

یادگیری در شبکه عصبی تصادفی با استفاده از آتماتای یادگیر

سیدمهدی وحیدی‌پور^۱، محمدرضا میبدی^۲

^۱ آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، vahidipour@aut.ac.ir

^۲ دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، mmeybodi@aut.ac.ir

چکیده

شبکه عصبی تصادفی که یک مدل شبکه عصبی است بر اساس مفهوم صف شکل گرفته است. هر نرون در این شبکه همانند یک صف است که تعداد بیشتر مشتریان داخل آن احتمال شلیک آنرا افزایش می‌دهد. در این شبکه مشتری منفی نیز تعریف می‌شود. رسیدن یک مشتری منفی به یک صف (نرون) تعداد مشتریان را کاهش می‌دهد (احتمال شلیک را کاهش می‌دهد). این شبکه دارای کاربردهای متفاوتی در مسائل مختلف می‌باشد. همانند شبکه عصبی الگوریتم‌های مختلف یادگیری در این شبکه نیز وجود دارد. در این مقاله با ارائه کلاس‌های متفاوت استفاده از آتماتای یادگیر، یافتن مقادیر بهینه نرخ یادگیری در راستای افزایش سرعت همگرایی بررسی شده است. بطورکلی استفاده از آتماتا در تنظیم نرخ یادگیری باعث افزایش سرعت همگرایی و فرار از نقاط بهینه محلی می‌شود. با پیاده‌سازی روش پیشنهادی بر روی چند مجموعه داده درستی این ادعا بررسی شده است.

واژه‌های کلیدی

آتماتای یادگیر، شبکه عصبی، RNN، کلاس‌بندی.

۱- مقدمه

آموزش شده و تغییر پارامترها را منجر می‌شود. این مراحل تا زمانی که شبکه به حالت مطلوب برسد ادامه می‌پاید.
موارد کاربرد زیادی برای شبکه‌های عصبی تصادفی وجود دارد. در حوزه‌ی مدلسازی این شبکه توانسته رفتارهای بیولوژیکی بدن را با روش مدلسازی صفت بیان نماید. این روش مدلسازی چنان مورد توجه قرار گرفت که سیستم صفت با نام *G-Network* بر اساس ایده وجود مشتری منفی در سیستم موجود آمد[۲]. در استفاده از این شبکه در بهینه‌سازی مسئله درخت مینیمم اشتاینر *MST* با ترکیب یک روش هیوریستیک و شبکه تصادفی حل شده است[۳]. وجود پارامترهای مختلف در داخل این شبکه و لزوم تطبیق پذیری آنها در طول آموزش شبکه باعث گردید تا این مقاله به بررسی استفاده از آتماتای یادگیر در داخل این شبکه بپردازد.
در این مقاله بعد از مرور آتماتای یادگیر در بخش دوم، بررسی شبکه عصبی تصادفی در بخش سوم آمده است. در بخش چهارم ترکیب میان این دو صورت می‌پذیرد. آزمایش‌های صورت گرفته در جهت نشان دادن درستی ایده مطرح شده در بخش پنجم آمده است.

۲- آتماتای یادگیر

آتماتای یادگیر [۶]، ماشینی است که می‌تواند تعداد متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به آتماتا داده می‌شود و

شبکه عصبی تصادفی (RNN) یک مدل شبکه عصبی است که بر اساس مفهوم شبکه صف شکل گرفته است [۱]. از سویی می‌توان این شبکه را بواسطه‌ی اتصالات میان نودها، شبکه عصبی نامید؛ البته نودها تنها قابلیت ارسال سیگنال را دارند و بر روی ورودی‌شان کاری نجام نمی‌دهند. از سوی دیگر با فرض وجود صف در هر نود شبکه‌ای از صفها شکل می‌گیرد؛ البته تنها تعداد عناصر داخل صف برای این شبکه اهمیت دارد.
حال نودی که تعداد عناصر بیشتری در داخل صف آن وجود دارد احتمال فعالیت (ارسال سیگنال محرک) بیشتری دارد. در این شبکه، سیگنال‌های محرک با دامنه یک میان نرون‌ها حرکت می‌کنند. این سیگنال‌ها یا از خارج سیستم وارد نرون شده و یا از یک نرون متصل دریافت می‌شوند. سیگنال مثبت نقش تشویق نود (افزایش تعداد عناصر داخل صف) و افزایش احتمال فعل شدن آن را در مرحله بعد دارد. در مقابل دریافت سیگنال منفی (کاهش تعداد عناصر داخل صف) احتمال فعالیت نرون را کاهش می‌دهد.

