دسترسی در سایتِ http://jnrm.srbiau.ac.ir
سال هفتم، شماره سی و دوم، مهر و آبان ۱۴۰۰
شماره شاپا: ۲۵۸۸-۵۸۸X

پژوهش‌های نوین در ریاضی

دانشگاه آزاد اسلامی، واحد علوم و تحقیقات

حل معادلات انتگرال ولترای تصادفی به روش شبکه عصبی مصنوعی فازی

سیدهادی ابطحی[۱]، حمیدرضا رحیمی[۲]، مریم مصلح[۳*]

[(۱،۲)] گروه ریاضی، دانشگاه آزاد اسلامی، واحد تهران مرکز، تهران، ایران
[(۳)] گروه ریاضی، دانشکده علوم پایه، دانشگاه آزاد اسلامی، واحد تهران غرب، تهران، ایران

تاریخ ارسال مقاله: ۱۳۹۹/۰۶/۱۸ تاریخ پذیرش مقاله: ۱۳۹۹/۰۷/۱۵

چکیده

معادلات انتگرال ولترا به عنوان خروجی مسائل مطرح شده در علوم پایه و مهندسی، کاربرد ویژه در پیشبرد حل مسائل پیچیده دارند. یکی از انواع پرکاربرد آن که متشکل از یک فرایند تصادفی تحت حرکت برونی ایجاد می‌شود، معادلات انتگرال ولترا تصادفی[۱] است. حل این نوع از معادلات همواره از چالش‌های محقیقن بوده است. از سوی دیگر با توسعه هوش مصنوعی و ارائه روش شبکه عصبی مصنوعی فازی به عنوان یک مدل الهام گرفته از فرایند تفکر و تجزیه و تحلیل در مغز انسان، مدل‌های پیشرفته‌ای از الگوریتم‌های طراحی شده است. برخی از این این الگوریتم‌های یادگیری در شبکه عصبی مصنوعی فازی در حل معادلات اسفاده شده است. در این مقاله با استفاده از این روش و طراحی یک الگوریتم یادگیری به حل معادلات انتگرال از نوع ولترای تصادفی می‌پردازیم. روش ارائه شده در این مقاله علاوه بر داشتن دقت بالاتر نسبت به روش‌های پیشین، سرعت بیشتری در حل مسئله را دارا است. این موضوع سبب ایجاد یک سطح اطمینان قابل قبول برای محققان در زمان برخورد با این نوع مسائل می‌شود.

واژه‌های کلیدی: شبکه عصبی مصنوعی فازی، معادلات انتگرال ولترای تصادفی.

---

* عهده‌دار مکاتبات: Email: maryammosleh79@yahoo.com
1. Stochastic Volterra Integral Equations

