

ارائه روشی نو مبتنی بر اتوماتان‌های یادگیر برای تطبیق پارامترهای الگوریتم یادگیری VLR¹ برای آموزش شبکه‌های عصبی MLP²

محمد باقر منهاج
دانشیار
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

بهبود مشعوفی
دانشجوی دکتری
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

محمد رضا میبدی
دانشیار
دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

سید احمد معتمدی
دانشیار
دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

چکیده

الگوریتم BP³ برای طیف وسیعی از کاربردها مورد استفاده قرار گرفته است. یکی از مشکلات عمده این الگوریتم، پائین بودن نرخ همگرایی آن می باشد. دلیل عمده، اشباع شدن تابع فعالیت می باشد. به محض اینکه خروجی یک واحد در ناحیه اشباع قرار می گیرد، گردایان متناظر دارای مقدار کوچکی خواهد بود. اگر از نرخ یادگیری ثابت کوچکی استفاده شود این مسئله می تواند باعث پائین آمدن سرعت همگرایی شود. یک راه حل برای افزایش سرعت همگرایی، استفاده از نرخ یادگیری بزرگ می باشد. متأسفانه این روش در مواقعی که در نقطه ای از سطح خطا قرار داریم که شیب تندی دارد بسیار خطرناک بوده و می تواند باعث نوسان و واگرایی شود. لذا نیاز به الگوریتم‌هایی داریم که بتواند بطور پویا نرخ همگرایی را بر اساس تغییرات مقادیر گردایان تغییر دهد. در این مقاله روش‌های مختلف تغییر دینامیک نرخ یادگیری مورد بررسی قرار گرفته است. الگوریتم نرخ یادگیری متغیر (VLR) و الگوریتم‌های تطبیق نرخ یادگیری مبتنی بر اتوماتان‌های یادگیر توضیح داده شده و بر روی مسائل مختلف پیاده و با هم مقایسه شده اند. نظر به اینکه در روش VLR پارامترهای مورد استفاده در این روش تأثیر بسزایی در عملکرد آن دارد لذا در این مقاله از اتوماتان یادگیر برای تنظیم پارامترهای آن استفاده کرده ایم. در الگوریتم جدید که تحت عنوان الگوریتم نرخ یادگیری متغیر تطبیقی² (AVLR) نامگذاری کرده ایم. پارامترهای الگوریتم VLR بطور دینامیک توسط اتوماتان یادگیر بر اساس تغییرات خطا تغییر داده می شود. نتایج مشابه سازی‌ها بر روی مسائل مختلف نشان می دهد الگوریتم پیشنهادی نسبت به روش‌های دیگر از سرعت همگرایی بالایی برخوردار می باشد.

کلمات کلیدی

شبکه‌های عصبی چند لایه، الگوریتم پس انتشار خطا، نرخ یادگیری متغیر، اتوماتان‌های یادگیر، نرخ یادگیری متغیر.

Introducing A Novel Learning Automata Based Method For Adapting VLR Learning Algorithm Parameters For Learning MLP Neural Networks

B. Mashoufi
Ph. D. Student
Electrical Engineering Department,
Amirkabir University of Technology

M. B. Menhaj
Associate Professor
Electrical Engineering Department,
Amirkabir University of Technology

S.A. Motamedi
Associate Professor
Electrical Engineering Department,
Amirkabir University of Technology

M. R. Meybodi
Associate Professor Computer
Engineering Department, Amirkabir
University of Technology

Abstract

BP algorithm has been used for wide range of applications. One of the most important limitations of this algorithm, is the low rate of convergence. The important reason behind this, is the saturation property of its activation functions. Once the output of a unit lies in the saturation area, the corresponding decent gradient would take a very small value. This will result in very little progress in the weight adjustment, if one takes a fixed small learning rate parameter. To avoid this undesired phenomenon, one may consider a relative large learning rate. Unfortunately this would be dangerous, because it may take the algorithm diverges, especially when the weight adjustment happens to fall into the surface regions with a large steepness. So, we require algorithms capable of tuning dynamically learning rate according to changes of gradient values. In this paper, different methods of dynamic changing of learning rate has been considered. Variable Learning Rate (VLR) algorithm and learning automata based learning rate adaptation algorithms are considered and compared with each other. Because the VLR parameters have important influence in its performance, so we use learning automata for adjusting them. In the proposed algorithm called Adaptive Variable Learning Rate (AVLR) algorithm, VLR parameters are adapted dynamically by learning automata according to error changes. Simulation results on various problems highlight better the merit of the proposed AVLR.

Keyword

Multilayer Neural Network, Backpropagation, Variable Learning Rate, Learning Automata, Adaptive Variable Learning Rate

مقدمه

کافی کوچک باشند انجام می گیرد. در آغاز به تمامی وزن ها مقادیر تصادفی کوچک داده می شود.

الگوریتم BP برای رنج وسیعی از کاربردها مورد استفاده قرار گرفته است. متأسفانه نرخ همگرایی BP مخصوصاً برای شبکه های با بیش از یک لایه مخفی بسیار پایین می باشد. دلیل اصلی پائین بودن نرخ همگرایی BP اشباع شدن تابع فعالیت مورد استفاده در واحدهای مخفی و خروجی می باشد. بمحض اینکه خروجی یک واحد در ناحیه اشباع قرار می گیرد، گرادیان متناظر اگر هم خطای خروجی خیلی بزرگ باشد دارای مقدار کوچکی خواهد بود. اگر از نرخ یادگیری ثابت کوچکی استفاده شود این مسئله می تواند باعث نرخ پائین همگرایی شود.

بمنظور جلوگیری از این پدیده نامطلوب یک راه حل استفاده از نرخ یادگیری بزرگ می باشد. این راه حل بویژه در مواقعی که در نقطه ای از سطح خطا قرار داریم که شیب تندی دارد بسیار خطرناک بوده و می تواند باعث واگرایی شود. تعدادی از محققین سعی کرده اند سطوح خطا شبکه های MLP با الگوریتم یادگیری BP را توصیف کنند [۵]-[۲]. نشان داده شده است موقعیکه شکل سطوح خطا از سطوح خطا درجه دوم خیلی فاصله داشته باشد در این صورت شامل مناطق مسطح و مناطق با شیب تند زیادی خواهد بود. در این

مسئله آموزش پس انتشار خطا (BP) به روش دسته ای^۵ را در نظر می گیریم. فرض کنیم که تعداد p الگوی آموزش در دسترس می باشد. هدف انتخاب وزن های شبکه است بطوریکه مجموع مربعات خطای زیر حداقل گردد.

$$J(W) = \frac{1}{p \cdot S_M} \sum_{s=1}^p \sum_{i=1}^{S_M} [t_{s,i} - a_{s,i}^M]^2$$

که در آن $t_{s,i}$ و $a_{s,i}^M$ به ترتیب i امین خروجی مطلوب و واقعی متناظر با s امین الگوی آموزش می باشد. W بردار وزن های شبکه و S_M تعداد واحدهای خروجی می باشد. به منظور دستیابی به بردار وزن بهینه به روش تکراری، یک الگوریتم از نوع نزولی^۶ برای او لین بار توسط Werbos در سال ۱۹۷۴ ارائه گردید. سپس در سال ۱۹۸۲ توسط Parker و در سال ۱۹۸۶ توسط Rumelhart توسعه داده شد. برای هر الگوی آموزش، این الگوریتم شامل دو مرحله می باشد. در مرحله اول، خروجی های واقعی شبکه بطرف جلو از لایه ورودی بطرف لایه خروجی حساب می شود. در مرحله دوم، گرادیان حساب شده و وزن ها تنظیم می شوند. این روند بطور تکراری برای هر الگوی آموزش تا زمانیکه تمامی سیگنال های خطا بین خروجی های مطلوب و واقعی به اندازه

صورت الگوریتم BP با نرخ یادگیری ثابت دارای راندمان پائینی خواهد بود. دلیل این مسئله این است که یمنظور جلوگیری از نوسان در مناطق سطح خطا با شیب زیاد بایستی نرخ یادگیری کوچک انتخاب شود. در نتیجه بردار وزن، موقعی که در مناطق مسطح قرار داریم بدلیل کوچک بودن گرادیان خیلی کند حرکت خواهد کرد. بنابراین نیاز به یک الگوریتم یادگیری کار آمد می باشد که بتواند بطور پویا نرخ یادگیری را بر اساس تغییرات مقادیر گرادیان تغییر دهد.

تحقیقات در زمینه تغییر پویای نرخ یادگیری الگوریتم BP در مراجع [۸]-[۶] گزارش شده است. اساساً تمامی این روش ها بطور پویا با یک ضریب ثابتی نرخ یادگیری و مومنتم را بر اساس مشاهدات سیگنال خطا افزایش یا کاهش می دهند. گرچه اثبات شده است این روش ها در موارد زیادی جواب های خوبی می دهند با اینحال در مواردی وارد نواحی نامطلوب فضای وزن ها شده و باعث واگرایی می شوند.

علاوه بر این روش های دیگری برای افزایش سرعت یادگیری ارائه شده است از قبیل استفاده از روش های مرتبه دوم (مثل روش نیوتون [۱۰]-[۹]، روش های Broyden-Levenberg-Marquardt و Shanno-Gold Farb-Fletcher [۱۱]-[۱۲]) گروهی دیگر از روش ها مبتنی بر اتوماتان های یادگیر می باشد (LA) [۲۱]-[۱۳]. LA نخستین بار توسط Tsetlin در اوائل دهه ۱۹۶۰ در اتحاد جماهیر شوروی سابق مطرح شد. او که در آغاز علاقه به مدل سازی رفتار سیستم های بیولوژیکی داشت به مطالعه و تحقیق روی سیستم های مهندسی پرداخت (شناسایی حروف، سیستم های تطبیقی، ...).

