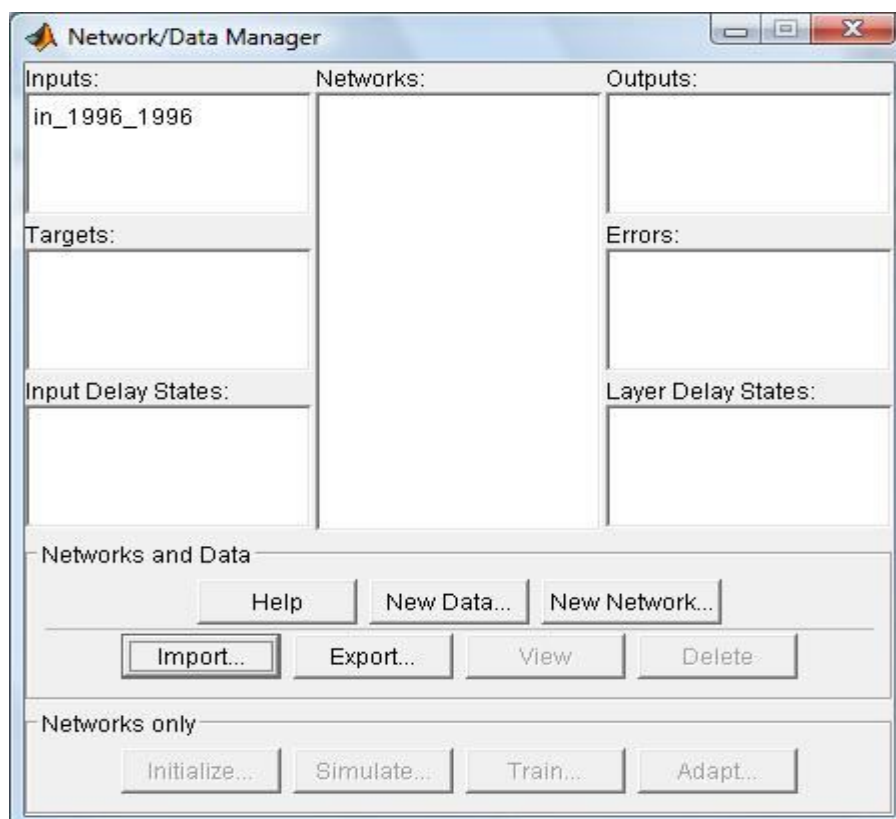


همچنان که در شکل مشخص است برای کار با شبکه عصبی ابتدا باید ورودیها (**Inputs**) و اطلاعات هدف (**Targets**) را مشخص کنیم. برای این کار بر روی کلید **Import...** کلیک

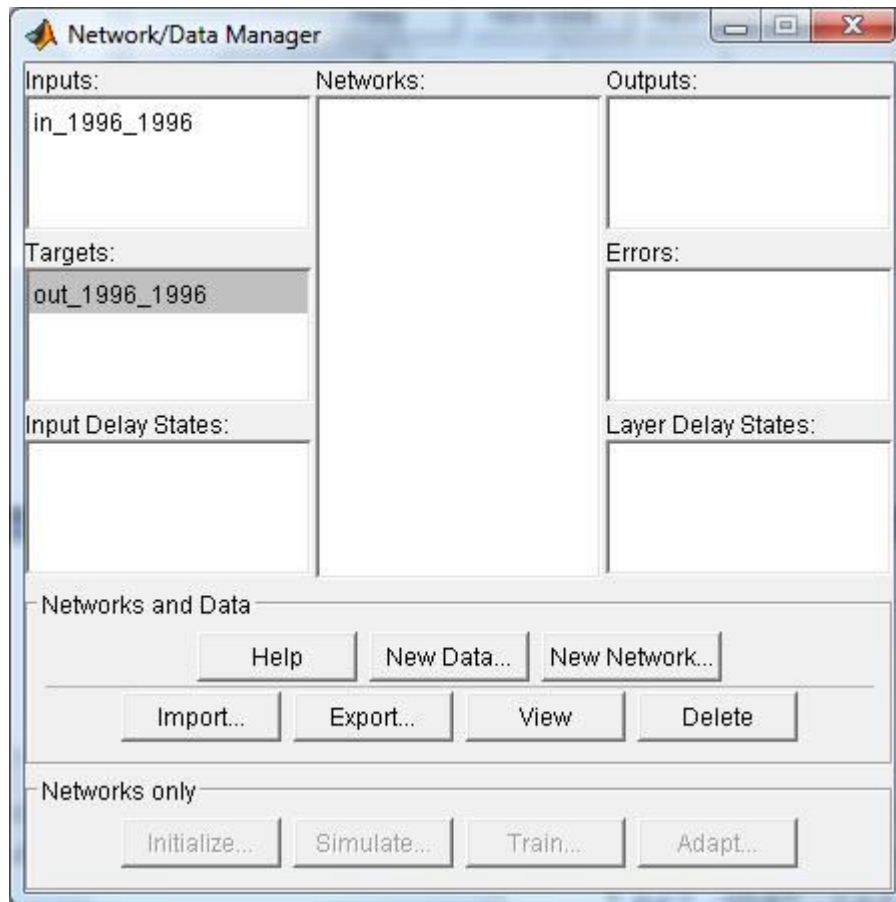
می کنیم ، تا صفحه Import or Load to Network/Data Manager ظاهر شود.



در این صفحه اطلاعات ورودی به شبکه را انتخاب کرده سپس در قسمت Destination ، Inputs را انتخاب کرده و بر روی کلید Import کلیک می کنیم. با این کار اطلاعات به قسمت ورودی شبکه عصبی اضافه می شود (شکل زیر).



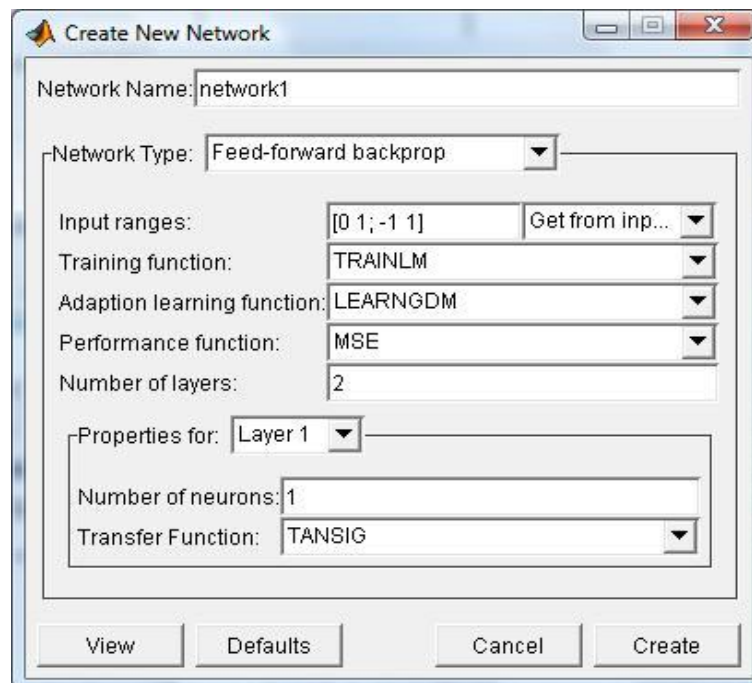
سپس مجدداً کلید **Import** را کلیک کرده اطلاعات هدف را مشخص می‌کنیم و در قسمت **Destination** ، **Targets** را انتخاب کرده و کلید **Imports** را کلیک می‌کنیم تا اطلاعات هدف نیز اضافه شود (شکل زیر).



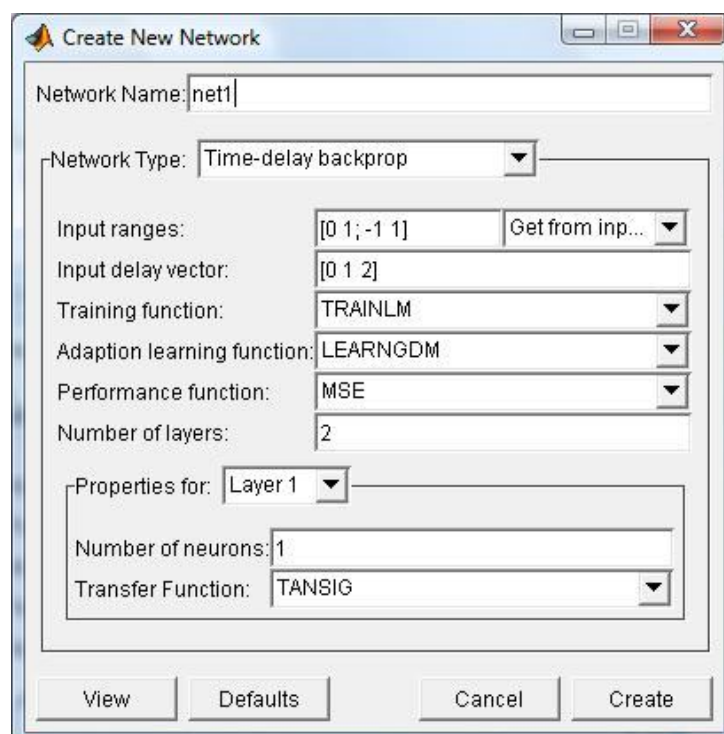
لازم به ذکر است که اگر اطلاعات ورودی یا اطلاعات هدف در فایل متنی (txt) یا به هر صورت قابل فهم برای شبکه بر روی هارد ذخیره شده باشند در صفحه **Import or Load to** **Network/Data Manager** باید گزینه **Load from disk file** را انتخاب کرده و با زدن کلید **Browse** مسیر فایل را مشخص کنیم و بعد از انتخاب مراحل مشابه قبل ادامه می‌یابد.

