

تقریب پارامترهای مدل‌های تصادفی: رویکرد شبکه‌های عصبی پیچشی

امیرحسین ابراهیمی^۱، هدی وفایی صفت^۲، دارا رحمتی^{۳*}

^۱ پژوهشگر، مرکز پردازش سریع، پژوهشگاه دانش‌های بنیادی، تهران

Amirh.ebrahimi1377@gmail.com

^۲ پژوهشگر، مرکز پردازش سریع، پژوهشگاه دانش‌های بنیادی، تهران

Hoda.vafaieisefat@gmail.com

^۳ استادیار، دانشکده مهندسی و علوم کامپیوتر، دانشگاه شهید بهشتی، تهران

d_rahmati@sbu.ac.ir

چکیده

امروزه مدل‌های زیادی وجود دارند که هرکدام رفتار یک فرایند طبیعی را توصیف می‌کنند. برای بررسی این رفتارها از مدل‌های ریاضی مانند معادلات دیفرانسیل معمولی کمک گرفته می‌شود. این مدل‌ها معمولاً بسیار پیچیده و شامل تعداد زیادی از پارامترهای نامعلوم هستند و تابع درست‌نمایی آن‌ها از لحاظ محاسباتی قابل حل نیست. تحت این شرایط تقریب پارامترها بدلیل پیچیدگی و نبودن داده‌های کافی کار نچندان ساده‌ای می‌باشد. در شبکه‌های عصبی روش‌های مختلفی برای حل تقریب محاسبات بیزی وجود دارد. مانند یادگیری ماشین، یادگیری عمیق و ... در این مقاله با استفاده از یک نوع معماری خاص در شبکه‌های عصبی به نام شبکه‌های عصبی رزنت که با رفتار غیر-خطی مدل‌ها سازگارتر است، عملکرد مناسبی در تخمین پارامترهای مدل‌های فرایندها بدست آمده است. طبق معیار واریانس ناهم‌هنگ معماری رزنت توانسته است برای یکی از مدل‌های پیچیده به نام مدل رانش انتشار فروپاشی به مقدار ۰.۸۷ نسبت به معماری استنتاج-عمیق بهتر عمل کند. اگر به سراغ مدل‌های ساده تر برویم این پیشرفت به مقدار بالاتری هم می‌رسد.

کلمات کلیدی

تقریب پارامتر، مدل‌های سری زمانی، یادگیری عمیق، شبکه پیچشی.

۱- مقدمه

به طور کلی در طبیعت مدل‌های مختلفی وجود دارند که هرکدام رفتار یک پدیده طبیعی را توجیه می‌کنند. هدف سازگار کردن این مدل‌ها بر روی داده‌ها است. از آن جایی که مدل‌ها اکثراً دارای فرایند به نسبت پیچیده‌ای هستند، پارامترهای ناشناخته‌ای را نیز دارا هستند و نیاز است تا از روش‌هایی برای تخمین پارامترها استفاده شود. به عنوان مثال روند پیشرفت انتشار یک بیماری مانند کوید-۱۹ [12,22]، صعود و تنزل در عرصه بورس، تحلیل مدل‌های تصمیم‌گیری در حوزه‌ی علوم شناختی [4,15] و علم ستاره‌شناسی برای تقریب پارامترهای امواج گرانشی در فضا [7,9] که هر کدام دارای پارامترهای خاص خود هستند را می‌توان نام برد. از این رو تقریب پارامتر توجه محققان را به خود جلب کرده است تا که بتوانند در زمینه‌های مختلف، پارامترهای مدل‌های مختلفی را تقریب بزنند. در آزمایش‌ها و تحقیق‌های علوم شناختی برای آنکه

روند تصمیم‌گیری یک آزمودنی را مورد بررسی قرار دهند از مدل‌های مختلفی استفاده می‌کنند. یکی از این مدل‌ها رانش-انتشار می‌باشد که با در دست داشتن زمان‌های پاسخ آزمودنی پارامترهای بخصوصی را بدست می‌آورد که با آن می‌توان فرایند تصمیم‌گیری فرد را مورد بررسی قرار داد [18]. به این دلیل استفاده از مدل‌های شبکه عصبی برای تقریب این پارامترها با محاسبه همزمان-با-وقوع امری حیاتی است که بتوان در حین اجرای تکلیف شناختی، پاسخ مناسب در کمترین زمان به آزمودنی داده شود. لذا شبکه‌های عصبی طراحی شده باید تا حد امکان ساده و از لحاظ بار محاسباتی قابل قبول باشند. روشی که در این تحقیق استفاده شده است، یادگیری عمیق است. ابتدا با استفاده از پیاده‌سازی معماری رزنت [10] برای مثال ساده‌ای از دو مدل توزیع نرمال چند متغیره و رگرسیون چندگانه نتایج را به دست آورده و نتایج با مدل دیگری مقایسه می‌شود. مدل‌های زیادی در این زمینه تحقیقاتی معرفی شده‌اند، از جمله : استنتاج-عمیق [16] و BAYESFLOW [17] و Likelihood و Approximation networks (LANs) [6] که هرکدام از شبکه‌های مطرح شده، دارای مشکلات و محدودیت‌هایی می‌باشند که به آن‌ها اشاره خواهد شد. عملکرد معماری پیاده‌سازی شده در این تحقیق بر روی مدل‌های پیچیده تر مانند مدل‌های سری زمانی، مدل تصمیم‌گیری انباشته کننده رقابتی نشی [1] و مدل پیش‌بینی روند انتشار بیماری بررسی شده است. نتیجه‌ای که از این مرحله به دست آمد این است که مدل‌هایی که با زمان مرتبط هستند نیاز به یک روشی کاربردی‌تر دارند، که ما را به استفاده از شبکه‌های عصبی مبتنی بر معادل دیفرانسیل معمولی [3] که با مدل‌های زمانی سازگار هستند و ارتباط نزدیکی با معماری رزنت دارند، سوق می‌دهد. ما به دنبال روشی هستیم که بتوان محدودیت‌های ذکر شده را دور بزنند و به مدلی کاربردی‌تر دست پیدا کرد.