آغاز بکار شبکه با بارگذاری تعداد در داخل صف هر نود می‌باشد. روابط حاکم بر شبکه باعث تغییر تعداد عناصر در داخل صفها می‌شود. این تغییرات تا زمانی پیگیری می‌شود که شبکه صفت به حالت پایدار خود برسد. مقایسه حالت پایدار شبکه با وضع مطلوب، باعث انجام الگوریتم

از یک توزیع نمایی تبعیت می‌نماید^[۲]. بر اساس ساختار شبکه نرون‌ها و نحوه اتصالات آن‌ها، شلیک حاصل از نرون \bar{z}_m به عنوان یک سیگنال به نرون \bar{z}_m وارد می‌شود. این سیگنال با احتمال p_{ij}^+ تشویقی و با احتمال p_{ij}^- تنبیه‌ی است. البته احتمال دیگری نیز وجود دارد. این سیگنال می‌تواند با احتمال d_i به خارج از شبکه منتقل شود. رابطه زیر میان مقادیر احتمالی مذکور برقرار است:

(۱)

برای ساده‌تر شدن کار می‌توان با تعریف $w_{ij}^- = r_i p_{ij}^-$ و $w_{ij}^+ = r_i p_{ij}^+$ نرخ شلیک هر نرون را به صورت زیر تعریف نمود:

(۲)

نرخ ورود سیگنال مثبت و سیگنال منفی که از محیط بیرونی وارد نرون \bar{z}_m می‌شوند را بترتیب با نماد A_i و λ_i نشان می‌دهیم. شکل ۲-نمایش یک نرون را در شبکه عصبی تصادفی نشان می‌دهد. حال به حرکت و تغییرات شبکه عصبی تصادفی بمنظور تحلیل انتقال حالات آن می‌پردازیم. این تغییرات در مقدار $K_i(t)$ در بازه Δt بررسی می‌گردد. این نرخ درونی نرون \bar{z}_m ، $K_i(t)$ ، بعد از شلیک کاهش می‌یابد. همچنین با دریافت سیگنال تنبیه‌ی از خارج شبکه یا یک نرون دیگر این کاهش نیز انجام می‌گردد. با شرایط فوق مقدار انرژی در لحظه $t + \Delta t$ یک واحد کاهش می‌یابد، یعنی $1 - K_i(t) \cdot \Delta t$. مقدار $K_i(t)$ در زمان $t + \Delta t$ افزایش یک واحدی خواهد داشت بشرطی که سیگنال تشویقی را از خارج شبکه و/یا یک نرون دیگر دریافت نماید. در غیر اینصورت مقدار انرژی درونی تغییری نخواهد داشت.

قیود دیگری نیز بر مقدار انرژی درونی برقرار است: نرون با انرژی درونی مثبت می‌تواند شلیک نماید، چنانچه این مقدار صفر باشد هیچ کاهشی بر روی آن انجام نمی‌ذیرد. بدین سان مقدار انرژی درونی مقداری غیر منفی خواهد بود.

فرض کنید $\mathbf{k}(t) = k_1(t), k_2(t), \dots, k_n(t)$ بردار نشان‌دهنده انرژی درونی نرون‌ها در شبکه عصبی باشند. مقدار $K = k_1, k_2, \dots, k_n$ را بعنوان یک مقدار مشخص از این بردار در نظر بگیرید. حال احتمال $\mathbf{k}(t) = \mathbf{K}$ قابل تعريف است. از آنجاییکه $t \geq 0$ $P(\mathbf{k}(t) = \mathbf{K}) = Pr[\mathbf{k}(t) = \mathbf{K}]$ یک زنجیره مارکف زمان پیوسته می‌باشد، معادله چاپمن-کلوموگروف را ارضاء نماید. پس تساوی میان ورودی و خروجی برای حالت \mathbf{K} بصورت زیر نوشته می‌شود.

در معادله (۳) مقدار خروجی تابع $[x]_1$ به ازای مقادیر $x > 0$ یک خواهد بود و در غیر اینصورت صفر می‌باشد. این رابطه تساوی بایستی در حالت پایدار برای شبکه صفت برقرار باشد.