بعد از تعیین ورودیها و اطلاعات هدف باید شبکه عصبی مورد نظر را انتخاب کنیم.

برای اینکار بر روی کلید New Network... کلیک می کنیم تا صفحه Create New Network باز شود (شکل زیر).



در این صفحه در بخش Network Name نام شبکه عصبی را تعیین می کنیم.
از بخش Network Type نوع شبکه عصبی را تعیین می کنیم که در این تحقیق از Time-Delay backprop استفاده می کنیم (شکل زیر).



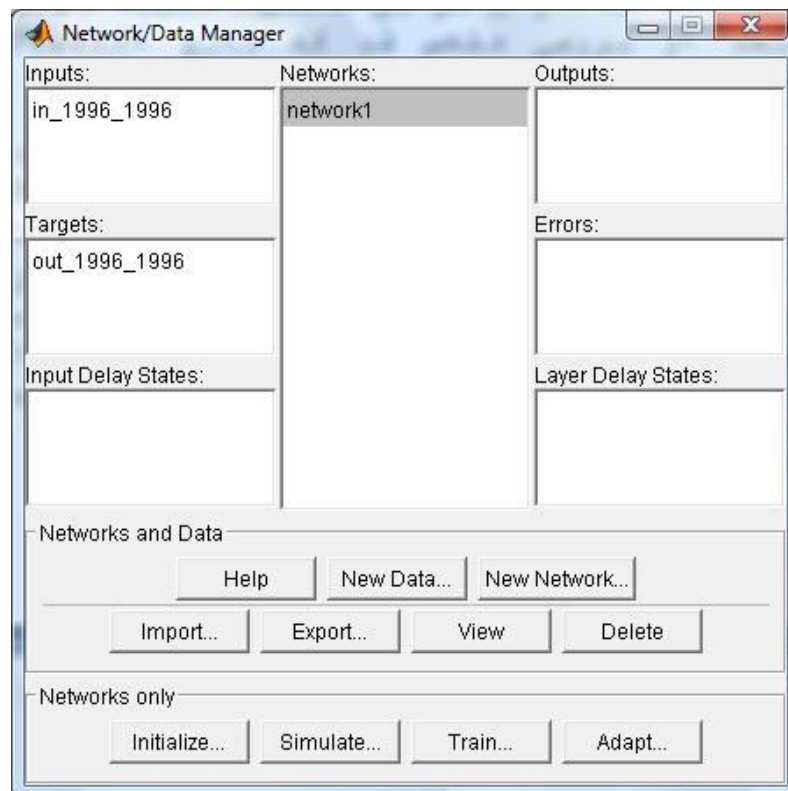
چنانچه در شکل فوق مشخص است این نوع از شبکه عصبی شامل بخشی به نام **Input delay vector** است که انواع دیگر شبکه این حالت را دارا نمی باشند.

در قسمت **Input ranges:** از **combo box** مقابل آن اطلاعات ورودی را مشخص می کنیم تا رنج اطلاعات مشخص شود. (منظور از رنج اطلاعات حداقل و حداکثر هر بردار است که این کار بصورت اتوماتیک صورت می پذیرد).

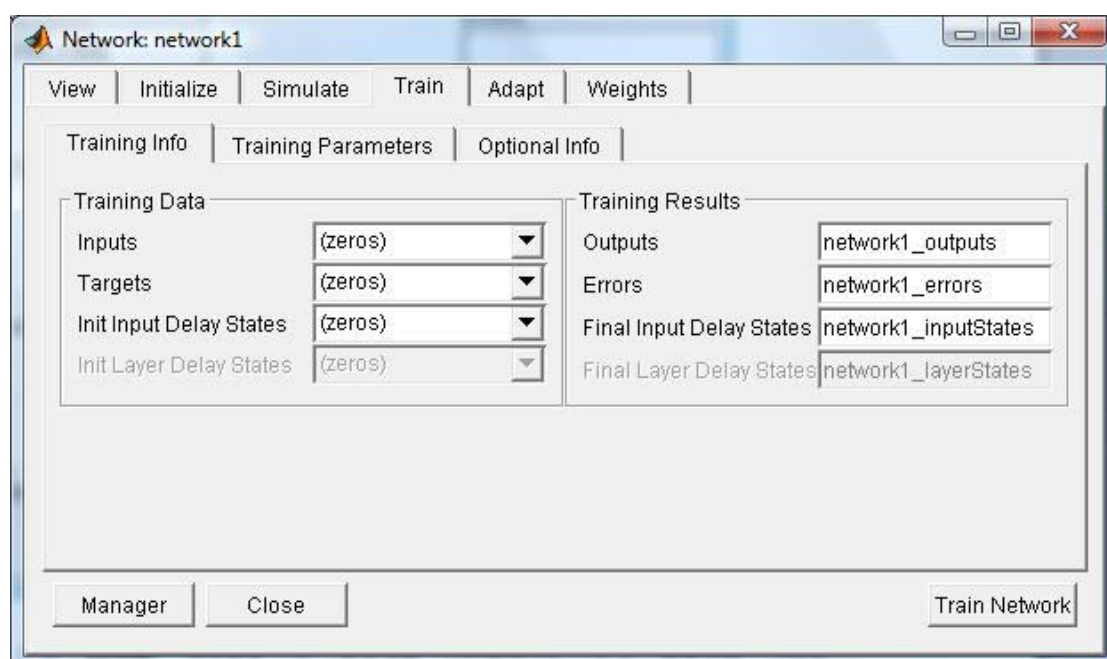
در قسمت **Input delay vector** هم با استفاده از یک بردار تعداد تاخیر زمانی را مشخص می کنیم. در حالت پیش فرض مطابق شکل بالا از یک بردار بصورت [0 1 2] استفاده شده است که به معنای سه تاخیر زمانی است. اگر بخواهیم بطور مثال از 5 تاخیر زمانی استفاده کنیم باید بردار را بصورت [0 1 2 3 4] تعریف کنیم ، ولی از آنجا که ما از اطلاعات سه روز قبل می خواهیم و اطلاعات هر روز در یک بردار ذخیره شده است بردار را بصورت پیش فرض آن که مربوط به سه تاخیر زمانی است انتخاب می کنیم. در قسمت **Properties for: Layer1** ، انتخاب کرده و سپس در قسمت **Number of neurons** تعداد نرونها لایه اول را مشخص می کنیم. مشخص کردن تعداد نرونها بصورت آزمون و خطا بدست می آید ، یعنی تعداد نرونها را آنقدر تغییر می دهیم و شبکه را آموزش می دهیم و نتایج را بررسی می کنیم تا به کمترین میزان خطا برسیم. در این تحقیق تعداد نرونها را از 1 تا 55 نرون تغییر دادیم و نتایج را بررسی کردیم و به این نتیجه رسیدیم که بهترین حالت به ازای 45 نرون در لایه اول تولید می شود. بعد از تعیین تعداد نرونها باید تابع فعالیت را مشخص کنیم که در این تحقیق از تابع **PURELIN** استفاده شده است. البته انتظار می رفت که به ازای توابع **Sigmoid** نتیجه بهتر باشد ولی بعد از بررسی مشخص شد که تابع **PURELIN** بهترین گزینه برای پیش بینی دما است.

بعد از این کار باید لایه دوم را هم تنظیم کنیم که این کار هم مشابه حالت قبل است. از قسمت **Properties for: Layer2** ، انتخاب می کنیم سپس در قسمت **Number of neurons** تعداد نرونها لایه دوم را مشخص می کنیم (در این تحقیق از 1 نرون در لایه دوم استفاده شده است)، سپس از بخش **Transfer Function** تابع فعالیت لایه دوم را مشخص می کنیم که مجدداً از تابع **PURELIN** در این لایه استفاده شده است.

در قسمت **Number of layers:** هم تعداد لایه ها مشخص می شود که نهایتاً دو لایه برای هر الگویی کفایت می کند. سایر بخشها را هم بصورت پیش فرض رها می کنیم و کلید **Create** را کلیک می کنیم تا شبکه ساخته شود (شکل زیر).

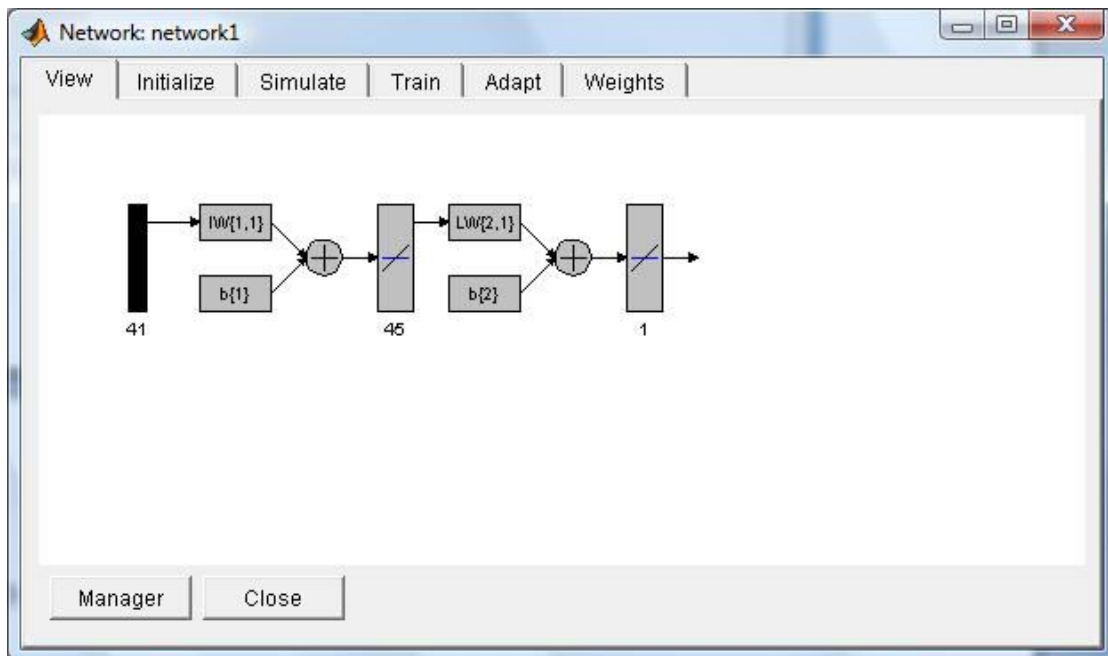


حال شبکه را انتخاب کرده و کلید **Train** را کلیک می کنیم پنجره **Network (network name)** ظاهر شود (شکل زیر).