۱- مقدمه

در این مقاله به دنبال ارائه حل معادلات انتگرال ولترای تصادفی با استفاده از روش شبکه عصبی مصنوعی هستیم. محققان فعال در مبحث هوش مصنوعی همواره در تلاش برای ارائه روشی الهام گرفته از فرایند تفکر در انسان بوده‌اند. روند پیشرفت این موضوع در گذر زمان به حد مناسبی رسیده است و از نتایج این تلاش‌ها می‌توان به ارائه مدل شبکه عصبی مصنوعی اشاره کرد. شروع هوش مصنوعی از اوایل دهه چهل میلادی توسط وارن مک کلاچ و والتر پیتز آغاز شد[1]. در سال ۱۹۵۰ ماروین مینسکی و دن ادمونز طراحی اولیه محاسبگر عصبی را انجام دادند[2]. می‌توان سال ۱۹۵۶ میلادی را سال تولد هوش مصنوعی بنامیم. در این سال مینکسی و دیگر محققان اولین کنفرانس در مورد هوش مصنوعی را برگزار کردند که در آن تمامی تلاش‌های انجام شده توسط دانشمندان به اشتراک گذاشته شد[3]. بعد از آن تحقیقات موثری در خصوص هوش مصنوعی انجام شد و در سال ۱۹۵۹ مک‌کاری اولین مدل کامل یه سیستم هوش مصنوعی را طراحی کرد[4]. یکی از قدم‌های مهم در طراحی شبکه عصبی مصنوعی توسط برایسون و هو درسال ۱۹۶۹ و با تشریح یک سیستم چندمرحله‌ای بهینه ساز انجام شد[5]. در سال ۱۹۸۲ هاپفیلد و در سال ۱۹۸۸ رام‌هارت الگوریتم‌های یادگیری شبکه عصبی مصنوعی را توسعه دادند و بعد از آن در اواخر دهه نود با طراحی اولیه شبکه عصبی توسط وان شکل قابل قبول از یک شبکه عصبی که مورد استقبال محققان قرار گرفت را ارائه کرد[6]. در طی این مدت با ارائه نظریه مجموعه‌های فازی توسط دکترزاده، تلاش‌هایی برای ترکیب این دو ایده انجام شد که محققان توانستند با ارائه مدل‌های فازی سازی و غیر فازی سازی در تمام مراحل یک شبکه عصبی، ایده شبکه عصبی مصنوعی فازی را پیاده سازی نمایند. در سال ۱۹۸۵ کلار و هانت با استفاده از تابع عضویت فازی در الگوریتم‌های یادگیری شبکه عصبی اولین توسعه در این خصوص را انجام داد[7]. در نتیجه در سال ۱۹۹۳ طبقه بندی کامل از یک شبکه عصبی مصنوعی فازی توسط خان ارائه شد[8]. در سال‌های اخیر این روش ترکیبی به عنوان یک راه حل مناسب برای حل معادلات پیچیده استفاده شده. از سوی دیگر حل معادلات انتگرال همواره یک چالش برای محققان بوده است که بر حسب پیچیدگی آن این مورد جدی‌تر شده است[10-9]. معادلات انتگرال ولترا که متشکل از یک یا چند انتگرال با کران پائین صفر و کران بالای پارامتری است که بر حسب قرار گرفتن مجهول در معادله به دو نوع تقسیم می‌شود. استفاده از توابع فازی در حل معادلات انتگرال ولترا سبب بهبود فرایند حل مسائل شده است که برخی تحقیقات در این خصوص انجام شده است [12-11]. همچنین بر حسب توابع به کار برده شده در انتگرال انواع مختلفی از این نوع معادلات توسعه یافته است که ما در اینجا معادلات انتگرال ولترای تصادفی را بررسی می‌نمائیم. از آنجا که تابع تغییرات معادله انتگرال بر حسب حرکت تصادفی براونی می‌باشد این نوع از معادله را معادلات انتگرال ولترای تصادفی می‌نامند. روش‌های مختلفی برای حل این معادله وجود دارد که روش ترکیبی ارائه شده در این مقاله با خطای کمتر و سرعت بیشتر به حل آن می‌پردازد. این روش در شکل‌های مختلفی از معادلات مطرح و استفاده شده است[13].

۲- مفاهیم اساسی

۲-۱- اعداد فازی

در این بخش به ارائه تعاریف مرتبط در خصوص اعداد فازی با روش ارائه شده می‌پردازیم:

تعریف۱: فرض کنیم U یک مجموعه مرجع باشد. مجموعه فازی X در U با یک تابع عضویت به صورت $\mu_x: U \rightarrow [0.1]$ مشخص سازی می‌شود

که به هر عضو در X یک عدد در بازه $[0.1]$ را اختصاص می‌دهد. اشتراک و اجتماع در این مجموعه‌ها به صورت زیر تعریف می‌شود:

$$\forall\, x \in U:\ \mu_{X\cap Y} = min\{\mu_X(x).\mu_Y(x)\}$$
$$\forall\, x \in U:\ \mu_{X\cup Y} = max\{\mu_X(x).\mu_Y(x)\}$$

تعریف۲: یک عدد را به صورت زوج $(X.\mu)$ تعریف نموده اگر دارای شرایط زیر باشد:

- X در R فشرده باشد
- μ پیوسته در R باشد
- به ازاء هر $x \in X$ به طوری که:

$$\mu(x) \geq 0.\forall\, x \in R - X$$

داشته باشیم:

$$\mu(x) = 0$$

- وجود داشته باشد $x_0 \in X$ به طوریکه $\mu(x_0) = 1$
- تابع μ در بازه $[-\infty.x_0]$ صعودی و در بازه $[x_0.+\infty]$ نزولی است.