در دو دهه اخیر محققین زیادی در کشورهای مختلف از اتوماتان های یادگیر، برای نمایش سیستم های گسسته معین و استوکستیک^۸ که عملکردشان را در محیط های تصادفی بهبود می دهند استفاده کرده اند. علاوه بر این محققین فعال در زمینه شبکه های عصبی از اتوماتان های یادگیر در مدل هایشان استفاده کرده اند. در برخی از این روش ها از اتوماتان یادگیر با ساختار متغیر (VSLA) یا اتوماتان یادگیر با ساختار ثابت^{۱۰} (FSLA) برای پیدا کردن مقادیر مناسب برای پارامترهای الگوریتم یادگیری BP استفاده شده است [۱۹]-[۱۳]. در مراجع [۲۰] و [۲۱] Handa, Baba و Sato از اتوماتان های یادگیر سلسله مراتبی با ساختار متغیر^{۱۱} (HSSA) برای تعیین پارامترهای یادگیری BP بهره جسته اند. در این مقاله روش های مختلف تغییر دینامیک نرخ یادگیری مورد بررسی قرار گرفته است. الگوریتم نرخ یادگیری متغیر (VLR) و الگوریتم های تطبیق نرخ یادگیری مبتنی بر اتوماتان های یادگیری توضیح داده شده و بر روی

مسائل مختلف پیاده و با هم مقایسه شده اند. در اینجا نظر به اینکه در روش VLR پارامترهای مورد استفاده در این روش تأثیر بسزایی در عملکرد آن دارد لذا از اتوماتان یادگیر برای تنظیم پارامترهای آن استفاده کرده ایم. در الگوریتم جدید که تحت عنوان الگوریتم نرخ یادگیری متغیر تطبیقی (AVLR) نامگذاری کرده ایم. پارامترهای الگوریتم VLR بطور دینامیک توسط اتوماتان یادگیر براساس تغییرات خطا تغییر داده می شود. نتایج مشابه سازی ها بر روی مسائل مختلف نشان می دهد الگوریتم پیشنهادی نسبت به روش های دیگر از سرعت همگرایی بالایی برخوردار می باشد.

بخش های بعدی مقاله بصورت زیر سازماندهی شده است. در بخش ۱ الگوریتم یادگیری پس انتشار خطا که برای آموزش شبکه های عصبی چند لایه مورد استفاده قرار می گیرد توضیح داده شده است. مقدمه ای بر اتوماتان های یادگیر و انواع مهم آن در بخش ۲ آورده شده است. در بخش ۳ الگوریتم نرخ یادگیری متغیر را توضیح داده ایم. روش های افزایش سرعت یادگیری شبکه های عصبی مبتنی بر اتوماتان های یادگیر را در بخش ۴ بررسی می نماییم. در بخش ۵ الگوریتم جدیدی تحت عنوان نرخ یادگیری متغیر تطبیقی (AVLR) معرفی شده است. نتایج آزمایش ها برای مسائل مختلف در بخش ۶ و در پایان نتیجه گیری آورده شده است.

۱- الگوریتم پس انتشار خطا

الگوریتم BP یک روش سیستماتیک برای آموزش شبکه های عصبی چند لایه می باشد. الگوریتم BP بطور مبسوط در مرجع [۲۶] توضیح داده شده است.

الگوریتم BP، دو مسیر محاسباتی دارد. مسیر اول مسیر پیشخور یا رفت و مسیر دوم مسیر پس خور یا مسیر برگشت نامیده می شود.

مسیر رفت: این مسیر با معادلات زیر توصیف می شود:

$$\underline{a}^0 = \underline{p}(k)$$

$$\underline{a}^{l+1}(k) = \underline{F}^{l+1} \left(\underline{W}^{l+1}(k) \underline{a}^l + \underline{b}^{l+1}(k) \right), l=0,1,\dots,L-1$$

$$\underline{a} = \underline{a}^L(k)$$

در این مسیر وزن ها و بایاس های شبکه تغییر داده نمی شوند. توابع محرک، روی تمامی نرون ها عمل می کند، یعنی:

۲- $\Phi = (\Phi_1, \dots, \Phi_s)$ مجموعه حالت هاست.

۳- $\beta = \{0, 1\}$ مجموعه ورودی هاست که در آن ۱ جریمه و ۰ پاداش را نشان می دهد.

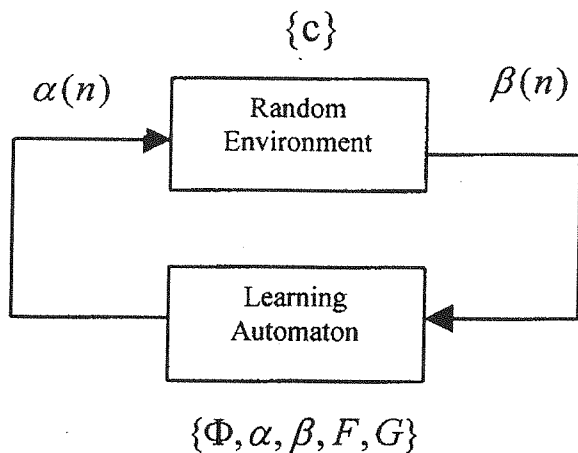
۴- $F: \Phi \times \beta \rightarrow \Phi$ نگاشت انتقال نامیده می شود. این نگاشت، انتقال حالت های اتوماتان پس از دریافت یک ورودی را تعریف می کند. F می تواند استوکستیک باشد.

۵- $G: \Phi \rightarrow \alpha$ نگاشت خروجی بوده و اقدام اتوماتان در حالت Φ_j را نشان می دهد.

اقدام انتخاب شده بعنوان ورودی به محیط اعمال شده و محیط در واکنش به آن پاسخ تصادفی $\beta(n)$ را در لحظه n تولید می کند.

$\beta(n)$ عنصری از $\beta = \{0, 1\}$ بوده و پاسخ فیدبک محیط به اتوماتان می باشد.

محیط اتوماتان را با احتمال c_j جریمه کرده ($\beta(n)=1$) که این احتمال وابسته به اقدام می باشد. بر اساس پاسخ $\beta(n)$ ، حالت اتوماتان $\Phi(n)$ تغییر یافته و اقدام جدیدی در لحظه $n+1$ انتخاب می شود. نحوه اتصال اتوماتان و محیط در شکل (۱) نشان داده شده است.



شکل (۱) اتصال فیدبک اتوماتان و محیط.

$\{c_j\}$ در ابتدا مشخص نبوده و مطلوب است در نتیجه تعامل با محیط، اتوماتان به اقدامی میل کند که احتمال جریمه متناظر با آن حداقل باشد. در قسمت های بعدی برخی از اتوماتان های یادگیر با ساختار ثابت و اتوماتان با ساختار متغیر را توضیح خواهیم داد.

اتوماتان با دو حالت (L_2)

این اتوماتان دارای دو حالت Φ_1, Φ_2 و دو اقدام α_1 و α_2 می باشد. اتوماتان از مجموعه $\{0, 1\}$ ورودی گرفته، با دریافت ورودی (۱) پاسخ نامطلوب (تغییر حالت داده و با دریافت

$$E^{l+1}(\underline{n}(k)) = [f^{l+1}(n_1^{l+1}(k)), \dots, f^{l+1}(n_{S^{l+1}}^{l+1}(k))]^T$$

مسیر برگشت: در این مسیر بردارهای حساسیت از لایه آخر به لایه اول برگشت داده می شوند. معادلات زیر، دینامیک مسیر برگشت را بیان می کند:

$$\underline{\delta}^L(k) = -2 \dot{F}^L(\underline{n}) \underline{e}(k)$$

$$\underline{\delta}^l(k) = \dot{F}^l(\underline{n}^l) (W^{l+1})^T \underline{\delta}^{l+1}, l = L-1, \dots, 1$$

$$\underline{e}(k) = \underline{t}(k) - \underline{a}(k)$$

در مسیر برگشت ابتدا با در دست بودن بردار هدف بردار خطا محاسبه می شود. سپس بردار خطا از سمت راست به چپ از لایه آخر به لایه اول توزیع شده، گرادیان محلی نرون به نرون با الگوریتم بازگشتی محاسبه می شود.

تنظیم پارامترها: در این مرحله ماتریس های وزن و بردارهای بایاس شبکه بصورت زیر تنظیم می شوند.

$$W^l(k+1) = W^l(k) - \alpha \underline{\delta}^l(k) (\underline{a}^{l-1}(k))^T$$

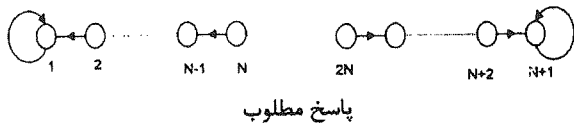
$$b^l(k+1) = b^l(k) - \alpha \underline{\delta}^l(k), l=1, 2, \dots, L \quad \text{و}$$

توقف: اگر میانگین مربعات خطا در هر epoch (جمع مربعات خطا برای تمامی الگوهای یادگیری) کمتر از مقدار از پیش تعیین شده ای بوده و یا اینکه فرم تغییرات در پارامترهای شبکه پس از هر سیکل خیلی کوچک باشد الگوریتم BP متوقف می شود.

