مجله کنترل
ISSN (print) 2008-8345
ISSN (online) 2538-3752
مقاله پژوهشی
جلد۱۷، شماره ۲، تابستان ۱۴۰۲، صفحه ۲۳-۱

مدل‌های مبتنی بر یادگیری عمیق در شناسایی سیستم‌های غیرخطی

وحید محمدزاده ایوقی[۱]، مهدی علیاری شوره‌دلی[۲]

[۱] دانشجوی دکتری مهندسی کنترل، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران vmohammadzadeh@email.kntu.ac.ir
[۲] دانشیار گروه مکاترونیک، دانشکده مهندسی برق، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران aliyari@kntu.ac.ir

دریافت: ۱۴۰۲/۰۵/۰۵ پذیرش: ۱۴۰۲/۰۶/۲۰

چکیده – مدل‌های مبتنی بر یادگیری عمیق عملکرد بسیار مناسبی در مدل‌سازی مسائل پیچیده در بینایی ماشین و پردازش زبان طبیعی از خود نشان داده‌اند که این موضوع به ماهیت غیرخطی و فراپارامتری این مدل‌ها نسبت داده می‌شود. روش‌های شناسایی سیستم‌های غیرخطی، می‌تواند از ابزارهای توسعه‌یافته در حوزه یادگیری عمیق بهره‌مند شوند که این امر موجب گسترده شدن ابزارهای موجود برای انتخاب یک مدل مناسب خواهد شد. ازاین‌رو، در این مقاله قصد داریم تا روش‌ها و ساختارهای موجود در یادگیری عمیق را از دیدگاه شناسایی سیستم‌های غیرخطی مرور کنیم. اگرچه مرور نسبتاً جامعی از ابزارهای قابل‌استفاده در حوزه شناسایی سیستم‌های غیرخطی ارائه خواهد شد، اما تمرکز اصلی این پژوهش بر کاربرد مدل‌های متغیر پنهان در شناسایی فضای حالت غیرخطی است. مدل‌های متغیر پنهان دسته‌ای از مدل‌های یادگیری عمیق هستند که در گروه مدل‌های مولد قرار می‌گیرند. نسخه اصلی این مدل‌ها، تنها قابلیت تولید داده‌های ایستا را داراست. اما با ترکیب شبکه‌های عصبی بازگشتی و خود رمزنگار تغییراتی، قابلیت تولید داده‌های ترتیبی برای این مدل‌ها نیز فراهم شده است. همچنین نسخه ساختاریافته‌ای از این مدل‌ها نیز که منطبق بر سیستم‌های کنترل است، ارائه خواهد شد. مطالعهٔ انجام شده نشان می‌دهد که عملکرد این مدل‌ها با مدل‌های پیشین و کلاسیک موجود، قابل‌قیاس است.

کلیدواژه: مدل‌های یادگیری عمیق، شناسایی سیستم‌های غیرخطی، مدل‌های متغیر پنهان، خود رمزنگار تغییراتی

Deep Learning based Models for Nonlinear System Identification

Vahid Mohammadzadeh Ayooghi and Mahdi Aliyari-Shoorehdeli

Abstract – Deep learning-based models appropriately perform in modeling complex problems in computer vision and natural language processing (NLP). With this in mind, nonlinear system identification methods can benefit from tools developed in deep learning, leading to enriched frameworks to choose from. For this purpose, we are going to review some potential structures and methods of deep learning that can be used in nonlinear system identification. Although we comprehensively review the applicable tools of deep learning to system identification, this paper mainly focuses on using latent variable models (LVM) for identifying nonlinear state space models. LVMs are powerful tools for extending generative models primarily developed for only generating static data, yet by a combination of recurrent neural network (RNN) and variational auto-encoders (VAE), they can also generate sequential data. A structured version of introduced models compatible with control systems will also be given. The study shows that the deep learning models have a comparative performance to traditional and classic ones.

Keywords: Deep learning, nonlinear system identification, latent variable models, VA