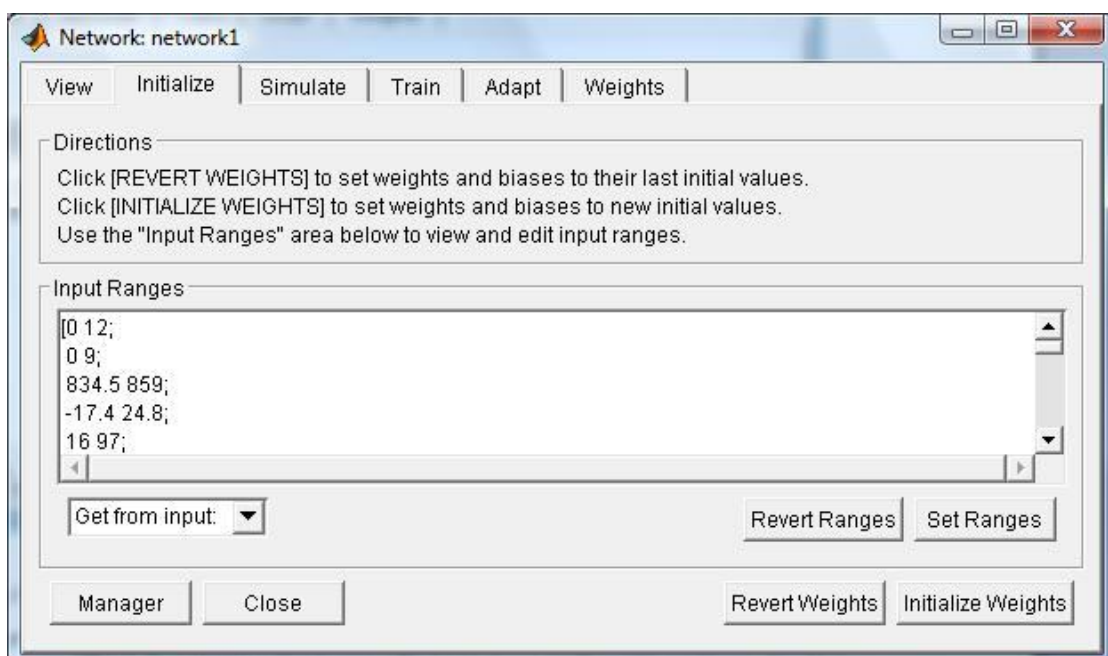


این پنجره شامل بخشهای مختلفی است که توضیح آنها می پردازیم.

در اولین Tab از این پنجره (گزینه view) شماتیک شبکه طراحی شده قابل بررسی است (شکل زیر).



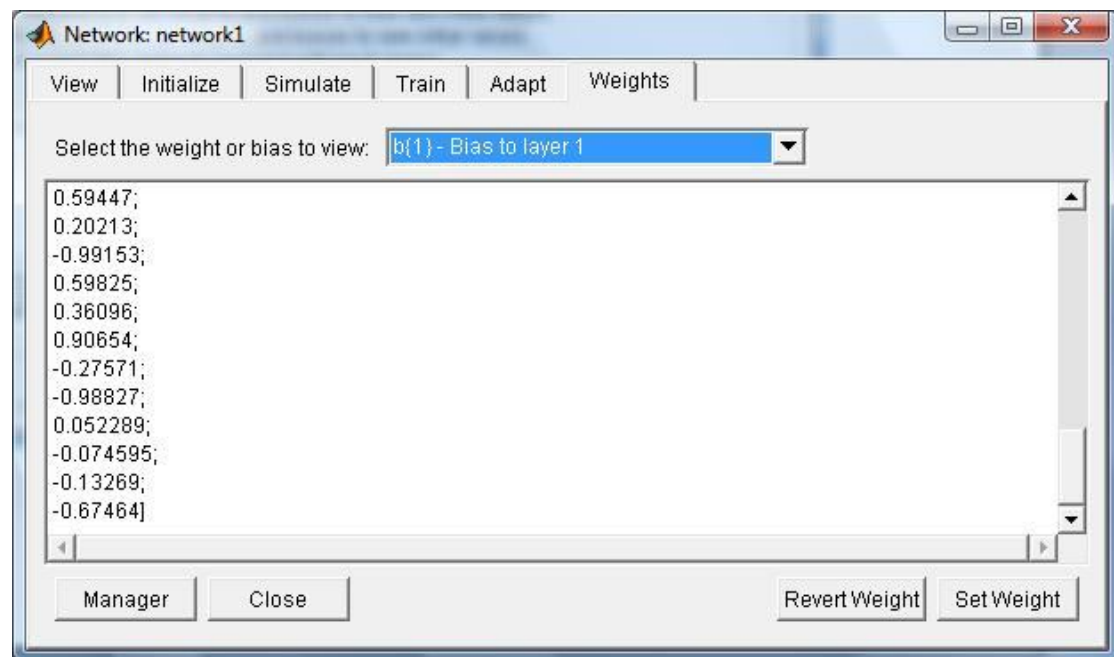
در دومین Tab از این پنجره (Initialize) مقادیر اولیه ورودی به شبکه قرار دارد که در هنگام طراحی شبکه به شبکه اعمال شده است (در بخش Input range) در این Tab میتوان این مقادیر را تغییر داد (شکل زیر).



Tab سوم بعد از آموزش شبکه عصبی برای پیش بینی استفاده می شود که در آینده مورد بررسی قرار می گیرد.

دو Tab بعدی یعنی Train و Adapt برای آموزش شبکه استفاده می شوند که در این تحقیق و بطور کلی در شبکه های عصبی تاخیر زمانی توصیه می شود که به جای Adapt از Train برای آموزش استفاده شود.

و در Tab آخر یعنی Weights مقادیر بایاس و وزنها قابل دسترسی هستند (شکل زیر).

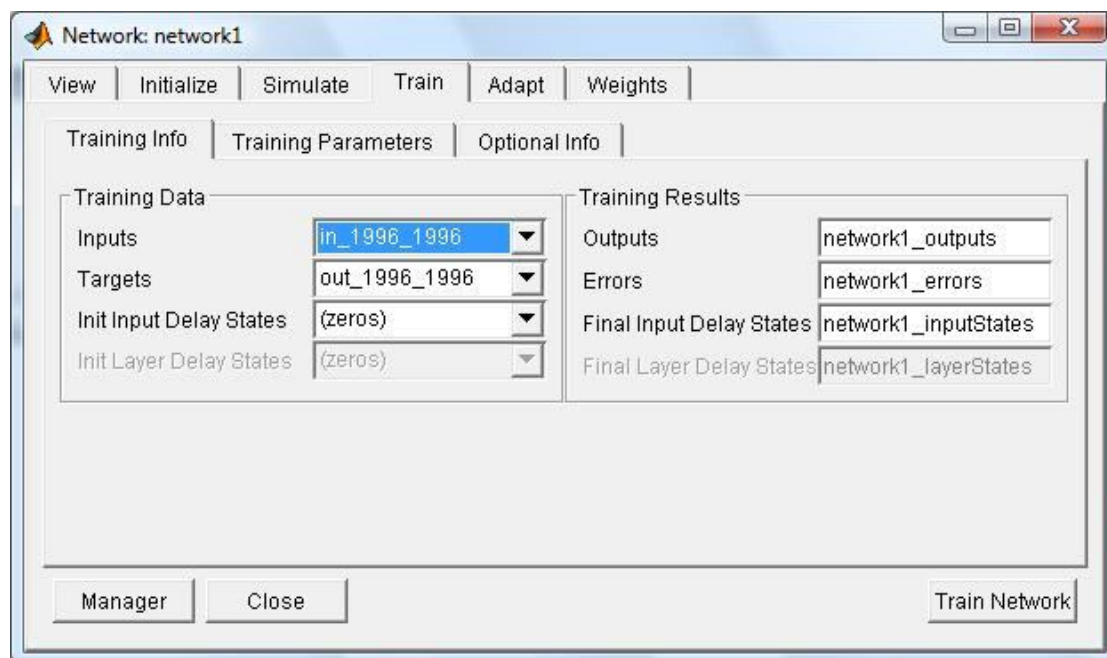


از این مقادیر در آینده شبکه برای پیش بینی استفاده می کند. در ضمن در طول آموزش شبکه مدام این مقادیر تغییر می کنند تا کمترین خطا حاصل شود.