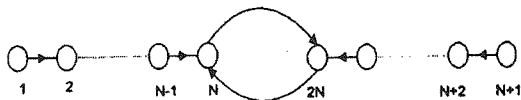
۲- اتوماتان یادگیر

اتوماتان یادگیر (LA) را می توان به دو گروه اصلی، اتوماتان یادگیر با ساختار ثابت (FSLA) و اتوماتان یادگیر با ساختار متغیر (VSLA) تقسیم کرد [۲۵]–[۲۲]. اگر احتمال انتقال از یک حالت به حالت دیگر و احتمال های اقدام و حالت ثابت باشند، اتوماتان با ساختار ثابت، در غیر این صورت اتوماتان با ساختار متغیر گفته می شود. بعضی از انواع FSLA عبارتند از اتوماتان های Krinsky, Tsetline و Krylov. اتوماتان یادگیر با ساختار ثابت یک پنج تایی بصورت $\langle \alpha, \Phi, \beta, F, G \rangle$ می باشد که در آن داریم:

۱- $\alpha = (\alpha_1, \dots, \alpha_R)$ مجموعه اقدام هاست که در هر مرحله یکی از این اقدام ها انتخاب می شود.

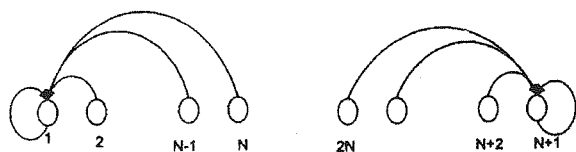


پاسخ مطلوب

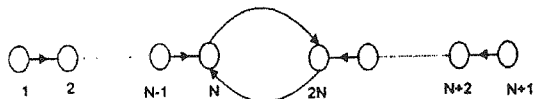


پاسخ نامطلوب

شکل (۳) گراف انتقال حالت برای $L_{2N,2}$



پاسخ مطلوب



پاسخ نامطلوب

شکل (۴) گراف انتقال حالت برای اتوماتان Krinsky.

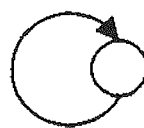
اتوماتان Krinsky

این اتوماتان موقعی که پاسخ محیط نامطلوب باشد، دقیقاً مثل $L_{2N,2}$ رفتار می کند. اما برای پاسخ مطلوب، هر حالت Φ_i (برای $i=1, \dots, N$) به حالت Φ_{i+1} می رود و هر حالت Φ_i (برای $i=N+1, \dots, 2N$) به حالت Φ_{N+1} می رود. لذا پس از هر پاسخ مطلوب تعداد N پاسخ نامطلوب لازم است تا اتوماتان از یک اقدام به اقدام دیگر سوئیچ کند. گراف انتقال حالت اتوماتان Krinsky در شکل (۴) نشان داده شده است.

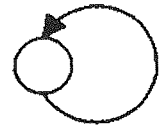
اتوماتان Krylov

این اتوماتان موقعی که خروجی محیط مطلوب باشد انتقال حالت مشابه $L_{2N,2}$ دارد. ولی موقعی که پاسخ محیط نامطلوب باشد حالت Φ_i ($i \neq 1, N, N+1, 2N$) با احتمال $0/5$ به حالت Φ_{i+1} و با احتمال $0/5$ به حالت Φ_{i-1} می رود. موقعی که $i=1$ یا $i=N+1$ می باشد، Φ_i با احتمال $0/5$ در همان حالت باقی می ماند. و با همان احتمال به حالت Φ_{i+1} می رود. موقعی که $i=N$ می باشد اتوماتان با احتمال $0/5$ به حالت $2N$ می رود و با همان احتمال به حالت $N-1$ می رود و بالاخره موقعی که $i=2N$ می باشد اتوماتان با احتمال $0/5$ به حالت N و با همان احتمال به حالت $2N-1$ می رود. گراف انتقال حالت اتوماتان

ورودی. (پاسخ مطلوب) در همان حالت باقی می ماند. اتوماتانی که از این استراتژی استفاده کند با $L_{2,2}$ نشان داده می شود. زیر نویس اول تعداد حالت ها و زیر نویس دوم تعداد اقدام ها را نشان می دهد. نحوه انتقال حالت در شکل (۲) نشان داده شده است.

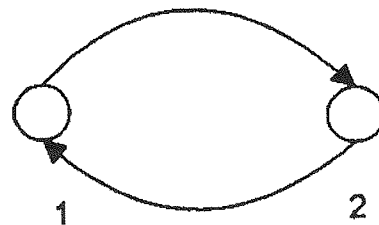


1



2

پاسخ مطلوب



1

2

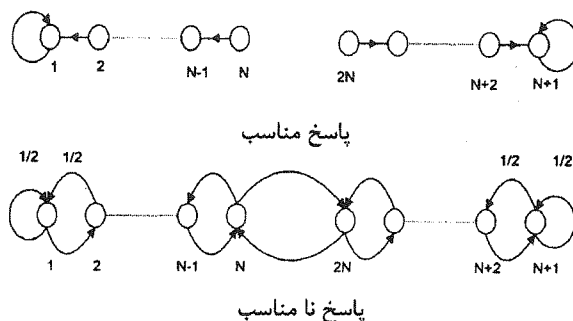
پاسخ نامطلوب

شکل (۲) گراف انتقال حالت برای $L_{2,2}$.

اتوماتان دو اقدامی با حافظه ($L_{2N,2}$)

این اتوماتان دارای $2N$ حالت و دو اقدام بوده و رفتار گذشته سیستم را در قانون تصمیم گیری برای انتخاب رشته اقدام دخالت می دهد. چنانچه دیدیم اتوماتان $L_{2,2}$ با دریافت پاسخ نامطلوب از محیط از یک اقدام به اقدام دیگر سوئیچ می کند. در صورتیکه اتوماتان $L_{2N,2}$ وقتی که در داخلی ترین حالت قرار گرفته باشد برای انتقال به اقدام دیگر نیاز به N پاسخ نامطلوب متوالی دارد. N عمق حافظه اقدام نامیده شده و می گوئیم اتوماتان دارای تعداد $2N$ حافظه می باشد. برای هر پاسخ مطلوب، حالت اتوماتان به سمت حالت های داخلی تر بیرونی تر میل می کند. این اتوماتان به اتوماتان با بیش از دو اقدام قابل توسعه بوده و با $L_{KN,K}$ نشان داده می شود که K نشان دهنده تعداد اقدام می باشد. گراف انتقال حالت اتوماتان $L_{2N,2}$ در شکل (۳) نشان داده شده است.

Krylov در شکل (۵) نشان داده شده است.



شکل (۵) گراف انتقال حالت برای اتوماتان Krylov.

اتوماتان با ساختار متغیر

اتوماتان با ساختار متغیر، احتمال‌های انتقال یا احتمال‌های اقدام را بر اساس ورودی تغییر می‌دهد. این نوع اتوماتان توسط شش تایی $\langle \beta, \Phi, \alpha, P, G, T \rangle$ نمایش داده می‌شود. که در آن β مجموعه ورودی‌ها، Φ مجموعه حالت‌های داخلی، α مجموعه خروجی‌ها و P بردار احتمال اقدام‌هاست که بر اساس آن در هر مرحله اقدامی انتخاب می‌شود. G نگاشت خروجی و T الگوریتم یادگیری را نشان می‌دهد. الگوریتم یادگیری یک رابطه برگشتی بوده و برای تغییر دادن بردار احتمال مورد استفاده قرار می‌گیرد. بدیهی است که الگوریتم یادگیری تأثیر بسزایی در عملکرد اتوماتان یادگیر با ساختار متغیر دارد. الگوریتم‌های یادگیری مختلفی در مرجع [۲۲] گزارش شده است. الگوریتم یادگیری پاداش - جریمه خطی L_{R-P} ¹² جزء اولین روش‌ها می‌باشد. فرض کنیم در لحظه k ، اقدام a_i از مجموعه اقدام‌ها با توجه به توزیع $p(k)$ انتخاب شود. در الگوریتم L_{R-P} معادله برگشتی برای تغییر P به صورت زیر تعریف می‌شود.

پاسخ مطلوب

$$\beta(n) = 0$$

$$p_i(n+1) = p_i(n) + a[1-p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i$$

پاسخ نامطلوب

$$\beta(n) = 1$$

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = b/(r-1) + (1-b)p_j(n) \quad j \neq i$$

پارامترهای a و b طول گام را نشان داده، مقدار افزایش (کاهش) احتمال اقدام‌ها را تعیین می‌کنند. الگوریتم یادگیری دیگری که اکثراً مورد استفاده قرار می‌گیرد الگوریتم پاداش - بیکرکت خطی L_{R-P} ¹³ می‌باشد. در L_{R-P} به ازای پاسخ مطلوب $\beta(n)=0$ احتمال متناظر با اقدام α_i افزایش یافته و بقیه احتمال‌ها کاهش می‌یابد. ولی به ازای پاسخ نامطلوب $\beta(n)=1$ هیچگونه تغییری در احتمال‌ها صورت نمی‌گیرد. معادله برگشتی برای تغییر P به صورت زیر می‌باشد.