۲- تقریب پارامتر

یکی از اهداف مهم در مدل‌سازی آماری آن است که فرایند تولید داده را از طریق مجموعه‌ای از پارامترها توصیف کرد. بدلیل وجود منابع مختلف عدم قطعیت، این توصیف به صورت ذاتی تصادفی است. یک روش برای نشان دادن این عدم قطعیت قرار دادن توزیع احتمال بر روی پارامترها است. در واقع، این ایده اساسی در آمار بیزی گنجانده شده است [8,16].

$$p(\theta, x) = \frac{p(x, \theta)\pi(x)}{\int p(x, \theta)\pi(x)d\theta} \quad (۱)$$

در فرمول (۱)، $p(\theta, x)$ بیانگر توزیع پسین پارامترهای مدل، $p(x, \theta)$ بیانگر تابع درست‌نمایی، $\pi(x)$ بیانگر توزیع احتمال قبلی و عبارت ثابت در مخرج توزیع حاشیه‌ای داده است که به دلیل بار محاسباتی بالا معمولاً در نظر گرفته نمی‌شود. بنابراین می‌توان فرمول (۱) را به صورت زیر بازنویسی کرد

$$p(\theta, x) \propto p(x, \theta) \times \pi(x). \quad (۲)$$

بسیاری از مدل‌های امروزی بسیار پیچیده هستند و یا از فرایندهای شبیه‌سازی برای تولید داده استفاده می‌کنند، در نتیجه تابع درست‌نمایی از نظر محاسباتی غیرقابل حل خواهد شد و یا نمی‌توان آن را به صورت جبری مشخص کرد. بنابراین برای انجام هر نوع استنتاج بیزی به روش‌های تقریبی نیاز است. در این بخش به بررسی روش‌های مختلف برای تقریب مقادیر پارامترهای مدل می‌پردازیم.

۲-۱- روش‌های تقریب پارامتر

روش محاسبه تقریبی بیزی [5] برای اولین بار در تحقیقات ژنتیکی با موفقیت پیاده‌سازی شد [21] و اهمیت و سودمندی این روش در سایر حوزه‌های تحقیقاتی گسترش یافت. ایده اصلی این روش آن است که تابع درست‌نمایی فرمول (۲) را از طریق نمونه‌گیری‌های مکرر از توزیع‌های قبلی تخمین بزند و به تولید داده‌های مصنوعی مشروط به پارامترهای نمونه‌گیری شده بپردازد [14, 19].

به دست آوردن تابع درست‌نمایی با دقت بالا از لحاظ محاسباتی امکان پذیر نیست؛ به همین علت به فرم $q(\theta)$ مشخص می‌شود. نقطه شروع تمامی الگوریتم‌های محاسبه تقریبی بیزی آن است که جدول مرجع ایجاد کنند. این جدول یک ساختمان داده مخصوص است که برای ذخیره سازی تعداد زیادی مجموعه داده‌های شبیه سازی شده است [14, 11, 19, 20].

جنگل‌های تصمیم تصادفی [2] یک روش یادگیری با نظارت برای دسته‌بندی رگرسیون است. به این الگوریتم دسته‌ای از جفت‌های متناظر ورودی - خروجی داده می‌شود و سعی می‌کند که یک نگاشت غیر خطی که ورودی را به خروجی تبدیل می‌کند بر روی درخت‌های تصمیم‌گیری یاد بگیرد که در حوزه محاسبه تقریبی بیزی، داده‌های شبیه‌سازی شده از جدول مرجع هستند و خروجی‌ها پارامترهای متناظر دیتاهای شبیه‌سازی شده است.

یکی از روش‌های پرکاربرد برای محاسبه تقریبی بیزی با بهره‌گیری از یادگیری ماشین، استفاده از شبکه‌های عصبی عمیق به منظور یادگیری ویژگی‌ها از جدول مرجع است [11, 20]. در حالت کلی یک شبکه عصبی با تعریف یک نگاشت $y = f_w(x)$ از ورودی x به خروجی y و یادگیری پارامترهای مدل، ما را به بهترین تقریب هدایت می‌کند. از مزیت شبکه‌های عصبی عمیق می‌توان به آن اشاره کرد که قادر به تقریب هر تابع پیوسته به یک درجه دقت دلخواه هستند.