۱-مقدمه

شناسایی سیستم‌ها[1] یکی از زمینه‌های حوزه کنترل خودکار[2] است که بر اساس داده‌های ورودی و خروجی جمع‌آوری شده از یک سیستم واقعی، تلاش در پیداکردن یک مدل مناسب دینامیکی دارد [۱]. اهمیت این مسئله و توسعه ابزارهای آن در یک چارچوب مشخص، برای اولین بار در [۲] برای سیستم خطی تغییر ناپذیر با زمان[3] آغاز شد و به مرور زمان منجر به توسعه ابزارهای متنوع حوزه شناسایی سیستم‌های خطی شده است [۱] ، [۳] و [۴]. محدودیت این روش‌ها در مدل کردن فرآیندهای صنعتی و واقعی که عمدتا دارای یک رفتار غیرخطی[4] هستند [۵] و [۳] ، پژوهشگران را بر آن داشت تا یک چارچوب برای شناسایی سیستم‌های غیرخطی نیز ایجاد شود که نمونه مناسبی از مجموعه این روش‌ها در [۳] به تفصیل آمده است. فارغ از عدم موفقیت تلاش‌های روزافزون، روش‌های توسعه داده شده برای مدل‌سازی غیرخطی، به طور کلی به سه دسته توسعه تابعی[5]، بلوک-محور[6] و جعبه سیاه[7] دسته‌بندی می‌شوند [۶]. روش توسعه ولترا[8] و وینر[9] از جمله رویکردهای مشهور توسعه تابعی است [۶]، [۷] و [۸]. این روش‌ها مسئله شناسایی سیستم را، به صورت یافتن یک تابعی[10] از ورودی‌های سیستم به خروجی‌های سیستم مشاهده می‌کنند، و این تابعی را به شیوه‌های خاص خود تجزیه می‌کنند. بر این اساس مسئله شناسایی سیستم، به مسئله تخمین پارامترهای هر بخش تجزیه شده تبدیل خواهد شد [۶]. رویکرد بلوک-محور، با فرض جداپذیر بودن فرآیند[11]، یک سیستم غیرخطی را به یک سیستم خطی دینامیکی و یک سیستم غیرخطی ایستاک[12] تجزیه می‌کنند که در آن سیستم غیرخطی هر نوع تقریب‌زنی اعم از شبکه‌های عصبی[13] و سیستم‌های فازی[14] می‌تواند باشد. بر اساس اتصال این دو بخش تجزیه شده، مدل‌های متفاوتی حاصل می‌شود که مدل‌های وینر[15]، همراشتاین[16] و وینر-همراشتاین[17] از جمله معروف‌ترین مدل‌های این رویکرد است [۹]. رویکرد جعبه سیاه مجموعه‌ای از دینامیک‌های ورودی را دریافت می‌کند و نگاشتی از آنها را بر روی مجموعه‌ای از داده‌های خروجی محاسبه می‌کنند. روش‌های مبتنی بر کرنل[18]، مدل‌های خطی محلی[19]، و استفاده از شبکه‌های عصبی بازگشتی[20] از جمله این روش‌ها هستند [۳].

از دیدگاه ریاضی، مسئله شناسایی سیستم، مسئله پیداکردن یک تابع ریاضی است که ورودی‌های این تابع ریاضی، متغیرهای دینامیکی ورودی، خروجی و یا خطای پیش‌بینی می‌تواند باشد. روش‌های شناسایی خطی، خروجی سیستم را در هر لحظه بر اساس ترکیب خطی‌ای از این ورودی‌ها بیان می‌کنند درحالی‌که روش‌های شناسایی غیرخطی، خروجی را در هر لحظه بر اساس ترکیب خطی‌ای از توابع غیرخطی که بر روی این ورودی‌ها اعمال می‌شوند، محاسبه می‌کنند که این رویکرد با نام مدل توابع پایه[21] شناخته شده است [۳]. از دیدگاه جبر خطی، می‌توان گفت هر دو روش سعی در پیداکردن یک فضای برداری دارند که در آن خروجی به‌سهولت ساخته می‌شود. برای روش‌های خطی این فضای مربوط به زیر فضای ورودی‌های مدل است درحالی‌که روش‌های غیرخطی فضایی متفاوت از فضای ورودی را به‌عنوان فضای حل مسئله انتخاب می‌کنند. در واقع دسته-بندی معرفی شده برای روش‌های شناسایی غیرخطی، تفاوت دیدگاه‌های موجود برای مدل‌سازی این نگاشت غیرخطی عنوان می‌کنند. به‌عنوان‌مثال برای مدل‌های جعبه سیاه، نوع نگاشت غیرخطی محاسبه شده اهمیت چندانی ندارد و تنها مسئله مهم نزدیک‌شدن خروجی مدل به خروجی سیستم است. یا در روش‌های مبتنی بر کرنل، در دید کلی، فرایندهای گوسی، این نگاشت غیرخطی بر اساس شباهت بین نمونه‌ها در فضای ورودی ساخته می‌شود [۳]، [۵]، [۱۰] و [۱۱].