پاسخ مطلوب

$$\beta(n) = 0$$

$$p_i(n+1) = p_i(n) + a[1-p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i$$

$$\beta(n) = 1 \quad \text{پاسخ مطلوب}$$

$$p_j(n+1) = p_j(n) \quad 1 \leq j \leq r$$

در روابط فوق r تعداد اقدام‌ها را نشان می‌دهد.

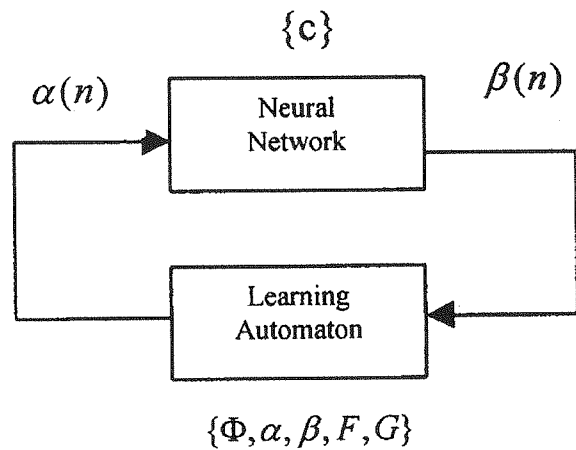
۳- نرخ یادگیری متغیر (VLR)

در الگوریتم BP استاندارد، نرخ یادگیری در طول یادگیری ثابت نگه داشته می‌شود. عملکرد الگوریتم به انتخاب مناسب نرخ یادگیری خیلی حساس می‌باشد. اگر نرخ یادگیری خیلی بزرگ انتخاب شود ممکن است الگوریتم نوسان کرده و ناپایدار شود. اگر نرخ یادگیری خیلی کوچک باشد زمان زیادی طول خواهد کشید تا الگوریتم همگرا شود. انتخاب نرخ یادگیری اپتیمم قبل از یادگیری عملی نبوده و در حقیقت نرخ یادگیری اپتیمم بهنگام پروسه آموزش همچنانکه الگوریتم بر روی سطح خطا حرکت می‌کند. دائماً تغییر می‌کند.

اگر اجازه دهیم نرخ یادگیری بهنگام پروسه آموزش تغییر کند عملکرد الگوریتم BP استاندارد را می‌توان بهبود بخشید. نرخ یادگیری تطبیقی سعی می‌کند که نرخ یادگیری را تا آنجائیکه ممکن است و سیستم ناپایدار نشده است افزایش دهد.

نرخ یادگیری تطبیقی نیاز به تغییراتی در الگوریتم BP استاندارد دارد. ابتدا خروجی و خطای شبکه محاسبه می‌شود. در هر تکرار با استفاده از نرخ یادگیری جاری وزن‌ها و بایاس‌های جدید محاسبه می‌شود. سپس خروجی‌ها و خطاهای جدید محاسبه می‌شود. اگر خطای جدید از خطای قبلی از یک نسبت از قبل تعیین

روش‌ها شبکه عصبی بعنوان محیط عمل می‌کند. نحوه اتصال شبکه عصبی و اتوماتان در شکل (۷) نشان داده شده است.



شکل (۷) نحوه اتصال اتوماتان یادگیر و شبکه عصبی.

مقادیر مختلف پارامتر BP مجموعه اقدام‌های اتوماتان را تشکیل داده در هر مرحله یک اقدام انتخاب شده و به محیط اعمال می‌شود. شبکه عصبی از این پارامتر استفاده کرده و به تعداد N گام الگوریتم BP را اجرا می‌کند. سپس تابعی از خطای شبکه با مقدار متناظرش در تکرار قبلی مقایسه می‌شود این تابع بعنوان مثال می‌تواند مقدار مینیمم خطا در N تکرار باشد. اگر کاهش در مقدار تابع داشتیم شبکه عصبی پاسخ مطلوب ($\beta(n)=0$) و اگر افزایش داشتیم پاسخ نامطلوب ($\beta(n)=1$) تولید می‌کند. اتوماتان یادگیر با استفاده از پاسخ شبکه نسبت به تغییر احتمال اقدام‌ها (در مورد اتوماتان‌های یادگیر با ساختار متغیر) یا تغییر حالت (در مورد اتوماتان‌های یادگیر با ساختار ثابت) اقدام می‌کند. الگوریتم‌های تطبیق نرخ یادگیری BP با استفاده از اتوماتان‌های یادگیر برای اتوماتان‌های با ساختار متغیر و ثابت به ترتیب در شکل‌های (۸) و (۹) نشان داده شده است.

د. نرخ یادگیری متغیر تطبیقی (AVLR)

همچنانکه قبلاً در بخش مربوط به نرخ یادگیری متغیر گفتیم این الگوریتم دارای سه پارامتر مهم می‌باشد. که عبارتند از ضرائب افزایش و کاهش نرخ یادگیری و ضریب نسبت خطای ماکزیمم. هرکدام از این پارامترها تأثیر بسزایی در سرعت همگرایی آموزش شبکه عصبی دارند. دلیل این تأثیر را می‌توان بصورت زیر بیان کرد.

سطح خطای شبکه‌های MLP با قانون یادگیری BP شامل مناطق مسطح و مناطق با شیب تند زیادی می‌باشد. در الگوریتم BP استاندارد، نرخ یادگیری در طول

شده $max-perf-inc$ بیشتر باشد از وزن‌ها و بایاس‌های جدید صرفنظر می‌شود. علاوه بر این نرخ یادگیری کاهش داده می‌شود. (معمولاً با ضرب کردن در یک ضریب $Lr-dec$ ¹⁵). در غیر این صورت وزن‌های جدید نگه داده می‌شوند. اگر خطای جدید از خطای قبلی کمتر باشد نرخ یادگیری افزایش داده می‌شود. (معمولاً با ضرب کردن در ضریب $Lr-inc$ ¹⁶) معمولاً مقادیر $1/0.5$ و $0.7/1.04$ به ترتیب برای ضرائب $max-perf-inc$ و $Lr-dec$ و $Lr-inc$ در نظر گرفته می‌شود.

این روش، نرخ یادگیری را به اندازه‌ای که بتواند بدون افزایش زیاد در خطای آموزش ببیند افزایش میدهد. بنابراین یک نرخ یادگیری نزدیک به اپتیمال بدست می‌آید. موقعی که یک نرخ یادگیری برای ضمانت کاهش در خطای خیلی بزرگ باشد نرخ یادگیری تا باز یافتن یادگیری پایدار کاهش داده می‌شود. الگوریتم نرخ یادگیری متغیر در شکل (۶) نشان داده شده است.

```

Initialize weights and biases
Set training parameters
for i=1:me
    if SSE < eg, i=i-1; break, end
    Feedforward;
    Backward;
    Compute New Weights and Biases;
    Compute new_SSE;
    if new_SSE > SSE * max_perf_inc
        lr = lr * Lr_dec;
        MC = 0;
    else
        if new_SSE < SSE
            lr = lr * Lr_inc;
        end
        w1 = new_w1; b1 = new_b1;
        a1 = new_a1;
        w2 = new_w2; b2 = new_b2;
        a2 = new_a2;
        e = new_e; SSE = new_SSE;
    end
end
    
```

شکل (۶) الگوریتم نرخ یادگیری متغیر.

ع. روش‌های مبتنی بر اتوماتان‌های یادگیر

در این بخش روش‌های مبتنی بر اتوماتان‌های یادگیر برای تطبیق پارامترهای BP را توضیح می‌دهیم. در این

```

Initialize network weights and biases
Set training parameters
Initialize automata parameters
for i=1:NO_OF_ITERATION
    Select a learning rate from
    learning rate set.
    For I=1:N
        Call BP
    End
    find minimum of training error
    in N iteration.
    if MinOfSSE_new >
    MinOfSSE*MAX_ERR_RATIO
        EnvirtRespnc = 1 % penalty
    else
        EnvirtRespnc = 0 % reward
        w1=w1_new;
        w2=w2_new;
        b1=b1_new;
        b2=b2_new;
        MinOfSSE= MinOfSSE_new;
        tr = tr_new;
    end
    Call UpdateActionProbability;
end

```

شکل (۸) الگوریتم تطبیق نرخ یادگیری با استفاده از VSLA.

```

Initialize network weights and biases
Initialize training parameters
Initialize automata parameters
for LoopCnt=1:NO_OF_ITERATION
    select a LEARNING_RATE from
    LEARNING_RATE set.
    For I=1:N
        Call BP
    end
    find minimum of training error in
    STEP_SIZE iteration.
    if minOfSSE_new >
    minOfSSE*MAX_ERR_RATIO
        EnvntRspnc=1 % penalty
    else
        EnvntRspnc= 0 % reward
    end
    call updates automata state
end

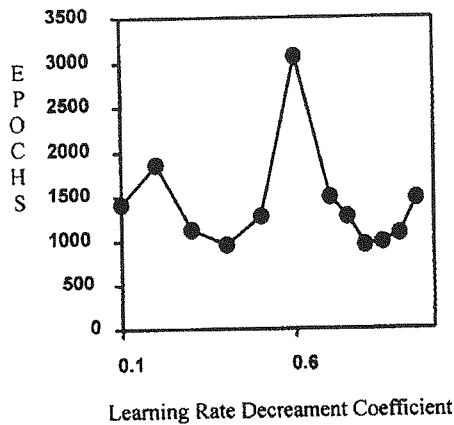
```

شکل (۹) الگوریتم تطبیق نرخ یادگیری با استفاده از FSLA.