در $[x]_1$ نشان داده می‌شود که شکل بسته‌ی $q_i^{k_i}$ مقدار k_i را ارضاء نماید. منظور از q_i احتمال شلیک نرون \bar{z}_m در حالت تساوی (۴) را ارضاء نماید. منظور از r_i احتمال شلیک نرون \bar{z}_m در حالت

اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۱ مشاهده می‌شود.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط را می‌توان توسط سه تابی $E = \{a, b, c\}$ نشان داد که در آن a مجموعه ورودی‌ها، b مجموعه خروجی‌ها و c مجموعه احتمال‌های جریمه می‌باشد. هرگاه مقادیر β از مجموعه دو عضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $\beta_i = 1$ به عنوان جریمه و $\beta_i = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q ، β می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ باشد. احتمال آنست که عمل a_i β متغیر تصادفی در فاصله $[0, 1]$ باشد. در محیط ایستا مقادیر c_i بدون تغییر می‌ماند، حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کند. اتوماتای یادگیر با ساختار ثابت توسط پنچ تابی $\{a, \beta, F, G, \varphi\}$ نشان داده می‌شود که در آن a مجموعه عمل‌های اتوماتا، β مجموعه ورودی‌های اتوماتا، φ مجموعه وضعیت‌های داخلی اتوماتا، $F: \varphi \rightarrow \beta$ تابع تولید وضعیت جدید اتوماتا و $G: \varphi \rightarrow a$ تابع خروجی می‌باشد که وضعیت کنونی اتوماتا را به خروجی بعدی نگاشت می‌نماید.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که α مجموعه عمل‌های اتوماتا، β مجموعه ورودی‌های اتوماتا، p بردار احتمال انتخاب هریک از عمل‌ها و $T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد.^[۷]

۳- شبکه عصبی تصادفی RNN

همانطور که گفته شد در این شبکه سیگنال‌های محرك با دامنه یک میان نرون‌ها حرکت می‌کنند. سیگنال مثبت نقش تشویق نرون و افزایش احتمال فعال شدن آن را دارد (شنیک) و در مقابل سیگنال منفی می‌کوشد تا این احتمال را کاهش دهد. نرون i در لحظه t پتانسیلی برای شلیک دارد. این مقدار غیرمنفی که با $K_i(t)$ مشخص می‌شود، با دریافت سیگنال تشویق یک واحد افزایش و با دریافت سیگنال منفی یک واحد کاهش می‌یابد.

هنگامیکه این مقدار بزرگتر از صفر باشد نرون فعال است و می‌تواند یک ضربه به بیرون ارسال نماید (انجام شلیک). شلیک توسط نرون \bar{z}_m در یک فرآیند پواسون با نرخ r_i تکرار می‌گردد بطوریکه فاصله بین دو شلیک

معادله‌ی (۵) و حذف عناصر همانند از طرفین تساوی، رابطه (۶) باقی ماند.

$$\begin{aligned} \sum_i (\Lambda_i + (\lambda_i + r_i) I[k_i > 0]) = \\ \sum_i [q_i r_i d_i + (r_i + \lambda_i^-) I[k_i > 0] + \lambda_i^- q_i \\ + (\lambda_i^- - \lambda_i) I[k_i = 0]] \end{aligned} \quad (5)$$

$$\sum_i \Lambda_i = \sum_i [q_i r_i - \lambda_i^+ + \Lambda_i + \lambda_i^- q_i] \quad (6)$$

با استفاده مجدد از رابطه‌ی $q_i = \lambda_i^+ / (\lambda_i^- + r_i)$ سمت راست تساوی

$$P(x) = \prod_{i=1}^n (1 - q_i) q_i^{k_i} \quad (6)$$

رابطه‌ی چاپمن-کلوموگروف را ارضا می‌نماید.

$$\begin{aligned} \sum_i [q_i r_i - \lambda_i^+ + \Lambda_i + \lambda_i^- q_i] = \\ \sum_i [\frac{\lambda_i^+ r_i}{r_i + \lambda_i^-} - \lambda_i^+ + \Lambda_i + \frac{\lambda_i^+ \lambda_i^-}{r_i + \lambda_i^-}] = \sum_i \Lambda_i \end{aligned} \quad (7)$$

بعارت ساده‌تر، با در نظر گرفتن همسایگی میان نرون‌ها و با استفاده از معادله‌ی (۴) مقدار q برای تمامی نرونها قابل محاسبه است.