دیدگاه بیان شده برای روش‌های شناسایی سیستم‌ها، دقیقاً منطبق بر فرایندی است که در حوزه یادگیری ماشین[22] برای حل یک مسئله داده - محور انجام می‌شود. داده‌هایی که در اختیار طراح برای حل یک مسئله مشخص قرار می‌گیرد، ابتدا به یک فضایی نگاشت می‌شوند که در فضای جدید حل مسئله سادگی قابل‌انجام باشد. به نمایش داده‌ها در فضای جدید، در ادبیات یادگیری ماشین، بازنمایی[23] گفته می‌شود. تمام تلاش‌ها در یک مسئله یادگیری ماشین یافتن یک بازنمایی مناسب برای حل مسئله است [۱۲]. در یادگیری ماشین کلاسیک، این بازنمایی بر اساس مجموعه‌ای از نگاشت‌های ثابت اعم از چندجمله‌ای[24]، توابع پایه شعاعی[25]، توابع کرنل[26]،

[1] System identification
[2] Automatic control
[3] Linear time-invariant system
[4] Nonlinear behavior
[5] Functional expansion
[6] Block-oriented models
[7] Black box
[8] Volterra expansion
[9] Wiener expansion
[10] Functional
[11] Process separability
[12] Static nonlinearity
[13] Neural networks
[14] Fuzzy systems
[15] Wiener model
[16] Hammerstein model
[17] Wiener-Hammerstein model
[18] Kernel-based methods
[19] Locally linear models
[20] Recurrent neural networks
[21] Basis function models
[22] Machine learning
[23] Representation
[24] Polynomial mapping
[25] Radial basis function
[26] Kernel functions

تبدیلات حوزه فرکانس[1] انجام می‌شود که به دلیل شکل و محدودیتی که در اعمال آن‌ها به مجموعه داده‌های ورودی دارد [۱۲] و [۱۳]، انتخاب و استفاده از آنها تا حد زیادی به دانش طراح بستگی دارد. مهم‌تر اینکه این نگاشت‌های ثبات در برخی از شرایط، مانند داده‌های با بعد بالا[2] و سیستم‌ها پیچیده[3]، عملکرد مناسبی از خود نشان نمی‌دهند و به بیان بهتر، در مواجه با برخی از مسائل نیاز است تا بازنمایی‌های پیچیده‌تری از متغیرهای ورودی مسئله پیدا کرد که طبیعتا امکان این مسئله به صورت دستی وجود ندارد. زیرا تعداد نامتناهی تابع وجود دارد که می‌تواند به داده‌ها اعمال شود. این مشکل دقیقا فلسفه ایجاد مدل‌هایی موسوم به مدل‌های عمیق[4] است که نسخه معین[5] آن شبکه‌های عصبی عمیق[6] است [۱۳]. شبکه‌های عصبی عمیق نوعی از مدل‌های عمیق هستند که بر روی داده‌های ورودی نگاشت‌های غیرخطی پی‌درپی اعمال می‌کند که در نتیجه اعمال این نگاشت‌های غیرخطی، نگاشت پیچیده‌تری از متغیرهای ورودی ساخته می‌شود که با تکیه بر تکنولوژی‌های یادگیری که روز به روز در حال پیشرفت هست، امکان کنترل این بازنمایی نیز وجود دارد [۱۳]. کنترل بازنمایی به معنای شفافیت در یادگیری لایه‌های پنهان یک شبکه عصبی است که این موضوع درک مدل را از داده‌ها نشان می‌دهد.