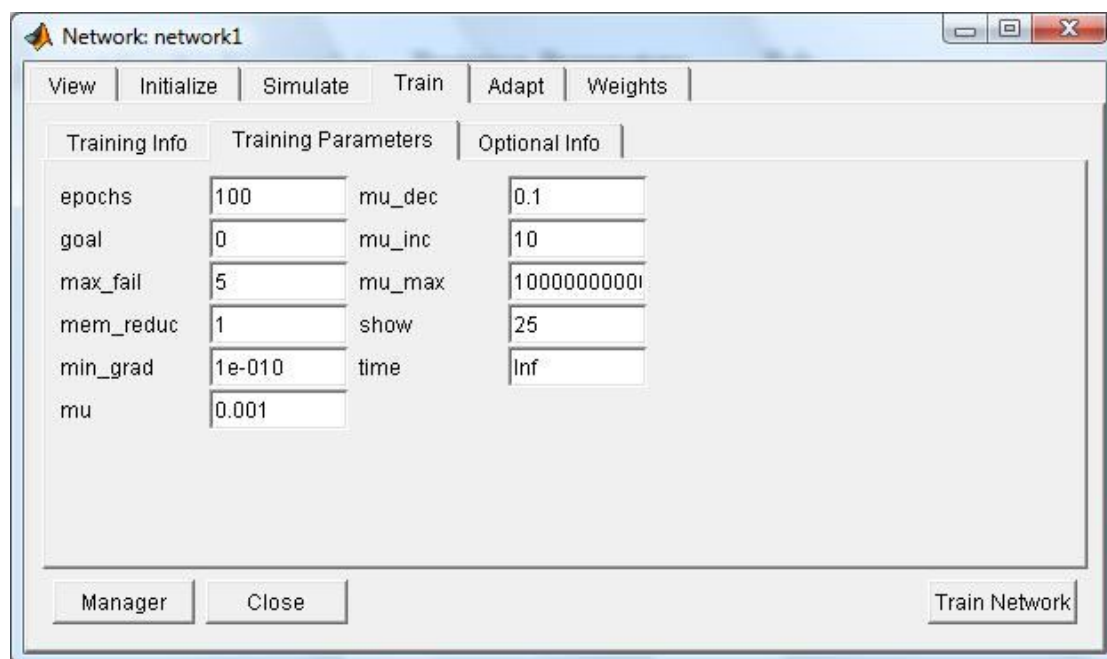
2-6 آموزش شبکه :

بعد از تعیین ورودیها و خروجیها و همچنین طراحی خود شبکه ، شبکه باید آموزش ببیند ، برای اینکار بعد از زدن کلید Train در صفحه Network/Data Manager که باعث باز شدن صفحه Network(network name) می شود به قسمت Train رفته (که بطور پیش فرض در این قسمت قرار داریم)

سپس از بخش Inputs ورودیهای شبکه عصبی را تعیین کرده و از بخش Target اطلاعات هدف را مشخص می کنیم (شکل زیر).



سپس به Tab Training Parameters ، رفته و تنظیمات مورد نظر برای آموزش شبکه را اعمال می کنیم.



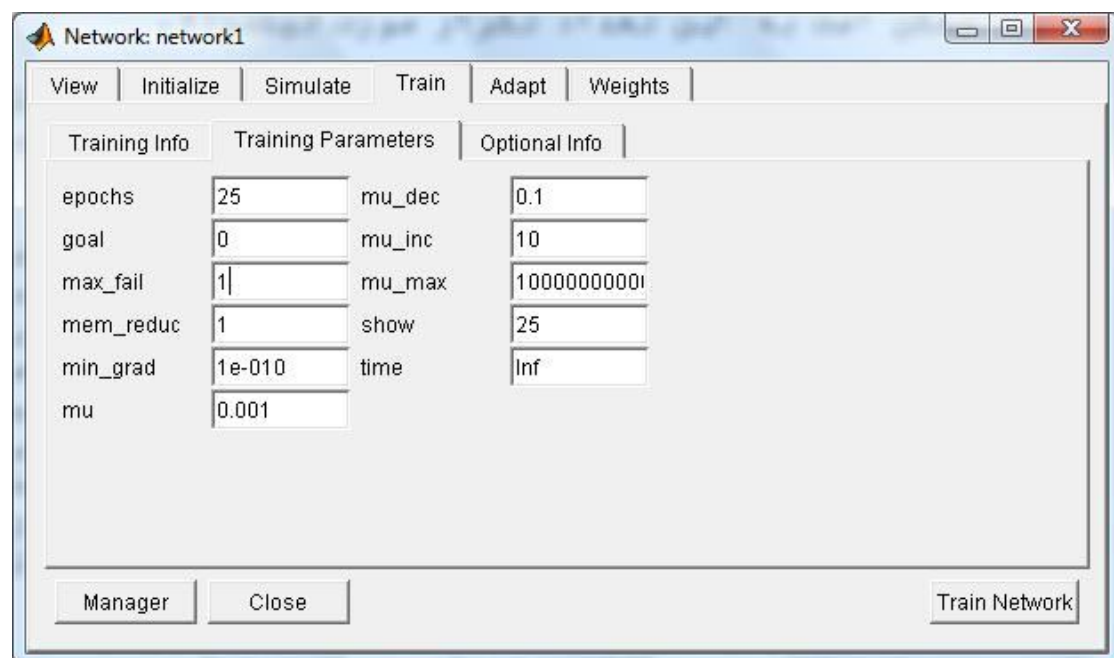
در این صفحه دو پارامتر کلیدی وجود دارد که بیشتر این دو پارامتر را مورد بررسی قرار می دهیم و از توضیح سایر بخشها خودداری می کنیم.

اولین پارامتر فیلد **epochs** می باشد که مربوط به تعداد تکرار شبکه است ، به این منظور که شبکه در حین آموزش حداکثر به تعداد **epochs** تکرار شود (دقت شود که از کلمه حداکثر استفاده شده به این معنی که ممکن است به این تعداد تکرار صورت نپذیرد).

دومین پارامتر فیلد **max_fail** است که بیانگر حداکثر خطای مورد قبول ماست و شبکه تا زمانی که به این میزان خطا نرسیده تکرار می شود.

حال با توجه به این دو پارامتر مشخص است که شبکه یا به تعداد **epochs** تکرار می شود یا تا زمانی که به میزان خطای تعریف شده در آن یعنی **max_fail** برسد و به هر کدام از این دو مورد رسید شبکه از آموزش باز می ایستد. لازم به ذکر است که اگر بعد از چند مرتبه تکرار شبکه بهبودی در نتایج حاصل نشود (نتایج تغییری نکنند) شبکه از آموزش باز می ایستد هر چند که تعداد تکرار آن تکمیل نشده و به میزان خطای مورد قبول هم نرسیده باشد ، چراکه از آن به بعد آموزش بیهوده است.

در این تحقیق حداکثر خطا 1 درجه سانتیگراد در نظر گرفته شده است و تعداد تکرار هم 25 مرتبه.



بعد از این تنظیمات بر روی کلید **Train Network** کلیک می کنیم تا شبکه آموزش خود را شروع کند.

در این تحقیق شبکه به ازای اطلاعات 9 سال (از سال 1985 تا 2004) آموزش دید و برای سال 2005 پیش بینی انجام داد.

در کامپیوتری با مشخصات زیر:

Ram 1GB

Cpu 3.00GHZ

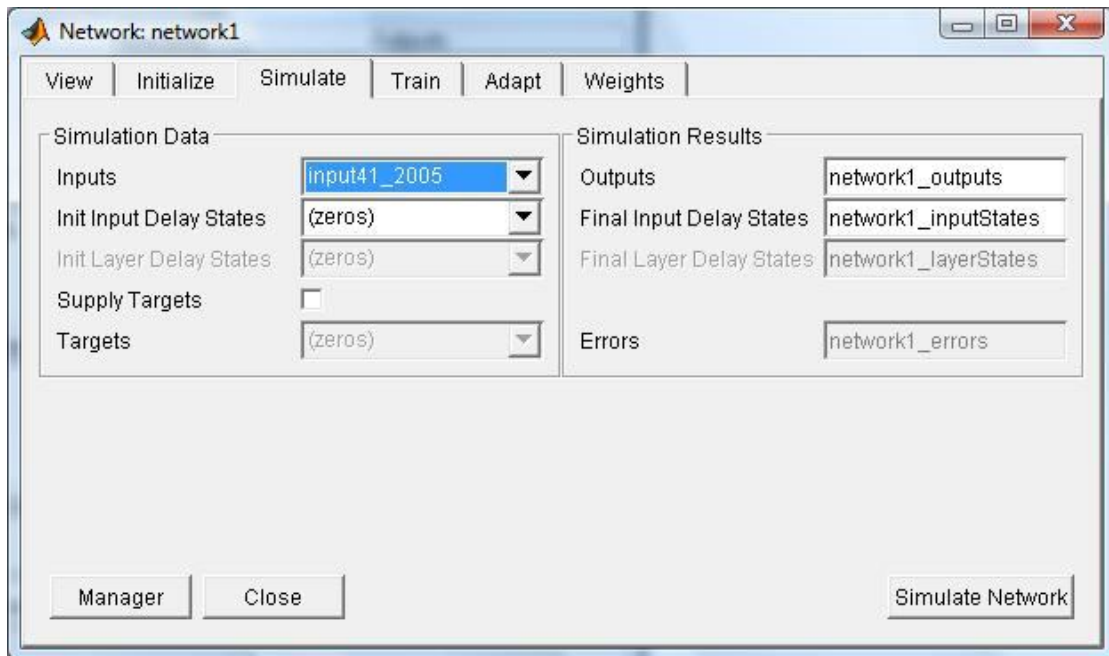
مدت زمان آموزش 32 ساعت بود که بیانگر حجم زیاد پردازش می باشد.