۱-۲. یادگیری در RNN

علاوه بر معرفی مشتری منفی در RNN، یادگیری وزنه‌ای تعریف شده‌ی w^+ و w^- از نوآوری‌های مطرح شده در آن است. فرض کنید برای حل یک مسئله کلاس‌بندی می‌خواهید از این شبکه استفاده نماییم. بعد از تعیین شکل شبکه (تعداد ورودی‌ها، تعداد لایه‌ها و نرون‌های آنها و تعداد نرون خروجی) بایستی نمونه‌های ورودی X در اختیار شبکه قرار گیرد. بر اساس نمونه ورودی x_k بردار $x_k = \{\Lambda_{k1}, \dots, \Lambda_{km}\}$ نرخ سیگنال مثبت ورودی به m نرون لایه اول به ازای ورودی k ام، و بردار $\lambda(k) = \{\lambda_{k1}, \dots, \lambda_{km}\}$ نرخ سیگنال منفی از نمونه ورودی به m نرون لایه اول به ازای ورودی k ام، شکل می‌گیرد. مقدار این دو نرخ برای سایر نرونها صفر است.

یکی از ساده‌ترین راه‌های تعیین نرخ از روی ورودی شبکه به قرار زیر است:

$$\begin{aligned} \text{if } x_{ik} > 0, \text{then } \Lambda_{ik} = |x_{ik}| \text{ and } \lambda_{ik} = 0 \\ \text{if } x_{ik} \leq 0, \text{then } \Lambda_{ik} = 0 \text{ and } \lambda_{ik} = |x_{ik}| \end{aligned}$$

که منظور از x_{ik} عنصر آم از نمونه ورودی x_k است. راه‌های دیگر برای این تبدیل در [۵] آمده است.

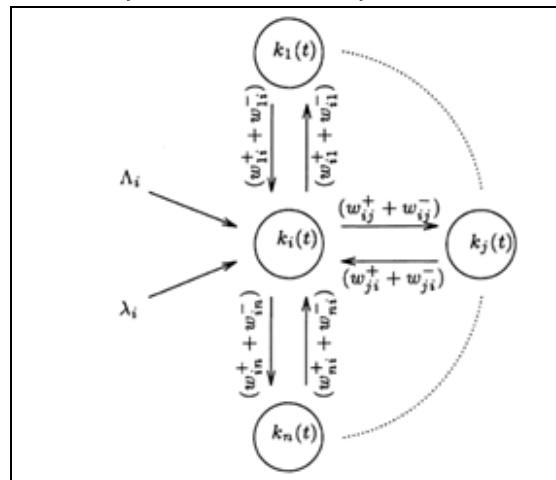
حال برای استفاده از الگوریتم‌های یادگیری در شبکه عصبی، خطای محاسباتی تعریف می‌شود. اختلاف میان خروجی مطلوب y و خروجی حاصل از هر نرون، یا همان احتمال شلیک در حالت پایدار q ، مقدار خطای محاسباتی را تعریف می‌نماید (معادله ۸).

(8)

پایدار است. بر اساس تئوری صفت روابط زیر در شبکه عصبی تصادفی برقرار است:

$$\begin{aligned} P(K) \sum_i (\Lambda_i + (\lambda_i + r_i) I[K_i > 0]) = \\ \sum_i [P(K + e_i) r_i d_i + P(K - e_i) \Lambda_i + P(K + e_i) \lambda_i] + \\ \sum_i \sum_j P(K + e_i - e_j) r_i p_{ij}^+ I[K_j > 0] + \\ \sum_i \sum_j P(K + e_i + e_j) r_i p_{ij}^- + \\ \sum_i \sum_j P(K + e_i) r_i p_{ij}^- I[K_j = 0] \end{aligned} \quad (3)$$

$$\begin{aligned} q_i = \lambda_i^+ / (\lambda_i^- + r_i), \\ \lambda_i^+ = \sum_j q_j r_j p_{ji}^+ + \Lambda_i, \lambda_i^- = \sum_j q_j r_j p_{ji}^- + \lambda_i \end{aligned} \quad (4)$$



آن: نرخ سیگنال مثبت ورودی به نرون آم از خارج سیستم

آن: نرخ سیگنال منفی ورودی به نرون آم از خارج سیستم

احتمال دریافت سیگنال مثبت از نرون آم: p_{ij}^+

احتمال دریافت سیگنال منفی از نرون آم: p_{ij}^-

$w_{ij}^+ = r_i p_{ij}^+$: نرخ سیگنال مثبت ارسالی از نرون آم به نرون j

$w_{ij}^- = r_i p_{ij}^-$: نرخ سیگنال منفی ارسالی از نرون آم به نرون j

$$r_i = \sum_{j=1}^n (w_{ij}^+ + w_{ij}^-)$$

: نرخ شلیک نرون آم

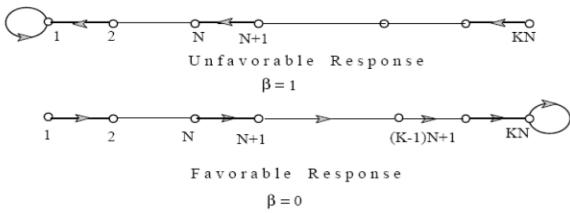
شکل ۲: نمایش یک نرون و پارامترهای RNN

با تقسیم طرفین معادله‌ی (۳) بر مقدار $P(K)$ و استفاده از شکل بسته‌ی

$$\lambda_i^+ / q_i = (\lambda_i^- + r_i) \quad \text{و رابطه‌ی} \quad P(x) = \prod_{i=1}^n (1 - q_i) q_i^{k_i}$$