الگوریتم‌های متفاوتی برای ایجاد درک مناسب از داده‌ها، و به بیان بهتر کنترل بازنمایی، در حوزه یادگیری ماشین ایجاد شده‌اند که عمدتاً جزو دسته یادگیری بدون نظارت هستند [۱۴]. این روش‌ها در حالت کلی به سه‌دسته قابل تقسیم هستند، نگاشت مستقیم[7]، مدل‌های احتمالاتی[8]، و یادگیری رویه[9] [۱۵]. روش نگاشت مستقیم مبتنی بر یادگیری بازنمایی‌ای از داده‌هاست که این بازنمایی اولا از یک نگاشت معین حاصل می‌شود و ثانیا این بازنمایی توانایی بازگشت به فضای اصلی را دارد. این روش‌های به صورت مستقیم از متغیرهای ورودی، ویژگی‌هایی را در فضایی با بعد پایین‌تر استخراج می‌کنند. مهم‌ترین ابزاری که در این دسته‌بندی قرار می‌گیرد خودرمزنگارها[10] هستند که از دو شبکه عصبی کدکننده[11] و کدگشا[12] تشکیل می‌شوند و شبکه با حداقل کردن خطای بازیابی آموزش می‌بینند [۱۳] و [۱۵]. خودرمزنگارها انواع متفاوتی دارند که تمام انواع به جز خودرمزنگار تغییراتی[13] [۱۶] ، مثال‌هایی از رویکرد نگاشت مستقیم برای یادگیری بازنمایی هستند [۱۵]. رویکرد نگاشت مستقیم، از طریق یک نگاشت معین، بازنمایی‌ای را آموزش می‌بینند که این بازنمایی قابلیت برگشت‌پذیری به فضای اصلی متغیرهای ورودی را داراست. بنابراین مسئله کنترل بازنمایی در این رویکرد منجر به یادگیری فضایی خواهد شد که مشخصات ثابت و معین متغیرهای را ورودی را داشته باشد. با استناد به این اصل که در متغیرهای یک مسئله پیچیده مانند تشخیص گفتار، هر دو مشخصات تصادفی و معین وجود دارد، این رویکرد قابلیت استخراج تمام مشخصات مرتبط با یک متغیر را ندارد و تنها مشخصات ثابت را استخراج می‌کند [۱۷]، [۱۸] و [۱۹]. مدل‌های احتمالاتی، متغیرهای مربوط به یک ساختار را به دو بخش تقسیم می‌کنند: متغیرهای مشاهده شده[14] و متغیرهای پنهان[15] [۱۸]. این دسته از مدل‌ها تحت عنوان مدل‌های متغیر پنهان[16] نیز شناخته می‌شوند و نوع بازنمایی آموزش دیده شده توسط این مدل‌ها با رویکرد نگاشت مستقیم تفاوت‌های زیادی دارد. این مدل‌ها بر این فرض استوار هستند که متغیرهای یک مسئله دارای دسته‌ای از ویژگی‌های پنهان هستند که این ویژگی‌ها تغییرات داده‌ها را توصیف می‌کنند. مسئله یادگیری بازنمایی در این مدل‌ها به صورت یک نگاشت تصادفی از متغیرهای پنهان به متغیرهای مشاهده شده تعریف می‌گردد. این مدل‌ها توانایی استخراج مشخصات تصادفی در متغیرهای ورودی را دارا هستند. ماشین بولتزمن محدود شده[17] [۲۰]، [۲۱] و [۱۳] و خودرمزنگار تغییراتی از جمله این روش‌ها هستند. ساختار خودرمزنگار تغییراتی دقیقا با ساختار خودرمزنگار عادی در رویکرد مستقیم یکسان است. بنابراین امکان استخراج مشخصات ثابت و تصادفی به صورت توامان توسط این ساختار وجود ندارد. مدل‌هایی در حوزه یادگیری ماشین توسعه داده شده‌اند که با ترکیب دو نگاشت معین و تصادفی هر دو مشخصات را از یک دسته ورودی محاسبه می‌کنند. به عنوان مثال در [۱۷] شبکه عصبی بازگشتی در کنار خودرمزنگار تغییراتی، امکان استخراج هر دو مجموعه از ویژگی‌ها را دارد به این صورت که شبکه عصبی بازگشتی ویژگی‌های ثابت و وابسته به زمان، و خودرمزنگار تغییراتی مشخصات تصادفی را استخراج می‌کند. روش یادگیری رویه، به طور مشابه با متغیرهای پنهان عمل می‌کند با این تفاوت که یادگیری بازنمایی بر اساس نزدیکی و شباهت بین نقاطی از داده‌های ورودی است که مدل مجبور به یادگیری آنها خواهد شد. یادگیری متضاد[18] [۲۲] از جمله روش‌هایی است که بر اساس شباهت بین نمونه‌های مجموعه‌ای از داده‌گان، بازنمایی مناسبی را از داده‌گان آموزش می‌بینند. اگرچه بسته به نوع کاربردها و مسائل متفاوت، هر یک از سه روش

[1] Frequency domain transformation
[2] High dimensional data
[3] Complex systems
[4] Deep models
[5] Deterministic version
[6] Deep neural network
[7] Direct mapping
[8] Probabilistic models
[9] Manifold learning
[10] Auto-encoders
[11] Encoder
[12] Decoder
[13] Variational auto-encoder
[14] Observed variables
[15] Latent variables
[16] Latent variable models
[17] Restricted Boltzmann machine
[18] Contrastive learning

یاد شده می‌تواند برای یادگیری بازنمایی مناسب باشد، اما یادگیری بازنمایی به کمک مدل‌های احتمالاتی از اهمیت بالایی برخوردار است. بدلیل ماهیت نگاشتی که این مدل‌ها استفاده می‌کنند، امکان تعمیم‌دهی[1] مدل به شرایط نادیده در این مدل‌ها بالاتر است زیرا این مدل‌ها تغییرات و عدم قطعیت‌ها موجود در مجموعه‌ای از داده‌ها را مدل می‌کنند. به بیان بهتر، این مدل‌ها قابلیت مدل کردن توزیع احتمال توام بین متغیرهای مشاهده شده و متغیرهای پنهان را داراست. در ادبیات یادگیری ماشین، این نوع مدل‌ها موسوم به مدل‌های مولد[2] است [۱۳] و [۱۶]. مدل‌های مولد به هدف‌های متفاوتی توسعه پیدا می‌کنند که یکی از این اهداف بهبود بازنمایی و بالا بردن ظرفیت تعمیم‌دهی مدل‌های یادگیری ماشین است [۱۳].