یادگیری ثابت نگه داشته می شود. عملکرد الگوریتم به انتخاب مناسب نرخ یادگیری خیلی حساس می باشد. اگر نرخ یادگیری خیلی بزرگ انتخاب شود ممکن است الگوریتم نوسان کرده و ناپایدار شود و اگر نرخ یادگیری خیلی کوچک باشد زمان زیادی طول خواهد کشید تا الگوریتم همگرا شود. انتخاب نرخ یادگیری اپتیمم قبل از یادگیری عملی نبوده و در حقیقت نرخ یادگیری اپتیمم بهنگام پروسه آموزش همچنانکه الگوریتم بر روی سطح خطا حرکت می کند دائماً تغییر می کند. اگر اجازه دهیم نرخ یادگیری بهنگام پروسه آموزش تغییر کند عملکرد الگوریتم BP استاندارد را می توان بهبود بخشید. نرخ یادگیری تطبیقی سعی می کند که نرخ یادگیری را تا آنجائی که ممکن است و سیستم ناپایدار نشده است افزایش دهد.

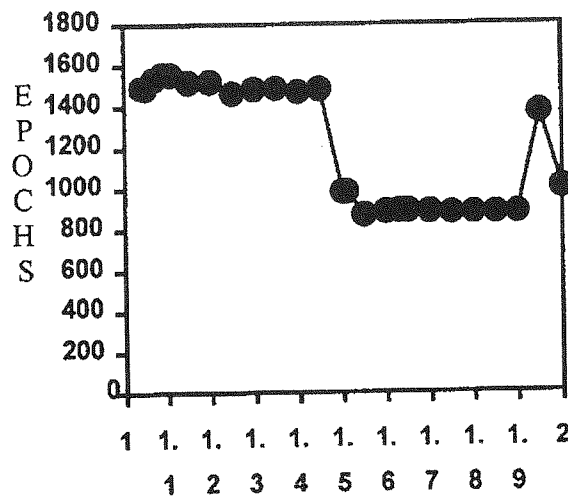
ابتدا خروجی و خطای شبکه محاسبه می شود. در هرتکرار با استفاده از نرخ یادگیری جاری وزن ها و بایاس های جدید محاسبه می شود. سپس خروجی ها و خطاهای جدید محاسبه می شود. اگر در نقطه ای از سطح خطا قرار گرفته باشیم که دارای شیب تندی می باشد خطای جدید از خطای قبلی از یک نسبت از قبل تعیین شده $max - inc - perf$ بیشتر خواهد شد در اینصورت از وزن ها و بایاس های جدید صرفنظر کرده علاوه بر این نرخ یادگیری را کاهش می دهیم. (معمولاً با ضرب کردن در یک ضریب Lr_dec). در غیر اینصورت وزن های جدید نگه داشته می شوند. اگر شیب سطح خطا خیلی تند باشد بدیهی است که بایستی با نرخ بیشتری نرخ یادگیری را کاهش دهیم و اگر شیب سطح خطا زیاد تند نباشد می توان با نرخ کمتری نرخ یادگیری را کاهش داد پس ملاحظه می کنیم با انتخاب دینامیک ضریب کاهش می توانیم با سرعت بیشتری از مناطق با شیب تند عبور کنیم. از طرف دیگر اگر در نقطه ای از سطح خطا قرار گرفته باشیم که مسطح باشد در اینصورت خطای جدید از خطای قبلی کمتر خواهد بود و نرخ یادگیری افزایش داده می شود. (معمولاً با ضرب کردن در ضریب Lr_inc) اگر سطح خطا خیلی مسطح باشد بدیهی است که بایستی با نرخ بیشتری نرخ یادگیری را افزایش دهیم و اگر سطح خطا زیاد مسطح نباشد می توان با نرخ کمتری نرخ یادگیری را افزایش داد پس ملاحظه می کنیم با انتخاب دینامیک ضریب افزایش می توانیم با سرعت بیشتری از مناطق مسطح عبور کنیم. در شکل های ۱۰ الی ۱۲ نحوه تغییرات سرعت یادگیری برحسب پارامترهای مختلف الگوریتم نرخ یادگیری متغیر نشان داده شده است.

Epochs versus Lr_Dec



شکل (۱۲) نحوه تغییرات سرعت همگرایی بر حسب پارامتر Learning Rate Decrement.

Epochs versus Maximum Error Ratio



شکل (۱۰) نحوه تغییرات سرعت همگرایی بر حسب پارامتر Maximum Error Ratio.

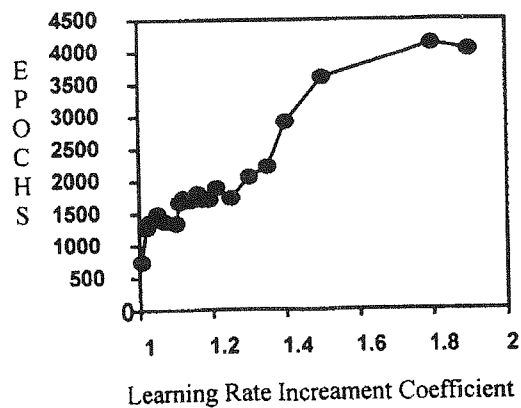
```

Initialize automata parameters
Initialize Network Weights and Biases
Set Training Parameters
for i=1:me
    Select_VLR_Parameters
    for j=1:STEP_SIZE
        if SSE < eg, i=i-1; break, end
        Feedforward;
        Backward;
        Compute New Weights and Biases;
        Compute new_SSE ;
        if new_SSE > SSE* max_perf_inc
            lr = lr * Lr_dec ;
            MC = 0;
        else
            if new_SSE < SSE
                lr = lr * Lr_inc ;
            end
            w1 = new_w1; b1 = new_b1;
            a1 = new_a1;
            w2 = new_w2; b2 = new_b2;
            a2 = new_a2;
            e = new_e; SSE = new_SSE;
        end
    end
    if new_MinOfSSE >=
        MinOfSSE*COEFF.
        EnvntRespc=1 %penalize
    else
        EnvntRespc=0 % reward
    end
    UpdateActiveState_Tsetline
end
    
```

شکل (۱۳) الگوریتم AVLRL.

در این مقاله الگوریتم جدیدی را تحت عنوان الگوریتم نرخ یادگیری متغیر تطبیقی ارائه می کنیم. در این روش بر خلاف روش نرخ یادگیری متغیر که در آن، پارامترهای ضرائب افزایش و کاهش نرخ یادگیری و همچنین ضریب نسبت خطای ماکزیمم مقادیر ثابتی فرض می شوند، در الگوریتم پیشنهادی جدید این پارامترها بطور دینامیک در طی آموزش تغییر داده می شوند. برای تطبیق پارامترها از یک اتوماتان با ساختار ثابت Tsetline استفاده کرده ایم. نتایج شبیه سازی بر روی مسائل مختلف نشان می دهد که الگوریتم پیشنهادی دارای بیشترین سرعت همگرایی می باشد. الگوریتم AVLRL در شکل ۱۳ نشان داده شده است.

Epochs versus Lr_Inc



شکل (۱۱) نحوه تغییرات سرعت همگرایی بر حسب پارامتر Learning Rate Increase.

۶ نتایج شبیه سازی ها

در این بخش روش های مختلف تطبیق نرخ یادگیری را با هم مقایسه کنیم. برای این منظور این روش ها را بر روی چند کاربرد مختلف پیاده می کنیم. ابتدا هر کدام از این کاربردها را توضیح می دهیم.

تقریب تابع سینوسی

اولین مسئله ای که مورد آزمایش قرار گرفته تقریب تابع سینوسی می باشد. در این مسئله ۴۱ نقطه از یک تابع سینوسی معادل ۲ سیکل بعنوان الگوهای آموزش به شبکه اعمال می شوند. تابع سینوسی مورد استفاده بصورت زیر می باشد.

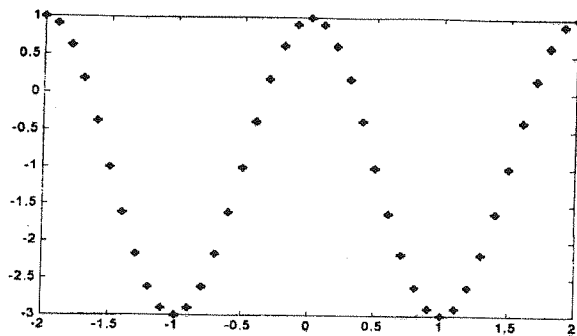
$$\{(p_i, t_i) \mid 1 \leq i \leq 41\} = \text{الگوهای آموزش}$$

$$t = 2 \cos \pi p - 1$$

 داده شده است. برای تقریب تابع از یک شبکه عصبی ۳ لایه با یک نرون در لایه ورودی، ۵ نرون در لایه مخفی و یک نرون در لایه خروجی استفاده شده است.

مسئله پریتی فرد^{۱۷}

در این مسئله هدف مشخص کردن زوج یا فرد بودن تعداد یک ها در الگوهای ورودی می باشد. اگر تعداد یک ها زوج باشد خروجی شبکه یک خواهد بود در غیر این صورت خروجی شبکه صفر خواهد بود. در این مسئله یک منطقی را با ۱ و صفر منطقی را با ۱ - نمایش داده ایم. ساختار انتخاب شده یک شبکه سه لایه با N نرون در لایه ورودی، N نرون در لایه مخفی و یک نرون در لایه خروجی می باشد. در تمامی کاربردها از روش دسته ای^۲ برای آموزش شبکه استفاده کرده ایم.