تعریف۳: یک عدد فازی مثلثی نمایشی به صورت $\tilde{A} = (a_1.a_2.a_3)$ دارید و دارای شرایط زیراست:

- توابع حد فاصل a_1 و a_2 صعودی و توابع حد فاصل a_2 و a_3 نزولی است
- $a_1 \leq a_2 \leq a_3$
- تابع عضویت به صورت زیر است:

$$\mu_{\tilde{A}(x)} = \begin{cases} 0\ .x < a_1\ and\ x > a_3 \\ \frac{x-a_1}{a_2-a_1}\ a_1 \leq x \leq a_2 \\ \frac{a_3-x}{a_3-a_2} a_2 \leq x \leq a_3 \end{cases}$$

تعریف۴: یک α-برش بر روی یک عدد فازی مثلثی یک تابع بر آن است که نمایش عدد فازی را به صورت $A_\alpha = [a_\alpha^L.a_\alpha^R]$ و با شرایط زیر ایجاد می‌کند:

- $\alpha \in (0.1]$
- $[a_\alpha^L.a_\alpha^R] = [(a_2.a_1)\alpha + a_1.-(a_3 - a_2)\alpha + a_3]$

تعریف۵: [14] یک h-سطح روی عدد فازی $\tilde{A}$ به صورت زیر تعریف می‌شود:

$[\tilde{A}]_h = \{x|\mu_{\tilde{A}}(x) \geq h.x \in R\}$ برای $0 < h \leq 1$

۲-۲- شبکه عصبی فازی مصنوعی

یکی از از کارآمدترین و پرکاربردترین متدهای برنامه نویسی برای حل مسائل، روش شبکه عصبی مصنوعی است. بهره‌گیری از منطق فازی در طراحی شبکه عصبی سبب بالارفتن دقت مدل شده است. در زیر به تعاریف مورد نیاز برای طراحی شبکه مدنظر در این مقاله می‌پردازیم.

تعریف۶: نمایش $(N.V.w)$ را یک شبکه از عصبی می‌نامیم که در آن N مجموعه نرون‌ها و V مجموعه $\{(i.j)|i.j\ \in N\}$ که نقش ایجاد اتصال بین دو نرون i و j را دارد. تابع $w: V \to R$ وزن اتصال بین دو نرون i و j را تعیین می‌کند که به اختصار به صورت $w_{i.j}$ نیز نمایش داده می‌شود.

یک شبکه عصبی متشکل از سه لایه ورودی، لایه پردازش (مخفی) و لایه خروجی است که هر لایه برحسب شرایط دارای تعدادی نرون هست. ارتباط نرون‌های هر لایه با لایه بعد به وسیله تابع وزن دهی مشخص می‌شود. تابع انتشار و تابع فعال سازی نقش اساسی در ارتباط بین لایه‌ها را ایفا می‌نمایند. فرایند ایجاد شده ایجاد یک تابع آموزش برای شبکه طراحی شده می‌نماید که نتیجه آن پیدا کردن جواب صحیح بعد از طی یک فرایند آزمون و خطا است. به‌طور مشخص هر لایه را می‌توان حاوی مقادیر زیر فرض کرد:

لایه ورودی:

$$x = (o_{i1}\ .\dots.\ o_{in})$$

لایه پردازی:

$$z_{ij} = f(net_{ij})$$
$$net_{ij} = x \cdot w_{ij} + b_{ij}$$

لایه خروجی:

$$N_i = f(Net_i)$$
$$Net_i = \sum_{j=1}^{m} z_{ij} \cdot v_{ij}$$

که در آن f تابع فعالسازی و b_{ij} بایاس است.
در یک شبکه عصبی فازی مقادیر ورود، پردازش و تحلیل با استفاده از یک h–سطح و بر حسب اعداد فازی مثلثی طراحی می‌شود. در این صورت وزن‌های لایه‌ها به‌صورت $w_{ij} = \left(w_{ij}^{L}.w_{ij}^{C}.w_{ij}^{R}\right)$ شده و سایر لایه‌ها به صورت زیر تعریف خواهد شد:

لایه ورودی:

$$x = (o_{i1} \ldots . o_{in})$$

لایه پردازش:

$$z_{ij} = f(net_{ij})$$
$$net_{ij} = \sum_{i=1}^{n} x \cdot w_{ij} + b_{ij}$$

لایه خروجی:

$$[N_i]_h = \left[[N_i]_h^L . [N_i]_h^U\right] = \left[f[Net_i]_h^L . [Net_i]_h^U\right]$$