معادله‌ی (۵) بدست می‌آید. با جایگذاری $(p_{ij}^+ + p_{ij}^-)$ در

پارامتر یادگیری به چند قسمت شکسته می‌شود. برای هر قسمت این فضا یک کنش در آتوماتا در نظر گرفته می‌شود.



شکل ۳: گراف تغییر وضعیت آtomاتای $J_{KN,K}$

procedure A_Scheme_BP (Automata)

```

Initialize the weights to small random values.
initialize the parameters for automaton Automata.
repeat
    for all training patterns (X, T) in the training set do
        call FeedForward
        call ComputeGradient
        if |E(k) - E(k-1)| < Threshold then
            η = call Automata (0)      // β is 0
        else
            η = call Automata (1)      // β is 1
        end if
    end for
    call UpdateWeights
    // Batch weights updating is used
    until k > N. // N is maximum training epoch number
end procedure

```

شبه کد ۱: شبه کد مورد استفاده در مدل A و B

بسته به آنکه نوع آtomاتا را ثابت و یا متغیر فرض نمایید نحوه انتخاب محدوده‌ی پارامتر متفاوت خواهد بود. در حالت ساختار متغیر برای بروز رسانی بردار احتمالات از شبه کد ۳-۳ استفاده می‌شود. در این الگوریتم یادگیری که با نام P-LR شناخته می‌شود پارامترهای a و b ضرایب تشویق و تنبیه می‌باشند و باقیتی قبل از اجرا برنامه قدر آنها مشخص شود. با توجه به واکنش محیط به آtomاتا احتمالات انجام کنش‌ها (حال درونی سیستم) بر اساس شبه کد ۳-۳ تغییر می‌نماید.

۵- آزمایشها

جهت انجام بررسی مدل‌های مختلف پیشنهادی آزمایش‌هایی بر روی مجموعه داده‌های موجود در جدول ۱-۱ جهت طبقه‌بندی اطلاعات انجام شده است. برای انجام آزمایش بر روی داده‌های موجود شبکه‌های جدول ۲-۲ مورد استفاده قرار گرفته است. آزمایشها در دو دسته انجام شده است. نخست آtomاتای ساختار ثابت در سه مدل A، B و C بر روی داده‌ها پیاده‌سازی شده است. در این مقاله از آtomاتای ساختار ثابت $J_{25,5}$ استفاده شده است. بعد از مرور نتایج این آزمایشها که در بخش ۱-۵ آمده است، از آtomاتای ساختار متغیر نیز برای مقایسه حالات مختلف استفاده شده است. شکل ۴-۴ نتایج روند کاهشی خطا را بر روی مجموعه داده Adult نشان می‌دهد. این روش برروی داده‌های Thyroid نیز پیاده‌سازی شده است.

در معادله‌ی (۸) منظور از E_k خطای ناشی از نمونه k است. چنانچه a_j نقش نرون j در محاسبات نقشی نداشته باشد می‌توان با قراردادن در یادگیری که در [۵] پیشنهاد شده است و بر اساس گرادیان توسعه یافته است، وزن موجود در شبکه چنان تغییر می‌باید تا خطای شبکه به مقدار کمینه خود نزدیک شود.

برای مشخص شدن نحوه تعیین خروجی از شبکه، همان کاربرد کلاس‌بندی با دو کلاس را فرض نمایید. با ارایه‌ی ورودی x_k و محاسبه‌ی مقادیر q برای نرون‌های خروجی، نسبت مقدار دو خروجی شبکه می‌تواند برای تعیین کلاس نمونه استفاده شود. اگر این نسبت کمتر از یک باشد نمونه متعلق به یک کلاس و در غیر این صورت به دیگری متعلق است [۵].