اگرچه زمینه شناسایی سیستم‌ها به لحاظ روش‌های موجود غنی است، اما استفاده از شبکه‌های عصبی عمیق به‌منظور شناسایی سیستم‌ها مزیت‌های فراوانی مانند حل مشکل بالابودن بعد متغیرهای ورودی و مدل‌سازی سیستم‌های پیچیده را به دنبال خود دارد. همچنین استفاده از تکنولوژی‌ها و الگوریتم‌های آموزشی در این حوزه، نه‌تنها روش‌های شناسایی سیستم‌ها را توسعه می‌دهد بلکه می‌تواند موجب به بهبود آنها شده و در برخی موارد تفسیرپذیری و شفافیت مدل‌های جعبه سیاه را تضمین کند. در این مقاله قصد داریم، تا ابزارهای یادگیری عمیق که در حوزهٔ شناسایی سیستم‌ها مورداستفاده قرار گرفته‌اند، مورد بررسی و توسعه قرار دهیم. باتوجه‌به اهمیت مدل‌های احتمالاتی برای یادگیری بازنمایی، این مدل‌ها در شناسایی ورودی – خروجی و شناسایی فضای حالت می‌توانند منفعت‌های زیادی برای جامعه مهندسی کنترل ایجاد کند. تمرکز اصلی این پژوهش بر روی مدل‌های متغیر پنهان است که امکان شناسایی فضای حالت برای سیستم‌های غیرخطی را فراهم می‌کند.

ظرفیت‌های استفاده از یادگیری عمیق در حوزه شناسایی سیستم‌ها مسئله جدیدی نیست و ارتباط بین دو حوزه شناسایی سیستم و یادگیری عمیق در مقاله مروری [۲۳] و مقاله [۲۴] آورده شده است. در [۲۴] و [۲۵] از شبکه‌های عصبی عمیق و خاصیت یادگیری ویژگی سلسله‌مراتبی آن‌ها استفاده شده است و مدل‌های عمیق NARX[3] و NFIR[4] توسعه داده شده‌اند. در عمده روش‌های شناسایی سیستم، فرض بر این است که متغیرهای خروجی از یک توزیع نرمال پیروی می‌کنند حال‌آنکه ممکن است در برخی کاربردها این فرض نادرست باشد. در [۲۶] با استفاده از شبکه‌های عصبی عمیق و مدل‌های مبتنی بر انرژی[5] [۲۷]، مسئله شناسایی سیستم را برای یک حالت کلی که در آن متغیرهای خروجی می‌توانند از هر توزیع احتمالی پیروی کنند، بیان شده است. چون ترکیب شبکه‌های عصبی عمیق و مدل‌های مبتنی بر انرژی می‌توانند هر نوع توزیع دلخواهی را تقریب بزنند، این رویکرد برای هر توزیعی از متغیرهای خروجی عملکرد خوبی خواهد داشت [۱۳]. در [۲۸] از ساختار شبکه‌های عصبی پیچشی زمانی[6] برای شناسایی سیستم‌ها استفاده شده است. شبکه عصبی پیچشی زمانی همانند شبکه کانولوشنالی معمولی عمل می‌کند با این تفاوت که به‌منظور حفظ گام‌های زمانی ورودی و خروجی هر لایه از آن باید دارای بعد برابر باشند و عملگر کانولوشن باید علی[7] باشد [۲۹]. استفاده از شبکه‌های کانولوشنالی زمانی می‌تواند در دسته روش‌های شناسایی بلوک – محور باشد چرا که به دلیل اعمال عملگر کانولوشن علی، می‌توان متصور شد که دنباله دادهٔ ورودی ابتدا از یک فیلتر خطی عبور می‌کند و سپس یک عملگر غیرخطی، تابع سیگموئید[8] و یا یک‌سوساز خطی[9]، بر روی آن اعمال می‌شود. چون برای هر شبکه چندین فیلتر در نظر گرفته می‌شود، می‌توان گفت فرایند فیلترکردن خطی و اعمال تابع غیرخطی چندین بار صورت می‌پذیرد و خروجی‌ها با هم تجمیع می‌شوند تا خروجی مدل را تخمین بزند. این فرایند مشابه مدل موازی وینر[10] است [۹]. در [۳۰] و [۳۱] از مدل‌های عمیق مبتنی بر کرنل[11] برای شناسایی سیستم‌های غیرخطی استفاده شده است. روش‌های مبتنی بر کرنل خروجی را بر اساس شباهتی که بین نمونه‌های متفاوت در فضای ورودی وجود دارند، خروجی‌های متناظر با هر ورودی با شباهت به‌دست‌آمده ترکیب خطی می‌کنند. در مدل‌های عمیق مبتنی بر کرنل، تابع شباهت سنج با استفاده از شبکه‌های عصبی عمیق به وجود آمده است. در [۳۲] از رویکردی مشابه با خودرمزنگارها استفاده شده است و فضای حالت سیستم‌های خطی تغییر ناپذیر با زمان شناسایی شده است. در [۲۴] و [۳۳] و [۳۴] و [۳۵] با استفاده از ترکیب شبکه‌های عصبی عمیق و مدل‌های فضای حالت، مدلی موسوم به فضای حالت عمیق برای سیستم‌های غیرخطی شناسایی شده است. در [۳۶] با استفاده از یک شبکه عصبی بازگشتی پارامترهای یک مدل خطی فضای حالت تغییر با زمان[12] را تخمین می‌زند. بر اساس این رویکرد مدل انتقال حالت و مدل مشاهدات هر دو از یک مدل خطی تغییرپذیر با زمان تبعیت می‌کنند اما چون نیاز است داده‌های زیادی از یک رویداد زمانی در دست باشد تا پارامترهای متغیر با زمان را تقریب بزند، از یک شبکه عصبی بازگشتی برای تقریب پارامترها استفاده می‌کند. فلسفه استفاده از آن به این دلیل است که اگر پارامترها مدل با زمان تغییر کند اثر این تغییرات در