2-7 پیش بینی:

بعد از آموزش شبکه مرحله بهره برداری از شبکه عصبی است.

در این مرحله فقط نیاز به اطلاعات ورودی است و شبکه با استفاده از وزنهایی که در حین آموزش بدست آورده برای آینده پیش بینی را انجام می دهد.

بعد از انتخاب ورودی در صفحه Netwrok/Data Manager روی کلید Simulate... کلیک می کنیم تا صفحه Netwrok(network name) باز شود سپس در Tab ، Simulate در قسمت Inputs ورودی را مشخص می کنیم (شکل زیر).

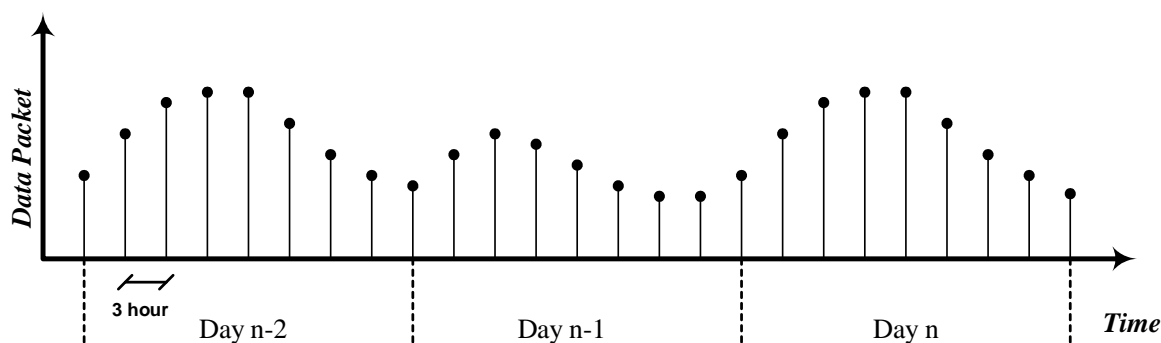


سپس بر روی کلید Simulate Network کلیک می کنیم تا شبکه با استفاده از وزنهای بدست آمده در حین آموزش برای ورودیهای مشخص شده خروجی تولید کند که خروجی تولید شده همان پیش بینی است که مورد استفاده قرار می گیرد.

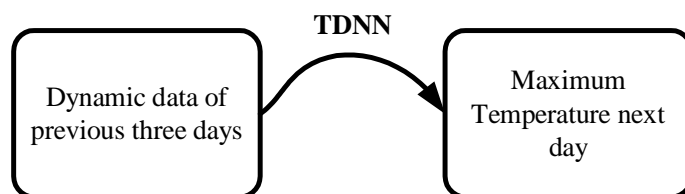
خروجیها در صفحه Network/Data Manager در بخش Outputs: قابل بررسی هستند.

2-8 بررسی نتایج بدست آمده :

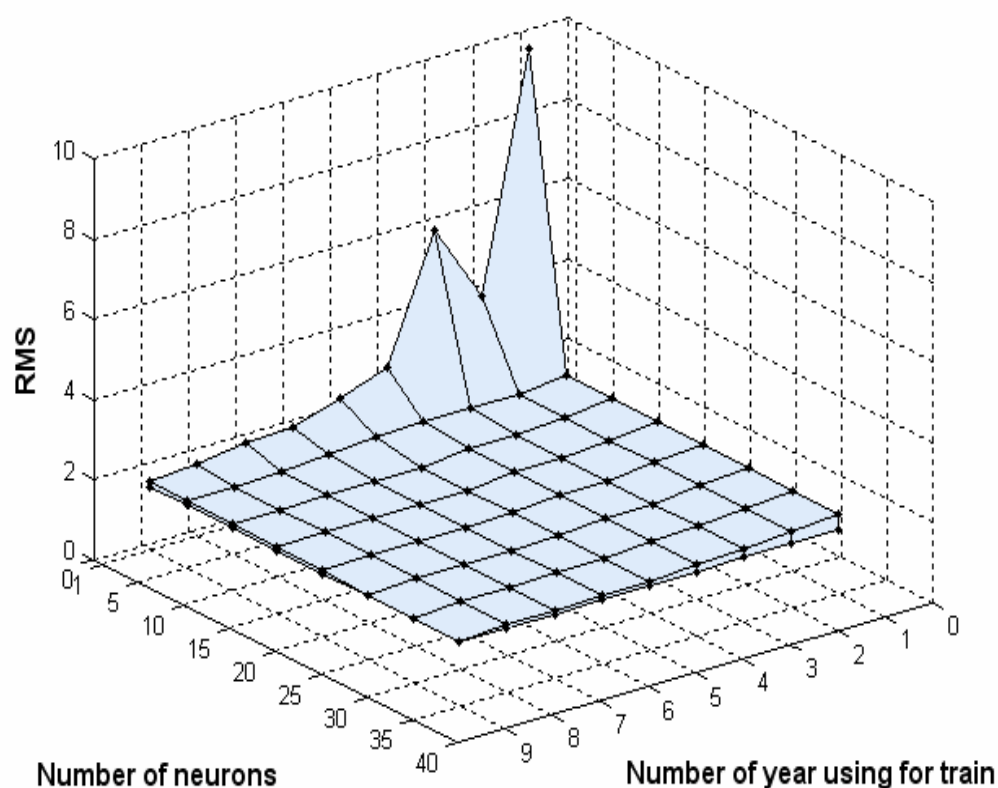
بعد از بررسی نتایج بدست آمده (پیش بینی برای سال 2005) با آنچه که در سال 2005 اتفاق افتاده بود متوجه شدیم که در بعضی از روزها خطای زیادی داریم و بعد از بررسی متوجه شدیم که باید عامل دیگری را در ورودی لحاظ کنیم که این عامل پارامتر ساعت آفتابی بود با اعمال این پارامتر به نتایج قابل قبولی دست یافتیم که در ادامه نتایج را بررسی می کنیم. در نهایت اطلاعات ورودی به فرم زیر برای شبکه تعریف شد.



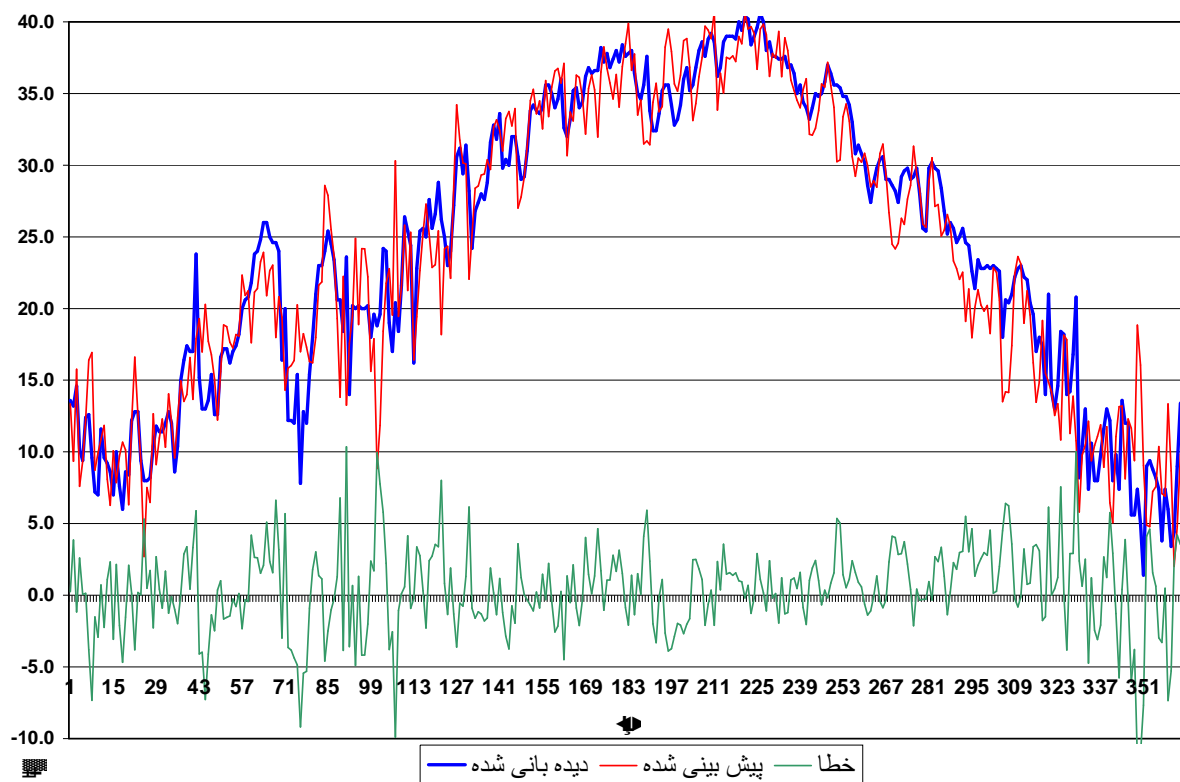
Data Packet = { Temperature, Pressure, Humidity, Wind Speed, Wind Direction, Atmospheric Phenomena, Cloud Covering }