شکل (۱۴) الگوهای آموزش برای تقریب تابع سینوسی.

مسئله تقارن

در این مسئله هدف آموزش شبکه عصبی بنحوی است

که شبکه توانایی تمیز دادن بین الگوهای متقارن و نامتقارن را داشته باشد. الگویی را که نسبت به مرکز الگو متقارن باشد الگوی متقارن می نامیم، در غیراینصورت الگو را نامتقارن می گوئیم. شبکه طوری آموزش داده می شود که به ازای الگوهای متقارن، خروجی شبکه ۱ و به ازای الگوهای نامتقارن خروجی شبکه ۱- باشد. در مشابه سازی های انجام شده الگوهای ورودی ۸ بیتی، در نظر گرفته شده و از یک شبکه سه لایه با ۸ نرون در لایه ورودی، ۲ نرون در لایه مخفی و یک نرون در لایه خروجی استفاده شده است.

مسئله شناسایی اعداد

در این مسئله می خواهیم شبکه ای را آموزش بدهیم که بتواند اعداد از ۰ تا ۹ را تشخیص دهد. برای این منظور اعداد را مطابق شکل (۱۵) نمایش می دهیم. برای هر عدد یک ماتریس ۸×۸ در نظر می گیریم. خانه های سیاه را با ۱ و خانه های سفید را با ۱- نشان می دهیم. اعداد ۰ تا ۹ را می توان مطابق جدول (۱) با چهار بیت کد کرد. لذا خروجی شبکه دارای ۴ نرون می باشد. از یک شبکه سه لایه با ۶۴ نرون در لایه ورودی، ۶ نرون در لایه مخفی و ۴ نرون در لایه خروجی استفاده کرده ایم.

مسئله تقریب تابع غیر خطی زمان گسسته درجه دوم

تابع غیر خطی زمان گسسته از مرتبه دوم با رابطه زیر مفروض می باشد.

$$y_{k+1} = \frac{1.5 y_k y_{k-1}}{1 + y_k + y_{k-1}} + 0.35 (y_k + y_{k-1}) + 1.2 u_k$$

می خواهیم با استفاده از یک شبکه عصبی سه لایه تابع فوق را با تقریب قابل قبولی شبیه سازی کنیم. برای این منظور ورودی هایی بین ۱- و ۱+ بطور تصادفی ایجاد کرده و به تابع اعمال می کنیم شرایط اولیه را نیز بطور تصادفی بین ۱- و ۱+ انتخاب می کنیم بدین ترتیب یک سری الگوی آموزش ایجاد می شود که از این الگوها برای آموزش شبکه استفاده می کنیم. در رابطه فوق u_k و y_k به ترتیب ورودی و خروجی در لحظه k بوده و y_{k+1} و y_{k-1} به ترتیب خروجی در لحظات k-1 و k+1 می باشد. برای تقریب تابع از یک شبکه عصبی سه لایه با ۳ نرون در لایه ورودی، ۸ نرون در لایه مخفی و ۱ نرون در لایه خروجی استفاده کرده ایم.

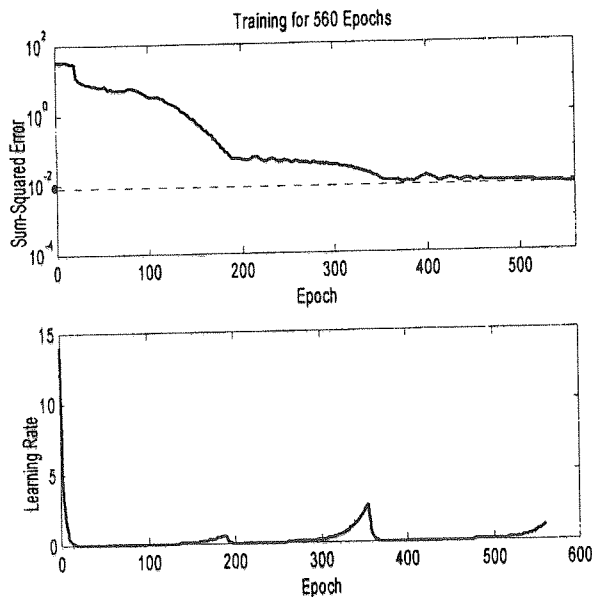
دسته‌ای استفاده کرده‌ایم. در زیر هر کدام از روش‌ها را توضیح می‌دهیم.

BP استاندارد

در این روش برای کلیه کاربردها نرخ یادگیری را $0/01$ در نظر گرفته‌ایم. از روش دسته‌ای برای آموزش شبکه استفاده کرده‌ایم. ضریب ممنتم نیز $0/98$ انتخاب شده است.

نرخ یادگیری متغیر (VLR)

برای مسائل مختلف، الگوریتم از نرخ اولیه 14 شروع کرده و بطور دینامیک در طی آموزش نرخ یادگیری را تغییر می‌دهد. در شکل (۱۶) منحنی تغییرات خطا و نرخ یادگیری برحسب تکرار برای مسئله تقریب تابع سینوسی نشان داده شده است.

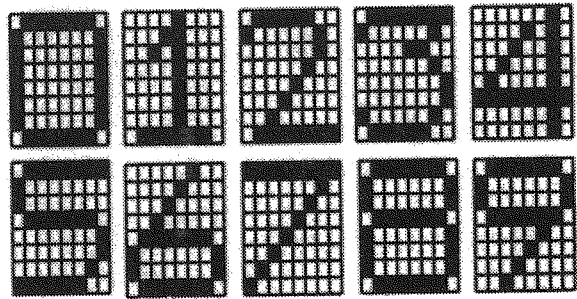


شکل (۱۶) منحنی تغییرات خطا و نرخ یادگیری برحسب epochs.

اتوماتان با ساختار متغیر

در این روش از یک اتوماتان با ساختار متغیر جهت تطبیق نرخ یادگیری استفاده کرده‌ایم. مجموعه اقدام‌های اتوماتان که در حقیقت همان مجموعه نرخ‌های یادگیری می‌باشد به صورت زیر انتخاب شده است.

$\alpha = \{0/01, 0/038, 0/066, 0/094, 0/1\}$
 ضرایب افزایش و کاهش احتمال اقدام‌ها به ترتیب $0/01$ و $0/001$ می‌باشد. اندازه گام 50 انتخاب شده است. بدین معنی که به ازای یک اقدام، الگوریتم BP، 50 بار اجرا شده سپس حداقل مقدار خطا در 50 بار تکرار با حداقل مقدار خطا



شکل (۱۵) نحوه نمایش اعداد از ۰ الی ۹.

جدول (۱) نحوه کد کردن اعداد از ۰ الی ۹.

Digit	Bit 3	Bit 2	Bit 1	Bit 0
0	-1	-1	-1	-1
1	-1	-1	-1	-1
2	-1	-1	1	-1
3	-1	-1	1	1
4	-1	1	-1	-1
5	-1	1	-1	1
6	-1	1	1	-1
7	-1	1	1	1
8	1	-1	-1	-1
9	1	-1	-1	1

مسئله شناسای فونمهای فارسی

در این مسئله هدف تشخیص فونمهای فارسی می‌باشد. دادگان گفتار مورد استفاده در این آزمایش، دادگان فارس دات می‌باشد. ابتدا، سیگنال صحبت با فرکانس 1 KHZ / 44 نمونه برداری می‌شود. سپس فرکانس تا 16 KHZ کاهش داده می‌شود. از هر فریم با طول 20 میلی ثانیه و شیفت 10 میلی ثانیه، ضرایب کپسترال LPC از مرتبه 14 ، بعلاوه مشتق ضرایب کپسترال LPC و لگاریتم انرژی بعنوان ویژگی‌های گفتار استخراج می‌شود.

بعد بردار ویژگی 25 می‌باشد. قبل از محاسبه ضرایب کپسترال LPC نمونه‌های گفتار بایک فیلتر با ضریب $0/97$ و پنجره همینگ فیلتر می‌شود. 938 الگوی آزمایش در مرحله آموزش مورد استفاده قرار گرفته است. یک شبکه عصبی با تأخیر زمانی 18 سه لایه با تأخیر 3 فریم مورد استفاده قرار گرفته است.

پس از توضیح مسائل مختلف حال نتایج حاصل از شبیه‌سازی‌ها را ارائه می‌کنیم. روش‌های مختلف تطبیق پارامتر نرخ یادگیری برای کاربردهای مختلف مورد استفاده قرار گرفته است. در تمامی آزمایشات ضریب ممنتم 98% انتخاب شده است. همچنین در تمامی روش‌ها از آموزش

در تکرار قبلی مقایسه شده، در صورت افزایش خطا، شبکه اتوماتان را جریمه کرده در غیر اینصورت پاداش می دهد.

اتوماتان های با ساختار ثابت

در اینجا از اتوماتان های Krylov و Krinsky، Tsetline برای تطبیق نرخ یادگیری بهره گرفته ایم. مشابه اتوماتان با ساختار متغیر ۵ اقدام بصورت زیر انتخاب کرده ایم.
 $\alpha = \{0/01, 0/038, 0/066, 0/094, 0/1\}$
 عمق حافظه ۳ و اندازه گام ۵۰ تکرار در نظر گرفته شده است.