۲-۲- استنتاج-عمیق

این روشی است که از مزایای دو روش محاسبه تقریبی بیزی با استفاده از جنگل‌های تصمیم تصادفی [19] و یادگیری عمیق [11] استفاده می‌کند. به طور کلی این الگوریتم به تعلیم شبکه‌های عصبی تمام پیچشی یا همگشتی [13] بر روی داده‌های شبیه‌سازی شده، با هدف تقریب میانگین پارامترهای مدل و همزمان تقریب عدم قطعیت و یا واریانس از طریق کمینه کردن تابع ضرر می‌پردازد.

معماری پیاده‌سازی شده شامل سه لایه مخفی می‌باشد که آخرین لایه میانگین مقادیر خروجی هر هسته را حساب می‌کند. اگر m تعداد پارامترهای مدل تولید شده باشد، لایه خروجی شبکه‌ی عصبی تمام پیچشی شامل m واحد فعال سازی خطی برای پیش‌بینی هر یک از اجزاء پارامتر θ و سایر m واحد بعدی برای پیش‌بینی واریانس $\sigma^2(\hat{x})$ است. آخرین واحدها نیز از یک تابع سافت پلاس (فرمول (۳)) عبور کرده‌اند. علت استفاده از این تابع در واحدهای آخر برای آن است که از مثبت بودن تقریب واریانس $\sigma^2(x)$ اطمینان حاصل شود. برای تمامی مثال‌ها از تعداد بیست epoch با الگوریتم بهینه سازی Adam و نرخ یادگیری ۰.۰۰۱ تعلیم یافته اند [16]

$$\text{softp}(z) = 1 + \log(1 + e^z). \quad (۳)$$

مدل استنتاج-عمیق دارای معماری بسیار ساده‌ای است که با وجود این سادگی از دو مدل محاسبه تقریبی بیزی با استفاده از جنگل‌های تصمیم تصادفی و یادگیری عمیق بهتر عمل می‌کند. اما سادگی این مدل در همه‌ی جهات مناسب نیست. به طور مثال مدل‌هایی همچون تصمیم‌گیری انباشته کننده رقابتی نشتی و روند انتشار بیماری که از مدل‌های سری‌زمانی می‌باشند. پارامترهای این نوع مدل‌های سری‌زمانی به خوبی توسط مدلی ساده مثل استنتاج-عمیق بازیابی نمی‌شوند. همچنین با پیچیدگی مدل یا افزایش پارامترها، نمی‌تواند به خوبی عمل کند.

۲-۳- BAYESFLOW, LANS

مدل BAYESFLOW دارای شبکه بسیار پیچیده‌ای است و از لحاظ زمانی و هزینه محاسباتی هزینه بر است. زیرا که معماری بسیار پیچیده‌ای دارد، که شامل یک شبکه عصبی معکوس‌پذیر عمیق برای بازیابی پارامترها و یک شبکه انکودر برای خلاصه کردن داده‌های ورودی می‌باشد، چنین شبکه‌ی عصبی دارای پارامترهای بسیار زیادی می‌باشد. همچنین مدل LANS نیز کاربرد عام ندارد و فقط مخصوص مدل‌های تصمیم‌گیری می‌باشد.

۳- معرفی معماری رزنت

قبل از رزنت، محقق‌ها با عمیق کردن شبکه‌های عصبی برای بدست آوردن خروجی‌های چشم‌انگیزتر، به مشکلات مختلفی از جمله محوشدگی گرادیان و انفجار گرادیان و در نهایت کاهش دقت عملکرد شبکه بر می‌خوردند. شیوکی‌نگ رن و همکارانش [10] با ارائه معماری رزنت کمک بزرگی به علم یادگیری ماشین کرده‌اند و باعث تحول این حوزه شده‌اند. رن و همکارانش با معرفی اتصالات میانبر راه حل مشکل‌های شبکه‌های بسیار عمیق را پیدا کرده‌اند. اتصال میانبر، همان طور که از اسمش پیداست، از یک یا چند لایه عبور می‌کند

$$\Sigma \sim \text{InvWisHart}(\Lambda_0, v_0), \mu | \Sigma \sim N_d(\mu_0, \frac{\Sigma}{k_0})$$

$$x_i | \mu, \Sigma \sim N_d(\mu_0, \Sigma) \rightarrow i = 1 \dots m. \quad (4)$$

که در این مدل $\text{InvWisHart}(\Lambda_0, v_0)$ توزیع معکوس Wishart پارامتر شده توسط یک ماتریس مقیاس Λ_0 به ابعاد $d \times d$ و درجه آزادی v_0 ، همچنین $N_d(\mu_0, \frac{\Sigma}{k_0})$ توزیع نرمال چند متغیره با بردار میانگین d بعدی μ و ماتریس کوواریانس Σ به ابعاد $d \times d$ است. با تعریف پارامترهای زیر، صد هزار داده از فرمول فوق تولید شد.