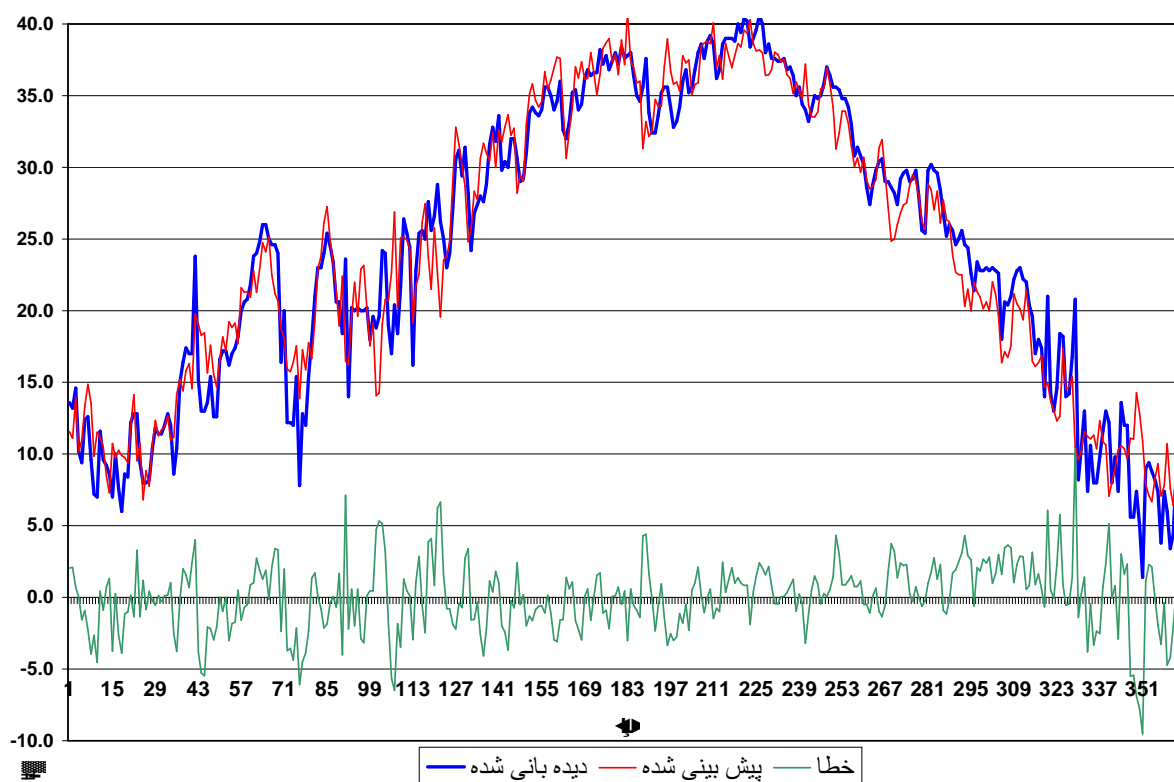
اولین نتیجه ای که بدست آمد یافتن تعداد نرونها در لایه میانی و تعداد سالهای مورد نیاز برای آموزش شبکه بود که این کار با استفاده از آزمودن تعداد نرونها مختلف به ازای تعداد سالهای متفاوت بدست آمد که نتایج در شکلهای زیر بیانگر روند بدست آوردن این اطلاعات می باشد.



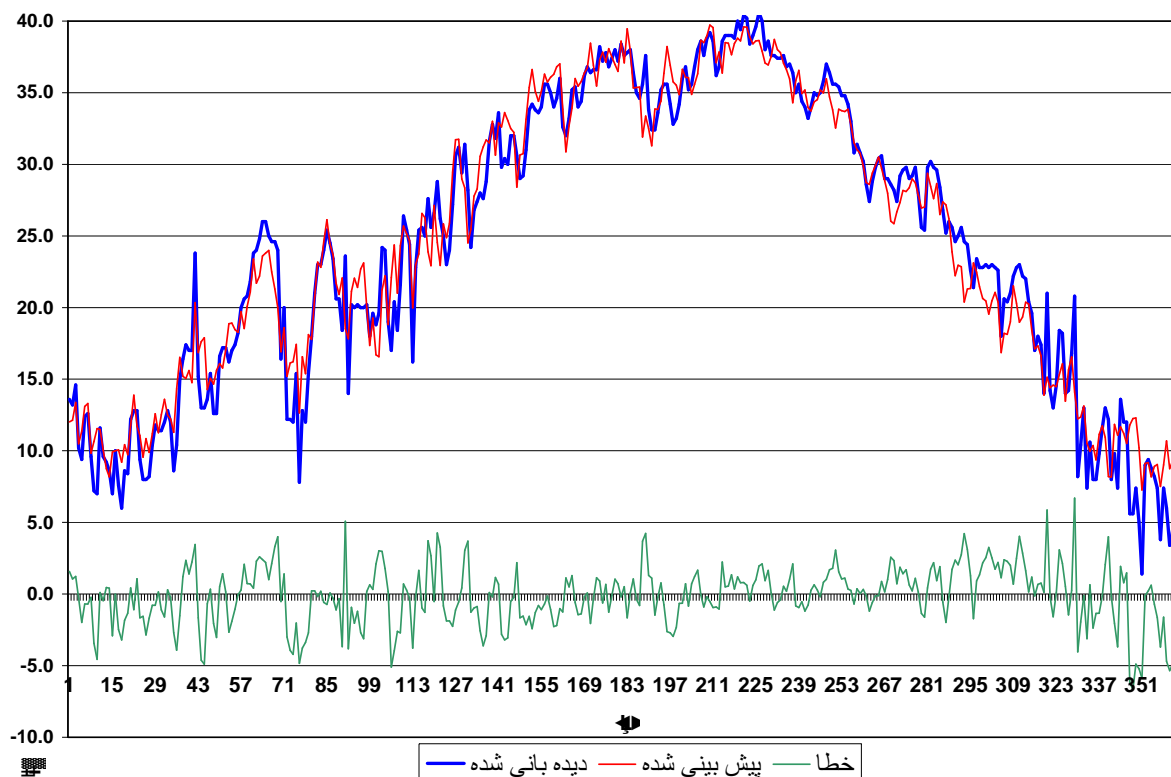
مقدار انحراف معیار خطای دمای مشاهده شده و پیش‌بینی شده در سال 2005 به ازای تعداد نرون‌های لایه مخفی و تعداد سال‌های آموزش داده شده متفاوت.



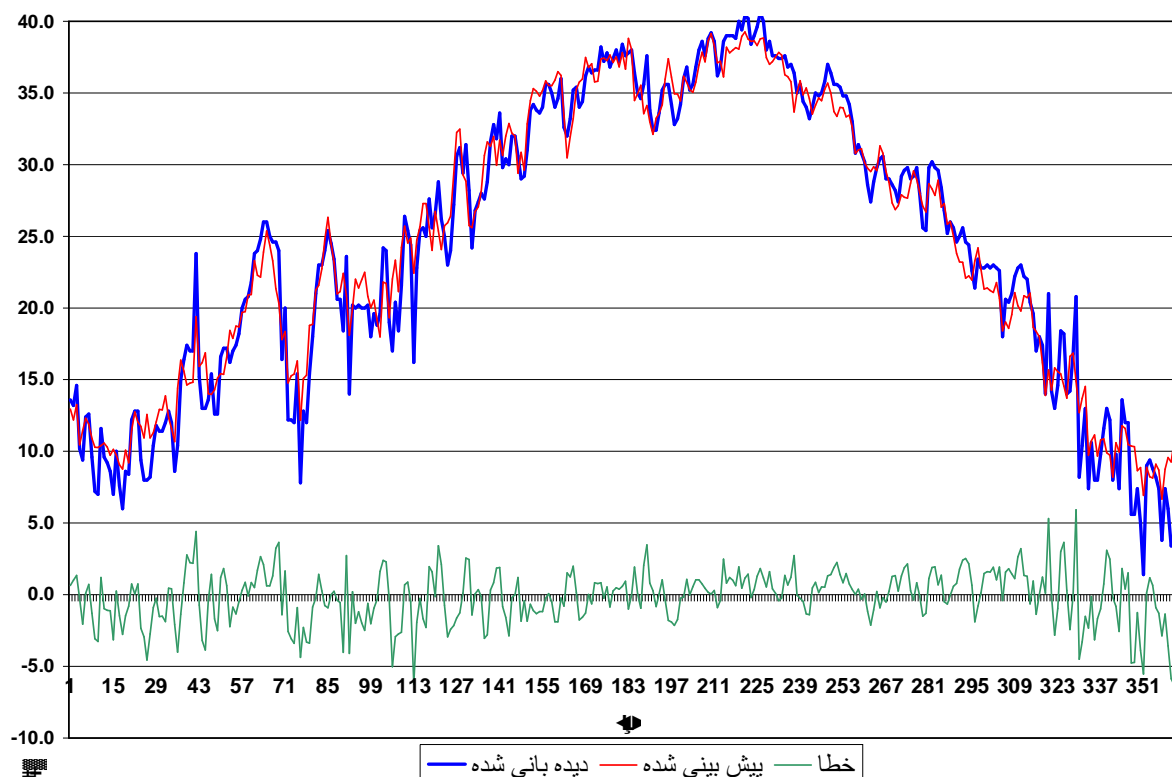
مقایسه دمای بیشینه واقعی سال 2005 با دمای پیش بینی شده توسط شبکه عصبی با 40 نرون در لایه میانی و آموزش يك ساله 2004.