۴- ترکیب آtomاتای یادگیر و RNN

یکی از راههای استفاده از آtomاتای یادگیر در این شبکه یافتن مقدار بهینه نرخ یادگیری در حین آموزش شبکه است. این نرخ که گام حرکتی تغییرات را مشخص می‌کند بهتر است رفتار پویایی از خود داشته باشد. در تنظیم مقدار بهینه برای پارامترهای دخیل در یادگیری شبکه عصبی تلاشهایی متفاوتی صورتی گرفته است [۸، ۹، ۱۰]. در این مقاله جهت استفاده از آtomاتای یادگیر در یافتن مقدار بهینه نرخ یادگیری در RNN سه مدل کلی در نظر گرفته شده است:

- مدل A: دو آtomاتای یادگیر برای کل شبکه: ضرب یادگیری وزن‌های مثبت و وزن‌های منفی.
- مدل B: یک آtomاتای یادگیر برای نرخ یادگیری کل شبکه
- مدل C: دو آtomاتا برای هر کدام از یالهای شبکه

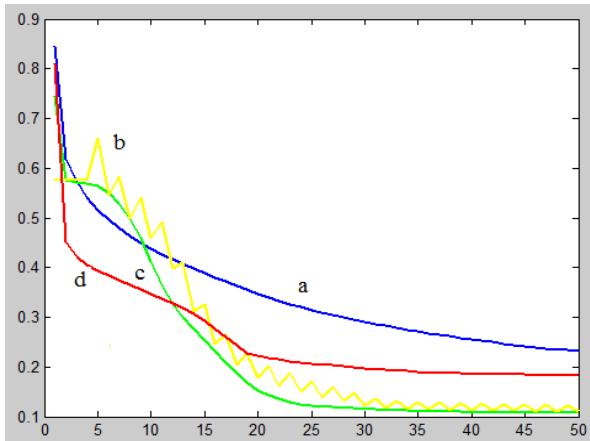
حال با توجه به وجود آtomاتای ساختار ثابت و متغیر در هر مدل دو نوع آtomاتا در نظر گرفته می‌شود: نوع Automata-F، آtomاتا با ساختار ثابت و نوع Automata-P-V، آtomاتا با ساختار متغیر و الگوریتم LR-P. آtomاتای ساختار ثابت مورد استفاده در [۱۱] معرفی شده است. این آtomاتا که با نماد $J_{KN,K}$ دارای گراف انتقال وضعیت معادل شکل ۳-۳ می‌باشد. مقدار مجاز برای نرخ یادگیری به k قسمت تقسیم شده و هر بازه دارای عمق حافظه N می‌باشد. حالتهایی که شماره‌ی آنها در بازه $(i-1) \times N + 1, iN], i = 1, \dots, K$,

دارند.

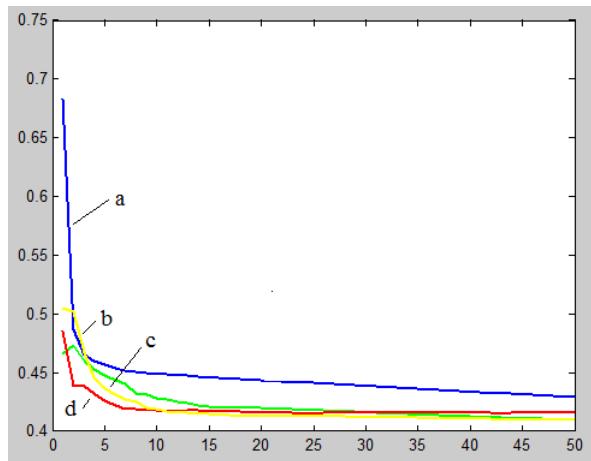
در مدل A و B هنگامی که اختلاف متوسط خطای دو مرحله متوالی از حدآستانه مشخصی کمتر باشد یک سیگنال پاداش برای آtomاتای یادگیر لحاظ می‌شود. مقدار سیگنال تشویق یک می‌باشد. این موضوع در شبه کد ۱-۱ نشان داده شده است.

در مدل C هرگاه علامت مشتق جزیی نسبت به وزن یال مشخص تغییر علامت ندهد یک سیگنال تشویق برای آtomاتا ارسال می‌گردد. شبه ۲-۲ نحوه عملکرد مدل C را نشان می‌دهد.