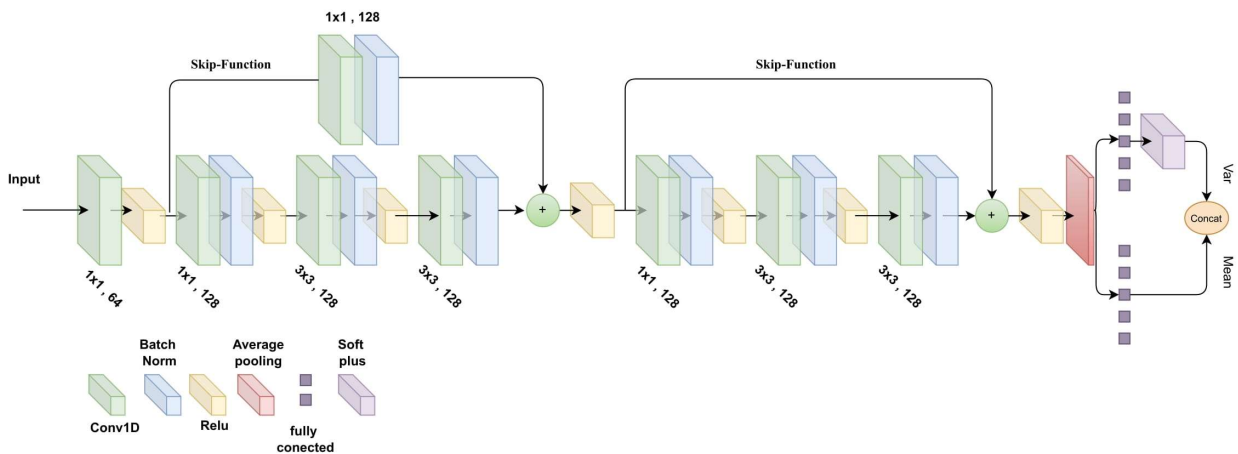
$$d = 2, \quad m = 100, \quad \Lambda_0 = I_2, \quad v_0 = 5, \\ k_0 = 3, \quad k_0 = 3$$

اگر مثال را کمی پیچیده تر کنیم، با تعریف داده های شبیه سازی شده ماتریس کوواریانس $X = (x_1 \dots x_d), m \times d$ می توان مدل را به صورت زیر تعریف کرد:

$$\sigma^2 \sim IG(a_0, b_0), \\ \beta | \sigma^2 \sim N_d(0, n\sigma^2(X^T X)^{-1}), \\ y | \beta, \sigma^2 \sim N_m(x\beta, \sigma^2 I),$$

که $IG(a_0, b_0)$ نشان دهنده توزیع گاما معکوس با بعد a_0 و مقیاس b_0 و وکتور $\beta = (\beta_1 \dots \beta_d)$ وزن های رگرسیون با توزیع نرمال d بعدی هستند و تابع درست نمایی توزیع نرمال m بعدی با ماتریس کواریانس ایزوتروپیک است.

با تعریف پارامترهای $d = 2, a_0 = 4, b_0 = 3, m = 100$ و تولید یک جدول مرجع شامل صد هزار ردیف از نمونه های یکسان توزیع شده $\{\tilde{x}_k^{(i)}\}_{k=1}^{100}$ که هر کدام شامل یک سه تایی از دو متغیر و یک خروجی حاصل از آن دو (x_1^i, x_2^i, y^i) است، تولید می کنیم و سپس به کمک شبکه عصبی پارامترهای داده های تولید شده متناظر را بدست می آوریم $(\beta_1^i, \beta_2^i, \sigma^2(i))$.



شکل (۱): معماری طراحی شده که شامل دو بلوک میانبر و اتصال میانبر اول دارای لایه های پیچشی و نرمال سازی دسته ای است.

و آن ها را در نظر نمی گیرد. با این کار (اضافه کردن خروجی لایه های پیشین به خروجی لایه فعلی)، ثابت می شود با وجود شبکه های بسیار عمیق، وزن های مدل مقادیر غیر صفر خواهند داشت و به دنبال آن مشکل محو شدن گرادیان حل می شود.

در این بخش معماری ساده رزنت تعریف شده است. دلیل انتخاب ساده بودن مدل این است که سعی داشتیم بتوانیم تا حد امکان پیچیدگی معماری را کاهش دهیم و تعداد پارامترهای حاصل از شبکه عصبی قابل مقایسه با سایر مدل ها همچون استنتاج-عمیق باشد. پس از تعریف معماری، نتایج آن را در مقایسه با نتایج معماری [16] در تخمین پارامتر دو مثال ساده نرمال چند متغیره و رگرسیون چندگانه و همچنین بر روی دو مدل تصمیم گیری رانش-انتشار و رانش-انتشار فروپاشی گزارش خواهد شد.