۳- معادلات انتگرال ولترای تصادفی

تعریف ۷: فرایند واینر یک فرایند تصادفی پیوسته زمانی هست که دارای شرایط زیر است:

- به ازاء هر t متغیر تصادفی W_t یک توزیع نرمال با میانگین صفر و واریانس t است
- برای هر $t_1 < t_2$ مقدار $W_{t2} - W_{t1}$ مستقل از متغیر تصادفی W_{t1} است.
- فرایند وینر W_t می‌تواند با یک تابع پیوسته نمایش داده شود که مشتق پذیر نیست

تعریف۸: نمایش $\int_0^t X_s \, dW_s$ را یک انتگرال ایتو می‌نامند هرگاه X یک فرایند تصادفی و w یک تابع واینر باشد.
یک معادله انتگرال ولترای تصادفی ترکیب از یک انتگرال ولترای شامل مولفه تابع واینر و یک انتگرال ولترای معمولی هست که به صورت زیر نمایش داده می‌شود:

$$f(x) = u(x) + \int_0^x k_1(x.y)u(y)dy + \int_0^x k_2(x.y)u(y)\, dW_y$$

که در آن $x \in [0.1]$ و توابع $u(x)$ و $k_1(x.y)$ و $k_2(x.y)$ برای $x.y \in [0.1]$ فرایندهای تصادفی تعریف شده در یک فضای احتمالی هستند. همچنین W_y یک فرایند واینر تعریف می‌شود. در اینجا فرض می‌شود $k_1(x.y)$ و $k_2(x.y)$ توابعی جدایی‌پذیر باشند.
برای تفکیک ورودی‌های به شبکه عصبی نیاز به خطی‌سازی معادله انتگرال فوق داریم: از آنجا که $k_1(x.y)$ و $k_2(x.y)$ جدایی‌پذیر هستند، داریم:

$$k_1(x.y) = \sum_{j=1}^{n} g_j(x) h_j(y)$$
$$k_2(x.y) = \sum_{i=1}^{m} m_i(x) n_i(y)$$

در نتیجه:

$$f(x) = u(x) + \sum_{j=1}^{n} g_j(x) \int_0^x h_j(y)u(y)dy + \sum_{i=1}^{m} m_i(x) \int_0^x n_i(y)u(y)\, dW_y$$

با توجه به ویژگی‌های فرایند واینر فرض می‌کنیم دوره‌های $0 = y_0 < y_1 < \cdots < y_z = x$ وجود دارد. در نتیجه می‌توان انتگرال ایتو را به شکل زیر نمایش داد:

$$f(x) = u(x) + \sum_{j=1}^{n} g_j(x) \sum_{k=1}^{z-1} \int_{y_k}^{y_{k+1}} h_j(y)u(y)dy + \sum_{i=1}^{m} m_i(x) \sum_{k=1}^{z-1} n_i(y_z)u(y_z)(W_{y_{z+1}} - W_{y_z})$$

با توجه به فرمول نیوتن کاتس[15] انتگرال بخش اول معادله را به صورت زیر تبدیل می‌کنیم:

$$f(x) = u(x) + \sum_{j=1}^{n} g_j(x) \sum_{k=1}^{z-1} \left(\frac{2\,v}{45} \sum_{\beta=1}^{\frac{l}{4}} 7\, h_i\left(y_{4\,\beta-4}\right) u(y_{4\,\alpha-4})\right.$$

$$+32n_i(y_{4\,\beta-3})u(y_{4\,\beta-3}) + 12n_i(y_{4\,\beta-2})u(y_{4\,\beta-2}) + 32n_i(y_{4\,\beta-1})u(y_{4\,\beta-1}) + 7n_i(y_{4\,\beta})u(y_{4\,\beta})) + \sum_{i=1}^{m} m_i(x) \sum_{k=1}^{z-1} n_i(y_z)u(y_z)(W_{y_{z+1}} - W_{y_z})$$

که در آن:

$$l = 4k'' \quad k'' \in N$$
$$v = \frac{x-a}{l} \qquad y_0 = y_k \quad y_l = y_{k+1}$$

۴- عملیات شبکه عصبی مصنوعی فازی بر معادله انتگرال ولترای تصادفی

فرایند محاسبات مقدار خروجی معادله در شبکه عصبی مصنوعی فازی طی یک فرایند فازی سازی اعداد ورودی، محاسبات فازی در بخش لایه پنهان و نتیجه‌گیری و غیر فازی سازی در لایه خروجی انجام خواهد شد. برای شروع ما معادله مدنظر را با کران‌های چپ و راست و یک h–سطح فازی سازی می‌کنیم.