نرخ یادگیری متغیر تطبیقی (AVLR)

در این روش که برای اولین بار در این مقاله ارائه شده است. از اتوماتان Tsetline برای تطبیق پارامترهای الگوریتم نرخ یادگیری متغیر استفاده کرده ایم. برای پارامترهای ضریب افزایش، ضریب کاهش و ضریب نسبت خطای ماکزیم مقادیری را بعنوان کاندید انتخاب کرده ایم. مقادیر انتخاب شده اقدام های اتوماتان را تشکیل می دهد. عمق حافظه ۲ و اندازه گام ۵۰ تکرار می باشد.

نتایج مشابه سازی های مختلف در شکل های ۱۹ الی ۲۴ نشان داده شده است. در شکل ۱۹ روش های مختلف تطبیق نرخ یادگیری برای مسئله تقریب تابع سینوسی مورد استفاده قرار گرفته است. در شکل ۲۰ همین مقایسه برای مسئله پرتی فرد انجام گرفته است. شکل ۲۱ مربوط به مسئله تقارن بوده، شکل ۲۲ این روش ها را برای کاربرد شناسایی اعداد ۰ الی ۹ مقایسه می کند، شکل ۲۳ نتایج مربوط به مسئله مشابه سازی تابع زمان گسسته غیر خطی از مرتبه ۲ و نهایتاً شکل ۲۴ نتایج مربوط به مسئله شناسایی فومنها را نشان می دهد. همچنانکه در شکل ها نشان داده شده است بیشترین سرعت مربوط به AVLR و کمترین سرعت مربوط به BP استاندارد می باشد.

با استفاده از الگوریتم یادگیری AVLR تابع گسسته غیر

$$y_{k+1} = \frac{1.5 y_k \hat{y}_{k-1}}{1 + y_k^2 + y_{k-1}^2} + 0.35 (y_k + y_{k-1}) + 1.2 u_k$$

را یکمک یک شبکه عصبی سه لایه با ۳ نرون در لایه ورودی، ۸ نرون در لایه مخفی و ۱ نرون در لایه خروجی تقریب زدیم. شبکه عصبی پس از ۸۰۰ تکرار با خطای ۰/۰۸۹۸ تابع را یاد می گیرید. برای آموزش شبکه از الگوهای آموزشی تصادفی در بازه $[-1, +1]$ استفاده کرده ایم. برای تست شبکه یک ورودی سینوسی به شبکه اعمال کردیم. شکل موج ورودی، خروجی مطلوب و خروجی واقعی در شکل ۱۷ نشان داده شده است. شکل موج مطلوب با ستاره و شکل موج واقعی بصورت پیوسته نشان داده شده است همچنانچه از شکل

مشخص می باشد خروجی شبکه عصبی با تقریب خوبی بر روی شکل موج مطلوب منطبق می باشد. برای تست هر چه بیشتر شبکه فرکانس شکل موج ورودی را دو برابر می کنیم. نتایج آزمایش در شکل ۱۸ نشان داده شده است. در این حالت نیز ملاحظه می کنیم خروجی شبکه با دقت خوبی خروجی مطلوب را دنبال می کند.

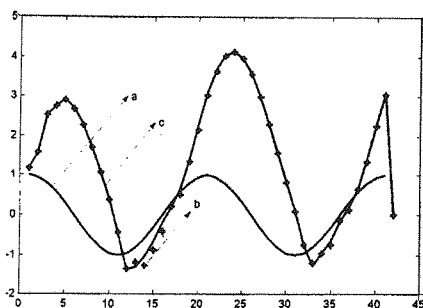
مقایسه روش های مختلف بر اساس تعداد عملیات

در آزمایش دیگری که انجام دادیم روش های مختلف را بر اساس تعداد عملیات با هم مقایسه کردیم نتایج اندازه گیری ها در جدول ۲ نشان داده شده است. همچنانکه در جدول نشان داده شده است روش FLR دارای بیشترین تعداد عملیات و روش AVLR دارای کمترین تعداد عملیات می باشد. با دقت در جدول مشاهده می کنیم با اینکه روش های مبتنی بر اتوماتان های Tsetline و Krinsky دارای تعداد کمتری Epoch نسبت به روش VLR می باشند ولی نظر به اینکه تعداد عملیاتی که در هر Epoch انجام می گیرد نسبت به روش VLR بیشتر می باشد لذا در مجموع روش های مبتنی بر اتوماتان های Tsetline و Krinsky دارای 19 MFLOPS بیشتر نسبت به روش VLR می باشند. که این خود نشان می دهد که از سرعت پائین تری برخوردار می باشد.

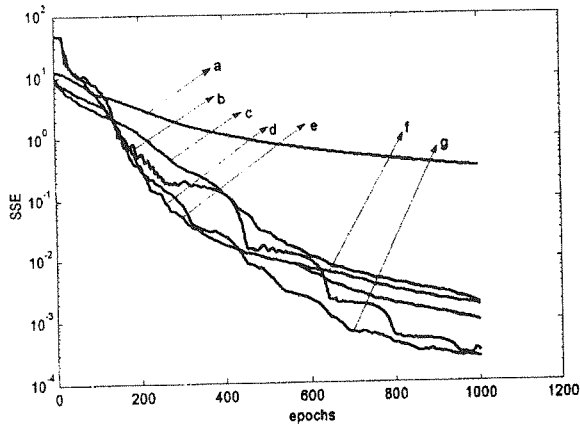
جدول (۲) مقایسه سرعت روش های مختلف برای

مسئله تقریب تابع سینوسی (۵/۰ eg=).

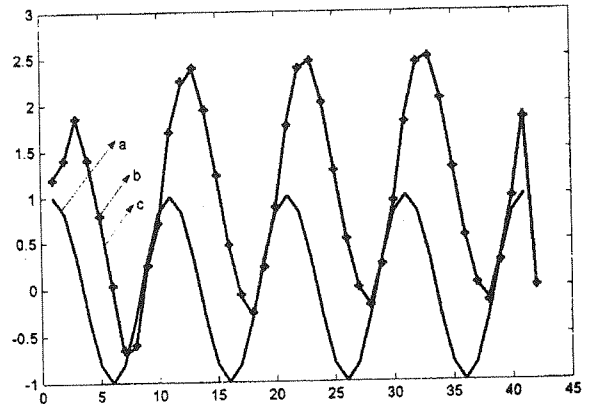
Method	Mflops	Epochs
FLR	12.57	2563
VLR	5.08	1042
VSLA	6.05	1156
F Tsetline(5,3)	5.27	1000
F Krinsky(5,3)	5.28	1002
F Krylov(5,3)	6.30	1202
AVLR	4.03	827



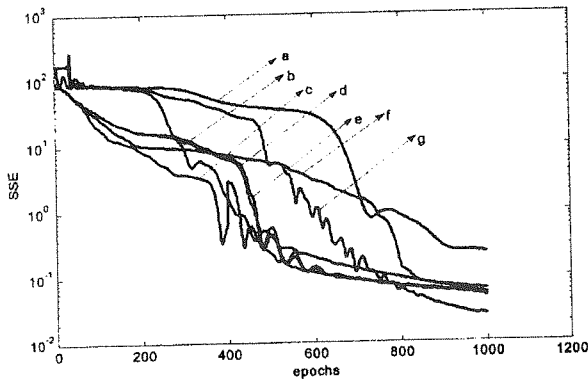
شکل (۱۷) شکل موج های مربوط به ورودی، خروجی مطلوب و خروجی واقعی (a) ورودی (b) خروجی مطلوب (c) خروجی واقعی.



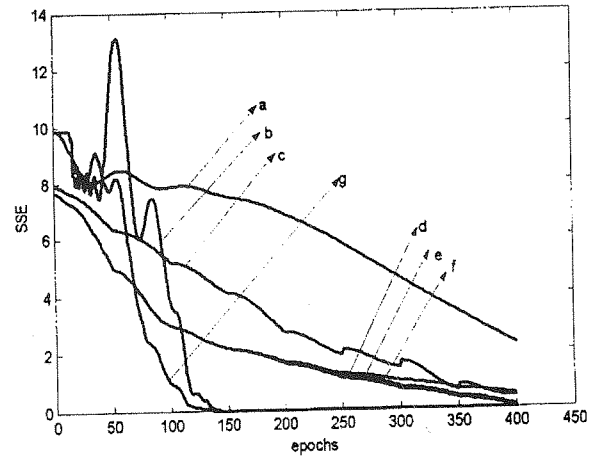
شکل (۲۱) مقایسه سرعت همگرایی روش های مختلف برای مسئله تقارن.



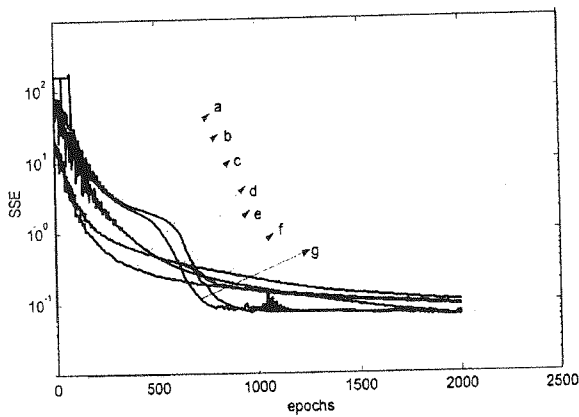
شکل (۱۸) شکل موج های مربوط به ورودی، خروجی مطلوب و خروجی واقعی. (a) ورودی (b) خروجی مطلوب (c) خروجی واقعی.