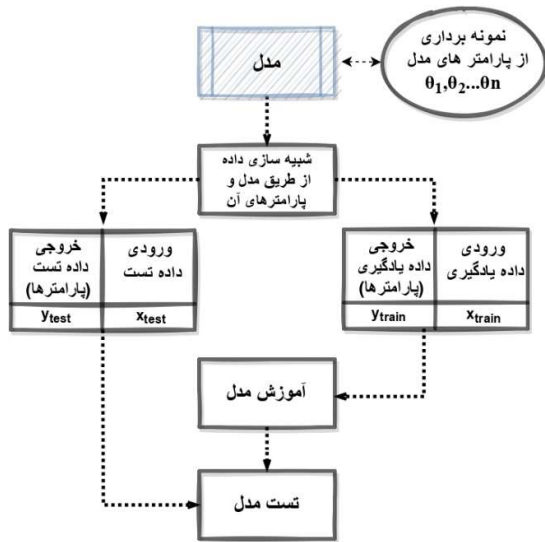
معماری رزنت که در شکل (۱) قابل مشاهده است. در ابتدا بر روی ورودی یک لایه کانولوشن با مقدار فیلتر ۶۴ و اندازه کرنل یک و سپس تابع فعال ساز اعمال شده است. در ادامه معماری شامل دو بخش است که در هر بخش سه لایه وجود دارد که به ترتیب در لایه ها از لایه پیچشی، لایه نرمال سازی دسته ای و تابع فعال ساز یکسوساز استفاده شده است. در تابع میانبر که برای مرحله اول می باشد به ترتیب از لایه پیچشی و لایه نرمال سازی دسته ای استفاده شده است و در مرحله دوم، تابع میانبر به صورت ساده است.

این تابع میانبر در انتهای هر بخش با خروجی آن بخش جمع می شود. تمامی مدل های شبکه های پیچشی از طریق واسط برنامه نویسی کاربردی مخصوص شبکه های عصبی، Tensorflow [23]، با زبان پایتون پیاده سازی شده است.

۳-۱- مقایسه نتایج استنتاج-عمیق و رزنت بر روی توزیع نرمال چند متغیره و رگرسیون چندگانه

با تعریف مدل توزیع نرمال چند متغیره با میانگین و انحراف معیار ناشناخته، خروجی معماری را بر روی آن می سنجیم. توزیع مزدوج برای میانگین μ و ماتریس کواریانس Σ به صورت فرمول (۴) است

۳-۲- مدل تصمیم‌گیری رانش-انتشار و رانش-انتشار فروپاشی



شکل (۳): روند کلی تولید داده برای مدل و استفاده از آن. یک فرایند طبیعی با استفاده از معادلات دیفرانسیل معمولی مدل می‌شود که شامل تعدادی پارامتر است. با مقداری مناسب به این پارامترها و جایگذاری آن‌ها در مدل تعدادی داده شبیه‌سازی می‌شود که از آن‌ها برای یادگیری استفاده می‌شود. پس از یادگیری برای ارزیابی مدل از داده‌های شبیه‌سازی شده‌ی متفاوتی استفاده می‌شود.

با استفاده از روش کار فوق چهار هزار داده تولید شد و سپس داده‌ها را بر روی مدل پیشنهادی خود رزنت و مدل استنتاج-عمیق اعمال کرده و نتایج مناسبی به دست آورده‌شد.

برخلاف مدل رانش-انتشار، مرزهای تصمیم‌گیری در رانش-انتشار فروپاشی ثابت نیستند و وابسته به زمان هستند و به صورت فرمول (۶) تعریف می‌شوند

$$l(t) = \frac{at}{2 * T_{\infty}} \quad , \quad u(t) = a - l(t). \quad (6)$$

که در آن a مرز تصمیم‌گیری است و وابسته به زمان است و T_{∞} پارامتری است که نشان می‌دهد مرزهای بالا و پایین چه زمانی به یکدیگر الحاق می‌شوند. مشابه قبل تعداد چهار هزار داده شبیه‌سازی شد و سپس داده‌ها بر روی مدل پیاده شد. از آنجایی که تعداد پارامترهای مدل رانش-انتشار فروپاشی بیشتر است و همچنین برای اثبات آنکه با افزایش لایه‌های رزنت خروجی بهتر می‌شود، معماری رزنت پیشین کمی عمیق‌تر در نظر گرفته شد.

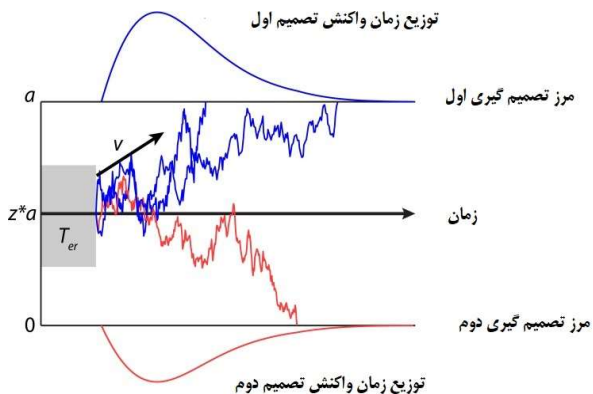
۴- مقایسه نتایج

نتایجی که به دست آوردیم نشان می‌دهد که در اکثر موارد رزنت یا مشابه معماری استنتاج-عمیق عمل کرده و یا نتایج بهتری به دست آورده است. در جدول (۱) نتایج به دست آمده توسط معماری رزنت بر روی مثال اول و در جدول (۲) نتایج بدست آمده توسط معماری رزنت بر روی مدل مثال دوم قابل مشاهده می‌باشد.