حال ما در ادامه در پی طراحی یک تابع آموزش در یک شبکه عصبی مصنوعی فازی سه لایه خواهیم بود. تابع آموزش ما به صورت زیر است:

$$y_T(x.P) = \alpha(x) + \beta[x.N(x.P)]$$

که در آن x مقدار ورود دلخواه، Y_t یک تابع وابسته به x و P یک پارامتر قابل تنظیم شامل وزن‌ها و بایاس هست. مقدار α نقش ایجاد شرایط اولیه و β تنظیم کننده مقادی ورود و خروجی در شبکه $N(x.P)$ را ایفا می‌کنند.

در اینجا ما یک شبکه عصبی ۱*۱*۱ بعدی طراحی می‌کنیم که در آن هر مقدار ورودی x با وزنی به نرون‌های موجود در لایه پردازش منتقل می‌شود. در نتیجه:

$$net_i = x \cdot w_i + b_i$$

که در آن b_i بایاس در لایه پردازش برای تقریب تابع می‌باشد. خروجی در لایه پردازش به صورت زیر است:

$$z_i = s(net_i) \quad i = 1. \ldots n$$

که در آن s تابع فعال‌سازی می‌باشد. درنتیجه خروجی شبکه به صورت

$$N_i = V_i z_i$$

که V_i وزن خروجی لایه پردازش به لایه خروجی است.

نمایش ترکیبی خروجی یک شبکه عصبی مصنوعی فازی بر حسب کران‌های چپ و راست در h–سطح به صورت زیر خواهد بود:

$$[N_i]_h = \left[[N_i]_h^L . [N_i]_h^R\right] = [[v_i]_h^L \cdot z_i . [v_i]_h^R \cdot z_i]$$

موضوع مهم دیگر در طراحی یک شبکه عصبی مصنوعی چگونه اجرا و طراحی الگوریتم یادگیری است که برای هر مسئله متفاوت است. روش مدنظر ما برای حل معادله انتگرال مطرح شده به صورت زیر طراحی شده است.

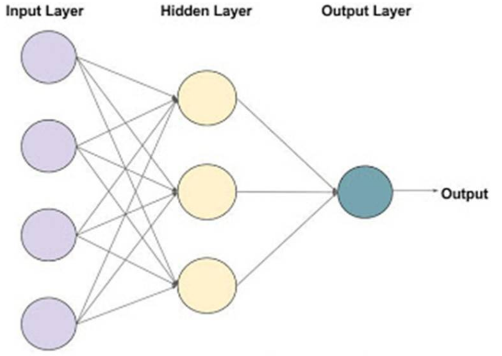
فرض کنید بردار فازی $t = (t_1. \ldots . t_c)$ بردار هدف و بردار فازی $o = (o_1. \ldots . o_c)$ خروجی شبکه عصبی منصوعی فازی طراحی شده باشد. برای آموزش شبکه باید تابع ثابتی تعریف شود که همواره بتوان تفاوت مقادیر بردارهای فوق را اندازه گیری کند. هدف الگوریتم یادگیری این است که این اندازه به سمت صفر میل کند. درنتیجه تابع خطایاب را به صورت زیر تعریف می‌کنیم:

$$e_h = \sum_{k=1}^{c} \frac{\left([t_k]_h^L - [o_k]_h^L\right)^2}{2} + \sum_{k=1}^{c} \frac{\left([t_k]_h^R - [o_k]_h^R\right)^2}{2}$$

که در آن $[t_k]_h = [[t_k]_h^L . [t_k]_h^R]$ و $[o_k]_h = [[o_k]_h^L . [o_k]_h^R]$.