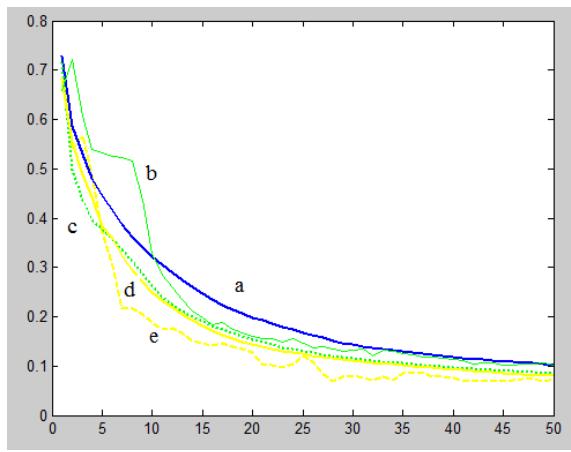
جهت افزایش کارایی روش استاندار، مقداردهی مناسب پارامتر یادگیری ۱/۱ به آtomاتای یادگیر محول شده است. محدوده‌ی قابل قبول برای



شکل ۴: نرخ کاهش خطای در داده‌های *Adult*. نمودار *a*: روش استاندارد، *b*: $J_{25,5}\text{-CF}$ و $J_{25,5}\text{-BF}$ و $J_{25,5}\text{-AF}$



شکل ۵: نرخ کاهش خطای در داده‌های *Thyroid*. نمودار *a*: روش استاندارد، *b*: $J_{25,5}\text{-CF}$ و $J_{25,5}\text{-BF}$ و $J_{25,5}\text{-AF}$



شکل ۶: نرخ کاهش خطای در داده‌های *Adult*. نمودار *a*: روش استاندارد، *b*: $J_{25,5}\text{-AF}$ و *c*: $L_{R\text{-}P}\text{-BV}$ و *d*: $J_{25,5}\text{-BF}$ و *e*: $L_{R\text{-}P}\text{-AV}$

۱-۵-نکات قابل توجه

الف. همانگونه که در نمودارهای یادگیری مشاهده می‌شود میزان خطای در

شکل ۵-رونده کاهش خطای بر روی این داده‌ها نشان می‌دهد. از آنجا که در هنگام استفاده از شبکه برای طبقه‌بندی از مقایسه مقدار دو خروجی شبکه استفاده می‌شود. جدول ۳ میزان درستی عملیات طبقه‌بندی را بر روی داده‌های متفاوت نشان می‌دهد.

procedure C_Scheme_BP (*Automata*)

Initialize the weights to small random values.

initialize the parameters for automaton *Automata*.

repeat

for all training patterns (*X*, *T*) in the training set **do**

call FeedForward

call ComputeGradient

for all layers in the network **do**

for all nodes in *i*th layer **do**

for all weights *w* for *n*th node in *i*th layer **do**

if $\text{Sign}\left(\frac{\partial E_p(k)}{\partial w}\right) = \text{Sign}\left(\frac{\partial E_p(k-1)}{\partial w}\right)$ **then**

 //The sign at iteration k and k-1 is the same

$\eta = \text{call Automata}(0)$ // β is 0

else

$\eta = \text{call Automata}(1)$ // β is 1

end if

end for

end for

end for

call UpdateWeights

 // Batch weights updating is used

until $k > N$. // N is maximum training epoch number
 end procedure

شبکه کد ۲: مراحل انجام کار در مدل *C*

$$\text{if } \beta(k) = 1 \quad p_j(k) = \begin{cases} p_j(k) + a(1 - p_j(k)) & \text{if } i = j \\ p_j(k)(1 - a) & \text{if } i \neq j \end{cases}$$

$$\text{if } \beta(k) = 0 \quad p_j(k) = \begin{cases} p_j(k)(1 - b) & \text{if } i = j \\ \frac{b}{r-1} + (1-b)p_j(k) & \text{if } i \neq j \end{cases}$$

شبکه کد ۳: الگوریتم *LR-P* که برای بروزرسانی مقادیر بردار احتمال انتخاب کنیش

جدول ۱: مجموعه داده‌های مورد استفاده جهت انجام آزمایشات

Data Names	#Instance	#Feature	#Class
<i>Thyroid</i>	215	5	2
<i>Adult</i>	20	4	2

جدول ۲: شبکه مورد استفاده در هر مجموعه داده

نام داده	تعداد سطوح شبکه	تعداد نود در هر لایه
<i>Thyroid</i>	3	5-2-2
<i>Adult</i>	3	4-2-2

جدول ۳: نتایج اجرای شبکه پس از یادگیری با الگوریتم‌های مختلف

	روش یادگیری استاندارد	<i>J25,5-AF</i>	<i>J25,5-BF</i>	<i>J25,5-CF</i>
<i>Thyroid</i>	76%	76%	76%	77%
<i>Adult</i>	95%	95%	95%	95%