مدل رانش-انتشار، یکی از مدل‌های مهم در زمینه مسائل تصمیم‌گیری است. این مدل رفتار دو انتخاب و یک زمان واکنش را توصیف می‌کند. این مدل فرض بر آن دارد که فرد برای گرفتن یک تصمیم برای خود دو مرز در نظر می‌گیرد؛ یک مرز برای گرفتن یک تصمیم و مرز دیگری را برای رد این تصمیم. برای رسیدن به این دو مرز با یک نرخ رانش از طریق قدم‌زنی تصادفی استفاده می‌شود. به این صورت که هر قدمی که به نفع تصمیم‌گیرنده باشد انتخاب می‌شود و به مرز مربوطه نزدیک می‌شود. این مدل شامل چهار پارامتر a ، v ، Z و t_0 است. پارامتر a ، جدا کننده آستانه است و نشان‌دهنده فاصله بین دو مرز بالا و پایین یا همان میزان اطلاعات لازم برای گرفتن تصمیم است. پارامتر v برای نمایش نرخ رانش و نشان‌دهنده میانگین شیب فرایند تصمیم‌گیری از نقطه شروع تا نقطه پایان است. در نهایت پارامتر Z ، نقطه شروع فرایند تصمیم‌گیری و نشان‌دهنده میزان تمایل به یکی از تصمیم‌ها و پارامتر t_0 برای نمایش زمان عدم تصمیم‌گیری است. عدم تصمیم‌گیری بیانگر آن است که فرد تصمیم‌گیرنده از همان ابتدا شروع به تصمیم‌گیری نمی‌کند بلکه زمانی را صرف پردازش اطلاعات می‌کند. علاوه بر آن زمان تاخیر آنکه تصمیم در مغز گرفته شود و دستور از آن به سایر اندام محرک انسان برسد نیز وجود دارد، به مجموع این دو زمان، پردازش اطلاعات و تاخیر صورت گرفته، زمان عدم تصمیم‌گیری گفته می‌شود. نمای کلی این مدل در شکل (۲) آورده شده است. به عنوان مثال نقطه شروع Z آغاز تصمیم‌گیری نام گذاری می‌شود که باید تصمیمی گرفته شود (سقف a) یا نشود (کف 0)، این مدل رفتار روند تصمیم‌گیری را در طی زمان به نمایش می‌گذارد

$$x(t + t') = x(t) + v \times t' + N(1, \alpha) \times t'. \quad (5)$$

از آنجایی که ما رفتار مدل را با مدل ریاضی، فرمول (۵)، می‌دانیم و به داده‌ای دسترسی نداریم، داده‌ها را تولید می‌کنیم. به این صورت که ابتدا مقادیر تصادفی و قابل قبولی از پارامترها را تولید می‌کنیم. سپس با مقادیر داده‌های مختلفی مقادیر Y یا خروجی را به دست آورده و ذخیره می‌کنیم. شمای کلی روند تولید داده و استفاده از آن را می‌توان در شکل (۳) مشاهده کرد.



شکل (۲): نمای از مدل رانش انتشار، T_{er} زمان عدم تصمیم‌گیری را نشان می‌دهد که تصمیم‌گیرنده از مرز تصمیم‌گیری خنثی با نرخ رانش v به سوی یکی از مرزها حرکت می‌کند.

جدول (۱) : نتایج دو معماری بر توزیع نرمال چند متغیره با معیار خطای جذر میانگین مربعات

	مدل رزنت	استنتاج-عمیق
$E[\mu_1 x]$	۰.۰۱۱	۰.۰۲۲
$E[\mu_2 x]$	۰.۰۰۹	۰.۰۱۷
$E[\sigma_1^2 x]$	۰.۰۵۳	۰.۰۴۲
$E[\sigma_2^2 x]$	۰.۰۳۰	۰.۰۳۸
$Var[\mu_1 x]$	۰.۰۰۲	۰.۰۰۲
$Var[\mu_2 x]$	۰.۰۰۰	۰.۰۰۱
$Var[\sigma_1^2 x]$	۰.۰۶۳	۰.۰۴۶
$Var[\sigma_2^2 x]$	۰.۰۰۳	۰.۰۱۳

جدول (۲) : نتایج دو معماری بر رگرسیون چندگانه با معیار خطای جذر میانگین مربعات

	مدل رزنت	استنتاج-عمیق
$E[\beta_1 x]$	۰.۰۵۹	۰.۱۰۳
$E[\beta_2 x]$	۰.۰۶۹	۰.۰۷۸
$E[\sigma^2 x]$	۰.۱۴۸	۰.۲۳۴
$Var[\beta_1 x]$	۰.۰۱۸	۰.۰۲۵
$Var[\beta_2 x]$	۰.۰۲۴	۰.۰۳۲
$Var[\sigma^2 x]$	۰.۲۴۵	۰.۲۳۵

همچنین نتایج حاصل بر روی دو مدل تصمیم‌گیری رانش-انتشار و رانش-انتشار فروپاشی نیز نشان‌دهنده آن است که مدل پیشنهادی می‌تواند به خوبی پارامترهای این دو مدل را تخمین بزند.

