مهندسی شبکه های عصبی توسط اتوماتانهای یادگیر: تعیین اندازه مطلوب برای شبکه های عصبی سه لایه

محمد رضا میبدی حمید بیگی

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیر کبیر
تهران، ایران

چکیده

هدف از مهندسی شبکه های عصبی بررسی معایب و مزایای شبکه های عصبی مصنوعی و ارایه روشهایی برای بهبود کارایی آنهاست. یکی از موضوعات مورد بحث در مهندسی شبکه های عصبی چندلایه، یافتن ساختار مناسب (نزدیک به بهینه) برای حل مسئله میباشد. معیار و نحوه انتخاب اندازه شبکه عصبی برای یک مسئله خاص هنوز شناخته شده نیست. در روشهای کلاسیک، طراح شبکه در ابتدای آموزش ساختاری را برای شبکه تعیین و سپس شبکه را آموزش میدهد و ساختار تعیین شده در حین آموزش ثابت نگاه داشته میشود. تعیین اندازه شبکه عصبی (تعداد واحدها و لایه های مخفی) تاثیر عمده ای بر روی کارایی شبکه های عصبی خواهد داشت. طراحی یک شبکه با ساختار بهینه یک مسئله ذاتا مشکل است. بهمین جهت بیشتر الگوریتم های ارایه شده برای تعیین ساختار شبکه های عصبی، الگوریتم های تقریبی هستند. این الگوریتم ها را میتوان به پنج گروه عمده الگوریتم های هرس ، الگوریتم های سازنده ، الگوریتم های ترکیبی ، الگوریتم های تکاملی و الگوریتم های بر اساس اتوماتانهای یادگیر تقسیم کرد. تنها الگوریتم گزارش شده براساس اتوماتان یادگیر، الگوریتم بقا نام دارد که توسط بیگی و میبدی ارایه گردیده است. این الگوریتم از یک اتوماتان یادگیر مهاجرت اشیا (بعنوان یک ابزار جستجوی عمومی) و الگوریتم یادگیری انتشار خطا به عقب استفاده میکند و در ضمن آموزش، تعداد واحدهای مخفی یک شبکه سه لایه را تعیین مینماید. در این مقاله سه الگوریتم بر اساس اتوماتانهای یادگیر مهاجرت اشیا و الگوریتم یادگیری انتشار خطا به عقب ارایه شده است که ساختار مناسب شبکه عصبی سه لایه (ساختاریکه دارای اندازه کوچک، پیچیدگی آموزش کم و قدرت تعمیم بالا باشد) را برای یک مجموعه از الگوهای آموزشی تعیین می نماید. الگوریتم های ارایه شده داری دو قسمت تعیین تعداد واحدهای مخفی مورد نیاز و تعیین تعداد وزنهای مورد نیازشبکه می‌باشند. در یکی از الگوریتم‌های پیشنهادی برای تعیین تعداد واحد های مخفی از الگوریتم بقا استفاده گردیده است. برای تعیین تعداد وزنها الگوریتم جدیدی با استفاده از اتوماتان یادگیر مهاجرت اشیا طراحی گردیده است. این الگوریتم با حذف وزنهایی که دارای اثر کمتری هستند نه تنها باعث کاهش پیچیدگی شبکه خواهد شد بلکه افزایش تعمیم شبکه را هم بدنبال خواهد داشت. در دو الگوریتم دیگر تعداد واحدهای مخفی بصورت صریح تعیین نمی‌گردند بلکه تعیین تعداد واحدهای مخفی از طریق تعیین تعداد وزنهای لایه مخفی انجام میگیرد. اغلب روشهای گزارش شده برای تعیین ساختار شبکه از الگوریتم های کوهنوردی استفاده میکنند و مشکل گرفتاری در حداقل محلی را دارند. در روش پیشنهادی در این مقاله بدلیل استفاده از روشهای جستجوی عمومی، امکان گرفتاری در حداقل های محلی فضای ساختارها کاهش می یابد. الگوریتم های ارایه شده روی مسائل متنوعی از جمله شناسایی اعداد فارسی و شناسایی اعداد انگلیسی آزمایش شده اند. از طریق شبیه سازی نشان داده شده است که شبکه های تولید شده توسط الگوریتم های پیشنهادی دارای کارایی بالایی هستند. همچنین الگوریتم های پیشنهادی با الگوریتم هرس کارنین مقایسه گردیده است. نتایج الگوریتم های پیشنهادی هم از لحاظ ساختار تولیدشده، هم از نظر زمان مورد نیاز برای تعیین ساختار شبکه عصبی، بسیار بهتر از الگوریتم هرس کارنین عمل کرده‌اند.

کلمات کلیدی: مهندسی شبکه‌های عصبی، شبکه‌های عصبی چندلایه، ساختار شبکه‌های عصبی، اتوماتانهای یادگیر مهاجرت اشیا

۱- مقدمه

هدف از مهندسی شبکه‌های عصبی بررسی معایب و مزایای شبکه‌های عصبی مصنوعی و ارایه روشهایی برای بهبود کارایی آنهاست. یکی از موضوعات مورد بحث در شبکه‌های عصبی چندلایه یافتن ساختار مناسب (نزدیک به بهینه) برای حل مسئله میباشد. در روشهای کلاسیک، طراح شبکه در ابتدای آموزش ساختاری را برای شبکه تعیین و سپس شبکه را آموزش میدهد. ساختار تعیین شده در حین آموزش ثابت نگاه داشته میشود. تعیین تعداد واحدهای لایه مخفی و وزنهای آن تاثیر بسزایی بر روی کارایی شبکه دارد. یک

شبکه‌بااندازه‌کوچک، خروجی‌دقیق‌تولیدنمیکندوشبکه‌ای‌بااندازه‌بزرگ، بسیارکندوپر هزینه‌خواهدبودوبرای‌تعمیم‌مجموعه‌آموزشی، نیازبه مجموعه‌آموزشی بزرگی‌خواهدداشت. طراحی‌یک‌شبکه‌باساختاربهینه‌یک مسئله NP-Hard است[۱۴]. بهمین‌جهت‌بیشترالگوریتمهای‌ارائه‌شده برای‌تعیین‌ساختارشبکه‌های‌عصبی، الگوریتمهای تقریبی هستند. این الگوریتم ها قبل ، در حین یا بعد از یادگیری ساختار مناسب برای شبکه را تعیین می نمایند. بعضی از این الگوریتم ها از اطلاعات محلی و بعضی دیگر از اطلاعات عمومی برای یافتن ساختار مناسب شبکه استفاده میکنند. این الگوریتم ها را میتوان به پنج گروه عمده زیر تقسیم کرد.

الف) الگوریتم های هرس[1]: این الگوریتم ها از یک شبکه بزرگ شروع نموده و بتدریج در حین آموزش یا بعد از آن واحد ها یا وزنهای اضافی را از شبکه هرس می کنند. در یک دسته از الگوریتمهای هرس یک جمله جریمه به تابع هدف شبکه افزده میشود و باعث حذف وزنهای کم اهمیت تر در حین آموزش شبکه میگردد [۵۵]. در دسته‌ای دیگر از الگوریتم های هرس ابتدا شبکه بطور کامل آموزش داده میشود و در پایان آموزش شبکه عصبی، حساسیت تابع هدف شبکه نسبت به هر وزن یا واحد (تاثیر حذف یک وزن یا واحددر تابع هدف شبکه) محاسبه می‌گردد و وزنها یا واحدهایی حذف میگردند که دارای حساسیت کمتری باشند[۵۴]. در الگوریتم های هرس نیاز است که تعداد واحد های مخفی را در ابتدای آموزش مشخص نماییم. البته این مشکل جادی را ایجاد نمیکند زیرا کران بالای تعداد واحد های مخفی برای یک مسئله معین و برای بعضی از انواع شبکه ها مشخص است[۲][۱۰][۳۵]. این الگوریتم ها هم از مزایای شبکه های بزرگ (کم بودن پیچیدگی آموزش و دوری از حداقل های محلی) و هم از مزایای شبکه های کوچک (قدرت تعمیم بالا) بهره می برند. [۵][۱۱][۱۲][۳۰][۳۱].

ب) الگوریتم‌های‌سازنده[2]: این‌الگوریتمهابایک‌شبکه‌کوچک‌شروع به آموزش وبتدریج درحین‌آموزش‌شبکه، واحد یالایه مخفی به شبکه میافزایند. این‌الگوریتمهامعمولاشبکه‌های‌کوچک‌تولیدمیکنندکه دارای پیچیدگی‌آموزش‌بالاهستند[۳][۶] [۷][۱۳][۱۶][۱۹] [۲۲] [۳۲] [۳۹].

ج) الگوریتم های ترکیبی : این گروه از الگوریتم ها، از ترکیبی از الگوریتم های سازنده و الگوریتم های هرس برای تعیین ساختار شبکه استفاده میکنند. در این الگوریتم ها برای رسیدن به شبکه مطلوب میتوان وزن، واحد یا لایه مخفی را کم یا زیاد نمود[۹][۲۴].

د) الگوریتم های تکاملی[3]: دراین الگوریتم ها تعیین ساختار بهینه برای شبکه از طریق جستجو در فضای ساختارها انجام میگیرد. هر نقطه از این فضا نماینده یک ساختار شبکه است. الگوریتم جستجو با استفاده از یک معیار کارایی مانند حداقل خطا و یا پیچیدگی آموزش بدنبال مناسبترین ساختار میباشد[۱][۱۵][۳۳][۳۷][۳۸].

هـ) الگوریتم های بر اساس اتوماتانهای یادگیر[4]: تنها الگوریتم گزارش شده که براساس اتوماتان های یادگیر طراحی شده است الگوریتم‌بقانام‌داردکه‌توسط‌بیگی‌ومیبدی [۴۳] ارایه‌گردیده‌است. این الگوریتم‌ازیک‌اتوماتان‌یادگیرمهاجرت‌اشیا[5] بعنوان‌یک‌ابزارجستجوی عمومی‌والگوریتم‌یادگیری‌انتشارخطابه‌عقب‌استفاده‌میکندودرضمن آموزش، ساختارمناسبی‌برای‌شبکه‌عصبی‌سه‌لایه(ساختاریکه‌دارای‌اندازه کوچک، پیچیدگی آموزش کم و قدرت تعمیم بالا باشد) تعیین مینماید. در الگوریتم بقا، آموزش از یک شبکه عصبی‌سه‌لایه‌بزرگ‌شروع شده واتوماتان یادگیربا افزودن و کاستن واحدهای مخفی، تعدادواحدهای‌لایه‌مخفی‌این شبکه را تعیین‌میکند. اغلب‌روشهای گزارش‌شده‌برای‌تعیین‌ساختارشبکه‌از الگوریتم های کوهنوردی استفاده میکنند[۱] و مشکل گرفتاری در حداقل محلی را دارند. در الگوریتم بقا بدلیل استفاده از روشهای جستجوی عمومی(اتوماتانهای یادگیر)، امکان گرفتاری در حداقل های محلی فضای ساختارها کاهش می یابد. الگوریتم بقا بعدا در این مقاله بطور خلاصه شرح داده خواهد شد.

برای‌اطلاعات‌بیشتردرموردالگوریتمهای‌فوق‌می‌توانیدبه‌مرجع [۴] مراجعه نمایید.

در این مقاله سه الگوریتم بر اساس اتوماتانهای یادگیرمهاجرت اشیا و الگوریتم یادگیری انتشار خطا به عقب ارایه شده است که ساختار مناسب برای شبکه عصبی سه لایه را برای یک مجموعه از الگوهای آموزشی تعیین می نماید. الگوریتم های ارایه شده داری دو قسمت میباشند: تعیین تعداد واحدهای مخفی مورد نیاز و تعیین تعداد وزنهای مورد نیازشبکه. در الگوریتم۱، برای تعیین تعداد واحد های مخفی از الگوریتم بقا استفاده گردیده است. الگوریتم‌بقا تعداد واحد های لایه میانی شبکه را که قدرت تعمیم قابل قبولی را ارایه و قادر به مدلسازی مسئله گردد تعیین می کند. برای تعیین تعداد وزنها الگوریتم جدیدی با استفاده از اتوماتان یادگیری مهاجرت اشیا طراحی گردیده است. این الگوریتم با حذف وزنهایی که دارای اثر کمتری هستند نه تنها باعث کاهش پیچیدگی شبکه خواهد شد بلکه افزایش تعمیم شبکه را هم بدنبال خواهد داشت. الگوریتم تعیین وزنها با خاموش و روشن کردن وزنها سعی در پیدا کردن وزنهای مناسب مینماید. در دو الگوریتم دیگر هدف فقط تعیین تعداد وزنهای لایه مخفی شبکه است و تعداد واحدهای لایه مخفی را از طریق تعیین تعداد وزنهای شبکه مشخص می‌گردد. این دو الگوریتم با بررسی های محلی یا عمومی وزنهای لایه مخفی سعی در پیدا کردن ساختار مناسب برای شبکه را دارند.

بخشهای‌بعدی‌مقاله‌بصورت‌زیرسازماندهی‌شده‌است. رابطه‌بین‌ساختار شبکه‌های‌عصبی‌چند لایه وشکل‌تابع‌تمایزدربخش۲ موردمطالعه‌قرار گرفته‌است. این‌مطالعه‌در بررسی‌الگوریتمهای‌ارایه‌شده دراین‌مقاله‌مورد استفاده قرارخواهدگرفت. سپس‌دربخش ۳ معیارهای کارایی‌شبکه‌را بررسی‌مینماییم. مقدمه‌ای‌بر اتوماتانهای‌یادگیروالگوریتم بقا دربخش ۴ آمده‌است. الگوریتمهای پیشنهادی‌دربخش۵ بیان‌شده‌اند. نتایج آزمایشها برای‌مسائل مختلف در بخش ۶ ودرپایان‌نتیجه‌گیری‌آورده شده است.

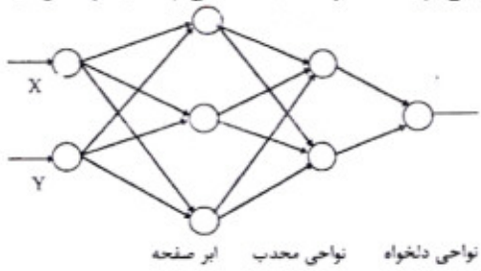
۲- رابطه‌بین ساختار شبکه عصبی و شکل تابع تمایز

یکی‌ازروشهای‌تعیین‌تابع‌تمایز، استفاده‌ازشبکه‌عصبی‌چندلایه‌است. یک شبکه‌عصبی‌چندلایه‌قادراست‌تاهرنگاشت‌غیرخطی(تابع تمایز) راازروی نمونه‌های‌آموزشی‌تقریب‌بزند. کیفیت‌دسته بندی، تعدادوشکل‌تابع‌تمایز

1- Pruning Algorithms
2- Constructive Algorithms
3- Evolutionary Algorithms
4- Learning Automata
5- Object Migrating Learning Automata

تعیین شده توسط این روش بستگی به تعداد لایه های مخفی، تعداد واحد هادر هر لایه مخفی، شکل تابع فعالیت واحدها، وزنها و غیره دارد.

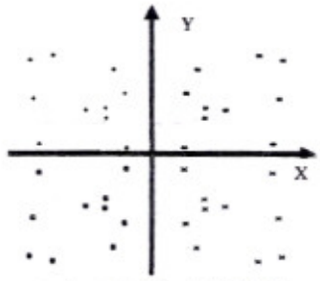
در یک شبکه عصبی چند لایه با n ورودی $X = \{x_1, x_2, ..., x_n\}$ را در که واحدهای آن دارای تابع فعالیت f(.) باشند هر واحد در لایه مخفی اول یک ابر صفحه با معادله $P_k = f(\sum_{i=1}^{n} x_i w_{ik} + b_k)$ تولید میکنند که w_{ik} وزن اتصال بین واحد ورودی x_i و واحد مخفی شماره k و b_k مقدار آستانه این واحد میباشد. واحد های موجود در لایه مخفی دوم با استفاده از ابرصفحه های تولید شده توسط واحد های لایه مخفی اول، نواحی محدبی تولید میکنند که ممکن است باز یا بسته باشند مانند ناحیه داخل مربع یا مثلث یا ناحیه بین دو خط موازی. اضافه نمودن واحد به لایه مخفی اول سبب افزایش تعداد لبه های موجود در مرز نواحی محدب تولید شده توسط این لایه میگردد. بنابر این تعداد لبه های موجود در مرز هر ناحیه حداکثر برابر با تعداد واحد های لایه مخفی اول است. بسیاری از این ابرصفحه‌ها تقریبا موازی و نزدیک بهم هستند و اطلاعات تقریبا یکسانی در وزنهای آنها ذخیره شده است که می‌توانند براحتی حذف شوند. اگر یک لایه دیگر به شبکه اضافه شود واحد های لایه جدید با استفاده از نواحی محدب تولید شده توسط واحد های لایه مخفی دوم، ترکیبی از این نواحی را (اشتراک، اجتماع و ...) تولید میکنند که الزاما محدب نیستند و میتوانند هر شکل دلخواهی به خود بگیرند. پیچیدگی شکل ناحیه های تولید شده به تعداد واحد های موجود در هر لایه مخفی از شبکه بستگی دارد. بنابراین با استفاده از یک شبکه عصبی چهار لایه (با احتساب لایه ورودی) میتوان ناحیه هایی با هر شکلی را تولید نمود. با توجه به اینکه مرز این نواحی همان تابع تمایز میباشد در نتیجه تعداد لایه های مخفی و تعداد واحددر هر لایه مخفی از شبکه ، شکل تابع تمایز را مشخص میکند. اگر تابع فعالیت واحدها، تابع پله‌باشدمرزاتصال بین دوابرصفحه‌درهرناحیه‌شکسته میگردد مانندرئوس مربع یامثلث. برای رفع شکستگی در مرز نواحی میتوان از تابع سیگموئیدبرای تابع فعالیت واحدها استفاده کرد. استفاده از تابع سیگموئیدبرای تابع فعالیت واحدهاباعث هموار شدن مرز نواحی تولید شده میگرددهرچندشکل ناحیه‌هاثابت باقی میماند[۴۱]. شکل ۱ خلاصه‌ای از انواع نواحی تولید شده توسط شبکه عصبی چند لایه را نشان میدهد.