[1] Generalization
[2] Generative models
[3] Deep nonlinear ARX
[4] Deep nonlinear FIR
[5] Energy-based models
[6] Temporal convolutional networks (TCN)
[7] Causal
[8] Sigmoid function
[9] Rectified linear units
[10] Parallel wiener model
[11] Deep kernel-based models
[12] Linear time-variant system

خروجی منعکس خواهد شد. بنابراین یک شبکه عصبی بازگشتی در نظر گرفته می‌شود که در هر لحظه از زمان خروجی سیستم را به عنوان ورودی می‌پذیرد و پارامترهای مدل را برمی‌گرداند. در [۳۷] و [۳۸] متغیرهای حالت و پارامترهای یک مدل فضای حالت غیرخطی با استفاده از روش‌های تغییراتی[1] تخمین زده می‌شوند.

عملکرد مدل‌های یادگیری عمیق بر اساس گزارش‌های ارائه شده غیرقابل انکار است. اما استفاده از آنها در حوزه شناسایی سیستم‌ها نیز دارای معایبی هست. شبکه‌های عصبی عمیق به دلیل ماهیت فراپارامتری خود، عملکرد مناسبی دارد که این موضوع استفاده و استقرار آنها را برای کاربردهای بلادرنگ محدود می‌کند [۳۹]. این محدودیت نه تنها در حوزه شناسایی سیستم‌ها بلکه در سایر حوزه‌ها نیز وجود دارد. بسته به محیط استقرار[2] این مدل‌ها، روش‌های برای کاهش تعداد پارامترهای مدل و فشرده‌سازی[3] مدل‌های آموزش دیده شده وجود دارد که امکان استفاده از این مدل‌ها را در سیستم‌هایی با منابع سخت‌افزاری محدود، مانند ظرفیت حافظه و توان پردازشی پردازنده در سیستم‌های یکپارچه[4]، ممکن می‌سازد. این روش‌ها در حالت کلی به سه دسته‌ی کوانیتزه کردن[5]، هرس کردن[6] و تقطیر کردن دانش[7] طبقه‌بندی می‌شوند [۴۰]، [۴۱]، [۴۲] و [۴۳]. مدل‌های یادگیری ماشین دارای دو جز ساختار و پارامتر هستند که پارامترهای مدل پس از آموزش به صورت اعشاری و در ۳۲ بیت[8] ذخیره می‌شوند. کوانتیزه کردن دقت این پارامترها می‌توانند بعضاً تا اعداد صحیح و در ۸ بیت[9] کاهش دهند که این کاهش دقت منجر به از دست رفتن عملکرد می‌شود. برای جلوگیری از این مسئله، کوانتیزه کردن مدل را حین آموزش انجام می‌دهند تا عملکرد مدل افت شدیدی نداشته باشد. روش‌های کوانتیزه کردن مدل‌ها در [۴۱]، [۴۴]، [۴۵]، [۴۶] و [۴۷] به تفصیل آورده شده است. دسته‌ای دیگر از روش‌های فشرده‌سازی مدل، ارتباطات اضافه و غیرمهم بین واحدهای محاسباتی گوناگون را می‌تواند حذف کند. این روش که موسوم به هرس کردن است با ایده از تکامل مغز انسان در مرحله نوزادی بیان می‌کند که تعداد ارتباطات عصبی بین نرون‌های مغز در یک نوزاد ابتدا پایین است که این تعداد در یک سن مشخص افزایش خواهد یافت و سپس دوباره کاهش می‌یابد. این کاهش ارتباطات عملکرد مغز را تحت‌تأثیر قرار نمی‌دهد [۴۸]. این روش‌ها در [۴۴] و [۴۷] به‌تفصیل ارائه شده است. تقطیر دانش که برای اولین بار در [۴۹] ارائه شده است بیان می‌کند که رفتار یک مدل عمیق با تعداد پارامترهای زیاد، توسط مدل‌هایی با تعداد پارامترهای بسیار کم‌تر قابلیت تقریب زده شدن دارد. روش‌های متعددی برای توسعه این رویکرد وجود دارد که در [۵۰] خلاصه شده‌اند. اگرچه پژوهش‌های انجام شده به منظور فشرده‌سازی مدل، قابلیت استفاده برای مدل‌های توسعه داده شده در حوزه شناسایی سیستم‌ها را نیز دارد، اما به‌کارگیری مدل‌های عمیق در حوزه شناسایی سیستم‌ها نیاز است تا با دقت بیش‌تری صورت گیرد و این چالش مهم پاسخ داده شود. زیرا حوزه‌هایی مانند حوزه مهندسی کنترل، در بسیاری از کاربردها مانند طراحی کنترلر، نیازمند شفافیت بیش‌تری در فرآیند مدل‌سازی هستند و زمان استنتاج در این کاربردها از اهمیت بالایی برخوردار است.