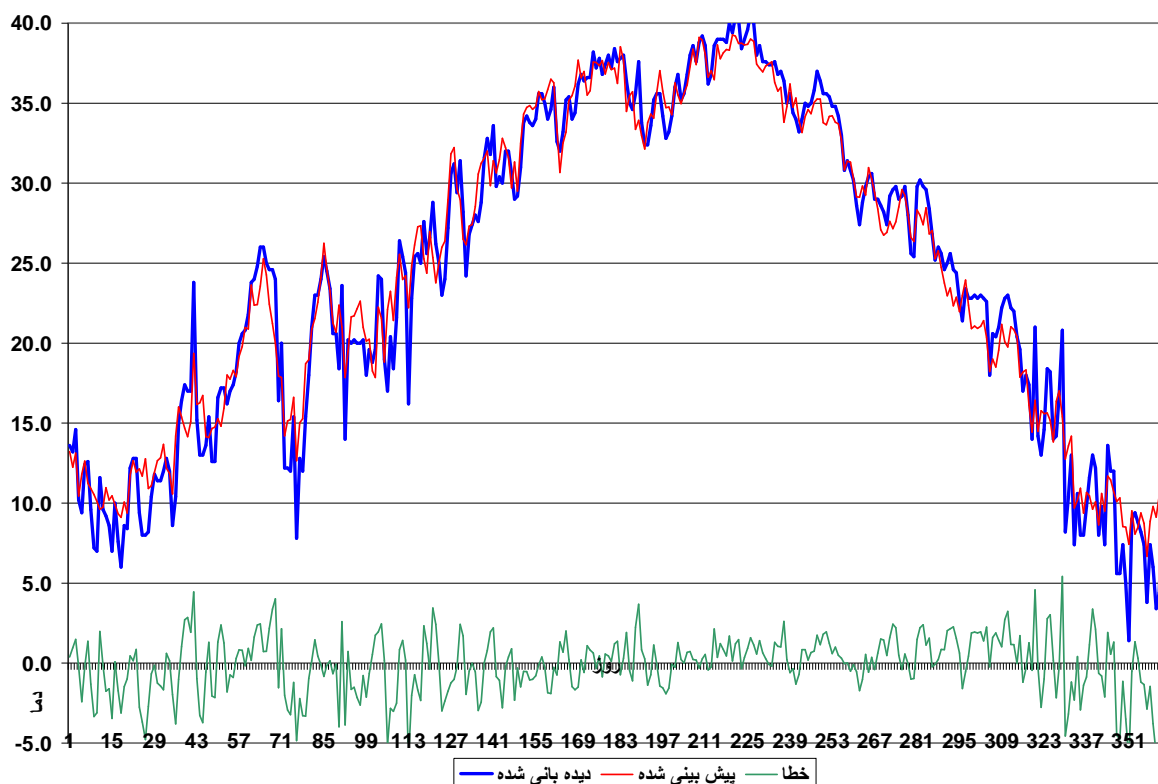
مقایسه دمای بیشینه واقعی سال 2005 با دمای پیش بینی شده توسط شبکه عصبی با 40 نرون در لایه میانی و آموزش دو ساله 2003 الی 2004.



مقایسه دمای بیشینه واقعی سال 2005 با دمای پیش بینی شده توسط شبکه عصبی با 40 نرون در لایه میانی و آموزش سه ساله 2002 الی 2004.



مقایسه دمای بیشینه واقعی سال 2005 با دمای پیش بینی شده توسط شبکه عصبی با 40 نرون در لایه میانی و آموزش هفت ساله 1998 الی 2004.



مقایسه دمای بیشینه واقعی سال 2005 با دمای پیش بینی شده توسط شبکه عصبی با 20 نرون در لایه میانی و آموزش هفت ساله 1998 الی 2004.

در جدول زیر هم میزان انحراف از معیار به ازای سالهای مختلف و تعداد نرونهاى مختلف مشخص شده که خود گویای نتایج حاصل از این تحقیق می باشد.

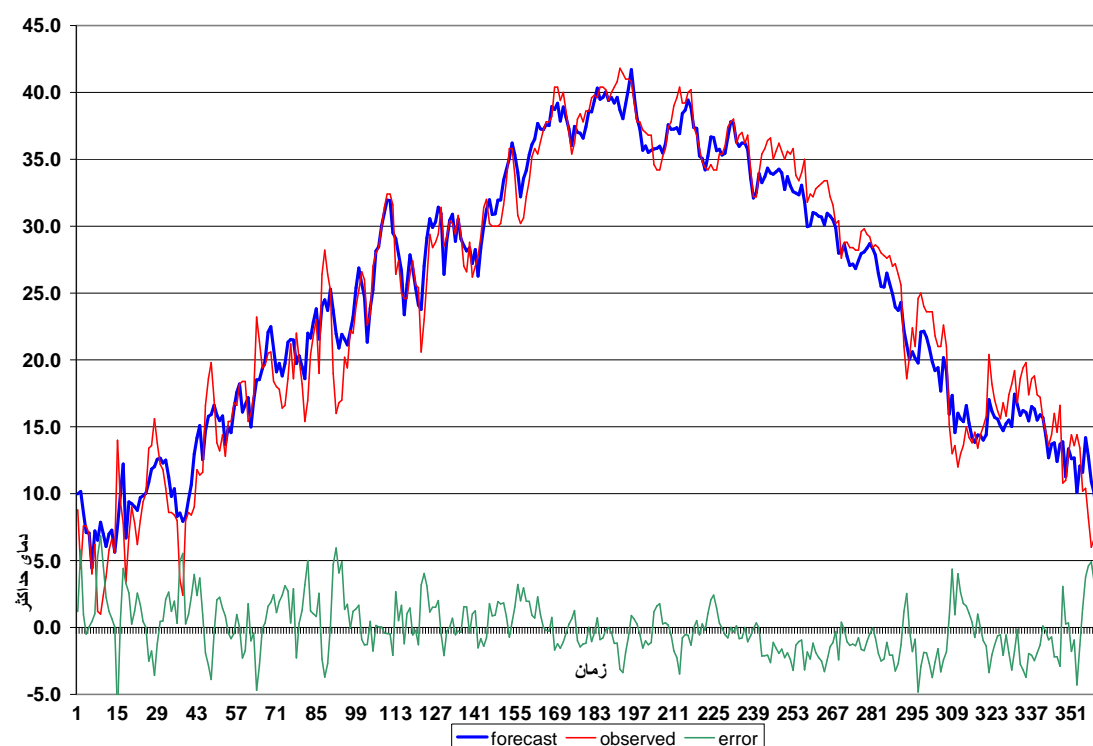
مقدار انحراف معیار خطا پیش‌بینی عددی با اطلاعات مشاهده شده در سال 2005

با دوره‌های آموزش‌های چند ساله و تعداد نرونهاى لایه میانی متغیر

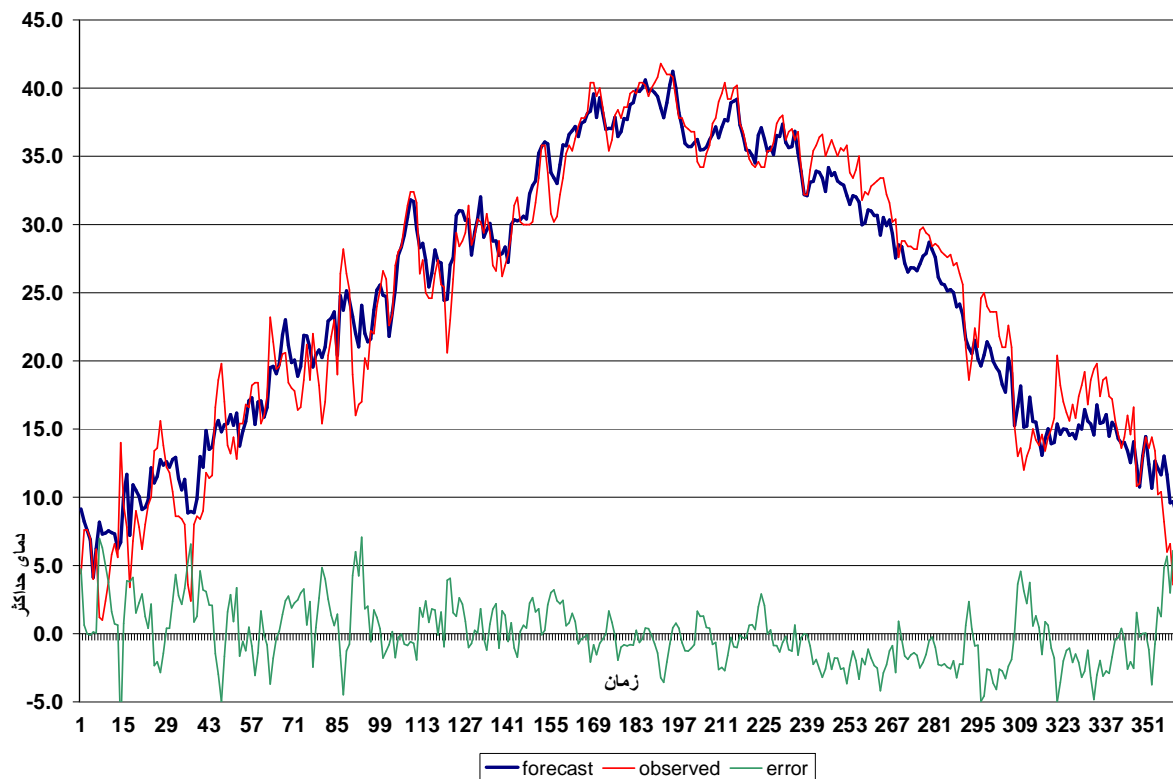
آموزش ← تعداد نرون ↓	یکساله 2003	دوساله 2002- 2003	سه ساله 2001- 2003	چهارساله 2000-2003	پنج ساله 1999- 2003	شش ساله 1998- 2003	هفت ساله 1997- 2003
5	8.07	6.28	4.92	4.02	3.81	3.54	3.02
10	5.01	4.21	3.46	2.86	2.72	2.46	2.29
15	4.92	3.56	2.91	2.68	2.54	2.37	2.24
20	4.68	3.14	2.73	2.54	2.46	2.30	2.18
25	4.43	2.94	2.57	2.41	2.37	2.24	2.13
20	4.21	2.69	2.46	2.32	2.28	2.19	2.09
30	3.56	2.52	2.36	2.21	2.20	2.11	2.03
35	3.12	2.46	2.14	2.08	2.08	2.02	1.93
40	3.08	2.44	2.07	1.99	1.98	1.91	1.86

هنگامی که 40 نرون در لایه میانی داشته باشیم و دوره آموزش 7 سال باشد میزان انحراف معیار $1/86$ بوده که در مقایسه با سایر روشها مقدار مناسبی به نظر می‌آید همچنین تعداد روزهایی که مقدار خطا حداکثر دما پیش‌بینی شده و دمای مشاهده شده حداکثر $\pm 2^{\circ}\text{C}$ است 83% کل ایام سال می‌باشد که این خود نیز در قیاس با سایر متدهای پیش‌بینی بسیار مناسب می‌باشد.