شکل ۱: نواحی تولید شده توسط یک شبکه عصبی چهار لایه

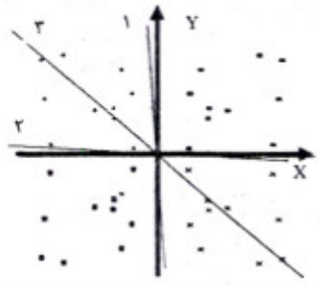
مثال زیر میتواند بعضی از نکات طرح شده فوق را روشن تر نماید.

مثال ۱: فرض کنید که چهار دسته الگو داشته باشیم (که در شکل ۲ توسط نمادهای "+"، "*"، "x" و "#" نشان داده شده‌اند) و هرالگو توسط دو ویژگی x و y نمایش داده شود. شکل ۲ نحوه قرار گرفتن الگوها در صفحه را نشان میدهد.



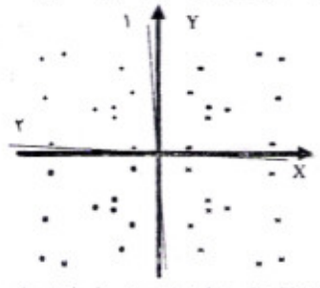
شکل ۲: توزیع الگوها در صفحه

حال فرض کنید که یک شبکه عصبی سه لایه با دو واحد در لایه ورودی، سه واحد در لایه مخفی با تابع فعالیت سیگموئید و چهار واحد در لایه خروجی با تابع فعالیت خطی برای دسته بندی الگوها استفاده گردد. بدلیل اینکه این شبکه دارای دو واحد ورودی است هر واحد در لایه مخفی یک خط (که با افزایش تعداد واحدهای ورودی خط به صفحه یا ابر صفحه تبدیل میگردد) در صفحه تولید میکنند. خطوط تولید شده توسط واحد های لایه مخفی این شبکه در شکل ۳ نشان داده شده است(لازم بذکر است که شبکه فوق توسط الگوریتم انتشار خطا به عقب آموزش داده شده است).



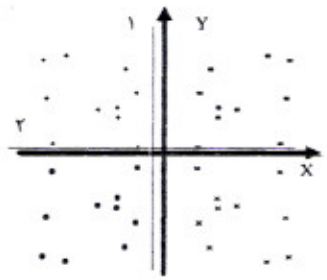
شکل ۳: خط های تولید شده توسط واحد های مخفی

با توجه به اینکه خروجی واحد مخفی ۳ (خطی که با شماره ۳ نشان داده شده است) در جداسازی دسته ها از یکدیگر تاثیری ندارد الگوهای فوق میتوانند توسط یک شبکه عصبی سه لایه که دارای دو واحد ورودی ، دو واحد مخفی و چهار واحد خروجی است دسته بندی گردند. خط های تولید شده توسط واحد های مخفی این شبکه در شکل ۴ نشان داده شده است.



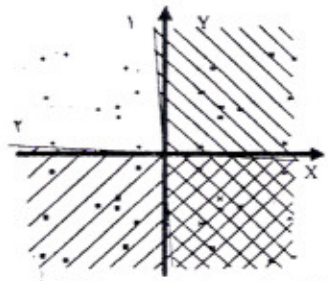
شکل ۴: خط های تولید شده توسط واحد های مخفی

همانطور که در این شکل دیده میشود دو خط ۱ و ۲ برای جداسازی دسته ها از یکدیگر کافی هستند. توجه کنید که بین لایه ورودی و لایه مخفی این شبکه شش اتصال وجود دارد. با توجه اینکه برای دسته بندی الگوها ویژگی y برای خط شماره ۱ و ویژگی x برای خط شماره ۲ چندان مهم نیست. میتوان این الگوها را با یک شبکه عصبی سه لایه با دو واحد ورودی، دو واحد مخفی، چها واحد خروجی و ۴ اتصال بین لایه ورودی و لایه مخفی از هم جدا نمود. خطوط حاصله از این شبکه در شکل ۵ نشان داده شده است.



شکل ۵: خط های تولید شده توسط واحد های مخفی

باتوجه‌به‌مثال‌فوق، تعدادلایه‌های مخفی، تعدادواحدهای مخفی درهر لایه و نوع‌اتصال وزنها، شکل‌نواحی‌تولیدشده‌توسط‌شبکه‌عصبی را مشخص میکنند. اگرتعدادواحدهای‌مخفی‌زیاد باشدتعداد ابر صفحه‌های موجوددر مرزتشکیل‌دهنده‌نواحی‌زیادترمیشوند. وزنهای‌متصل‌به‌یک‌واحد مخفی وابستگی‌ابرصفحه‌تولیدشده‌رابه‌ویژگیهای‌الگوهای ورودی‌تعیین میکنندویابه‌عبارتی‌دیگرتعدادوزنهای‌ورودی‌به‌هرواحدمخفی، ابعادابر صفحه‌تولیدشده‌توسط‌این‌واحدرامشخص میکند. مثلابرای‌شبکه‌ای‌که خطوط‌تولیدشده‌توسط‌واحدهای‌مخفی‌آن‌درشکل ۴نشان‌داده‌شده‌است نواحی‌تشکیل‌شده‌توسط‌شبکه(تابع‌تمایز)درشکل ۶ نشان‌داده‌شده‌است.



شکل ۶: نواحی تولید شده توسط شبکه عصبی

۳- معیارهای کارایی شبکه

در این قسمت رابطه پیچیدگی آموزش و قدرت تعمیم شبکه با ساختار شبکه های عصبی را مورد بررسی قرار میدهیم.

پیچیدگی‌آموزش‌شبکه: پیچیدگی‌آموزش‌شبکه درواقع همان پیچیدگی زمانی‌الگوریتم‌آموزش‌میباشدکه‌بستگی‌به‌کاربردمورد‌نظر، ساختار شبکه عصبی والگوریتم‌یادگیری‌دارد. پیدا نمودن وزنهای مناسب برای‌شبکه‌برای یک مسئله مشخص یک مسئله ذاتا مشکل میباشد ودر گروه‌مسایل NP-Complete قراردارد[۵۰]. بهمین‌دلیل‌برای‌تعیین‌وزنهای شبکه بمنظورحصول‌به‌یک نگاشت‌صحیح، الگوریتمهای تقریبی پیشنهاد شده‌است که‌الگوریتم‌یادگیری‌انتشار خطابه‌عقب که‌براساس‌جستجودر امتدادگرادیان‌بناشده‌است‌یک‌نمونه‌از‌آنهامیباشد. الگوریتمهای‌جستجودر امتدادگرادیان‌دارای‌زمان‌اجرایی‌ازمرتبه‌نمایی‌نیستند امابسیارکندعمل میکنند. یک‌دلیل‌برای‌کندی‌این‌الگوریتمهاشکل خاص‌تابع‌موردجستجو (مجموع مربعات خطا) میباشد. بنابراین‌پیچیدگی‌آموزش‌یک‌شبکه به‌شکل‌سطح‌خطاوابسته است که‌خودبه‌ساختار شبکه‌عصبی بستگی دارد [۵۱]. هنوزشکل‌سطح‌خطا برای انواع شبکه ها بررسی نشده است. تنها نتیجه ای که در این رابطه در دسترس است برای شبکه های سه لایه میباشد. اثبات شده است که اگر تعداد واحدهای مخفی در یک شبکه عصبی‌سه‌لایه به اندازه یک واحد کمتر از تعداد داده های آموزشی انتخاب شود سطح خطای این شبکه بدون حداقل محلی خواهد بود[۴۰]. متاسفانه‌ازاین‌نتیجه‌درعمل نمیتوان استفاده نمود زیرا شبکه تولید شده بسیار بزرگ ودارای قدرت تعمیم بسیار کم خواهد بود.

قدرت‌تعمیم‌شبکه : قدرت‌تعمیم‌شبکه‌معیاریست که مشخص میکند یک شبکه‌پس‌ازاتمام‌دوره‌آموزش‌روی‌داده‌هایی که‌درمجموعه‌آموزشی موجود نبوده‌اندچگونه‌عمل‌میکند. هرچه‌تعداد پاسخهای‌مناسب شبکه به اینگونه ورودیهابیشترباشدقدرت‌تعمیم‌شبکه‌بیشترخواهد بود. قدرت تعمیم‌یک شبکه‌برای‌یک‌مسئله‌به‌سه‌عامل‌اندازه‌وکارایی مجموعه آموزشی، معماری‌و ساختارشبکه‌وپیچیدگی‌فیزیکی‌مسئله‌بستگی‌دارد. از بین سه‌عامل‌فوق‌طراح‌هیچ‌نوع کنترلی‌روی‌عامل پیچیدگی‌فیزیکی‌مسئله‌ندارد. بنابراین میتوان‌قدرت‌تعمیم‌شبکه‌راازدونقطه‌نظرمورد بررسی قرار داد.

الف)ساختار شبکه را ثابت در نظر گرفته و با تعیین اندازه مناسب برای مجموعه آموزشی قدرت تعمیم قابل قبولی را برای شبکه ایجاد کرد.

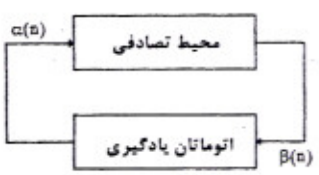
ب)اندازه مجموعه آموزشی را ثابت در نظرگرفته و بدنبال ساختار مناسب (تعداد لایه ها، تعداد واحد های هر لایه، محدودیت روی مقدار وزنها و غیره) برای حصول به قدرت تعمیم قابل قبول بود.

در این‌مقاله‌مسئله‌راازنقطه‌نظردوم‌بررسی‌میکنیم وسعی میکنیم برای یک مجموعه آموزشی مشخص یک ساختار مناسب برای شبکه را پیدا کنیم. در این بررسی توجه به این نکته حائز اهمیت است که در صورتیکه تعداد واحدهای مخفی و تعداد وزنهای شبکه کم باشد مرز نواحی تولید شده توسط شبکه هموار بوده و بهمین دلیل شبکه دارای قدرت تعمیم بالا خواهد بود(شبکه ویژگیهای مجموعه آموزشی را یاد میگیرد). در صورت بالا بودن تعداد واحدهای لایع مخفی و تعداد وزنهای لایه مخفی مرز نواحی تولید شده توسط شبکه غیر هموار و دارای شکستگی هایی بوده وبهمین دلیل شبکه دارای قدرت تعمیم پایین می‌باشد(چنین شبکه ای مشابه یک جدول عمل می‌کند و تنها الگوهای آموزشی را بخاطر می‌سپارد). لازم بذکر است که تعداد زیاد وزنها نه تنها باعث کاهش‌قدرت تعمیم شبکه میگردد بلکه زمان آموزش را افزایش می‌دهد.

۴- اتوماتانهای یادگیر

یادگیری‌دراتوماتانهای‌یادگیر،انتخاب‌اقدام[1] بهینه‌ازمیان‌مجموعه اقدامهای مجازمیباشد. این‌اقدام‌روی‌یک‌محیط‌تصادفی‌اعمال‌میشودو محیط‌به این اقدام‌اتوماتان‌بوسیله‌یک‌پاسخ‌ازمجموعه‌پاسخهای‌مجازجواب‌میدهد. پاسخ محیط‌بصورت‌آماری‌به‌اقدام‌اتوماتان‌وابسته‌است. اصطلاح محیط شامل‌اجتماع‌تمام‌شرایط‌خارجی‌وتاثیرات‌آنهاروی‌عملکرداتوماتان میباشد.

یک محیط بصورت‌یک‌سه تایی $\langle \alpha, \underline{C}, \beta \rangle$ نشان داده میشود. مجموعه $\underline{\alpha}=\{\alpha_1,\ldots,\alpha_r\}$ مجموعه‌ورودیها،مجموعه $\underline{C}=\{c_1,\ldots,c_r\}$ مجموعه احتمالات(c_i احتمال‌شکست‌اقدام α_i است) ومجموعه $\beta=\{\beta_1,\beta_2\}$ خروجی دودویی محیط میباشد[۲۵]. اتصال یک اتوماتان با محیط در شکل ۷ نشان داده شده است.



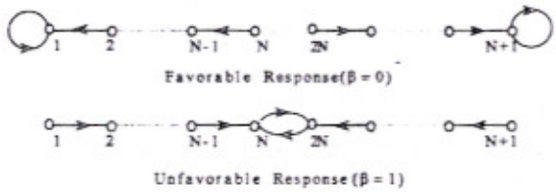
شکل ۹: اتصال اتوماتان یادگیری با محیط

1-Action

اتوماتان های یادگیر به دو خانواده اتوماتان های یادگیر با ساختارثابت[1] و اتوماتان‌های یادگیربا ساختار متغیر[2] دسته بندی میشوند. اتوماتا ن‌هـای *Tsetline* ، *Krinsky* و *Krylov* مثالهایی از خانواده اتوماتان‌های یادگیر با ساختار ثابت هستند. یک اتوماتان یادگیربا ساختار ثابت را میتوان با یک پنج تایی $\langle \underline{\alpha}, \underline{\Phi}, \underline{\beta}, F, G \rangle$ نشان داد. که $\underline{\alpha}=\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعه اقدام های مجاز برای اتوماتان یادگیر، $\underline{\Phi}=\{\Phi_1, \Phi_2, \ldots, \Phi_s\}$ مجموعه وضعیت های اتوماتان یادگیر، $\underline{\beta}=\{0, 1\}$ مجموعه ورودیهای اتوماتان یادگیر (در این مجموعه **یک** نمایانگر شکست[3] و **صفر** نمایانگر موفقیت[4] میباشد)، $F: \Phi \times \beta \to \Phi$ تابع نگاشت وضعیت ها و $G: \Phi \to \alpha$ تابع نگاشت خروجی میباشد.

اقدام اتوماتان بعنوان ورودی به محیط داده میشود و محیط بعد از اعمال اقدام داده شده توسط اتوماتان یک پاسخ تصادفی که میتواند موفق یا نا موفق باشد را تولید میکند که بعنوان ورودی به اتوماتان داده میشود. اتوماتان با توجه به پاسخ محیط اقدام مربوطه را جریمه میکند و یا به آن پاداش میدهد. اگر احتمال تغییر وضعیت ها در اتوماتان ثابت باشد آنرا اتوماتان یادگیری با ساختار ثابت و در غیر اینصورت آنرا اتوماتان یادگیری با ساختار متغیر مینامند. درادامه بطور مختصر یک نمونه از اتوماتان های یادگیر با ساختار ثابت بنام اتوماتان Tsetline شرح داده میشود. برای اطلاعات بیشتر در مورد اتوماتانهای با ساختار متغیر به مراجع [۱۷][۲۵][۴۴][۴۵][۴۶][۴۷] مراجعه نمایید.

اتوماتان Tsetline ($L_{2N,2}$) : این اتوماتان 2N وضعیت و ۲ اقدام دارد و سعی دارد از رفتار گذشته محیط برای تصمیم گیری های آینده خود استفاده نماید. این اتوماتان تعداد پاسخ های دریافت شده برای هر اقدام را میشمارد و در صورتی اقدام خود را تغییر میدهد که تعداد پاسخ های ناموفق بیشتر از تعداد پاسخ های موفق و یا عدد ثابت N (عمق حافظه) باشد. برای هر پاسخ دریافت شده موفق (نا موفق) ، اتوماتان بسمت وضعیت های داخلی تر (وضعیت های خارجی تر) اقدام انتخاب شده حرکت‌میکند. گراف‌تغییرحالت‌اتوماتان‌درشکل ۸ نشان داده شده است.



شکل ۸: اتوماتان Tsetline

اتوماتانهای یادگیر دارای کاربردهای فراوانی میباشند بعضی از این کاربردها عبارتند از: مسیر یابی در شبکه های ارتباطی[۱۸]، فشرده سازی تصاویر[۸]، شناسایی الگو [۳۶]، برنامه ریزی فرایندها[5] در یک شبکه کامپیوتری[۲۵]، تئوری صف [۲۱] ، کنترل دسترسی در شبکه های انتقال ناهمزمان[6] [۱۷]، کمک به آموزش شبکه های عصبی [۲۰] دسته بندی و افراز اشیاء [۲۶] .

1- Fixed Structure Learning Automata (FSLA)
2- Variable Structure Learning Automata (VSLA)
3- Unfavorable
4- Favorable
5- Process
6- Asynchronous transfer mode (ATM)

در صورتیکه از اتوماتانهای یادگیر برای دسته بندی یا افراز اشیاء استفاده شوند علیرغم سرعت همگرایی خوب، تعداد اقدام های اتوماتان بسیار زیاد خواهد بود[۲۶]. برای کاهش تعداد اقدام های اتوماتان، اتوماتان مهاجرت اشیا توسط ***اومن***[7] و ***ما***[8] پیشنهاد شده است[۲۶]. تعداد خروجی های این اتوماتان بمراتب کمتر از اتوماتانهای قبلی است ولی در عوض سرعت همگرایی آن پایین‌تر میباشد.

اتوماتان مهاجرت اشیا : اتوماتان مهاجرت اشیاء توسط پنج تایی $\langle \underline{\alpha}, \underline{\Phi}, \underline{\beta}, F, G \rangle$ نشان داده میشود[۲۶]. که $\underline{\alpha}=\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ مجموعه اقدام‌های مجاز، $\underline{\Phi}=\{\Phi_1, \Phi_2, \ldots, \Phi_s\}$ مجموعه وضعیتها، $\underline{\beta}=\{0, 1\}$ مجموعه ورودیها، $F: \Phi \times \beta \to \Phi$ تابع نگاشت وضعیت ها و $G: \Phi \to \alpha$ تابع نگاشت خروجی اتوماتان میباشند. این نوع اتوماتان برای دسته بندی اشیاء[۲۶]، انتساب حروف به کلیدها[۲۷] ، افراز گراف[9] [۲۸]، تناظر گراف [۴۲] و تعیین تعداد واحد های مخفی شبکه های عصبی [۴۳] مورد استفاده قرار گرفته است.