تمرکز اصلی این مقاله بر روی استفاده از مدل‌های متغیر پنهان به منظور شناسایی سیستم‌های غیرخطی است. در بخش ۲ ادبیات مربوط به حوزه یادگیری عمیق و ابزارهای مورد استفاده برای مدل‌سازی سیستم‌های دینامیکی ارائه خواهد شد و سپس در بخش ۳ مدل‌های متغیر پنهانی که در حوزه شناسایی سیستم‌ها ورود کرده‌اند مورد بررسی قرار می‌گیرد. در بخش ۴، نسخه تغییر یافته‌ای از مدل‌های متغیر پنهان که دارای ساختار هستند است ارائه می‌شود، و نهایتا موثر بودن مدل معرفی شده از طریق آزمایش بر روی سه نوع سیستم معیار[10] غیرخطی، به نمایش گذاشته خواهد شد.

۲- ابزارهای یادگیری عمیق برای مدل‌سازی سیستم‌های دینامیکی

یکی از مسائل مهم یادگیری ماشین، تخمین تابع چگالی احتمال توام[11] بر روی مجموعه‌ای از متغیرهای به هم مرتبط[12] است. مسئله استنتاج[13] و یادگیری توزیع توام بر روی این متغیرها بدلیل ناتوانی در کشف همبستگی بین متغیرها به صورت مستقیم، مسئله بسیار دشواری است که معمولا برای حل آن از مدل‌های متغیر پنهان[14] استفاده می‌شود که در آن فرض می‌شود، یک مکانیزم پنهان وجود دارد که می‌تواند تغییرات موجود در متغیرها را مدل کند. این مکانیزم پنهان به صورت لایه‌ای از متغیرهای تصادفی مدل می‌شود که دارای توزیع از پیش مشخص $p(z)$ است که هر نمونه از آن توسط مدل $p(x|z)$ توانایی تولید هر نمونه x را خواهد داشت. فرض می‌کنیم متغیر پنهان z یک متغیر تصادفی پیوسته[15] است اما بدون از دست دادن عمومیت می‌توان مباحث مطرح شده را به متغیرهای گسسته نیز تعمیم داد [۵۱].