در این تحقیق همچنین پیش‌بینی به ازای 4 روز گذشته و 5 روز گذشته هم صورت پذیرفت که علارقم افزایش میزان زمان پردازش نتایج بهبود نیافت (شکلهای زیر).



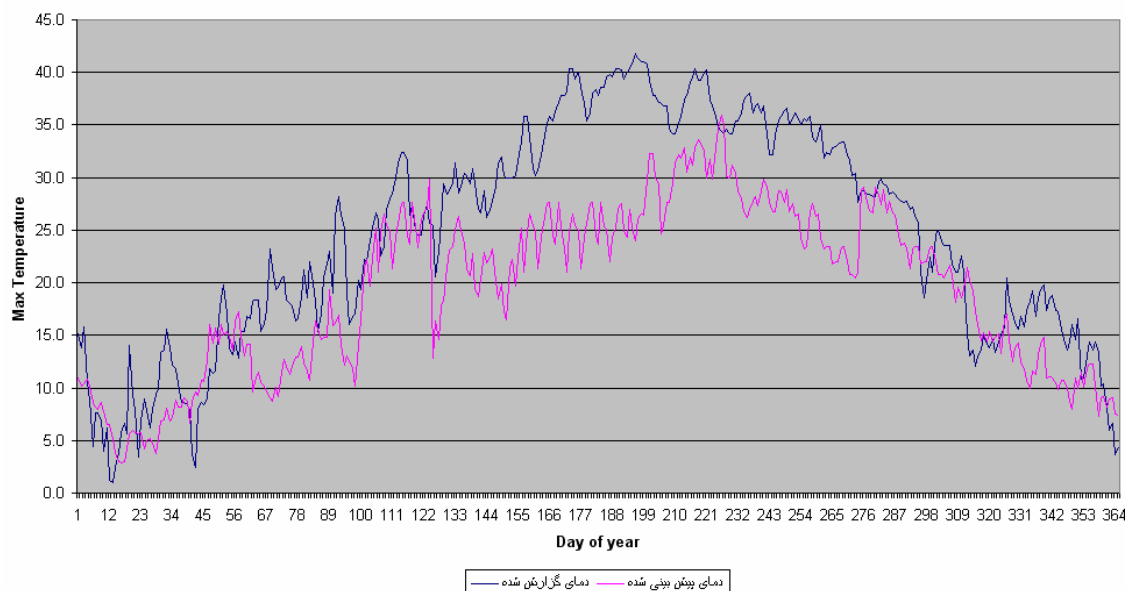
مقایسه دمای پیش‌بینی شده با دیده‌بانی شده با استفاده از سابقه 4 روز گذشته و 35 نرون لایه مخفی.



مقایسه دمای پیش‌بینی شده با دیده‌بانی شده با استفاده از سابقه 5 روز گذشته و 30 نرون لایه مخفی.

همچنین از روش Short term training هم برای آموزش استفاده کردیم که نتایج قابل قبول نبود. در این مرحله از آموزش یک هفته ای استفاده شده است (شکل زیر).

نمودار مقایسه دمای گزارش شده از ایستگاه شرق اصفهان با دمای پیش‌بینی شده توسط شبکه عصبی در short term training با آموزش یک هفته ای



مقایسه دمای گزارش شده با دمای پیش‌بینی شده با استفاده از سابقه 3 روز گذشته و 40 نرون لایه مخفی با آموزش یک هفته ای.

نتایج آزمایشات پس از آموزش شبکه کارایی آن را تایید می‌نماید. افزودن پارامتر پدیده جوی در شبکه عصبی تاثیر به سزایی در اصلاح مقدار موثر خطا بدلیل وقوع پدیده‌ها ندارد. بدلیل آن که میزان وقوع این پدیده‌ها بسیار کم بوده و ضرایب این ورودی‌ها بسیار کوچک می‌باشد. پیشنهاد می‌شود که با استفاده از منطق فازی و یا شبکه‌های نروفازی میزان خطاهای ایجاد شده به هنگام وقوع پدیده‌های جوی را کاهش دهیم.

9-2 مراجع

- 1- رحيم زاده ف.، "تحليل سري هاي زماني با داده هاي گمشده بوسيلة صافي كالمن"، پايان نامه كارشناسي ارشد دانشگاه شهيد بهشتي تهران، 1370.
- 2- پدram م. ن.، ابوفاضلي، "پيش بيني ديناميكي - آماري دماي ماكزيم مينيمم شهر تهران"، مجموعه مقالات هفتمين كنفرانس آمار ايران دانشگاه علامه طباطبائي تهران، 1383.
- 3- آزادي، م.، 1380، مروري بر فرآيند پيش بيني عددي وضع هوا، مجموعه مقالات اولين كنفرانس پيش بيني عددي وضع هوا، سازمان هواشناسي كشور.
- 4- A. Jain, "Frost prediction using artificial neural networks: A temperature prediction approach," M.S. thesis, Artificial Intelligence Center, University of Georgia, Athens, GA, 2003.
- 5- Brian A. Smith, Ronald W. McClendon, and Gerrit Hoogenboom; "Improving Air Temperature Prediction with Artificial Neural Networks". International Journal of Computational Intelligence: 3(3) , 2006 , pp179-186. online , [www.griffin.uga.edu/aemn/Smith McClendon Hoogenboom.pdf](http://www.griffin.uga.edu/aemn/Smith_McClendon_Hoogenboom.pdf).
- 6-Krishnamurti, T. N. & L. Bounoua, 1996 : An Introduction to Numerical Weather Prediction Techniques, CRC Press, Inc., 293 pp.
- 7- Daniel S. Wilks; Thomas M. Hamil; "Comparison of Ensemble-MOS Methods Using GFS Reforecasts". Mon. Wea. Rev., vol 125, JUN 2007, pp 527-548.
- 8-Kohzady, N., M.S. Boyd, I. Kaastra, B.S. Kermanshahi, D. Scuse, 1995. Neural networks for forecasting: an introduction, Canadian Journal of Agricultural Economics, 43, 463-474.
- 9-Kartalopoulos, S.V., 1996. Understanding neural networks and fuzzy logic. Basic Concepts and Applications. IEEE Press, New York, pp. 205.
- 10-Patterson, D.W., 1996. Artificial Neural Networks: theory and applications. Simon & Schuster, Singapore, pp. 477.

11-Bachisio Arca¹, Fabrizio Benincasa , Matteo De Vincenzi and Gianni Fasano;” A Neural Model to Predict the Daily Minimum of Air Temperature”, 7th CCTA - International Congress For Computer Technology in Agriculture, "Computer technology in agricultural management and risk prevention" 15-18 novembre 1998, Firenze pp 485-493.

12-Müller, B., and J. Reinhardt, 1991: *Neural Networks: An Introduction*. Vol. 2, the Physics of Neural Networks Series, Springer-Verlag, 266 pp.

13-Hornik, K. M.; Stinchcombe, M.; & White, H.;"Multilayer Feed forward Networks are Universal Approximators", Neural Networks, vol. 2 , 1989.

14- L. Bodri, V. Cermák; “Prediction of Surface Air Temperatures by Neural Network, Example Based on Three-Year Temperature Monitoring at Spořilov Station” , Studia Geophysica et Geodaetica , Vol 47, Number 1 , JAN 2003, pp173-184

15- Hippert, H.S.; Pedreira, C.E.; Souza, R.C. “Combining neural networks and ARIMA models for hourly temperature forecast”, Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. Volume 4, Issue, 2000 Page(s):414 - 419 vol.4

16- Ramyaa, “Frost prediction using artificial neural networks: A classification approach,” M.S. thesis, Artificial Intelligence Center, University of Georgia, Athens, GA, 2004.

17-Christoph C. Raible,Gorge Bischof,Klaus Fraedrich, etal; “Statistical Single-Station Short-term Forecasting of Temperature and Probability of Precipitation: Area Interpolation and NWP Combination”. Weather and Forecasting, vol 14, Apr 1999, pp 203–214.

18-Kenneth F. Dewey;“Daily Maximum and Minimum Temperature Forecasts and the Influence of Snow Cover”, Mon. Wea. Rev., vol 117 , Dec 1977 , pp 1594-1597.