در این اتوماتان هر اقدام یک دسته از الگوها را نشان میدهد. در اتوماتانهای با ساختار ثابت پاسخ محیط به اتوماتان سبب تغییر وضعیت اتوماتان میگردد در صورتیکه در اتوماتان مهاجرت اشیا، اشیا به وضعیت های اتوماتان انتساب داده میشوند و پاسخ محیط به اتوماتان سبب گردش اشیا در بین وضعیت‌های اتوماتان میگردد. از طریق این گردش دسته بندی اشیا انجام میگیرد. اگر شی W_i دراقدام شماره j اتوماتان مهاجرت اشیا قرار داشته باشد این شی متعلق به دسته شماره jاست. برای اقدام α_k مجموعه وضعیت $\{\Phi_{(k-1)N+1}, \ldots, \Phi_{kN}\}$ در نظر گرفته میشود که N عمق حافظه را نشان میدهد. بدون از دست دادن عمومیت بحث میتوان $\Phi_{(k-1)N+1}$ را داخلی ترین وضعیت و Φ_{kN} راخارجی ترین وضعیت این اقدام در نظر گرفت. اگر دو شی w_i و w_j بترتیب در وضعیت های $\Phi_{(k-1)N+1}$ و $\Phi_{(k-1)N+m}$ (برای $m > 1$) قرار داشته باشند در اینصورت احتمال تعلق شی w_i به این دسته از احتمال تعلق شی w_j بیشتراست. بنابر این برای اقدام α_k، وضعیت $\Phi_{(k-1)N+1}$، وضعیت با بیشترین‌احتمال‌و وضعیت Φ_{kN}، وضعیت باکمترین‌احتمال‌نامیده میشود.

اتوماتان یادگیرتعیین تعداد واحدهای لایه مخفی[10] : در این قسمت اتوماتان یادگیرتعیین تعداد واحدهای مخفی که از نوع مهاجرت اشیا میباشد و اساس الگوریتم بقا را تشکیل میدهد بطور خلاصه شرح داده میشود[۴۳]. وظیفه این اتوماتان دسته بندی واحد های لایه مخفی یک شبکه عصبی سه لایه به دو دسته واحد های مناسب و واحد های نا مناسب است. اتوماتان یادگیرتعیین تعداد واحدهای لایه مخفی بصورت یک شش تایی $\langle \underline{\alpha}, H, \underline{\Phi}, \underline{\beta}, F, G \rangle$ نشان داده میشود که در آن

۱- $\underline{\alpha}=\{\alpha_1, \alpha_2\}$ مجموعه اقدام های مجاز برای اتوماتان یادگیر است. این اتوماتان دو اقدام دارد که اقدام شماره یک آن اقدام مناسب یا واحد های روشن نام دارد. واحد هایی که در وضعیت های این اقدام قرار دارند برای آموزش شبکه عصبی مورد استفاده قرار میگیرند. اقدام شماره دو آن اقدام نامناسب یا واحد های خاموش نام دارد. واحدهایی

7-Oommen
8-Ma
9- Graph Partitioning
10- Hidden Unit Learning Automata (HULA)

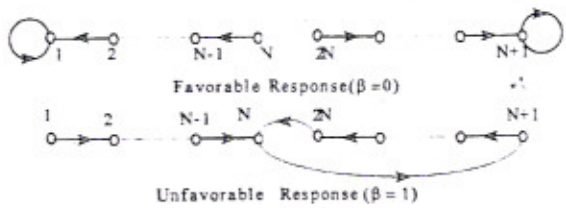
که در وضعیت های مربوط به این اقدام قرار میگیرند برای آموزش شبکه عصبی مورد استفاده قرار نمی گیرند.

۲- $\underline{H}=\{H_1, H_2, \ldots, H_n\}$ مجموعه واحدهای مخفی موجود در وضعیت‌های اتوماتان میباشد. اگر واحد H_i دراقدام شماره یک اتوماتان ظاهر شود این واحد بعنوان واحدمخفی‌مناسب(روشن) ودرغیر اینصورت این‌واحدبعنوان‌واحد مخفی‌نامناسب(خاموش)درنظرگرفته میشود.

۳- $\underline{\Phi}=\{\Phi_1,\Phi_2,\ldots,\Phi_{2N}\}$ مجموعه‌وضعیتها و N عمق‌حافظه برای‌اتوماتان میباشد. مجموعه‌وضعیتهای‌این‌اتوماتان‌به‌دوزیرمجموعه $\{\Phi_1, \ldots, \Phi_N\}$ و $\{\Phi_{N+1}, \ldots, \Phi_{2N}\}$ افراز میشودوبراساس‌اینکه‌واحد های مخفی‌درچه وضعیتی‌قرارداشته‌باشند، دسته‌بندی‌میشوند. براین‌اساس‌واحدهای‌روشن با مجموعه $ON=\{H_i \mid 1\le State(H_i)\le N\}$ و واحد های‌خاموش‌با مجموعه $OFF = \{H_i \mid N+1 \le State(H_i) \le 2N\}$ نشان داده میشوند و $State(H_i)$ نشاندهنده وضعیتی‌است‌که واحد H_i درآن‌قرار دارد.

۴- $\underline{\beta}=\{0, 1\}$ مجموعه ورودیهای اتوماتان میباشد. در این مجموعه **یک** نمایانگر شکست و **صفر** نمایانگر موفقیت میباشد.

۵- $F: \Phi \times \beta \to \Phi$ تابع نگاشت وضعیت‌ها میباشد. این تابع از روی وضعیت فعلی و ورودی اتوماتان وضعیت بعدی آنرا تولید می نماید. در واقع این تابع چگونگی گردش واحد های مخفی رادر وضعیت های اتوماتان مشخص میکند. شرح کارکرد این تابع در [۴۳] آمده است. برای سهولت نمایش در ارائه مطا لب، اتوماتان یادگیر تعیین تعداد واحدهای لایه مخفی با K اقدام، عمق حافظه N و انتساب M واحد مخفی توسط HULA (K, N, M) نشان داده میشود که گراف تغییر حالت برای اتوماتان HULA (2, N, 1) در شکل ۹ نشان داده شده است.



شکل ۹: اتوماتان HULA (2, N, 1)

۶- $G: \Phi \to \alpha$ تابع‌نگاشت‌خروجی‌میباشد. این‌تابع‌تصمیم‌میگیرد که به‌ازای‌هروضعیت، اتوماتان چه‌اقدامی‌راانجام‌دهد. اگرواحد H_i متعلق‌به مجموعه‌وضعیتهای $\{\Phi_1, \Phi_2, \ldots, \Phi_N\}$ باشداین‌واحدروشن‌درنظر گرفته‌میشود. اگرواحددروضعیت Φ_i قرارداشته‌باشد مناسبترین واحد است و بیشترین اهمیت را دارا میباشد. اگردروضعیت Φ_N قرار داشته‌باشد دارای‌کمترین‌اهمیت‌میباشد. اقدام‌شماره‌دونیزبهمین‌صورت میباشد. اگر واحد متعلق به مجموعه وضعیت های $\{\Phi_{N+1}, \ldots, \Phi_{2N}\}$ باشد واحد را خاموش در نظر میگیریم. اگر واحد در وضعیت Φ_{N+1} قرار داشته باشد بیشترین اهمیت را داردواگردروضعیت Φ_{2N} قرارداشته باشد دارای‌کمترین‌اهمیت میباشد.

۵- الگوریتم های پیشنهادی

این‌قسمت‌ابتدا به‌شرح‌مختصری‌ازالگوریتم‌بقا می‌پردازد و سپس الگوریتمهای‌جدیدی‌براساس‌اتوماتانهای‌یادگیرمهاجرت‌اشیاآرایه‌میکند.

الگوریتم بقا از یک اتوماتان یادگیر تعیین تعداد واحد های لایه مخفی (بعنوان یک ابزار جستجوی عمومی) و الگوریتم یادگیری انتشار خطا به عقب استفاده میکند و در ضمن آموزش، تعداد واحدهای مخفی مورد نیاز برای یک شبکه عصبی سه لایه را تعیین مینماید. در الگوریتم بقا، آموزش از یک شبکه بزرگ شروع شده و اتوماتان یادگیر با افزودن و کاستن واحدهای مخفی، تعداد واحدهای لایه مخفی یک شبکه عصبی سه لایه را تعیین میکند. در این الگوریتم، مناسب بودن یا نامناسب بودن یک واحد مخفی بر اساس واریانس فعالیت (انرژی مصرف شده) آن واحد برای الگوهای مختلف میباشد.

در الگوریتمهای پیشنهادشده‌دراین‌مقاله‌ازدواتوماتان استفاده‌میگرددر روی‌یکی‌ازاین‌اتوماتانهاوزنهاوروی‌اتوماتان‌دیگر واحدهای‌مخفی شبکه قرار میگیرند. برای‌سادگی‌درارایه مطالب، دواتوماتان‌موجوددرالگوریتم بصورت HULA(2, N_1, H)-HWLA(2, N_2, W) نشان داده میشوند که HULA(2, N_1, H) اتوماتان‌تعیین‌تعدادواحدهای‌لایه‌مخفی است که دارای دواقدام وعمق‌حافظه N_1 است‌وروی‌وضعیتهای‌آن H واحدمخفی قراردارد و HWLA(2, N_2, W) وزنهای‌لایه‌مخفی[1] است که دارای‌دواقدام‌وعمق حافظه N_2 است‌که‌روی وضعیتهای آن مجموعه‌وزن $W=\{w_1,w_2,w_3, \ldots\}$ قراردارد. گراف تغییر حالت برای اتوماتان تعیین تعداد وزنهای لایه مخفی شبکه در شکل ۹ نشان داده شده است. همچنین برای تنظیم فعالیت اتوماتانها، الگوریتم‌هایی بنام الگوریتم‌های زمانبندی[2] معرفی شده است. وظیفه این الگوریتم‌ها خاموش یا روشن کردن این دو اتوماتان است. انواع مختلفی ازالگوریتم‌های زمانبندی میتواند مورد استفاده قرار گیرد که در این مقاله تنها به دو مورد اشاره شده است. لازم به ذکر است که الگوریتم های زمانبندی مختلف منجر به کارایی های متفاوت میگردند. حال به شرح االگوریتم‌های پیشنهادی میپردازیم.

الگوریتم ۱: این‌الگوریتم‌داری‌دومرحله‌میباشد: مرحله‌تعیین‌تعداد واحدهای موردنیازلایه‌مخفی‌ومرحله‌تعیین‌تعدادوزنهای موردنیازلایه مخفی‌شبکه. برای‌تعیین‌تعدادواحدهای‌لایه‌مخفی‌ازالگوریتم بقااستفاده گردیده‌است. الگوریتم‌بقاحداقل‌تعدادواحدهای‌لایه‌میانی‌شبکه‌عصبی‌سه لایه که قدرت تعمیم قابل قبولی ارایه و قادر به مدلسازی مسئله گردد راتعیین‌میکند. درالگوریتم‌بقا، درابتداهمه واحدهای مخفی روشن‌بوده ودر وضعیت Φ_1 قرار میگیرند و به همه واحد های روشن مدتی اجازه داده میشود تا درآموزش شبکه شرکت نمایند. واحد هایی که عملکرد آنها در این مدت مناسب نبوده جریمه میشوند، واحد هایی که عملکرد آنها خیلی خوب بوده پاداش میگیرند و واحدهایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییروضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک واحد از متوسط انرژی مصرف شده آن واحد استفاده میشود. چگونگی تغییر فعالیت یک واحد به ازای مجموعه الگوهای‌آموزشی، انرژی مصرف شده آن واحد نامیده میشود.

پس از اتمام مرحله اول ، مرحله دوم الگوریتم که تعیین تعداد وزنهای لازم برای لایه مخفی شبکه میباشد شروع میشود. این مرحله از وزنهای متصل به واحد های تعیین شده در مرحله اول الگوریتم استفاده میکند

1-Hidden Weight Learning Automata (HWLA)
2–Timming Algorithms

و تنها طریقه اتصال وزنهای بین لایه ورودی و لایه مخفی را تعیین مینماید. در این مرحله اتوماتان تعیین واحد های لایه مخفی خاموش و اتوماتان تعیین تعدادوزنهای لایه مخفی شبکه روشن میگردد. در ابتدا همه وزنهای موجود در شبکه (وزنهای متصل به واحد های مخفی روشن) در وضعیت Φ_1 قرار میگیرند. به همه وزنهای روشن مدتی اجازه داده میشود تا درآموزش شبکه شرکت نمایند. وزنهایی که عملکرد آنها در این مدت مناسب نیست جریمه میشوند، وزنهایی که عملکرد آنها خیلی خوب است پاداش میگیرند و وزنهایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییروضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک وزن ازقدرت انتشاراین وزن (قدر مطلق مقدار وزن) استفاده میشود که توسط دو قانون مکاشفه ای زیر بیان میشود.

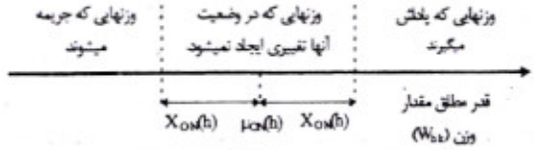
عملکرد خوب یک وزن: وزنی دارای عملکرد خوب میباشد که سیگنال وارد شده به این وزن بمیزان زیادی تضعیف نگردد و یا به عبارتی دیگر قدر مطلق آن کوچک نباشد. این بدان معنی است که اطلاعات ذخیره شده دراین وزن مهم میباشد.

عملکردبدیک وزن: وزنی دارای عملکردبداست اگرسیگنال واردشده به این وزن بمیزان زیادی تضعیف گرددویابه عبارتی دیگرقدرمطلق آن کوچک باشد. این بدان معنی است که اطلاعات ذخیره شده در وزن این واحدمهم نمیبا شد.

تشخیص نحوه عملکرد یک وزن روشن: اگر طی یک دوره آموزش شبکه توسط الگوریتم یادگیری انتشار خطا به عقب، قدرمطلق مقدار یک وزن از یک مقدار آستانه ای کمتر باشد این وزن بد و اگر قدرمطلق مقدار آن از مقدار آستانه ای دیگر بیشتر باشد وزن را خوب مینامیم. برای تعیین مقادیر آستانه ای از واریانس قدرمطلق مقدار وزنها استفاده میشود. بدلیل اینکه در مرحله اول این الگوریتم، الگوریتم بقا تعداد واحدهای مخفی شبکه را تعیین کرده است و در اینجا تنها هدف تعیین اتصال وزنهای بین لایه ورودی و لایه مخفی است لذا برای محاسبه واریانس قدرمطلق مقدار وزنها، واریانس بصورت مجزا و برای وزنهای متصل به هریک از واحدهای مخفی روشن و بصورت زیر محاسبه میگردد.

$$\sigma_h = \sqrt{\frac{\sum \left(|W_{hk}| - \mu_{ON}(h)\right)^2}{|ON_w(h)|}} \qquad h \in ON$$

که در آن W_{hk} وزن اتصال بین واحد ورودی k به واحد مخفی h، ON مجموعه واحدهای روشن، $ON_w(h)$ مجموعه وزنهای ورودی روشن متصل به واحد مخفی h و $\mu_{ON}(h)$ میانگین قدرمطلق مقدار وزنهای روشن متصل به واحد مخفی h میباشد. پس از محاسبه واریانس وزنهای متصل به واحدهای روشن، وزنهای روشنی که قدرمطلق مقدار آنها از یک مقدار آستانه ای کمتر باشد جریمه میشوند و وزنهای روشنی که قدرمطلق مقدار آنها از یک مقدار آستانه ای دیگر بیشتر باشد پاداش میگیرند و وزنهای روشنی که قدرمطلق مقدار آنها بین دو مقدار آستانه ای فوق قرار میگیرند تغییری در وضعیت آنها ایجاد نمیشود.(شکل ۱۰)

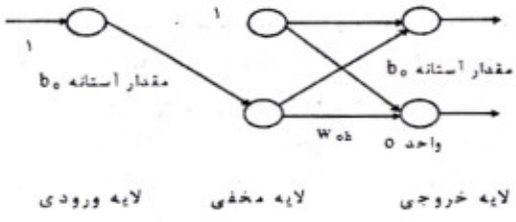


شکل ۱۰: تشخیص مقدار آستانه وزنهای روشن

$X_{ON}(h)$ درشکل فوق پهنای روشنی نامیده میشودوبصورت زیر محاسبه میگردد.

$$X_{ON}(h) = \lambda_{ON} \frac{|OFF_w(h)|}{\#INPUT + 1}$$

ثابت $\lambda_{ON} > 0$ ضریب پهنای روشنی، #INPUT تعدادواحدهای لایه ورودی و $OFF_w(h)$ مجموعه وزنهای خاموش متصل به واحد مخفی روشن h است. وزنهایی که قدرمطلق مقدار آنها از $\mu_{ON}(h) - X_{ON}(h)$ کمتر باشد جریمه میشوند. وزنهایی را که قدرمطلق مقدار آنها از $\mu_{ON}(h) + X_{ON}(h)$ بیشتر باشند پاداش میگیرند و وزنهایی که قدرمطلق مقدار آنها در فاصله $[\mu_{ON}(h) - X_{ON}(h), \mu_{ON}(h) - X_{ON}(h)]$ قراردارندتغییروضعیت باقی میمانند. اگریک واحدمخفی (مطابق شکل ۱۱) دارای تنها یک وزن آستانه باشداین واحد ووزن آنرا میتوان حذف کردواثرخروجی این واحدرادر مقدار آستانه واحدهای لایه خروجی اعمال نمود. در اینحالت مقدار $f(b_h) * W_{oh} + b_o$ را بعنوان مقدار آستانه واحد خروجی (b_o) قرار میدهیم و وزنهای b_h و W_{oh} و واحد مخفی h حذف میگردند.



شکل ۱۱: حذف وزن بایاس یک واحد

تشخیص نحوه عملکرد یک وزن خاموش: بدلیل اینکه در این مرحله اتوماتانهای تعیین تعداد واحد های مخفی خاموش است بنابراین هیچکدام از واحدهای مخفی تغییر وضعیت نمیدهند و تنها وزنهای خاموش متصل به واحد های روشن بررسی میگردند. یک وزن خاموش در آموزش شبکه شرکت نمیکند بنابراین مقدار آنرا از آخرین زمان روشن بودن آن محاسبه میکنیم. اگر یک وزن برای یک مدت طولانی خاموش باشد از اهمیت و در نتیجه از مقدار آن کاسته میشود. بنابراین مقدار یک وزن خاموش بصورت زیر محاسبه میشود.