نتایج حاصل از شبکه بر روی این دو مدل در جدول (۳ و ۴) به ترتیب آورده شده است که نشان می‌دهد از رزنت برای مدل‌های پیچیده‌تر همچون این مدل می‌توان استفاده کرد و نتایج بهتری دریافت کرد.

جدول (۳) : نتایج دو معماری بر روی مدل رانش-انتشار با معیار سنجش ضریب همبستگی پیرسون

	مدل رزنت	استنتاج-عمیق
a	۰.۹۱	۰.۸۴
v	۰.۹۸	۰.۸۹
zr	۰.۸۶	۰.۸۱
Ndt	۰.۹۵	۰.۸۷

جدول (۴) : نتایج دو معماری بر روی مدل رانش-انتشار فروپاشی با معیار سنجش ضریب همبستگی پیرسون

	مدل رزنت	استنتاج-عمیق
a	۰.۷۳	۰.۶۹
v	۰.۷۹	۰.۶۴
zr	۰.۷۲	۰.۷
ndt	۰.۸۸	۰.۸۳
T _{inf}	۰.۶۹	۰.۶۴

۵- کارهای آینده

مسأله‌ای که وجود دارد این است که ممکن است در مواقعی به سراغ مدل‌هایی برویم که به زمان وابستگی دارند مانند مدل‌هایی که برای پیش‌بینی وضعیت شیوع یک بیماری در یک جامعه در طی زمان به کار می‌روند و یا اگر لازم باشد که وضعیت بورس کشور در طی زمان را مدل کنیم. در این شرایط به دنبال خانواده‌ی خاصی از شبکه‌های عصبی به نام شبکه‌های عصبی دیفرانسیل معمولی [3] می‌رویم و آن را مورد بررسی قرار می‌دهیم.

در شبکه عصبی میانبر تعدادی لایه مخفی وجود دارند که به صورت گسسته و مجزاء می‌باشند. اگر گام‌های شبکه را به سمت بی‌نهایت سوق دهیم و مدل از حالت گسسته به پیوسته تبدیل شود می‌توان امید داشت که از این شبکه برای مدل‌های وابسته به زمان استفاده کرد. از طرف دیگر برای مسائلی که با داشتن یک وضعیت ابتدایی بتواند یک روند در آینده را پیش‌بینی کند که برای حل شدن نیاز به یک حل‌کننده دارند (فرمول (۷)). به دلیل شباهتی که بین رابطه روش اویلر که یک حل‌کننده است با بلوک میانبر وجود دارد می‌توان نتیجه گرفت که از آن می‌توان به عنوان یک حل‌کننده ساده استفاده کرد

$$z(t_1) = z(t_0) + \int_{t_1}^{t_0} f(z(t_0), t, \theta) dt. \quad (7)$$

شبکه میانبر شامل چندین بلوک میانبر است و طبق مطالب گفته شده هر بلوک به عنوان یک حل‌کننده عمل می‌کند. حال اگر از یک حل‌کننده معادلات دیفرانسیل معمولی به جای تمامی بلوک‌ها استفاده کنیم تعداد ارزیابی‌های بین لایه‌ها افزایش می‌یابند. در این شبکه هرچه زمان بیشتری صرف آموزش شبکه شود در بخش تست دقت بالاتری خواهیم داشت و به دلیل اینکه شبکه سنگین و با تعداد لایه‌های بسیار زیادی است برای بهینه‌سازی پارامترها نمی‌توان از پس انتشار استفاده کرد و از روش حالت الحاقی برای این کار استفاده می‌شود که این کار باعث می‌شود میزان مصرف از حافظه به شدت کاهش پیدا کند. با جای‌گذاری تمامی بلوک‌های میانبر با یک حل‌کننده معادله دیفرانسیل معمولی به یک شبکه‌ی عصبی دیفرانسیل معمولی می‌رسیم.

۶- نتیجه گیری

با توجه به مفاهیمی که تا به الان گفته شد، مدل‌هایی وجود دارند که پارامترهای آن‌ها برای ما مهم است و اگر این مدل‌ها پیچیده شوند تقریب این پارامترها کار آسانی نیست پس باید به سوی روش‌های تقریب پارامتر رفت. یادگیری عمیق و استفاده از شبکه‌های عصبی یکی از این روش‌ها است. در این روش به بررسی معماری رزنت پرداختیم که خروجی مطلوبی را در مقایسه با موارد اشاره شده داشت. همچنین طبق ویژگی‌های اصلی شبکه‌های عصبی میانبر، هر مقدار این شبکه‌ها را عمیق‌تر کنیم، نتایج بهتری را به دست خواهیم آورد که این مورد در استنتاج-عمیق قابل مشاهده نبود. با این حال مدل‌های زیاد و مهمی وجود دارد که وابسته به زمان هستند. مدل ساده مطرح شده نمی‌تواند پارامترهای این مدل را به خوبی بازایی کند، مگر این که از شبکه‌های بسیار عمیق رزنت استفاده شود، که بدلیل بار محاسباتی بسیار بالا و طولانی بودن فرآیند یادگیری توجه ما را جلب نمی‌کند. توضیح داده شد که معماری جدیدی به نام شبکه‌های عصبی دیفرانسیل معمولی وجود دارد که به دلیل نوع طراحی این مدل بسیار با مدل‌های زمانی سازگار است و همچنین جایگزین معماری رزنت است. روشی با نام روش حالت الحاقی وجود دارد که نسبت تغییرات تابع خسارت به پارامتر را می‌سنجد. روش حالت الحاقی یک روش عددی برای محاسبه بهینه گرادیان یک تابع یا عملگر در یک مسئله بهینه سازی عددی است. در ادامه قرار است که نسبت تابع خسارتی را که در رزنت استفاده کردیم را به پارامترهایی که می‌خواهیم تقریب بزنیم، بسنجیم و معماری را با تبدیل به شبکه‌های عصبی دیفرانسیل معمولی برای سازگاری با مدل‌های زمانی تغییر دهیم.