- [3] E. Gelenb, A. Ghanwani. and V. Srinivasan, *Improved neural heuristics for multicast routing*. IEEE J. Sel. Areas Commun., 15, 147–155, 1997.
- [4] H. Bakirciglu and T. Kocsak, *Survey of random neural network applications*, European Journal of Operational Research 126, 319-330, 2000.
- [5] E. Glembe, *Learning in the Recurrent Random Neural Network*, Neural Computation 5, 154-164, 1993.
- [6] K. S. a. T. Narendra, M. A. L. , *Learning Automata: An Introduction*: Prentice Hall, 1989.
- [7] M. A. L. Thathachar, P. S. Sastry, "Varieties of learning automata: an overview," *Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE Transactions on, vol. 32, pp. 711-722, 2002.
- [8] R. A. Jacobs, *Increased rates of convergence through learning rate adaptation*, Neural Networks 1, 295-307, 1988.
- [9] M. R. Devos and G. A. Orban, *Self learning backpropagation*, Proc. of NeuroNimes, 1988
- [10] J. D. Schaer, D. Whitley and L. J. Eshelman, *Combination of genetic algorithms and neural networks: A survey of the state of the art*, Proc. of Int. Workshops on Combination of Genetic Algorithms and Neural Networks, COGANN-92, 1{37}, 1992.
- [11] M. R. Meybodi and H. Beigy, *New Learning Automata Based Algorithms for Adaptation of Backpropagation Algorithm Parameters*, International Journal of Neural Systems, Vol. 12, No. 1, 45-67, 2002.

انتهای این مرحله قدری بالاتر است. اما نتیجه استفاده از شبکه برای داده‌های آموزشی قبل قبول است. به یک مثال عددی دقت کنید: فرض کنید که داده‌ای از کلاس اول به شبکه موجود داده شده و مقادیر خروجی شبکه $[0.8 \quad 0.4]$ بدست آمده باشد. در مرحله استفاده از شبکه ورودی متعلق به کلاس اول است(مقدار خروجی اول بزرگتر است). اما در هنگام آموزش، خطای اساس اختلاف این خروجی با $[1 \quad 0]$ محاسبه می‌گردد.

ب. توجه به شرایط ذکر شده می‌توان حالتی را متصور شد که مقدار خروجی‌های شبکه یکسان باشد و در این مورد تصمیم‌گیری قابل انجام نیست. برای چنین حالتی می‌توان خطای بزرگتری در مرحله آموزش برای شبکه متصور شد.

ج. برای انجام بهتر مراحل یادگیری، تغییر در مقادیر پارامتر یادگیری به صورت دسته‌ای انجام پذیرفت. یعنی پس از یکبار آموزش با مقادیر مشخص پارامتریک، از متوسط خطای بدست آمده برای تغییر در پارامتر استفاده گردید.

۵. هنگامیکه مقدار N و K در آتماتای ساختار ثابت افزایش یابد تاثیر خود را از دست می‌دهد و رفتار ثابتی از خود نشان می‌دهد.

۶. برای داده‌های *Adult* نرخ سقوط خطای برای حالت آتماتای ثابت از حالت متغیر بهتر عمل می‌کند(شکل ۶) ز. رفتار حالت *B* در اکثر مواقع از حالت *A* بهتر است و در برخی از مواقع به همان خوبی است. (شکل‌های ۴ تا ۶)

۶- نتیجه‌گیری و کارهای آینده

در این مقاله ضمن بررسی ساختار شبکه و الگوریتم‌های یادگیری مختلف، به بررسی استفاده از آتماتای یادگیر در این فرآیند پرداخته شد. بر اساس آزمایش‌های صورت گرفته استفاده از آتماتای یادگیر در تنظیم پارامترها، مانند پارامتر یادگیری، می‌تواند سرعت همگرایی روش گرادیان را افزایش دهد. در استفاده از آتماتا جهت انتساب مستقیم مقادیر وزن می‌توان از مقادیر خروجی نرون‌های شبکه نیز استفاده نمود و این مقادیر را به عنوان ورودی به آتماتا وارد نمود. این دسته از آتماناهای یادگیر که با نام آتماتای یادگیر تعمیم‌یافته شناخته می‌شوند قابلیت برقراری ارتباط میان ورودی و کنش آتماتا را دارند.

۷- سپاسگزاری

با تشکر از پژوهشگاه فضای مجازی که با حمایتهای مالی خود ما را در انجام این پژوهش یاری رسانده‌اند.

مراجع

- [1] E. Gelenbe, *Random neural networks with negative and positive signals and product form solution*. Neural Comput., 1, 502–510, 1989.
- [2] E. Gelenbe, *The first decade of G-networks*. Eur. J. Oper.Res., 126, 231–232, 2000.