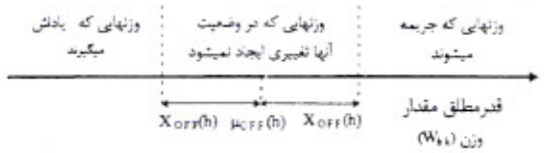
$$W_{hK}(n+1) = W_{hK}(n)\,\lambda_d^{\,n}$$

که ثابت $0 < \lambda_d < 1$ ضریب کاهش مقدار وزن نامیده میشود و n شاخص زمان را نشان میدهد. به این ترتیب مقدار یک وزن خاموش بتدریج کم میشود. واریانس وزنهای خاموش بصورت زیر محاسبه میشود

$$\sigma_h = \sqrt{\frac{\sum \left(|w_{hk}| - \mu_{OFF}(h)\right)^2}{|OFF_w(h)|}} \qquad h \in ON$$

که در آن W_{hk} وزن اتصال بین واحد ورودی k به واحد مخفی h، ON مجموعه واحدهای روشن و $\mu_{OFF}(h)$ میانگین قدرمطلق مقدار وزنهای خاموش متصل به واحد مخفی روشن h میباشد. پس از محاسبه واریانس وزنهای متصل به واحد های روشن، وزنهای خاموشی که قدرمطلق مقدار آنها از یک مقدار آستانه ای کمتر باشد پاداش میگیرند و وزنهای خاموشی که قدرمطلق مقدارآنها از یک مقدار آستانه ای دیگر بیشتر باشد جریمه میشوند و وزنهای خاموشی که قدرمطلق مقدار آنها

بین دو مقدار آستانه ای فوق قرار میگیرند تغییری در وضعیت آنها ایجاد نمیشود.(شکل ۱۲)



شکل ۱۲: تشخیص مقدار آستانه وزنهای خاموش

$X_{OFF}(h)$ در شکل فوق پهنای خاموشی نامیده میشود بصورت زیر محاسبه میگردد.

$$X_{OFF}(h) = \lambda_{OFF} \frac{|ON_w(h)|}{|\#INPUT + 1|}$$

که ثابت $\lambda_{OFF} < 0$ ضریب پهنای خاموشی و #INPUT تعدادواحدهای لایه ورودی است. وزنهایی که قدرمطلق مقدار آنها از $\mu_{OFF}(h)$ -$X_{OFF}(h)$ کمتر باشد پاداش میگیرند. وزنهایی که قدر مطلق مقدار آنها از $\mu_{OFF}(h)$ +$X_{OFF}(h)$ بیشتر باشند جریمه میشوند. و وزنهایی که قدر مطلق مقدار آنها در فاصله [$\mu_{OFF}(h)$ -$X_{OFF}(h)$, $\mu_{OFF}(h)$ -$X_{OFF}(h)$] قرار دارد وضعیت آنها تغییر نمیکند. زمانیکه یک وزن از حالت خاموش به حالت روشن میرود مقدار آن بصورت تصادفی مقداردهی میگردد. این الگوریتم در شکل ۱۳ نشان داده است. در الگوریتم ۱ ، الگوریتم زمانبندی در ابتدا اتوماتان تعیین تعداد واحدهای لایه مخفی را روشن و اتوماتان تعیین تعداد وزنهای لایه مخفی را خاموش میکند. پس از تعیین تقریبی تعداد واحدهای لایه مخفی شبکه ، اتوماتان تعیین تعداد واحدهای لایه مخفی خاموش و اتوماتان تعیین تعداد وزنهای لایه مخفی شبکه را روشن میکند. در بعضی از مواقع و با استفاده از اطلاعات موجود درباره مسئله، تعداد تقریبی واحدهای لایه مخفی را میتوان از قبل تعیین نمود و بهمین دلیل مرحله تعیین تعداد واحدهای لایه مخفی مورد نیاز نخواهد بود. در الگوریتم ۲ این مسئله مد نظر گرفته شده است.

الگوریتم ۲: درصورتیکه تعداد تقریبی واحدهای لایه مخفی از قبل مشخص باشد میتوانیم از شکل خلاصه تری از الگوریتم ۱ استفاده نماییم و ساختار شبکه را تعیین کنیم. الگوریتم ۲ حالت خاصی از الگوریتم ۱ است که در آن اتوماتان تعیین تعداد واحدهای لایه مخفی همیشه خاموش است. معیار تشخیص خوب یا بد بودن عملکرد یک وزن و همچنین نحوه جریمه کردن و پاداش دادن به یک وزن همانند الگوریتم ۱ است.

الگوریتم ۳: این الگوریتم در یک مرحله تعداد وزنها را تعیین میکند. در این الگوریتم برای تعیین اهمیت هر وزن ورودی یک واحد مخفی، آن وزن باکلیه وزنهای ورودی واحدهای روشن موجود در لایه مخفی شبکه مقایسه میگردد. (در این الگوریتم حذف یک واحد بطور غیر مستقیم از طریق حذف وزنهای آن انجام خواهد گرفت) در صورتیکه در الگوریتمهای ۱ و ۲ برای تعیین اهمیت یک وزن ، مقدار ان وزن با مقدار وزنهای ورودی همان واحد مقایسه میگردد. در این الگوریتم ، در ابتدا همه وزنهای موجود در شبکه در وضعیت Φ_1 اتوماتان HWLA قرار میگیرند و به همه وزنهای روشن مدتی اجازه داده میشود تا درآموزش شبکه شرکت نمایند. وزنهایی که عملکرد آنها دراین مدت مناسب نبوده جریمه میشوند، وزنهایی که عملکرد آنها خیلی خوب بوده پاداش میگیرند و وزنهایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییر وضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک وزن از دو قانون مکاشفه ای بیان شده در الگوریتم ۱ استفاده شده است.

```
algorithm _1
  Input:
    Training Set (X, T), X is input and T is desired output
    Maximum No. of Hidden Units: Hmax
  Output:
    Network Weight Vector : W
    Network Tolpology: ON, ONW
  call Survival
  repeat
    // After K steps of BP weights are examined
    for m := 1 To K do
      call BP
    end for
    for all h ∈ ON do
      if w ∈ OFFW then
        w = w * λd
      Compute XON, XOFF
      if w ∈ OFFW then
        if abs (w)<(μOFF - XOFF) then RewardOFFWeight(w)
        if abs (w) > (μOFF + XOFF)then PenalizeOFFWeight(w)
      else
        if abs (w) < (μON - XON)   then PenalizeOnWeight (w)
        if abs (w) > (μON + XOFF)  then RewardOnWeight(w)
      end for
      if unit h has only bias weight then
        remove this weight and its unit and change the output
        units biases
    end for
  until
end

procedure PenalizeOnWeight (I)
  inc (State (I))
end procedure

procedure RewardOnWeight (I)
  if State (I) > 1 then
    dec (State (I))
  end if
end procedure

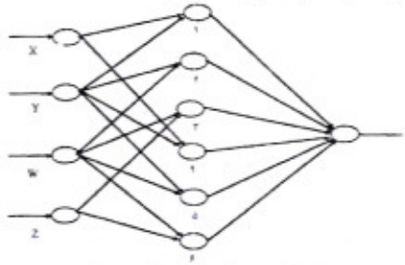
procedure PenalizeOffWeight (I)
  if State (I) ≠ 2 * N then
    inc (State (I))
  else
    State (I) := N
  end if
end procedure

procedure RewardOffWeight (I)
  if State (I) ≠ N + 1 then
    dec (State (I))
  end if
end procedure
```

شکل ۱۳: الگوریتم ۱

تذکر ۱: در بسیاری از کاربرد ها ممکن است یک واحد مخفی مسئول پردازش قسمتی از الگوی ورودی باشد. مثلا در شناسایی نوری حروف، یک واحد مخفی ممکن است پردازش ناحیه کوچکی از تصویر حرف را عهده دار باشد. اگر قسمتی از تصویر برای همه حروف یکسان باشد در فرایند شناسایی بی تاثیر خواهد بود بعنوان مثال در شناسایی اعداد فارسی ۱، ۲، ۳، ۴ نیمه پایین تصاویر اعداد یکسان است و هیچ تاثیری

در شناسایی ندارد بنابراین واحدهای مخفی برای شناسایی این اعداد نیاز به پردازش این ناحیه از تصویر ندارند. بطور مثال برای شناسایی دو عدد ۳ و ۴ از بین اعداد فوق، واحد مخفی که باید ۴ را شناسایی می‌کند می‌بایست ناحیه‌ای را پردازش نماید که واحد مخفی تشخیص دهنده عدد ۳ برای تشخیص ۳ بدان نیاز ندارد. اتصالات در چنین شبکه ای کامل نیست وتنها تعدادی از اتصالات بین لایه ورودی و لایه مخفی موجود است. چنین شبکه‌ای را شبکه با اتصالات محلی[1] میگویند. یک نمونه از اینگونه شبکه در شکل ۱۴ نشان داده شده است.



شکل ۱۴: شبکه با اتصالات محلی

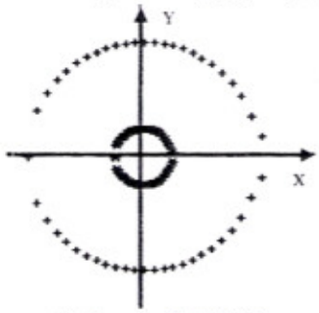
در شبکه شکل ۱۴ واحد شماره ۱ دو وزون ورودی به دو واحدهای X و Y دارد و وزنهای ورودی آن به واحد های W و Z حذف شده است و این بدان معنی است که واحد شماره ۱ تنها این ناحیه از ورودی (X و Y) را پردازش میکند. این نوع شبکه ها ممکن است دارای تعداد واحدهای مخفی بیشتری نسبت به شبکه های کاملا متصل باشد اما دارای تعداد وزنهای کمتر و قدرت تعمیم بیشتری هستند زیرا خروجی هر واحد مخفی به تعداد متغیر کمتری وابسته است و هموار تر است و در نتیجه قدرت تعمیم آن بیشتر است[۵۲]. این مسئله در الگوریتم ۲ (وقتی که تعداد واحدهای مخفی حدس زده شده زیاد باشد) و الگوریتم ۳ مد نظر میباشد. در این حالت وزنهایی از لایه مخفی شبکه حذف میگردند که ورودی برای آن وزنها در بیشتر الگوهای ورودی تقریبا یکسان باشد. نتایج آزمایشات که در قسمت بعد به آن میپردازیم نشان می‌دهند که هر سه الگوریتم ارایه شده، شبکه‌هایی با اتصالات محلی تولید میکنند. نتایج آزمایشات همچنین نشان میدهند که تعداد واحدهای مخفی که الگوریتم شماره ۱ تعیین میکند بمراتب کمتر از واحدهای مخفی تعیین شده توسط الگوریتم های ۲ و ۳ می‌باشد.

۶-نتایج شبیه سازیها

دراین قسمت چهار مسئله نمونه که الگوریتمهای پیشنهادی روی آنها پیاده سازی شده است معرفی میگردد و سپس نتایج آزمایشات ارایه میشود.

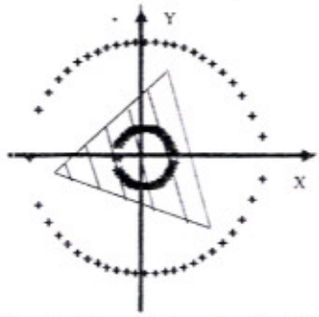
الف) دسته بندی چهار دسته از الگوها : دراین مسئله چهار دسته الگو(که در شکل ۲ با علایم "+"، "*"، "x"و "#" نشان داده شده اند) وجود دارد که هر الگو توسط دو ویژگی x و y نمایش داده می‌شوند و این دو ویژگی ورودیهای شبکه را تشکیل میدهند[۵۳]. هر دسته از این الگوها دارای۱۵ نمونه هستند که ۱۰ نمونه از آنها برای مجموعه آموزشی و از ۵ نمونه دیگر برای مجموعه آزمایشی استفاده گردیده است. این دسته ها بصورت خطی از هم جدایی پذیر نیستند و برای جدا سازی آنها یک شبکه‌عصبی سه‌لایه‌باحداقل دو واحد مخفی و ۴ اتصال مورد نیاز میباشد (که‌نواحی‌تولیدشده‌توسط‌چنین‌شبکه‌ای‌درشکل۵ نشان‌داده‌شده است).

ب) دسته بندی دو دسته از الگوها: در این مسئله دو دسته الگو وجود دارد که در شکل ۱۵ توسط علائم "*" و "+" نشان داده شده اند و بترتیب روی دایرهایی به شعاع های ۰/۵ و ۲ توزیع شده اند. الگوها توسط دو ویژگی x و y نمایش داده میشوند واین دو ویژگی ورودیهای شبکه عصبی را تشکیل میدهند[۵۲]. هر دسته از این الگوها دارای ۲۰۰ نمونه می‌باشد که ۱۰۰ نمونه از آنها برای مجموعه آموزشی و از ۱۰۰ نمونه دیگر برای نمونه آموزشی استفاده گردیده است.



شکل۱۵: الگوهای مثال ۲

این‌دسته‌هابصورت‌خطی‌ازهم‌جدایی‌پذیرنیستندوبرای‌جداسازی‌آنهایک شبکه‌عصبی‌سه‌لایه‌باحداقل۳ واحدمخفی و۹ اتصال‌موردنیازمیباشد(که نواحی‌تولیدشده‌توسط‌چنین‌شبکه‌ای‌درشکل ۱۶ نشان‌داده‌شده است).



شکل۱۶: خط های جدا کننده الگوهای مثال ۲

ج) مسئله تشخیص اعداد چاپی فارسی : در این مسئله تشخیص اعداد چاپی فارسی (شکل ۱۷) مد نظر میباشد که روی یک صفحه کاغذ چاپ شده اند. اعداد فارسی از طریق یک دستگاه پویشگر و با قدرت تفکیک ۳۰۰ نقطه در اینچ نمونه برداری گردیده است. از هر حرف ۱۷ نمونه وجود دارد که از ۱۶ نمونه برای مجموعه آموزشی و از ۱ نمونه دیگر برای مجموعه آزمایشی استفاده شده است.



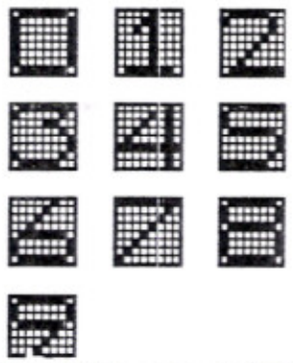
شکل ۱۷: اعداد چاپی فارسی

پس‌ازجداسازی‌حروف‌درتصویر، ثابتهای‌گشتاوری M_1 تا M_7 استخراج شده است و این ۷ ویژگی ورودیهای شبکه عصبی را تشکیل میدهند[۴۸].

د)مسئله تشخیص اعداد لاتین: دراین مسئله ده‌عدد وجوددارد که‌هرکدام ازآنهاتوسط یک‌ماتریس ۸×۸ ازنقطه‌های‌سیاه‌وسفیدنشان داده میشود[۳۴]. الگوهای‌آموزشی‌برای‌این‌مسئله‌درشکل۱۸ نشان‌داده‌شده است.

درادامه‌نتایج‌شبیه‌سازیهای‌انجام‌گرفته‌شده‌برای‌مسئله‌های‌فوق‌آورده‌شده است.

1- Local Connections



شکل ۱۸ : الگوهای آموزشی برای مسئله تشخیص اعداد لاتین

الف) الگوریتم ۱: این الگوریتم همانطور که قبلا اشاره شد از دو مرحله تعیین تعداد واحدهای لایه مخفی و تعیین تعداد وزنهای لایه مخفی تشکیل شده است. الگوریتم ۱ روی مسائل فوق پیاده سازی شده و نتایج آزمایشها در جداول ۱ تا ۴ آورده شده است. قسمت (الف) جداول ۱ تا ۴ شبکه تولید شده توسط الگوریتم بقا و قسمت (ب) این جداول نتیجه تولید شده بعد از مرحله دوم الگوریتم ۱ را نشان میدهد لازم بذکر است که در تمامی آزمایشهای انجام گرفته توسط الگوریتم ۱، مجموعه الگوهای آموزش ۱۰۰۰۰ بار به شبکه داده شده است.

دسته بندی چهار دسته از الگوها: الگوریتم ۱ روی ۶ شبکه باوزنهای اولیه تصادفی آزمایش شده و نتایج آن در جدول ۱ نشان داده شده است. برای تمامی این شبکه ها از اتوماتان HULA(2, 7, 39)-HWLA (2, 7, N_w) استفاده گردیده است. Nw تعداد وزنهای شبکه در پایان الگوریتم بقا میباشد. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹ ، ضریب پهنای روشنی برابر ۱۲، و ضریب پهنای خاموشی برابر ۱۲ در نظر گرفته شده است.

جدول ۱: نتیجه الگوریتم ۱ برای مسئله ۱

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۷	۲۱	۱۰۰	۱۰۰
۲	۱۲	۳۶	۱۰۰	۱۰۰
۳	۸	۲۴	۱۰۰	۱۰۰
۴	۱۱	۳۳	۱۰۰	۱۰۰
۵	۵	۱۵	۱۰۰	۱۰۰
۶	۴	۱۲	۱۰۰	۱۰۰
متوسط	۷/۸	۲۳/۵	۱۰۰	۱۰۰

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزشی)	درصدتشخیص (آزمایشی)
۱	۴	۸	۱۰۰	۱۰۰
۲	۶	۹	۱۰۰	۱۰۰
۳	۵	۱۰	۱۰۰	۱۰۰
۴	۶	۱۱	۱۰۰	۱۰۰
۵	۴	۸	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۴/۸	۸/۵	۱۰۰	۱۰۰

دسته بندی دو دسته از الگوها: الگوریتم ۱ روی ۶ شبکه باوزنهای اولیه تصادفی آزمایش شده و نتایج آن در جدول ۲ نشان داده شده است. برای انجام این آزمایش ها از اتوماتان HULA(2, 7, 60)-HWLA (2, 7, N_w) استفاده شده است. Nw تعداد وزنهای شبکه در پایان الگوریتم بقا میباشد. برای انجام این آزمایشها، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر ، ضریب کاهش وزن برابر ۰/۹ ، ضریب پهنای روشنی برابر ۱۰۰، و ضریب پهنای خاموشی برابر ۱۰۰ در نظر گرفته شده است.