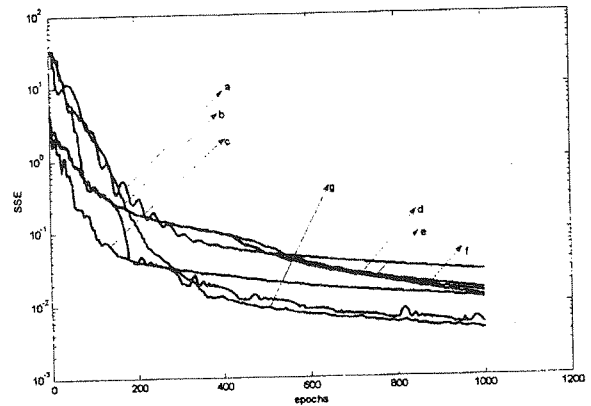
شکل (۲۲) مقایسه سرعت همگرایی روش های مختلف برای مسئله شناسایی اعداد از ۰ الی ۹.



شکل (۱۹) مقایسه سرعت همگرایی روش های مختلف برای کاربرد تقریب تابع سینوسی.



شکل (۲۳) مقایسه سرعت همگرایی روش های مختلف برای مسئله تقریب تابع غیر خطی زمان گسسته درجه دوم.

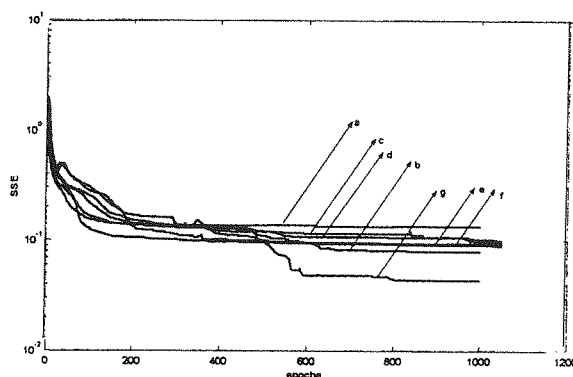


شکل (۲۰) مقایسه سرعت همگرایی روش های مختلف برای مسئله پریپی فرد.

داده می شود. نتایج شبیه سازی های مختلف بر روی مسائل تقرب تابع سینوسی، پریتمی فرد، تقارن، شناسایی اعداد، تقریب تابع غیر خطی زمان گسسته درجه دوم و شناسایی فونم های مصوت فارسی نشان می دهد الگوریتم پیشنهادی نسبت به روش های دیگر از سرعت همگرایی بالایی برخوردار می باشد.

زیر نویس ها

- 1- Variable Learning Rate
- 2- Multi Layer Perceptron
- 3- Back Propagation
- 4- Adaptive Variable Learning Rate
- 5- Batch-style
- 6- Descent-type
- 7- Learning Automata
- 8- Stochastic
- 9- Variable Structure Learning Automata
- 10- Fixed Structure Learning Automata
- 11- Hierarchical Structure Stochastic Automata
- 12- Linear Reward-Penalty
- 13- Linear Reward-Inaction
- 14- Maximum-Performance-increase
- 15- Learning rate-decrease
- 16- Learning rate-increase
- 17- Odd - Parity
- 18- Time Delay Neural Network
- 19- Million Floating Point Operation Per Second



شکل (۲۴) مقایسه سرعت همگرایی روش های مختلف برای مسئله شناسایی فونم.

۷- نتیجه گیری

یکی از مشکلات عمده الگوریتم یادگیری BP، پائین بودن نرخ همگرایی آن می باشد. برای افزایش سرعت یادگیری، الگوریتم هایی ارائه شده است که بطور پویا نرخ همگرایی را بر اساس تغییرات مقادیر گرادیان تغییر می دهند. نتایج مشابه سازی های انجام شده بر روی مسائل مختلف نشان داد الگوریتم یادگیری VLR بدلیل اینکه از امکان بالایی برای انتخاب نرخ یادگیری برخوردار می باشد لذا از سرعت بالایی نسبت به روش های مبتنی بر اتوماتان های یادگیر برخوردار می باشند. در این مقاله ضمن مطالعه و مقایسه این روش ها برای کاربردهای مختلف، الگوریتم جدیدی را تحت عنوان الگوریتم نرخ یادگیری متغیر تطبیقی (AVLR) ارائه کردیم. در این الگوریتم پارامترهای الگوریتم نرخ یادگیری متغیر بطور دینامیک بر اساس تغییرات خطا توسط اتوماتان یادگیر تطبیق

مراجع

- [1] D. E. Rumelhart and J. L. Mc Clelland, "Distributed processing: Explanation in the microstructure of cognition, VOL. I, II and III. Cambridge, MA:MIT Press, 1986 and 1987.
- [2] B. Widrow and M. A. Lehr, "30 Years of adaptive neural networks: perceptron, madalines, and back propagation," Proc. IEEE, Vol. 78, No. 9, pp. 1415-1441, 1990.
- [3] D. R. Hush and B. G. Horne, "Progress in supervised neural networks," IEEE Signal processing Mag., Vol. 10, No. 1, pp. 8-39, 1993.
- [4] D. R. Hush, J. M. Salas, and B. Horne, "Error surfaces for multilayer perceptrons," in Proc. Of IJCNN, Vol. I, June 1991, pp. 759-764.
- [5] X. -H. Yu, "Can back propagation error surface not have local minima," IEEE Trans. Neural networks, Vol.3, No. 6, pp. 1019-1021, 1992.
- [6] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," Neural Networks, Vol. 1, No. 4, pp. 295-308, 1988.
- [7] T. P. Vogel et al., "Accelerating the convergence of the back propagation method," Biol. Cybern., Vol.59, pp. 257-263, 1988.
- [8] L. G. Allred and G. E. Kelly, "Supervised learning techniques for backpropagation networks," in Proc. Of IJCNN, Vol. 1, San Diego, June 1990, pp. 702-709.
- [9] L. P. Ricoti, S. Ragazzini, and G. Martinelli, "Learning of word stress in a suboptimal second order back propagation neural networks," in Proc.

- 1st Int. Conf. Neural Networks, Vol. 1, pp. 355-361, New York, 1988.
- [10] R. L. Watrous, "Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization," in Proc. 1st Int. Conf. neural networks, Vol. 2, pp. 619-628, 1987.
- [11] M. S. Moller, "A scaled conjugate gradient algorithm for fast supervised learning" Neural Networks. Vol. 6, pp. 525-534, No. 4, 1993.
- [12] A. R. Webb, D. Lowe, and M. D. Bedworth, "A comparison of nonlinear optimization strategies for feed forward adaptive layered networks," Royal Signals and Radar Establishment, memorandum No. 4157, July 1988.
- [13] M. B. Menhaj and M. R. Meybodi, "Application of learning automata to neural networks", Proc. Of Second Annual CSI Computer Conference CSIC'96, Tehran, Iran, pp. 209-220, Dec. 1996.
- [14] M. B. Menhaj and M. R. Meybodi, "A novel learning scheme for feedforward neural networks," Proc. Of ICEE-95, University of science and Technology, Tehran, 1994.
- [15] M. B. Menhaj and M. R. Meybodi, "Flexible sigmodal type functions for neural networks using game of automata", Proc. Of Second Annual CSI Computer Conf. CSI'96, Tehran, Iran, pp. 221-232, Dec 1996.
- [16] M. B. Menhaj and M. R. Meybodi, Using learning automata in backpropagation algorithm with momentum", Technical Report, Computer Eng. Department, Amirkabir University of Technology, Tehran, Iran, 1997.
- [17] H. Beigy and M. R. Meybodi, "Adaptation of momentum factor and steepness parameter in backpropagation algorithm using fixed structure learning automata", Proc. Of 4th Int. Annual Conf. Of Computer Society of Iran, pp. 117-124, 199.
- [18] H. Beigy and M. R. Meybodi, and M. B. Menhaj, "Adaptation of learning rate in backpropagation algorithm using fixed structure learning automata", Proc. Of ICEE-98.
- [19] M. R. Meybodi, Hamid Beigy, "New learning automata based algorithms for adaptation of backpropagation algorithm parameters", Proc. Of th 7th Iranian Conference on Electrical Engineering, Iran Telecommunication Research Center, May 1999.
- [20] N. Baba and H. Handa, "Utilization of hierarchical structure stochastic automaton for the backpropagation method with momentum", Proc. Of IEEE, 1995.
- [21] N. Baba and K. Sato, "A consideration on the learning algorithm of neural network", Proc. Of IEEE, 1998.
- [22] K. S. Narendra and M. A. L. Thathachar, Learning automata: An introduction, Prentice Hall, Englewood cliffs, 1989.
- [23] M. R. Meybodi and S. Lakshmiarahan, "Optimality of a general class of learning algorithm," Information science, Vol. 28, pp. 1-20, 1982.
- [24] M. R. Meybodi and S. Lakshmiarahan, "On a class of learning algorithm which have a symmetric behavior under success and failure," Springer Verleg Lecture Notes in Statistics, pp. 145-155, 1984.
- [25] M. R. Meybodi, "Results on a strongly absolutely expedient learning automata," Proc. Of OU Inference Conf. 86, ed. D. R. Mootes and R. Butrick, Athens, Ohio: Ohio University Prees, pp. 197-204.
- [26] M. B. Menhaj, "Computational intelligence (vol.1) Fundamentals of neural networks," Professor Hessabi Publication, 1998.