مراجع

- [7] Gabbard, H., Messenger, C., Heng, I.S., Tonolini, F. and Murray-Smith, R., 2022. *Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy*. Nature Physics, 18(1), pp.112-117.
- [8] Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B., 1995. *Bayesian data analysis*. Chapman and Hall/CRC.
- [9] Green, S.R., Simpson, C. and Gair, J., 2020. *Gravitational-wave parameter estimation with autoregressive neural network flows*. Physical Review D, 102(10), p.104057.
- [10] He, K., Zhang, X., Ren, S. and Sun, J., 2016. *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [11] Jiang B, Wu TY, Zheng C, Wong WH. *Learning summary statistic for approximate Bayesian computation via deep neural network*. Statistica Sinica. 2017 Oct 1:1595-618.
- [12] Liu, F., Li, X. and Zhu, G., 2020. *Using the contact network model and Metropolis-Hastings sampling to reconstruct the COVID-19 spread on the "Diamond Princess"*. Science Bulletin, 65(15), pp.1297-1305.
- [13] Long, J., Shelhamer, E. and Darrell, T., 2015. *Fully convolutional networks for semantic segmentation*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
- [14] Mertens UK, Voss A, Radev S. ABrox—A user-friendly Python module for approximate Bayesian computation with a focus on model comparison. PloS one. 2018 Mar 8;13(3):e0193981.
- [15] Palestro JJ, Sederberg PB, Osth AF, Van Zandt T, Turner BM. *Likelihood-free methods for cognitive science*. Springer International Publishing; 2018 Feb 7.
- [16] Radev, S.T., Mertens, U.K., Voss, A. and Köthe, U., 2020. *Towards end-to-end likelihood-free inference with convolutional neural networks*. British Journal of Mathematical and Statistical Psychology, 73(1), pp.23-43.
- [17] Radev, S.T., Mertens, U.K., Voss, A., Ardizzone, L. and Köthe, U., 2020. *BayesFlow: Learning complex stochastic models with invertible neural networks*. IEEE transactions on neural networks and learning systems.
- [18] Ratcliff, R., 1978. *A theory of memory retrieval*. Psychological review, 85(2), p.59.
- [19] Raynal L, Marin JM, Pudlo P, Ribatet M, Robert CP, Estoup A. *ABC random forests for Bayesian parameter inference*. Bioinformatics. 2019 May 15;35(10):1720-8.
- [20] Sheehan, S. and Song, Y.S., 2016. *Deep learning for population genetic inference*. PLoS computational biology, 12(3), p.e1004845.
- [21] Tavaré, S., Balding, D.J., Griffiths, R.C. and Donnelly, P., 1997. *Inferring coalescence times from DNA sequence data*. Genetics, 145(2), pp.505-518.
- [22] Yin, M.Z., Zhu, Q.W. and Lü, X., 2021. *Parameter estimation of the incubation period of COVID-19 based on the doubly interval-censored data model*. Nonlinear Dynamics, 106(2), pp.1347-1358.
- [23] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. *{TensorFlow}: a system for {Large-Scale} machine learning*. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).
- [1] Bird, S.M., Sir David, C., Farewell, V.T., Harvey, G., Tim, H. and Peter C, S., 2005. *Performance indicators: good, bad, and ugly*. Journal of the Royal Statistical Society: Series A (Statistics in Society), 168(1), pp.1-27.
- [2] Breiman, L., 2001. *Random forests*. Machine learning, 45(1), pp.5-32.
- [3] Chen, R.T., Rubanova, Y., Bettencourt, J. and Duvenaud, D.K., 2018. *Neural ordinary differential equations*. Advances in neural information processing systems, 31.
- [4] Chen, Y. and Zhou, Y., 2020. *Machine learning based decision making for time varying systems: Parameter estimation and performance optimization*. Knowledge-Based Systems, 190, p.105479.
- [5] Csilléry, K., Blum, M.G., Gaggiotti, O.E. and François, O., 2010. *Approximate Bayesian computation (ABC) in practice*. Trends in ecology & evolution, 25(7), pp.410-418.
- [6] Fengler, A., Govindarajan, L.N., Chen, T. and Frank, M.J., 2021. *Likelihood approximation networks (LANs) for fast inference of simulation models in cognitive neuroscience*. Elife, 10, p.e65074.