جدول ۲: نتیجه الگوریتم ۱ برای مسئله ۲

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۷	۲۱	۱۰۰	۱۰۰
۲	۱۲	۳۶	۱۰۰	۱۰۰
۳	۸	۲۴	۱۰۰	۱۰۰
۴	۵	۱۵	۱۰۰	۱۰۰
۵	۸	۲۴	۱۰۰	۱۰۰
۶	۴	۱۲	۱۰۰	۱۰۰
متوسط	۷/۳	۲۲	۱۰۰	۱۰۰

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزشی)	درصدتشخیص (آزمایشی)
۱	۴	۸	۱۰۰	۱۰۰
۲	۶	۹	۱۰۰	۱۰۰
۳	۵	۱۰	۱۰۰	۱۰۰
۴	۴	۸	۱۰۰	۱۰۰
۵	۵	۹	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۴/۶	۸/۱۶	۱۰۰	۱۰۰

مسئله تشخیص اعداد چاپی فارسی: الگوریتم ۱ روی ۶ شبکه باوزنهای اولیه تصادفی آزمایش گردیده و نتایج آن در جدول ۳ نشان داده شده است. برای تمامی این شبکه ها از اتوماتان HULA(2, 7, 60)-HWLA (2, 7, N_w) استفاده شده است. Nw تعداد وزنهای شبکه در پایان الگوریتم بقا میباشد. برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۴، ضریب گشتاور برابر ۰/۵، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۳۰، وضریب پهنای خاموشی برابر ۳۰ درنظر گرفته شده است.

جدول ۳: نتیجه الگوریتم ۱ برای مسئله ۳

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

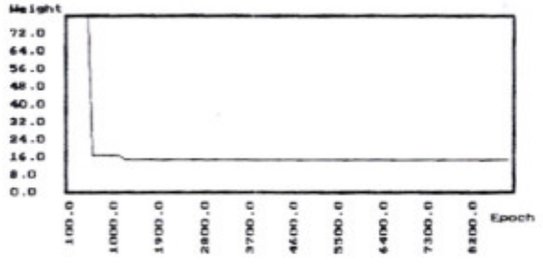
شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۰	۸۰	۹۹/۲۷۵	۱۰۰
۲	۸	۶۴	۹۶/۸۷۵	۱۰۰
۳	۸	۶۴	۹۶/۸۷۵	۱۰۰
۴	۷	۵۶	۸۷/۷۵	۹۰
۵	۱۱	۸۸	۹۹/۲۷۵	۱۰۰
۶	۷	۵۶	۸۳/۷۵	۸۰
متوسط	۸/۵	۶۸	۹۴	۹۵

جدول ۳(ادامه): نتیجه الگوریتم ۱ برای مسئله ۳

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزشی)	درصد تشخیص (آزمایشی)
۱	۸	۱۵	۹۹/۳۷۵	۱۰۰
۲	۸	۲۶	۹۵/۰	۱۰۰
۳	۸	۲۶	۹۵/۶۲۵	۱۰۰
۴	۷	۱۸	۹۶/۸۷۵	۱۰۰
۵	۹	۱۸	۹۹/۳۷۵	۱۰۰
۶	۷	۱۸	۹۶/۸۷۵	۱۰۰
متوسط	۷/۸۳۳	۲۰/۱۶	۹۷/۱۸۷	۱۰۰

شکل ۱۹ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو[1] برای شبکه شماره ۱ جدول ۳-ب نشان میدهد.



شکل ۱۹ :تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

مسئله تشخیص اعداد لاتین: الگوریتم ۱ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده اند و نتایج آن در جدول ۴ نشان داده شده است. برای انجام آزمایش های انجام شده از اتوماتان HULA(2, 20, 60)-HWLA (2, 20, N_w) استفاده گردیده است. N_w تعداد وزنهای شبکه در پایان الگوریتم بقا میباشد. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۰/۱، وضریب پهنای خاموشی برابر ۰/۱ درنظر گرفته شده است.

جدول ۴: نتیجه الگوریتم ۱ برای مسئله ۴

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

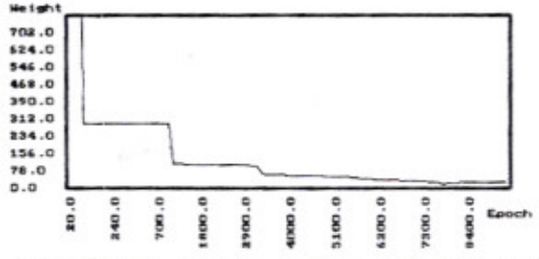
شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۲	۷۸۰	۱۰۰	۱۰۰
۲	۷	۴۵۵	۱۰۰	۱۰۰
۳	۹	۵۸۵	۱۰۰	۱۰۰
۴	۱۳	۸۴۵	۱۰۰	۱۰۰
۵	۱۱	۷۱۵	۱۰۰	۱۰۰
۶	۱۳	۸۴۵	۱۰۰	۱۰۰
متوسط	۱۰/۸۳	۷۰۴/۱۶	۱۰۰	۱۰۰

جدول ۴(ادامه): نتیجه الگوریتم ۱ برای مسئله ۴

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد شخیص (آموزشی)	درصدتشخیص (آزمایشی)
۱	۱۲	۲۴	۱۰۰	۱۰۰
۲	۷	۲۷	۱۰۰	۱۰۰
۳	۹	۵۱	۱۰۰	۹۰
۴	۱۳	۵۶	۱۰۰	۱۰۰
۵	۱۱	۵۸	۱۰۰	۱۰۰
۶	۱۳	۵۶	۱۰۰	۱۰۰
متوسط	۱۰/۸۳	۴۵/۳۳	۱۰۰	۱۰۰

شکل ۲۰ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول ۴-ب نشان میدهد.



شکل ۲۰: تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

همانطور که در جداول فوق مشخص است تعداد وزنها پس از مرحله تعیین تعداد وزنهای لایه مخفی بسیار کمتر از تعداد وزنها پس از مرحله تعیین واحدهای لایه مخفی میباشد و همچنین قدرت تعمیم شبکه در پایان مرحله تعیین وزنها نیز در بیشتر موارد بیشتر از قدرت تعمیم شبکه در پایان مرحله تعیین واحدها است. این بدان دلیل است که با کاهش نسبت متوسط تعداد وزن بر واحد مخفی، ابر صفحه های تولید شده توسط واحدهای مخفی بسیار هموار میگردند. نکته جالب دیگری که از جداول فوق میتوان نتیجه گرفت تعداد متوسط وزنهای متصل به هر واحد مخفی میباشد. در مسئله ۱ و۲ همه واحدهای مخفی کمتر از ۳ وزن ورودی دارند و در مسئله ۱ بیشتر خطوط تولید شده توسط واحد های مخفی موازی یکی از دو محور می باشد و تعداد کمی از آنها خطوط مورب هستند که در دسته بندی الگوها نقش ندارند. در مسئله ۳ هر واحد مخفی تقریبا ۳ وزن دارد یعنی هر واحد مخفی بطور متوسط بیشتر از نیمی از ویژگیهایی که برای شناسایی در نظر گرفته شده است استفاده نمیکند. در مورد مسئله ۴ نسبت متوسط تعداد وزن بر واحد مخفی برای آزمایش های مختلف ، متفاوت می باشد. شبکه‌های تولید شده توسط الگوریتم یک این واقعیت را نشان می دهند که بیشتر ویژگیهای مربوط به الگوها (ورودیهای شبکه عصبی) برای شناسایی مورد استفاده قرار نمی گیرند.

ب) الگوریتم ۲: الگوریتم ۲ شکل خلاصه تری از الگوریتم ۱ است که در آن تعداد واحدهای مخفی بصورت تقریبی ودر ابتدا توسط طراح شبکه مشخص میگردد. این الگوریتم فقط طریقه اتصال بین لایه ورودی و لایه مخفی را تعیین میکند. در این الگوریتم تعیین تعداد واحد های لایه مخفی بطور غیر مستقیم و از طریق تعیین تعداد وزنهای لایه مخفی شبکه انجام میگیرد. این الگوریتم نیز مانند الگوریتم ۱ شبکه ای

1- Epoch

با اتصالات محلی تولید میکند اگر تعداد واحدهای مخفی اولیه برای دو الگوریتم یکسان باشد تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بمراتب بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد این بدان دلیل است که حذف واحدهای لایه مخفی بطور غیر مستقیم از طریق حذف وزنهای لایه مخفی انجام می‌گیرد. الگوریتم ۲ روی چهار مسئله تعریف‌شده پیاده‌سازی شده و نتایج آزمایشات در جداول ۵ تا ۸ آورده شده است. برای آزمایشاتی که نتایج آنها در قسمت (الف) جداول ۵ تا ۸ آمده است تعداد واحدهای مخفی اولیه نسبت به تعداد واحدهای اولیه الگوریتم ۱ کوچک در نظر گرفته‌شده است. برای آزمایشاتی که نتایج آنها در قسمت (ب) جداول ۵ تا ۸ آمده است تعداد واحدهای مخفی اولیه الگوریتم ۲ مساوی تعداد واحدهای مخفی اولیه الگوریتم ۱ انتخاب شده است. لازم بذکر است که در تمامی آزمایشهای زیر ۱۰۰۰۰ بار مجموعه آموزشی به شبکه ارایه شده است.

دسته‌بندی چهاردسته‌ازالگوها: الگوریتم۲روی۶ شبکه‌باوزنهای‌اولیه تصادفی آزمایش‌شده‌اندونتایج‌آن‌درجدول۵ نشان‌داده‌شده‌است. برای تمامی‌این‌شبکه‌هادرجدول۵-الف‌ازاتوماتان (2, 7, 18) HWLA ودر جدول ۵-ب از اتوماتان (2, 7, 117) HWLA استفاده شده است. نرخ یادگیری‌برابر ۰/۱، ضریب‌گشتاور برابر صفر ، ضریب کاهش وزن ۰/۹، ضریب‌پهنای‌روشنی‌برابر۱۲، وضریب‌پهنای‌خاموشی‌برابر۱۲ می‌باشد.

جدول ۵ : نتیجه شبیه سازی الگوریتم ۲ برای مثال ۱

الف) نتیجه الگوریتم ۲ برای مثال ۱ برای اتوماتان HWLA (2, 7, 18)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزش)	درصد شخیص (آزمایشی)
۱	۴	۵	۱۰۰	۱۰۰
۲	۶	۷	۱۰۰	۱۰۰
۳	۶	۷	۱۰۰	۱۰۰
۴	۵	۶	۱۰۰	۱۰۰
۵	۶	۶	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۵/۸۳	۶	۱۰۰	۱۰۰

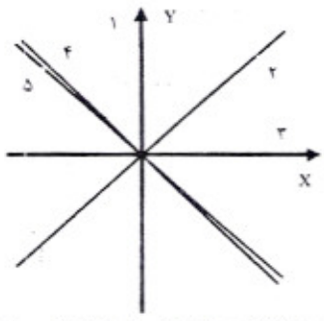
ب) نتیجه الگوریتم ۲ برای مثال ۱ برای اتوماتان HWLA (2, 7, 117)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزش)	درصدتشخیص (آزمایشی)
۱	۲۶	۴۹	۱۰۰	۱۰۰
۲	۱۴	۱۸	۱۰۰	۱۰۰
۳	۱۸	۱۹	۱۰۰	۱۰۰
۴	۱۵	۲۰	۱۰۰	۱۰۰
۵	۱۹	۳۵	۱۰۰	۱۰۰
۶	۲۲	۴۲	۱۰۰	۹۵
متوسط	۱۹	۳۰/۵	۱۰۰	۹۹/۱۶

با مقایسه جداول ۱ و ۵ میتوان به نکات جالب توجه ای اشاره نمود. اول آنکه متوسط تعداد وزن ورودی بر هر واحد مخفی در الگوریتم ۱ و الگوریتم ۲ تقریبا مساوی است. دوم آنکه هیچکدام از شبیه سازیها، شبکه با ساختار بهینه (شبکه سه لایه با دو واحد مخفی و چهار وزن بین لایه ورودی و لایه مخفی که در ابتدای مقاله بدان اشاره شد) را تولیدنکرده‌اند. علت‌این‌امر آنستکه‌تعدادزیادی‌ازواحدهای‌مخفی‌خروجیهای مشابه‌ای‌را تولیدمیکنندوبهمین‌دلیل‌تعدادی‌ازواحدهای‌مخفی‌زائد هستند. برای‌مثال‌شبکه‌شماره۱رادرجدول ۵-الف در نظر بگیرید. ماتریس وزنهای لایه مخفی در پایان اجرای الگوریتم این شبکه به صورت زیر میباشد.

$$W = \begin{bmatrix} 0 & 0 & -2.44 \\ 0 & 3.124 & -2.68 \\ 0 & -1.69 & 0 \\ 0 & -3.01 & -2.69 \\ 0 & -1.62 & -1.16 \end{bmatrix}$$

در ماتریس فوق عنصر w_{j0} مقدار آستانه واحد مخفی شماره j، عنصر w_{jm} وزن اتصال واحد مخفی شماره j به واحد ورودی m میباشد. در این شبکه واحد ورودی ۱ ویژگی x و واحد ورودی ۲ ویژگی y را نشان میدهند. خطوط تولید شده توسط واحد های مخفی این شبکه در شکل ۲۱ نشان داده شده است.



شکل ۲۱: خطوط تولید شده توسط واحد های مخفی شبکه ۱

در شکل ۲۱ خط شماره j توسط واحد مخفی شماره j تولید شده است (خطوط با شماره های ۱و ۳ بترتیب روی محورهای Y و X قرار گرفته‌اند). با توجه به شکل فوق خطوط با شماره های ۴و ۵ بسیار نزدیک بهم هستند. این بدان معنی است که اطلاعات تقریبا یکسانی در وزنهای آنها ذخیره شده است (این دو واحد دارای واریانس فعالیت بالا و وزنهای با مقادیر بزرگ میباشند) وبهمین دلیل میتوان یکی از این دو واحد را حذف نمود. در نتیجه شبکه هرس شده دارای ۴ واحد مخفی و ۶ اتصال خواهد بود. شبکه حاصله هنوز دارای دو واحد زائد با شماره های ۳ و ۴ می‌باشد که الگوریتم پیشنهادی با معیار بکار برده شده قادر به حذف آنها نشده است.

دسته‌بندی‌دودسته‌ازالگوها: الگوریتم۲روی۶ شبکه‌باوزنهای‌اولیه تصادفی‌آزمایش‌شده‌اندونتایج‌آن‌درجدول۶ نشان‌داده‌شده‌است. برای تمامی‌این‌شبکه‌هادرقسمت(الف)ازاتوماتان(2, 7, 18) HWLA و در قسمت(ب)ازاتوماتان(2, 7, 180) HWLA استفاده‌شده‌است. نرخ‌یادگیری برابر ۰/۱، ضریب‌گشتاوربرابرصفر، ضریب‌کاهش‌وزن‌برابر۰/۹، ضریب پهنای روشنی برابر ۱۵، و ضریب پهنای خاموشی برابر ۱۵ در نظر گرفته شده است.

جدول ۶ : نتیجه شبیه ساری الگوریتم ۲ برای مثال ۲

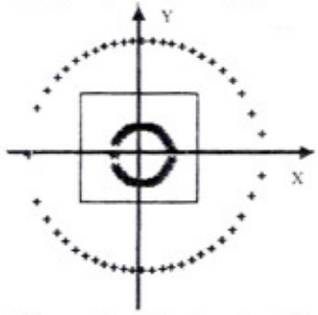
الف) نتیجه الگوریتم ۲ برای مثال ۲ برای اتوماتان HWLA (2, 7, 18)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵	۱۳	۱۰۰	۱۰۰
۲	۵	۱۱	۱۰۰	۱۰۰
۳	۵	۱۳	۱۰۰	۱۰۰
۴	۵	۱۲	۱۰۰	۱۰۰
۵	۴	۱۰	۱۰۰	۱۰۰
۶	۵	۱۲	۱۰۰	۱۰۰
متوسط	۴/۸۸	۱۱/۸۳	۱۰۰	۱۰۰

ب) نتیجه الگوریتم ۲ برای مثال ۲ برای اتوماتان HWLA (2, 7, 180)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۴۸	۹۶	۹۸	۹۶
۲	۳۹	۷۸	۱۰۰	۱۰۰
۳	۴۸	۹۶	۱۰۰	۱۰۰
۴	۴۴	۸۸	۱۰۰	۱۰۰
۵	۴۲	۸۴	۱۰۰	۱۰۰
۶	۳۸	۷۶	۱۰۰	۱۰۰
متوسط	۴۳/۱۶	۸۶/۳۳	۹۹/۶۶	۹۹/۳۳

با توجه به جداول ۲و ۶ میتوان به نکات زیر اشاره نمود. اول آنکه، نسبت متوسط تعداد وزن برواحد مخفی در دو جدول مذکور تقریبا یکسان است. این نسبت در جداول ۲-ب و ۶-الف حدود دو و در جدول ۶-ب مساوی ۲ است. یعنی شبکه های آموزش داده شده در جدول ۶-ب از یکی از وزنهای ورودی‌خود (وزن آستانه، وزن متصل به ورودی X ویا وزن متصل به ورودی Y) برای شناسایی دسته ها از یکدیگر استفاده نمیکند. با توجه به ناحیه رسم شده در شکل ۱۶ که دودسته الگوها را از هم‌جدا میسازد میتوان نتیجه گرفت که این خطوط یا موازی یکی از دو محور (معادلات خطوط بفرم $Y = C$ و یا $X = C$) و یا مورب (معادلات خطوط بفرم $Y = mX + C$) هستند. در صورتیکه معادله خط بصورت $X = C$ باشد، واحد مخفی نیاز به دو وزن (وزن‌آستانه و وزن متصل به ورودی X) دارد ودر حالتیکه معادله خط بصورت $Y = C$ است واحد مخفی نیاز به دو وزن (وزن‌آستانه و وزن متصل به ورودی Y) دارد و درحالتیکه معادله خط بصورت $Y = mX + C$ است واحد مخفی نیاز به سه وزن ورودی(وزن‌آستانه، وزن متصل به ورودی X و وزن متصل به ورودی Y) دارد. با توجه به آزمایشهای انجام شده، واحدهای مخفی تفکیک شده عموما یکی از دوخط $X = C$ یا $Y = C$ را نشان می‌دهند. ناحیه های تولید شده مطابق شکل ۲۲ خواهد بود. شبکه ای که این نواحی را تولید می‌کند بایستی‌حداقل چهار واحد مخفی داشته باشد. این ساختار از ساختار بهینه (شبکه عصبی سه لایه با ۳ واحد مخفی و ۸ اتصال) یک واحد مخفی بیشتر دارد اما تعداد وزنهای این دو شبکه یکسان است. اما شبکه های تولید شده دارای تعداد واحدهای بیشتری هستند که هر کدام از آنها یکی از چهار خط زیر را تولید می‌کنند و الگوریتم قادر به تشخیص و حذف واحدهای تکراری نیست.