در این مقاله ما از یک شبکه عصبی فازی سه لایه استفاده می‌کنیم. انجام محاسبه و تجزیه تحلیل

داده‌های ورود به شبکه در لایه پنهان توسط نرون‌ها انجام خواهد و خروجی این لایه به لایه پنهان منتقل خواهد شد. نمایش این شبکه به صورت زیر است:



با توجه به شرایط فوق عملیات فازی سازی عبارت خطی شده در قسمت ۳ را برای دامنه بالا و پائین اعمال می‌کنیم:

$$[f(x)]_h^L = [u(x)]_h^L + \sum_{j=1}^{n} [g_j(x)\,]_h^L$$
$$\sum_{k=1}^{z-1} \,(\frac{2\,v}{45} \sum_{\beta=1}^{\frac{l}{4}} 7\,[h_i\,(y_{4\,\beta-4})]_h^L [u(y_{4\,\alpha-4})]_h^L$$
$$+32[h_i(y_{4\,\beta-3})]_h^L [u(y_{4\,\beta-3})]_h^L$$
$$+12[h_i(y_{4\,\beta-2})]_h^L [u(y_{4\,\beta-2})]_h^L$$
$$+32[h_i(y_{4\,\beta-1})]_h^L [u(y_{4\,\beta-1})]_h^L$$
$$+7[h_i(y_{4\,\beta})]_h^L [u(y_{4\,\beta})]_h^L)$$
$$+ \sum_{i=1}^{m} [m_i(x)\,]_h^L$$
$$\sum_{k=1}^{z-1} \,[n_i(y_z)]_h^L [u(y_z)]\,_h^L([W_{y_{z+1}}]_h^L$$
$$-[W_{y_z}]_h^L)[f(x)]_h^R$$
$$= [u(x)]_h^R$$
$$+ \sum_{j=1}^{n} [g_j(x)\,]_h^R \sum_{k=1}^{z-1} \,(\frac{2\,v}{45} \sum_{\beta=1}^{\frac{l}{4}} 7\,[h_i\,(y_{4\,\beta-4})]_h^R$$
$$[u(y_{4\,\alpha-4})]_h^R +$$
$$32[h_i(y_{4\,\beta-3})]_h^R [u(y_{4\,\beta-3})]_h^R +$$
$$12[h_i(y_{4\,\beta-2})]_h^R [u(y_{4\,\beta-2})]_h^R +$$
$$32[h_i(y_{4\,\beta-1})]_h^R [u(y_{4\,\beta-1})]_h^R +$$
$$7[h_i(y_{4\,\beta})]_h^R [u(y_{4\,\beta})]_h^R) +$$
$$\sum_{i=1}^{m} [m_i(x)\,]_h^R \sum_{k=1}^{z-1} \,[n_i(y_z)]_h^R [u(y_z)]\,_h^R([W_{y_{z+1}}]_h^R -$$
$$[W_{y_z}]_h^R)$$

۵- مثال

معادله انتگرال ولترای فازی تصادفی زیر را در نظر بگیرید:

$$\widetilde{f(x)} = \frac{1}{10} + \int_0^x \frac{1}{\sqrt{x+y}} f(y)dy + \int_0^x \sin y\, f(y) dw_y$$

که در آن $x \in [0.1]$ باشد. با پیاده‌سازی الگوریتم یادگیری ارائه شده در بخش ۳ و تابع آموزش شبکه عصبی مصنوعی فازی با بعد ۱*۱*۱ خروجی زیر را خواهیم داشت:

X	جواب تقریبی $[f(x)]^L$	جواب دقیق $[f(x)]^L$	جواب تقریب $[f(x)]^R$	جواب دقیق $[f(x)]^R$
۰	۰,۰۸۲۳۲	۰,۰۸۴۴۵	۰,۰۸۴۲۳۱	۰,۰۸۴۹
۰,۱	۰,۰۴۴۱۷	۰,۰۴۴۲۳	۰,۰۴۴۲۷	۰,۰۴۴۲۹
۰,۲	۰,۰۳۴۱۹	۰,۰۳۴۱۷	۰,۰۳۴۹۷	۰,۰۳۵۰۶
۰,۳	۰,۰۲۶۶	۰,۰۶۶۳	۰,۰۲۶۶۹	۰,۰۲۶۷۲
۰,۴	۰,۰۲۲۷۴	۰,۰۲۲۸۱	۰,۰۲۲۸۳	۰,۰۲۲۸۹
۰,۵	۰,۰۲۰۶۱	۰,۰۲۰۶۷	۰,۰۲۰۸۷	۰,۰۲۰۹۰
۰,۶	۰,۰۱۹۵۶	۰,۰۱۹۵۲	۰,۰۱۹۶۰	۰,۰۱۹۷۶
۰,۷	۰,۰۲۲۰۱	۰,۰۲۲۰۲	۰,۰۲۲۳۹	۰,۰۲۲۴۱
۰,۸	۰,۰۱۶۰۰	۰,۰۱۶۰۱	۰,۰۱۶۱۰	۰,۰۱۶۱۲
۰,۹	۰,۰۱۴۰۱	۰,۰۱۴۰۴	۰,۰۱۴۰۹	۰,۰۱۴۱۱

۶- نتیجه‌گیری و کاربردها

استفاده از ابزارهای هوش مصنوعی در قالب شبکه عصبی مصنوعی فازی روش مناسب و منعطف در حل معادلات انتگرال است که در این مقاله از آن استفاده شده است. روش شبکه عصبی ضمن داشتن توانایی حل مسئله در مدت زمان کوتاه، دارای خروجی با کمترین خطا است. حل معادله انتگرال ولترای تصادفی با این روش، چالش‌های پیشرو محققان در حل این نوع از معادلات با روشی قابل اتکا را برطرف خواهد نمود. سایر معادلات انتگرال و یا معادلات دیفرانسیل با طراحی مدل آموزشی مناسب برای شبکه عصبی مصنوعی فازی قابل حل است که نیاز به بررسی مجزا دارد.

فهرست مراجع

[1] McCulloch, W.S. and W. Pitts, A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 1943. 5(4): p. 115-133

[2] Subramanian, R., Emergent AI, Social Robots and the Law: Security, Privacy and Policy Issues. Subramanian, Ramesh (2017)" Emergent AI, Social Robots and the Law: Security, Privacy and Policy Issues," Journal of International, Technology and Information Management, 2017. 26(3).

[3] Moor, J., The Dartmouth College artificial intelligence conference: The next fifty years. Ai Magazine, 2006. 27(4): p. 87-87.

[4] McCarthy, J., Programs with common sense. 1960: RLE and MIT computation center.

[5] Bryson, A.E. and Y.-C. Ho, Optimization, estimation and control. Ginn and Company, 1969.

[6] Wan, E.A. Time series prediction by using a connectionist network with internal delay lines. in SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-. 1993. Addison-Wesley publishing co.

[7] Keller, J.M. and D.J. Hunt, Incorporating fuzzy membership functions into the perceptron algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985(6): p. 693-699.

[8] Khan, E. and P. Venkatapuram. Neufuz: Neural network based fuzzy logic design algorithms. in [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems. 1993. IEEE.

[9] Babakhani, A., E. Enteghami, and H. Hosseinzade, Numerical solution of Voltra algebraic integral equations by Taylor expansion method. Journal of New Researches in Mathematics, 2020. 6(24): p. 53-64.

[10] Adabitabar Firozja, M. and B. Agheli, A simple algorithm for solving the Volterra integral equation featuring a weakly singular kernel. Journal of New Researches in Mathematics, 2017. 2(8): p. 29-36.

[11] فریبرزی عراقی, فرزانه جوان, عباسبندی, سعید. Solving Fuzzy Integral Equations of the Second Kind by using the Reproducing Kernel Hilbert Space Method. Journal of New Researches in Mathematics. 2020 Aug 27.

[12] Adabitabar Firozja, M. and Agheli, B. A simple algorithm for solving the Volterra integral equation featuring a weakly singular kernel. Journal of New Researches in Mathematics. 2017

[13] Mosleh, M. and M. Otadi, Simulation and evaluation of fuzzy differential equations by fuzzy neural network. Applied Soft Computing, 2012. 12(9): p. 2817-2827.

[14] Leondes, C.T., Fuzzy logic and expert systems applications. Vol. 6. 1998: Elsevier

[15] Balagurusamy, E., Computer Oriented Statistical and Numerical Methods. 1988: Macmillan India Limited.