شکل ۲۲: نواحی تولید شده برای شبکه های جدول ۶-ب برای مثال ۲

مسئله تشخیص اعداد چاپی فارسی: الگوریتم۲روی۶ شبکه‌باوزنهای اولیه‌تصادفی‌آزمایش‌شده‌اندونتایج‌آن‌درجدول۷ نشان‌داده‌شده‌است. برای تمامی‌این‌شبکه‌هابرای‌قسمت (الف)ازاتوماتان(HWLA (2, 7, 120 و برای قسمت (الف) ازاتوماتان HWLA (2, 7, 480) استفاده‌شده‌است. برای انجام این آزمایشها، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر ، ضریب کاهش وزن برابر۰/۹ ، ضریب پهنای روشنی برابر ۱۵، و ضریب پهنای خاموشی برابر ۱۵ در نظر گرفته شده است.

جدول ۷ : نتیجه شبیه سازی الگوریتم ۲ برای مثال ۳

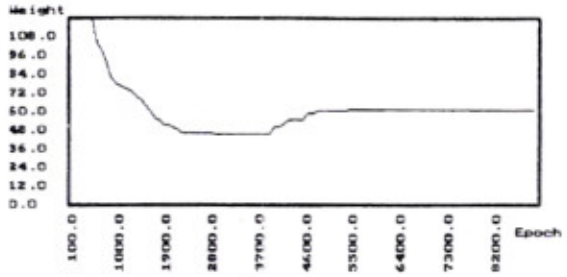
الف) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 120)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزش)	درصد شخیص (آزمایشی)
۱	۱۴	۶۱	۹۸/۷۵	۱۰۰
۲	۱۵	۶۲	۹۸/۱۲۵	۱۰۰
۳	۱۵	۷۰	۹۹/۳۷۵	۱۰۰
۴	۱۵	۶۱	۹۹/۳۷۵	۱۰۰
۵	۱۵	۶۰	۹۹/۳۷۵	۱۰۰
۶	۱۵	۷۴	۹۸/۷۵	۱۰۰
متوسط	۱۴/۸۳	۶۴/۶۶	۹۸/۹۵	۱۰۰

ب) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 480)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد شخیص (آموزش)	درصدتشخیص (آزمایشی)
۱	۲۸	۸۲	۱۰۰	۱۰۰
۲	۳۵	۹۸	۱۰۰	۱۰۰
۳	۳۰	۶۵	۱۰۰	۱۰۰
۴	۲۵	۶۸	۹۹	۱۰۰
۵	۳۴	۹۰	۱۰۰	۱۰۰
۶	۳۰	۷۰	۱۰۰	۱۰۰
متوسط	۳۰/۳۳	۷۸/۸۳	۹۹/۸۴	۱۰۰

شکل ۲۳ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول ۷-الف نشان میدهد.



شکل ۲۳ : تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

باتوجه‌به‌جداول ۳-ب وجدول ۷ میتوان به‌نکات زیر اشاره نمود. نسبت متوسط تعداد وزن ورودی به‌واحد مخفی در الگوریتمهای ۱ و ۲ تقریبا مساوی هستند. بدلیل اینکه در حین کاهش تعداد وزنهای شبکه، تعدادی از واحدهای مخفی، خروجیهای مشابه‌ای را تولید میکنند تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بسیار بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد و الگوریتم قادر به حذف واحدهای مشابه نیست (اطلاعات ذخیره شده در بردار وزن واحدهایی که خروجیهای مشابه را تولید میکنند تقریبا یکسان است و میتوان همه این واحدهای مشابه بجز یکی از آنها را حذف نمود. البته این کار میتواند بعد از تعیین ساختار شبکه توسط الگوریتم دیگری انجام گیرد).

مسئله تشخیص اعداد لاتین: الگوریتم ۲ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول ۸ نشان داده شده است. برای تمامی این شبکه‌ها برای قسمت (الف) از اتوماتان HWLA (2, 20, 975) و برای قسمت (ب) از اتوماتان HWLA (2, 20, 3900) استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱، و ضریب پهنای خاموشی برابر ۱/۵ در نظر گرفته شده است.

جدول ۸ : نتیجه شبیه سازی الگوریتم ۲ برای مثال ۴

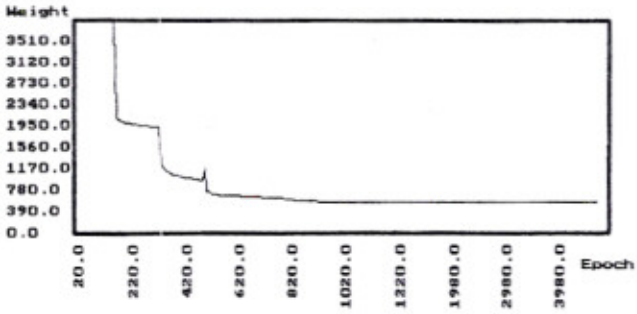
الف) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 875)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد شخیص (آزمایشی)
۱	۱۵	۵۱۲	۱۰۰	۱۰۰
۲	۱۵	۴۸۸	۱۰۰	۱۰۰
۳	۱۵	۵۲۷	۱۰۰	۱۰۰
۴	۱۵	۵۲۲	۱۰۰	۱۰۰
۵	۱۵	۵۰۷	۱۰۰	۱۰۰
۶	۱۵	۵۱۵	۱۰۰	۱۰۰
متوسط	۱۵	۵۱۱/۸۳	۱۰۰	۱۰۰

ب) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 3900)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد شخیص (آزمایشی)
۱	۶۰	۴۴۰	۱۰۰	۹۰
۲	۶۰	۵۳۷	۱۰۰	۱۰۰
۳	۶۰	۵۴۱	۱۰۰	۱۰۰
۴	۶۰	۵۸۲	۱۰۰	۱۰۰
۵	۶۰	۵۷۱	۱۰۰	۱۰۰
۶	۶۰	۵۳۱	۱۰۰	۱۰۰
متوسط	۶۰	۵۳۳/۶۶	۱۰۰	۹۸/۳۴

شکل ۲۴ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول ۸-ب نشان میدهد.



شکل ۲۴ : تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

با توجه‌به‌جداول ۴-ب و ۸ نتایج زیر را میتوان گرفت. همانند مسائل قبل تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بمراتب بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ می‌باشد. نکته دیگر اینکه شبکه های تولید شده توسط الگوریتمهای ۱ و ۲ برای مسائل ۱ و ۲ تقریبا از سی درصد از ورودیهایش وبرای مسئله ۳ بیشتر از نیمی از ورودیهایش و برای مسئله ۴ تقریبا از هشتاد درصد از ورودیهایش استفاده نمی‌کند. بنابراین میتوان انتظار داشت دو الگوریتم ۱ و ۲ برای شبکه‌هایی مناسب‌ترهستندکه دارای ورودیهای زیادی باشند.

ج) الگوریتم ۳: الگوریتم ۳ بمنظور تعیین تعداد وزنها، وزنها را بصورت عمومی با در نظر گرفتن همه وزنهای همه واحد های مخفی روشن مقایسه میکند و بر اساس نتایج این مقایسه‌ها عمل افزایش یا کاهش تعداد وزنهای لایه مخفی را انجام می‌دهد. در این الگوریتم تعیین تعداد واحد های لایه مخفی بطور غیر مستقیم و از طریق تعیین تعداد وزنهای لایه مخفی شبکه انجام می‌گیرد این الگوریتم نیز مانند دوالگوریتم ۱ و ۲ شبکه ای با اتصالات محلی تولید میکند(تذکر ۱). در صورتیکه تعداد واحدهای مخفی اولیه برای دو الگوریتم ۱و ۳ یکسان باشند تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۳ بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد زیرا در الگوریتم ۳ بصورت صریح واحدی حذف نمیگردد.بدلیل استفاده از مزایای آموزش شبکه‌های بزرگ، تعداد واحدهای مخفی در ابتدای آموزش شبکه توسط الگوریتم ۳ بیشتر از تعداد تعیین شده توسط الگوریتم ۱ در نظر گرفته شده است. این الگوریتم روی مسائل نمونه پیاده سازی شده و نتایج آزمایشات در جداول ۹ تا ۱۲ آورده شده است. لازم بذکر است که در تمامی آزمایشهای زیر برای آموزش و تعیین ساختار شبکه های عصبی ۱۰۰۰۰ بار مجموعه آموزشی به شبکه ارایه شده است.

دسته بندی چهار دسته از الگوها : الگوریتم ۳ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده اند و نتایج آن درجدول۹ نشان داده شده است. برای تمامی این شبکه ها از اتوماتان HWLA (2, 7, 117) استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۴، ضریب گشتاور برابر ۰/۶، ضریب کاهش وزن برابر ۰/۹ ، ضریب پهنای روشنی برابر ۰/۱۵، و ضریب پهنای خاموشی برابر ۰/۱۵ در نظر گرفته شده است.

جدول ۹ : نتیجه شبیه سازی الگوریتم ۳ برای مثال ۱

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۷	۲۹	۱۰۰	۱۰۰
۲	۱۷	۲۹	۱۰۰	۱۰۰
۳	۱۷	۳۰	۱۰۰	۹۵
۴	۱۷	۲۹	۱۰۰	۹۷/۵
۵	۲۰	۲۷	۱۰۰	۱۰۰
۶	۲۰	۳۴	۱۰۰	۱۰۰
متوسط	۱۸	۲۹/۶۶	۱۰۰	۹۸/۷۵

دسته بندی دو دسته از الگوها: الگوریتم ۲روی۶ شبکه با وزنهای اولیه تصادفی آزمایش شده اند و نتایج آن درجدول ۱۰ نشان داده شده است. برای تمامی این شبکه ها از اتومتان HWLA (2, 7, 180) استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر ، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۰/۲۵، و ضریب پهنای خاموشی برابر ۰/۲۵ در نظر گرفته شده است.

جدول ۱۰ : نتیجه شبیه سازی الگوریتم ۳ برای مثال ۲

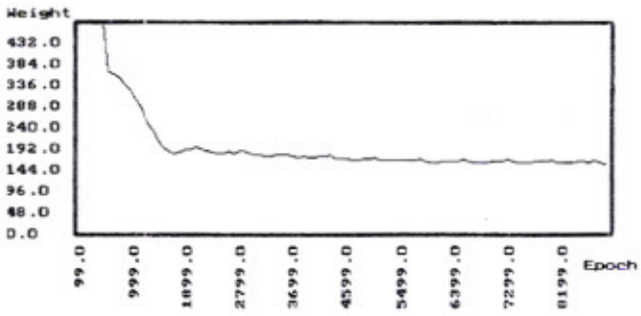
شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۵	۸۲	۱۰۰	۱۰۰
۲	۳۵	۹۸	۱۰۰	۱۰۰
۳	۳۰	۶۵	۱۰۰	۱۰۰
۴	۲۵	۶۸	۱۰۰	۱۰۰
۵	۳۴	۹۰	۱۰۰	۱۰۰
۶	۳۷	۸۳	۱۰۰	۱۰۰
متوسط	۳۱	۸۱	۱۰۰	۱۰۰

مسئله تشخیص اعداد چاپی فارسی : الگوریتم ۳روی۶ شبکه با وزنهای اولیه تصادفی آزمایش شده اند و نتایج آن درجدول ۱۱ نشان داده شده است. برای تمامی این شبکه ها از اتوماتان HWLA (2, 7, 480) استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۴، ضریب گشتاور برابر صفر ، ضریب کاهش وزن برابر ۰/۹ ، ضریب پهنای روشنی برابر ۰/۱۵، و ضریب پهنای خاموشی برابر ۰/۱۵ در نظر گرفته شده است.

جدول ۱۱: نتیجه شبیه سازی الگوریتم ۳ برای مثال ۳

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۶	۱۵۸	۹۹/۳۷۵	۱۰۰
۲	۵۷	۱۴۴	۹۹/۳۷۵	۱۰۰
۳	۵۵	۱۴۹	۹۹/۳۷۵	۱۰۰
۴	۵۶	۱۶۱	۹۹/۳۷۵	۱۰۰
۵	۵۰	۱۱۱	۹۸/۷۵	۱۰۰
۶	۵۶	۱۶۰	۹۸/۷۵	۱۰۰
متوسط	۵۵	۱۴۷/۱۶	۹۹/۱۶	۱۰۰

شکل ۲۵ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول ۱۱ نشان میدهد.



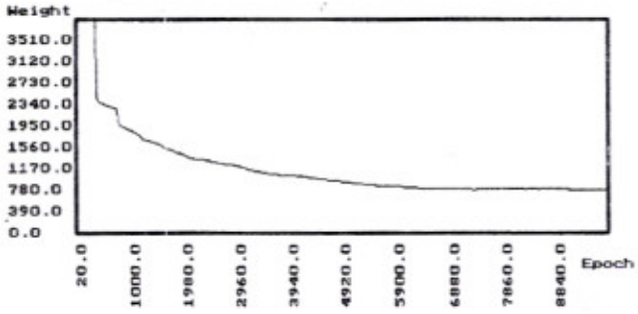
شکل۲۵ : تغییرات تعدادوزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

مسئله تشخیص اعداد لاتین: الگوریتم ۳ روی۶ شبکه با وزنهای اولیه تصادفی آزمایش شده اند و نتایج آن درجدول ۱۲ نشان داده شده است. برای تمامی این شبکه ها از اتوماتان HWLA (2, 20, 3900) استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر ، ضریب کاهش وزن برابر ۰/۹ ، ضریب پهنای روشنی برابر ۰/۱۲۵، و ضریب پهنای خاموشی برابر ۰/۱۲۵ در نظر گرفته شده است.

جدول ۱۲ : نتیجه شبیه سازی الگوریتم ۳ برای مثال ۴

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۶۰	۷۷۱	۱۰۰	۱۰۰
۲	۶۰	۸۰۲	۱۰۰	۱۰۰
۳	۶۰	۷۷۴	۱۰۰	۱۰۰
۴	۶۰	۷۸۴	۱۰۰	۱۰۰
۵	۶۰	۷۸۶	۱۰۰	۱۰۰
۶	۶۰	۷۸۷	۱۰۰	۱۰۰
متوسط	۶۰	۷۸۴	۱۰۰	۱۰۰

شکل ۲۶ منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول ۱۲ نشان میدهد.



شکل ۲۶: تغییرات تعدادوزنهای لایه مخفی شبکه براساس تکرار ارایه الگو

تذکر ۲: نتایج آزمایشات برای هرسه الگوریتم پیشنهادی نشان می‌دهند با افزایش تعداد واحدهای ورودی شبکه، متوسط نسبت تعداد وزنهای روشن به کل وزنهای آن واحد در ابتدای آموزش (تعداد واحدهای ورودی شبکه) برای واحدهای مخفی کاهش می‌یابد. با توجه به اینکه

حذف یک وزن بین یک واحد ورودی (ویژگی) و یک واحد مخفی بمنزله از بین رفتن وابستگی ابر صفحه تولید شده توسط واحد مخفی به آن ویژگی می‌گردد(زمانیکه وزنهای متصل به یک واحد ورودی حذف گردد آن واحد ورودی (ویژگی) برای شناسایی مهم نیست و می‌تواند حذف شود). این بدان معنی است که این الگوریتم‌ها مواقعی کارایی بالایی خواهند داشت که تعداد واحدهای ورودی شبکه زیاد باشد. از این خاصیت می‌توان برای تعیین یک زیر مجموعه از مجموعه ویژگیهای مناسب برای جداسازی الگوها استفاده نمود. بعنوان مثال، در شناسایی نوری حروف، ویژگیهای گشتاورها، گشتاورهای مرکزی، گشتاورهای مقیاس شده و ثابت های گشتاوری از تصویر حروف استخراج شده از تصویر میتوانند مورد استفاده قرار گیرند. برای تعیین بهترین زیرمجموعه از مجموعه ویژگیهای فوق، میتوان تمام این ویژگیهارا به شبکه ارایه نموده و از طریق الگوریتم تعیین ساختار شبکه، بهترین زیر مجموعه ویژگیها را برای کاربرد مورد نظر انتخاب نمود.

تذکر ۳: در الگوریتم بقا (قسمت اول الگوریتم ۱)، در ابتدا واریانس فعالیت واحدهای مخفی تقریبا مشابه هستند و بیشتر آنها مناسب تشخیص داده‌شده و روشن می مانند. اما بتدریج در حین آموزش، واریانش فعالیت تعدادی از واحدهای مخفی افزایش و تعدادی آزآنها کاهش می‌یابد (واحدهایی که واریانس فعالیت آنها افزایش میابد واحدهایی هستند که نقش‌موثر و مثبتی در عملکرد شبکه دارا هستند و واحدهایی که واریانس فعالیت آنها کاهش می یابد نقش موثری در عملکرد شبکه دارا نیستند) و بدین ترتیب قسمت هایی از شبکه که فعالیت بیشتری از خود نشان میدهند بتدریج کوچکتر میشود. همین پدیده در مورد وزنها نیز اتفاق میافتد. در ابتدا، مقادیر همه وزنها تقریبا یکسان هستند(زیرا در ابتدای آموزش وزنها بصورت تصادفی، از اعداد کوچکی انتخاب و مقدار دهی می‌شوند) که با پیشرفت آموزش ، مقادیر تعدادی از وزنها افزایش و مقادیر تعدادی از آنها کاهش‌می‌یابد. نتایج آزمایشات نشان داده است که در حین آموزش، از تعداد وزنهایی که مقدار آنها بزرگ است کاسته می‌شود. پدیده فوق را میتوان به این صورت نیز بیان نمود که در ابتدای آموزش همه قسمت های شبکه فعال هستند و بتدریج و با پیشرفت آموزش بخش هایی از شبکه غیر فعال میگردند و در انتها بخش کوچکی از شبکه فعال باقی می‌ماند. در این بخش کوچک از شبکه، واریانس فعالیت واحدها بالا و وزنها دارای مقادیر بزرگ هستند. با بررسی‌های انجام شده روی مغز انسان، دانشمندان به این نتیجه رسیده اند که پدیده ای مشابه پدیده فوق در مغز انسان در زمان یادگیری نیزرخ میدهد[1].

تذکر ۴: با توجه به نتایج شبیه سازیهای انجام گرفته برای مسائل ذکر شده میتوان نتیجه گرفت که : (۱) از بین سه الگوریتم ارایه شده، الگوریتم ۱ شبکه های کوچکتری را تولید می‌نماید. (۲) الگوریتم ۲ در صورتیکه تعداد تقریبی واحدهای مخفی از قبل مشخص باشد نتایجی مشابه الگوریتم ۱ را تولید میکند. (۳) شبکه های تولید شده توسط الگوریتم ۲ دارای کارایی‌پایین‌تری نسبت به شبکه های تولید شده توسط الگوریتم ۱ هستند. (۴) در الگوریتم ۲ در صورتیکه تعداد واحدهای مخفی اولیه نسبتا زیاد باشد، ساختار تعیین شده برای شبکه دارای تعداد واحد های مخفی زیادی خواهد بود. نتیجه (۳) بدین علت است که الگوریتم ۱، آموزش را از یک شبکه بزرگ شروع می‌کند و سپس در حین آموزش واحدهای مخفی را کم یا زیاد می‌نماید و در نتیجه از مزایای در حین آموزش شبکه های بزرگ (پیچیدگی آموزش کم، دوری از حداقل های محلی) بیشترین بهره را می‌برد.

تذکر ۵: با توجه به جداول ۵ تا ۸ مشاهده می‌گردد که نسبت متوسط تعداد وزن ورودی بر واحد مخفی در قسمت (الف) این جداول بیشتر از قسمت (ب) آنها می‌باشد و این تفاوت با افزایش تعداد واحدهای ورودی شبکه (تعداد ویژگیهای الگوها) افزایش می‌یابد. یک دلیل این افزایش این است‌که شبکه‌حداکثرکوشش‌خودرادرحفظ وزنهای موجود می‌نماید. در صورتیکه شبکه های بزرگ نسبت به حذف وزن مقاوم تر هستند و بهمین دلیل این نسبت کاهش می‌یابد.

تذکر ۶: در الگوریتم ۳ وزنها بصورت عمومی و با همه وزنهای روشن واحدهای روشن لایه مخفی مقایسه میگرند و یک وزن ممکن است بصورت اشتباه حذف گردد زیرا ممکن است یک وزن نسبت به وزنهای ورودی همان واحد مخفی دارای مقدار بزرگی و در نتیجه برای دسته بندی مورد نیاز باشد اما یک واحد مخفی دیگر دارای وزنهای نسبتا بزرگ باشد و یکی از وزنهای این واحدبرای‌دسته بندی لازم نباشد. مقایسه این دو وزن کوچک(وزن مناسب) و بزرگ (وزن نامناسب)، باعث حذف وزن کوچکتر گردد. در نتیجه الگوریتم ۳ با احتیاط بیشتری وزنها را حذف میکند که باعث تولید شبکه‌های بزرگتری می‌گردد.

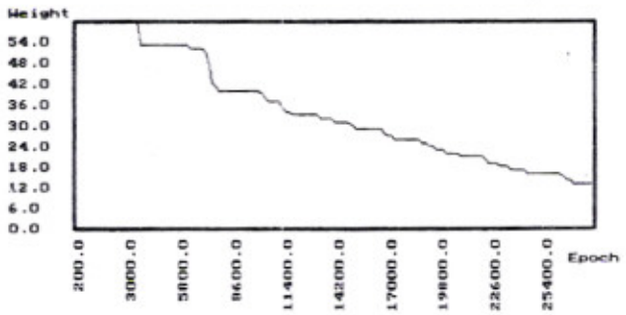
تذکر۷: در این قسمت الگوریتم زمانبندی دیگری برای نحوه فعالیت اتوماتانهای یادگیر تعیین تعداد واحدها و وزنهای لایه مخفی شبکه عصبی سه لایه بیان و نتایج آن برای یک مسئله نمونه ارایه میگردد. در این الگوریتم، درحین‌آموزش، هردواتوماتان‌تعیین‌تعداد واحدها و وزنهای‌لایه‌مخفی‌روشن‌هستندوبطورهمزمان‌سعی‌درپیدا نمودن ساختار مناسب شبکه عصبی‌رادارند. لازم بذکر است که اتوماتان تعیین تعداد وزنهای لایه مخفی در هر مرحله فقط وزنهای متصل به واحدهای مخفی روشن را بررسی می‌کند و با تغییر وضعیت آنها جهت یافتن ساختار مناسب حرکت می‌کند. این‌الگوریتم برای مسئله تشخیص اعداد لاتین روی ۶ شبکه آزمایش شده است و نتایج آزمایش‌ها در جدول ۱۳ نشان داده شده است. برای آزمایشهای انجام شده در این جدول، از اتوماتان HULA (2, 20, 60)-HWLA (2, 20, 3900) استفاده‌گردیده است. دراین آزمایشهامقدارنرخ‌یادگیری‌برابر ۰/۰۱ وضریب‌گشتاور برابر ۰/۱ در الگوریتم‌انتشارخطابه‌عقب، ضریب پهنای روشنی و ضریب پهنای خاموش بترتیب برابر ۰/۰۵ و ۰/۰۵ در برای تعیین تعداد واحدهای لایه مخفی و ضریب پهنای روشنی و ضریب پهنای‌خاموش بترتیب برابر ۰/۵ و ۰/۵ در برای تعیین تعداد وزنهای لایه مخفی استفاده گردیده است.

۱- در بررسی هایی که روی مغز انسان به انجام رسیده است مشاهده شده است که الگوی فعالیت گسترده در اوایل فراگیری و سپس کاسته شدن از این فعالیت گسترده پس از عادی شدن فعالیت، هم در مورد فعالیت های فیزیکی نظیر تایپ کردن و رانندگی کردن و امثال آن صادق است و هم در خصوص فعالیت های نظری و فکری مانند فراگرفتن اطلاعات تازه و یا خواندن مطالب جدید (نظیر دروس تازه) به منظور به خاطر سپردن آنها[روزنامه اطلاعات، تاریخ دوشنبه ۲۸ تیر ۱۳۷۸ صفحه ۳ ضمیمه].

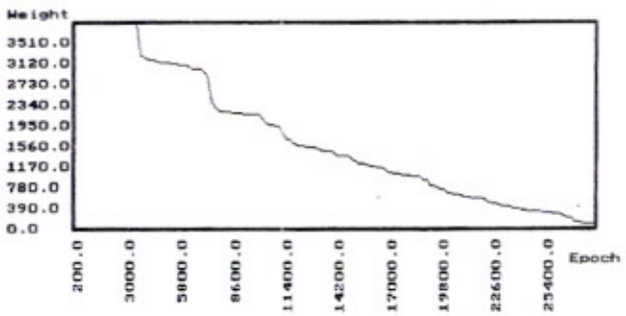
جدول ۱۳ : نتیجه شبیه سازی برای مسئله تشخیص اعداد لاتین

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۰	۵۷	۱۰۰	۱۰۰
۲	۹	۱۲۵	۱۰۰	۱۰۰
۳	۷	۱۵۳	۱۰۰	۱۰۰
۴	۱۳	۹۲	۱۰۰	۱۰۰
۵	۷	۹۶	۱۰۰	۱۰۰
۶	۱۰	۱۸۰	۱۰۰	۱۰۰
متوسط	۹/۳۳	۱۱۷/۱۶	۱۰۰	۱۰۰

شکلهای ۲۷ و۲۸ منحنی تغییرات تعدادواحدهاووزنهای لایه‌مخفی شبکه‌بر اساس تکرارارایه‌الگونشان‌داده‌شده‌است. شکل ۲۷ منحنی تغییرات تعداد واحدهای لایه مخفی و در شکل ۲۸ منحنی تغییرات تعدادوزنهای لایه مخفی شبکه را برای شبکه شماره ۱ جدول ۱۳ نشان میدهد.



شکل ۲۷: تغییرات تعدادواحدهای لایه‌مخفی شبکه براساس تکرارارایه‌الگو



شکل ۲۸: تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرارارایه‌الگو

باتوجه‌به‌جدول ۱۳ وجداول ۴، ۸ و۱۲ مشاهده‌میشودکه‌این‌الگوریتم‌بهتراز الگوریتمهای ۲ و۳ وکمی‌بدتراز‌الگوریتم ۱ عمل‌میکندوتعدادواحدهای مخفی‌درشبکه‌های‌تولیدشده‌بسیارکمتراز‌تعداد واحدهای مخفی تعیین شده در الگوریتم های ۲ و ۳ ومقداری بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ می‌باشد اما نسبت متوسط تعداد وزن بر واحد مخفی‌دراین‌الگوریتم بیشتراز‌الگوریتمهای‌پیشنهاد شده دیگر است.

تذکر ۸: هر واحد مخفی در شبکه های عصبی سه لایه یک ابر صفحه تولید[1] می‌کند و این ابر صفحه یک یا چند دسته یا قسمتی از یک دسته الگو را از بقیه دسته هایی جدا می‌کند که مجموعه ویژگیهای آنها تفاوت نسبتازیادی‌داشته باشند(مثلادر شکل ۶، خط شماره۱ با استفاده‌از ویژگی X دسته‌های {'x'، '*'} و {'#'، '+'} را ازهم‌جدا میکنددرصورتیکه ویژگی Y برای‌دسته‌های {'x'} و {'#'} ودسته‌های{'+'} و {'*'} یکسان می‌باشد. در شکل ۶ برای مجموعه ویژگی{X, Y} دو زیر مجموعه {X} و {Y} این خاصیت را دارا می باشند). بنابر این پیدا نمودن ساختار شبکه عصبی سه لایه، تبدیل به پیدا کردن کمترین تعداد زیر مجموعه‌از مجموعه ویژگیهای ورودی میگردد بطوریکه هر زیر مجموعه بیشترین اختلاف را برای دسته‌های دو طرف ابر صفحه ایجاد نماید. مثلا برای شناسایی اعداد لاتین مجموعه ویژگی $X=\{x_0, x_1, x_2, ..., x_{64}\}$ به شبکه اعمال می‌شود که x_0 دارای مقدار یک (واحد ورودی به وزن آستانه) و واحد ورودی x_k مقدار نقطه k ام تصویر را دریافت میکند. فرض کنید $H=\{h_1, h_2, . . ., h_M\}$ کوچکترین مجموعه واحدهای مخفی لازم برای یک دسته بندی صحیح الگوها باشد. مسئله پیدا کردن ساختاربهینه برای شبکه عصبی به مسئله پیدا نمودن M زیر مجموعه با کمترین تعداد اعضا از مجموعه X تبدیل می‌گردد(که k امین زیر مجموعه، وزنهای متصل به واحد مخفی شماره h_k را نشان می‌دهد) بطوریکه شبکه تولید شده بتواند دسته بندی با دقت بالارا انجام‌دهد. مثلا برای‌شبکه‌شماره۲جدول۴-ب که توسط الگوریتم یک بدست آمده است، مجموعه واحدهای مخفی تعیین شده برابر $H=\{۴،۱۲،۱۳،۱۸،۲۷،۳۸،۵۹\}$ میباشد. مجموعه‌وزنهای‌ورودی‌برای واحدهای مخفی شماره ۴، ۱۲، ۱۳، ۱۸، ۲۷، ۳۸، ۵۹ بترتیب عبارتند از :{۳۸ ، ۱۸، ۱۳، ۴}، {۱۸، ۶۴}، {۴،۱۶،۱۸،۳۰،۳۸،۵۶}، {۱۸،۵۹،۶۴}، {۶۱،۶۲،۶۴}، {۱۸،۳۸،۵۹،۶۱} و {۱۸،۳۸}.

مقایسه‌الگوریتمهای‌پیشنهادی‌باالگوریتم‌هرس‌کارنین[1]: دراین قسمت‌الگوریتمهای‌پیشنهادی‌باالگوریتم‌هرس که‌توسط کارنین‌پیشنهاد گردیده است[۵۴] مقایسه‌میگردند. این الگوریتم به شرح زیر میباشد.

در این‌الگوریتم، ابتداشبکه‌بطور کامل آموزش داده‌میشودودرپایان آموزش، حساسیت‌هروزن(تاثیرمنفی حذف یک‌وزن در تابع‌هدف‌شبکه)محاسبه میگرددووزنهایی که‌دارای‌حساسیت کمتری هستندحذف میگردند. در این الگوریتم مقدار حساسیت یک وزن (S_{ij}) بصورت زیر محاسبه میگردد.

$$S_{ij} = - \frac{E(w^f) - E(0)}{w^f - 0} w^f$$

که w^f مقدار نهایی‌وزن در پایان دوره آموزش، 0 مقدار وزن پس‌ازحذف وزن w^f، $E(w^f)$ مقدار خطای شبکه در هنگام موجود بودن وزن w^f و $E(0)$ مقدار خطا پس از حذف وزن w^f میباشد. برای محاسبه حساسیت یک وزن میتوان آن وزن را حذف نموده و مقدار خطا و در نتیجه مقدارحساسیت برای وزن حذف شده را محاسبه نمود. با توجه به اینکه محاسبه مقدار حساسیت برای وزن حذف شده نیاز به تعداد ارایه های مجموعه آموزشی از مرتبه $O(W^2)$ دارد که W تعداد وزنهای شبکه است. مقدار حساسیت تخمین زده شده بصورت زیر محاسبه میگردد.

$$\hat{S}_{ij} = - \sum_{n=0}^{N-1} \frac{\partial E}{\partial w_{ij}} \Delta w_{ij} \frac{w_{ij}^f}{w_{ij}^f - w_{ij}^i}$$

که N تعداد مراحل یک دوره آموزش، w_{ij}^f مقدار نهایی وزن w_{ij} و w_{ij}^i مقدار اولیه وزن w_{ij} میباشند. تمام اطلاعات مورد نیاز برای محاسبه فوق در زمان آموزش فراهم می‌باشد. پس از یک دوره آموزش ، برای هر وزن یک مقدار تخمینی از حساسیت محاسبه میشود و وزنی

1- Karnin

که دارای کمترین حساسیت هست حذف میگردد. قابل ذکر است اگر تمام وزنهای خروجی یک واحد حذف گردند آن واحد حذف میگردد. اگر همه وزنهای ورودی یک واحد حذف شوند خروجی این واحد دارای مقدار ثابتی میشود و میتوان این واحد را حذف نمود و اثر خروجی آنرا در مقدار آستانه واحد های لایه بعد اعمال نمود(شکل ۱۵). لازم بذکر است که در تمامی آزمایشهای انجام گرفته با استفاده از الگوریتم کارنین ابتدا شبکه بطور کامل آموزش داده شده است بطوریکه قادر به شناسایی کامل مجموعه آموزشی باشدو سپس در مرحله هرس ، پس از حذف یک وزن شبکه باز آموزی گردیده است. در مرحله هرس وزنها، ۵۰۰۰۰ بار مجموعه آموزشی به شبکه ارایه شده است ودر انتها شبکه ای را که بهترین ساختار رادارد انتخاب و در جداول ۱۳ تا ۱۶ گزارش شده است.

مسئلهدستهبندی چهار دسته از الگوها : برای پیاده سازی این مسئله ۶ شبکه با ۲ واحد ورودی ، ۳۹ واحد مخفی و ۴ واحد خروجی در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین، هرس گردیده اند. پس از حذف وزن با کمترین حساسیت، ۱۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن درجدول ۱۴ آورده شده است. برای آزمایشهای انجام شده مقدار نرخ یادگیری برابر ۰/۱ و مقدار ضریب گشتاور برابر۰/۶ میباشد.

جدول ۱۴ : نتیجه شبیه سازی الگوریتم کارنین برای مثال ۱

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۸	۸۳	۱۰۰	۱۰۰
۲	۲۷	۷۹	۱۰۰	۱۰۰
۳	۲۹	۸۶	۱۰۰	۱۰۰
۴	۳۱	۹۱	۱۰۰	۱۰۰
۵	۳۳	۹۷	۱۰۰	۱۰۰
۶	۳۲	۹۴	۱۰۰	۱۰۰
متوسط	۳۰	۸۸/۳۳	۱۰۰	۱۰۰

با توجه به جداول ۱-ب، ۵-ب، ۹ و ۱۴ مشاهده میشود که تعداد وزنهای تعیین شده توسط الگوریتم کارنین برای لایه مخفی تقریبا ۱۰ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۱، ۲ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۲ و ۳ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۳ میباشد که این خود تاییدی بر برتری الگوریتم های پیشنهاد شده میباشد. لازم بذکر است که در شبکههایتولید شده توسط الگوریتم هرس کارنین تقریبا همه واحدها دارای ۳ وزن ورودی هستند این بدان معناست که این لگوریتم در اینمورد شبکه ای با اتصالات محلی را تولید نمیکند.

مسئله دسته بندی دو دسته از الگوها : برای پیاده سازی این مسئله ۶ شبکه با ۲ واحد ورودی ، ۶۰ واحد مخفی و ۲ واحد خروجی دو نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین، هرس گردیده اند. پس از حذف یک وزن در شبکه، ۵۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن درجدول ۱۵ آورده شده است.

لازم بذکر است که درآزمایشهای جدول ۱۵ مقدار نرخ یادگیری برابر۰/۰۱ و مقدار ضریب گشتاور برابر صفر در نظر گرفته شده است.

جدول ۱۵ : نتیجه شبیه سازی الگوریتم کارنین برای مثال ۲

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۳	۱۵۸	۱۰۰	۱۰۰
۲	۵۱	۱۵۲	۱۰۰	۱۰۰
۳	۵۳	۱۵۷	۱۰۰	۱۰۰
۴	۴۶	۱۳۷	۱۰۰	۱۰۰
۵	۵۰	۱۵۰	۱۰۰	۱۰۰
۶	۴۹	۱۴۷	۱۰۰	۱۰۰
متوسط	۵۰/۳	۱۵۰/۱	۱۰۰	۱۰۰

با توجه به جداول ۲-ب، ۶-ب، ۱۰ و ۱۴ مشاهده میشود که تعداد وزنهای لایه مخفی تقریبا ، ۲۰ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۱، ۲ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۲ وبیشتراز ۲ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۳ میباشد. همچنین در شبکههایتولید شده توسط الگوریتم هرس کارنین تقریبا همه واحدها دارای ۳ وزن ورودی هستند. مشاهده شده است که برای مسائل ۱ و ۲ در بیشتر موارد الگوریتم کارنین، وزنها را بطریقی هرس مینماید که وزنهای هرس شده متوالیا متعلق بهیک واحد مخفی میباشند و تا حذف کامل این واحد ادامه پیدا میکند(جداول ۱۴ و ۱۵) و بهمین دلیل این الگوریتم در بیشتر موارد برای مسائل ۱ و۲ شبکه ای با اتصالات محلی را تولید نمینماید.

مسئله تشخیص اعداد چاپی فارسی : برای پیاده سازی این مسئله ۶ شبکه با ۷ واحد ورودی، ۶۰ واحد مخفی و ۱۰ واحد خروجی در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین هرس گردیده اند. پس از حذف یک وزن در شبکه، ۱۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن درجدول ۱۶ آورده شده است. برای آزمایشهای انجام شده مقدار نرخ یادگیری برابر۰/۰۵ و مقدار ضریب گشتاور برابر صفر میباشد.

جدول ۱۶ : نتیجه شبیه سازی الگوریتم کارنین برای مثال ۳

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۸	۲۷۵	۹۸/۱۲۵	۹۰
۲	۵۸	۱۳۰	۹۶/۶۲۵	۱۰۰
۳	۵۹	۲۴۰	۹۶/۸۷۵	۹۰
۴	۵۶	۱۵۴	۹۶/۶۲۵	۱۰۰
۵	۶۰	۲۱۶	۹۸/۱۲۵	۱۰۰
۶	۶۰	۱۸۰	۹۹/۳۷۵	۱۰۰
متوسط	۵۸/۵	۱۹۹/۱۶	۹۷/۶۲۵	۹۶/۶۶

با توجه به جداول ۳-ب، ۷-ب، ۱۱ و ۱۶ مشاهده میشود که تعداد وزنهای تعیین شده توسط الگوریتم کارنین برای لایه مخفی تقریبا ۱۰

برابر تعداد وزنهای تعیین شده توسط الگوریتم ۱، بیشتر از ۲ برابر تعداد وزنهای تعیین شده توسط الگوریتم ۲ و مقداری بیشتر از تعداد وزنهای تعیین شده توسط الگوریتم ۳ می‌باشد. نکته جالب توجه این است که در این مورد شبکه‌های تولید شده توسط الگوریتم هرس کارنین برای مسئله تشخیص اعداد چاپی فارسی و برخلاف مسائل قبل، شبکه هایی با اتصالات محلی هستند.

مسئله تشخیص اعداد لا تین : برای پیاده سازی این مسئله ۶ شبکه با ۶۴ واحد ورودی ۶۰ واحد مخفی و ۱۰ واحد در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین هرس گردیده اند. پس از حذف یک وزن در شبکه، ۵۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن درجدول ۱۷ آورده شده است. برای آزمایشهای جدول ۱۷ مقدار نرخ یادگیری برابر ۰/۰۱ و مقدار ضریب گشتاور برابر صفر می‌باشد.

جدول ۱۷ : نتیجه شبیه سازی الگوریتم کارنین برای مثال ۴

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصدتشخیص (آموزش)	درصدتشخیص (آزمایشی)
۱	۶۰	۳۸۴۰	۱۰۰	۱۰۰
۲	۶۰	۳۸۴۲	۱۰۰	۱۰۰
۳	۶۰	۳۸۴۰	۱۰۰	۱۰۰
۴	۶۰	۳۸۳۹	۱۰۰	۱۰۰
۵	۶۰	۳۸۵۰	۱۰۰	۱۰۰
۶	۶۰	۳۸۴۸	۱۰۰	۱۰۰
متوسط	۶۰	۳۸۴۳/۱	۱۰۰	۱۰۰

باتوجه‌به‌جدول ۱۷ مشاهده‌میشود که‌الگوریتم کارنین تعداد کمی از وزنهای لایه‌مخفی را حذف‌نموده‌است. درهنگام‌شبیه‌سازیها‌مشاهده گردید که‌با حذف‌اولین واحدمخفی، کارایی شبکه‌بشدت کاهش‌یافته‌وشبکه‌بهیچوجه قادربه‌بازآموزی‌نمی گردد. همچنین‌دربسیاری‌ازشبیه‌سازیهامشاهده‌شده‌که بدون‌اینکه‌واحدمخفی حذف گردد کاهش‌زیادی‌درکارایی شبکه‌رخ‌میدهد. شبکه‌های‌ارایه‌شده‌درجدول ۱۷ نماینده‌شبکه‌های تولیدشده‌توسط‌الگوریتم کارنین‌قبل‌ازشروع کاهش‌شدید کارایی‌میباشد. مثال فوق، عدم کارایی الگوریتم هرس کارنین‌رادراین‌مسئله‌نشان‌میدهد. الگوریتمهای‌پیشنهادی روی‌مسائل‌متنوعی‌آزمایش شده‌اند که‌نتایج کامل آنهادرمرجع [۴۹] آمده‌است.

۷- نتیجه گیری

دراین‌مقاله‌سه‌الگوریتم‌براساس‌اتوماتانهای‌یادگیرمهاجرت‌اشیابرای‌تعیین ساختار شبکه‌های عصبی‌سه‌لایه‌ارایه گردیده‌است. این‌الگوریتم‌ها تعداد واحدهاوتعدادوزنهای‌لایه‌مخفی‌یک‌شبکه‌عصبی سه‌لایه‌را‌تعیین‌میکنند. الگوریتمهای‌پیشنهادی‌ازاتوماتانهای‌مهاجرت‌اشیا والگوریتم یادگیری انتشارخطابه‌عقب‌برای‌تعیین ساختارشبکه‌استفاده می کنند. الگوریتمهای پیشنهادی، شبکه‌هایی‌باتصالات‌محلی که‌دارای‌قدرت‌تعمیم‌بالایی‌هستند تولیدمی‌کنند. کارایی‌الگوریتمهای‌پیشنهادی‌ازطریق‌شبیه سازی‌روی مسائل‌مختلف‌نشان‌داده‌شده‌است. اتوماتانهای‌یادگیربعنوان‌یک‌ابزار جستجوی‌عمومی، زمانی‌کارایی‌بالایی‌ازخودنشان‌میدهدکه‌فضایی که‌آنها جستجومیکنندبزرگ‌باشد. بهمین‌دلیل‌الگوریتمهای‌پیشنهادشده‌دراین مقاله‌که‌مبتنی‌براتوماتانهای‌یادگیرطراحی‌شده‌اندزمانی‌دارای‌کارایی‌بالایی هستندکه‌فضای‌جستجوی‌آنها (تعدادواحدهای‌لایه‌ورودی‌شبکه) بزرگ باشد. الگوریتمهای‌پیشنهادی‌باالگوریتم‌هرس کارنین‌مقایسه گردیده‌است و برتری‌الگوریتمهای‌پیشنهادی‌نسبت‌به‌این‌الگوریتم‌ازطریق‌شبیه سازی نشان داده شده است.

مراجع

[1] Angeline, P. J., Saunders, G. M., and Pollack, J. B. (1994). Evolutionary Algorithm that Construct Recurrent Neural Networks, IEEE Trans. on Neural Networks, Vol. 5, No. 1, pp. 54-65.

[2] Arai, M. (1993). Bounds on the Number of Hidden Units in Binary-Valued Three-Layer Neural Networks, Neural Networks, Vol. 6, pp. 855-860.

[3] Beigy, H. and Meybodi, M. R. (1998). A Fast Method for Determining the Number of Hidden Units in Feedforward Neural Networks, Proc. of CSIC-97, Tehran, Iran , pp. 414-420 (In Persian).

[4] Beigy. H. and Meybodi, M. R. (1998). Optimization of Topology of Neural Networks: A Survey, Technical Reports, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).

[5] Castellano, G., Fanelli, A. M., and Pelillo, M. (1997). A Iterative Pruning Algorithm for Feedforward Neural Networks, IEEE Trans. on Neural Networks, Vol. 8, No. 3, pp. 519-531.

[6] Fahlman, S. E. and Lebier, C. (1990). The Cascade - Correlation Learning Architecture, Advances in Neural Information Processing System, Vol. II, pp. 524-532.

[7] Frean, M. (1990). The Upstart: A Method for Constructing and Training Feedforward Neural Networks, Neural Computation, pp. 198-209.

[8] Hashim, A. A., Amir, S., and Mars, p. (1986). Application of Learning Automata to Data Compression, In Adaptive and Learning Systems, K. S. Narendra (Ed.), New York: Plenum Press, pp. 229-234.

[9] Hirose, Y., Yamashita, K., and Hijya, S. (1991). Back-Propagation Algorithm Which Varies The Number of Hidden Units, Neural Networks, Vol. 4, No. 1, pp. 61-66.

[10] Huang, S. C. and Huang, Y. F. (1991). Bounds on the Number Hidden Neurons in Multilayer Preceptrons, IEEE Trans. on Neural Networks, Vol. 2, No. 1, pp. 47-56.

[11] Kruschke, J. H. (1988). Creating Local and Distributed Bottlenecks in Hidden Layer of Backpropagation Networks, Proc. of Connectionist Models, Summer School, Eds. D. Tourestzky, G. Hinton, and T. Sejnowski, pp. 120-126.

[12] Kruschke, J. H. (1989). Improving Generalization in Backpropagation Networks, Proc. of Int. Joint Conf. on Neural Networks, Vol. I, pp. 443-447.

[13] Kwok, T. Y. & Yeung, D. Y. (1997). Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems, IEEE Trans. on Neural Networks, Vol. 8, No. 3 pp. 630-645.

[14] Lin, J. H. & Vitter, J. S. (1991). Complexity Results on Learning by Neural Nets, Machine Learning, Vol. 6, pp. 211-230.

[15] Maniezzo, V. (1994). Genetic Evolution of The Topology and Weight Distribution of Neural Networks, IEEE Trans. on Neural Networks, Vol. 5, No. 1, pp. 39-53.

[16] Marchand, M., Golea, M., and Rujan, R (1990). A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons, Europhysics Letters 11, pp. 487-492.

Algorithms, Proc. of Int. Joint Conf. on Neural Networks, Vol. I, pp. 134.

[38] Yao, X. and Liu, Y. (1997). A New Evolutionary System for Evolving Artifitial Neural Networks, IEEE Trans. on Neural Networks, Vol. 8, No. 3, pp. 694-713.

[39] Yeung, D. Y. (1991). Automatic Determination of Network Size for Supervised Learning, IEEE Int. Joint Conf. on Neural Networks , pp. 158-164.

[40] Yu, X. H. (1992). Can Backpropagation Error Surface Not Have Local Minima, IEEE Trans. on Neural Networks, Vol. 3, No. 6, pp. 1019-1021.

[41] Lippman, R. P. (1987). An Introduction to Computing with Neural Nets, IEEE ASSP Mag., Vol. 4, pp. 4-22.

[42] Beigy. H. and Meybodi, M. R. (1999), Graph Isomorphism Using Learning Automata, Technical Reports, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).

[43] Beigy. H. and Meybodi, M. R. (1999), Optimization of Topology of Neural Networks Using Learning Automata", Proc. of 4th Annual Int. Computer Society of Iran Computer Conf. CSICC-98, Tehran, Iran, pp. 417-428 (In Persian).

[44] Meybodi, M. R. and Lakshmivarhan, S. (1982), Optimality of a General Class of Learning Algorithm, Information Science, Vol. 28, pp. 1-20.

[45] Meybodi, M. R. and Lakshmivarhan, S. (1984), On a Class of Learning Algorithms which have a Symmetric Behavior Under Success and Failure, Springer Verlag Lecture Notes in Statistics, pp. 145-155.

[46] Meybodi, M. R. (1987), Results on Strongly Absoulutely Expedient Learning Automata, Proc. of OU Inference Conf. 86, ed. D. R. Mootes and R. Butrick, Athens, Ohio: Ohio University Press, pp. 197-204.

[47] Lakshmivarahan, S. (1981), Learning Algorithms: Theory and Applications, New York, Springer-verlag.

[48] Dastpak, V. (1992), Automatic Recognition of Farsi Printed Letters, Ms. Thesis, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).

[49] Meybodi, M. R. and Beigy, H. (1999), Neural Network Engineering Using Learning Automata: Determination of Desired Size for Three Layer Feedforward Neural Network, Technical Reports, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).

[50] Judd, J. S. (1990), Neural Network Design and the Complexity of Learning, Cambridge, MA: MIT Press.

[51] Frasconi, P., Gori, M. and Tesi, A. (1992), Successes and Failures of Backpropagation: A Theoritical Investigation, , Technical Reports, Dipartimento di Sistemi e Information, Universita di Firenze, Firenze, Italy.

[52] Hush, D. R. and Horne, B. G., (1993), Progress In Supervised Neural Networks, IEEE Signal Processing Magazine, Jan. 1993, pp. 8-39.

[53] Schalkoff, R. J. (1992), Pattern Recognition: Statistical, Structural and Neural Approaches, John Whiley, New York.

[54] Karnin, E. D. (1990), A Simple Procedure for Pruning Backpropagation Trained Neural Networks, IEEE Trans. on Neural Networks, Vol. 1, No. 2, pp. 239-242.

[55] Ji, C., Snapp, R. R., and Psalatis (1990), Generalizing Smoothness Constraints from Discrete Samples, Neural Computation, Vol. 2, No. 2, pp. 188-197.

[17] Mars, P., Chen, J. R., and Nambiar, R. (1998). Learning Algorithms: Theory and Application in Signal Processing, Control, and Communications, CRC press, New York.

[18] Mars. P. and Narendra. K. S., and Chrystall, M. (1983). Learning Automata Control of Computer Communication Networks, Proc. of Third Yale Workshop on Applications of Adaptive Systems Theory, Yale University.

[19] Meltser, M., Shoham, M., and Manevitz, L. M. (1996). Approximating Function by Neural Networks: A Constructive Solution in the Uniform Norm, Neural Networks, Vol 9, No. 6, pp. 965-978.

[20] Meybodi, M. R. and Beigy, H. (1998). New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters, Proc. of EUFIT-98, Achen, Germany, pp. 339-344.

[21] Meybodi. M. R. and Lakshmivarhan, S. (1983). A Learning Approach to Priority Assignment in a Two Class M/M/1 Queuing System with Unknown Parameters, Proc. of Third Yale Workshop on Applications of Adaptive Systems Theory, Yale University, pp. 106-109.

[22] Mezard, M. and Nadal, J. P. (1989). Learning in Feedforward Neural Networks: The Tiling Algorithm, Journal of Physics, pp. 1285-1296.

[23] Minor, J. M. (1993). Parity With Two Layer Feedforward Nets, Neural Networks, Vol. 6, No. 5, pp. 705- 707.

[24] Nabhan, T. M. and Zomaya, A. Y. (1994). Toward Neural Networks Structures for Function Appriximation, Neural Networks, Vol. 7, No. 1, pp. 89-99.

[25] Narendra, K. S. and Thathachar, M. A. L. (1989). Learning Automata : An Introduction, Prentice-hall, Englewood cliffs.

[26] Oommen, B. J. and Ma, D. C. Y.(1988). Deterministic Learning Automata Solutions to the Equipartitioning Problem, IEEE Trans. on Computers, No. 37, No. 1, pp. 2-13.

[27] Oommen, B. J., Valiveti, R. S., and Zgierski, J. R. (1991). An Adaptive Learning Solution to the Keyboard Optimization Problem, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 21, No. 6, pp. 1608-1618.

[28] Oommen, B. J. and Croix, E. V. de St. (1996). Graph Partitioning Using Learning Automata, IEEE Trans. on Computers, No. 45, No. 2, pp. 195-208.

[29] Rumelhart, D. E., Hinton, G. E., and Williams, R. J.(1986). Learning Internal Representations by Error Backropagation, In Parallel distributed processing, Cambridge, MA: MIT Press.

[30] Reed, R. (1993). Prunning Algorithms - A Survey, IEEE Trans. on Neural Networks, Vol. 4, No. 5, pp. 740-747.

[31] Sietsma, J. and Dow, R. J. F. (1991). Creating Artifitial Neural Networks that Generalize, Neural Networks, Vol. 4, No. 1, pp. 67-79.

[32] Sirat, J. A. and Nadal, J. P. (1990). Neural Trees: A New Tool for Classification, Preprint, Laboratories d'Electronique, Philips, Limeil Brevannes, France.

[33] Schaffer, J. D., Whitely, D., and Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: A Survey of the state of the art, IEEE Proc. COGANN-92, pp. 1-37.

[34] Sperduti, A. and Starita, A. (1993). Speed Up Learning and Network Optimization with Extended Backpropagation, Neural Networks, Vol. 6, pp. 365-383.

[35] Tamura, S. and Tateishi, M. (1997). Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three, IEEE Trans. on Neural Networks, Vol. 8, No. 2, pp. 251-255.

[36] Thathachar, M. A. L. and Sastry, P. S. (1987). Learning Optimal Discriminant Functions Throught a Cooperative Game of Automata, IEEE Trans. Syst., Man and Cybern., Vol. SMC-27, pp.73-85.

[37] Whitley, D. and Bogart, C. (1990). The Evolution of Connectivity: Pruning Neural Networks Using Genetic

دانشگاه شهید بهشتی

انجمن کامپیوتر ایران
Computer Society of Iran

مجموعه مقالات

پنجمین کنفرانس بین المللی سالانه

انجمن کامپیوتر ایران

دانشکده مهندسی برق و کامپیوتر

دانشگاه شهید بهشتی

۱۷ الی ۱۹ اسفند ۱۳۷۸