[1] Variational methods
[2] Deployment environment
[3] Model compression
[4] Embedded systems
[5] Quantization
[6] Pruning
[7] Knowledge distillation
[8] Floating-point 32
[9] Integer 8
[10] Benchmark
[11] Joint probability distribution
[12] Highly correlated high dimensional data
[13] Inference problem
[14] Latent variable models
[15] Continues random variable

احتمال توام بر روی این متغیرها را می‌توان به صورت زیر تعریف کرد:

$$p(x,z) = p(z)p(x|z) \quad (۱)$$

تعریف لایه پنهان به منظور کشف روابط همبستگی بین متغیرهای ورودی مسئله، این امکان را به ما می‌دهد که به جای رویارویی با توزیع پیچیده و نامشخص $p(x)$ با یک توزیع احتمال مشخص و ساده، از نظر نمونه‌گیری[1] و ارزیابی، روبه‌رو شویم [۱۹]. تمام اجزای این توزیع احتمال قابل تعیین و مشخص هستند. توزیع $p(x)$ بر اساس این مدل به سادگی از طریق حاشیه‌سازی[2] بر روی متغیر z قابل تعریف است.

$$p(x) = \int p(x|z)p(z)dz \quad (۲)$$

مسئله اصلی پیش‌روی این رویکرد این است انتگرال فوق قابل محاسبه[3] نیست و برای محاسبه پارامترهای مدل باید از روش‌های تقریب[4] استفاده کرد. در یک دسته‌بندی کلی روش‌های تقریب به دو گروه تقریب تصادفی[5] و تقریب معین[6] تقسیم‌بندی می‌شوند که در [۱۳] و [۱۹] به تفصیل آورده شده است. بدلیل بالا بودن بار محاسبات روش‌های تقریب تصادفی، در این مقاله از روش‌های تقریب معین تغییراتی استفاده شده است. لگاریتم تابع شباهت رابطه (۲) به صورت زیر است:

$$\log p_\theta(x) = \log \int p_\theta(z)p_\theta(x|z)$$

که در آن θ بردار مربوط به تمام پارامترهای متعلق به توزیع توام است. رابطه فوق را می‌توان به صورت زیر نوشت:

$$\log p_\theta(x) = \log \int \frac{p_\theta(x,z)}{q_\phi(z|x)} q_\phi(z|x)dz$$

که در آن $q_\phi(z|x)$ یک توزیع شرطی معتبر است که در هیچ‌جای فضا و برای هیچ مقداری از z صفر نیست. این توزیع، توزیع تغییراتی[7] نامیده می‌شود. با استفاده از ناتساوی Jensen[8]، یک کران پایین برای تابع بیشینه شباهت تعریف می‌شود [۱۲]:

$$\log p_\theta(x) \geq \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz = E_{q_\phi}\left(\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right)$$

رابطه فوق مشهور به کران پایین مشاهدات[9] است که به صورت زیر قابل بسط دادن است:

$$E_{q_\phi}\left(\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right) = E_{q_\phi}(\log p_\theta(x|z)) - D_{kl}\left(q_\phi(z|x)\middle|p_\theta(z)\right) = F(\theta,\phi) \quad (۳)$$

که در آن D_{kl} معیار KL[10] است که برای محاسبه تفاوت بین دو توزیع احتمال به کار برده می‌شود. مسئله آموزش پارامترهای مسئله از طریق روش بیشینه‌سازی تابع شباهت که یک مسئله دشوار بود، تبدیل به یک مسئله بهینه‌سازی معادل با ترم‌های ساده‌تر شد. دلیل این سادگی حضور توزیع تغییراتی است که این امکان را به طراح می‌دهد تا به‌جای کارکرد با توزیع احتمال موخر[11] با یک توزیع احتمال ساده‌تر کار کند. حداکثر سازی عبارت موفق به منظور تعیین پارامترها معادل حداقل کردن D_{kl} است. لازم بذکر است که عبارت فوق زمانی حداقل خواهد شد که $q_\phi(z|x)$ برابر با $p_\theta(z|x)$ باشد.

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} F(\theta,\phi) \quad (۴)$$

چون دو مجموعه پارامترهای تحت بهینه‌سازی به یکدیگر وابسته هستند، برای بهینه‌سازی از روش بیشینه‌سازی امید تغییراتی[12] باید استفاده کنیم. بنابراین داریم:

$$\text{E-step:} \quad \phi_{k+1} = \arg\max_{\phi} F(\theta_k,\phi)$$
$$\text{M-step:} \quad \theta_{k+1} = \arg\max_{\theta} F(\theta,\phi_{k+1}) \quad (۵)$$

مسیر طی شده تا به اینجای کار یک مسیر استاندارد برای حل هر مسئله از طریق مدل متغیرهای پنهان است. تفاوت مسائل حل شده توسط این ابزار، در نحوه‌ی مدل کردن توزیع‌های شرطی موجود در روابط است. به‌عنوان مثال ماشین بولتزمن محدود شده[13] از مدل‌های مبتنی بر انرژی[14] برای مدل کردن $p_\theta(x|z)$ و $p_\theta(z|x)$ استفاده می‌کند که بدلیل فرضیات اعمال شده بر روی ساختار مدل، هر دو توزیع احتمال شرطی قابلیت ارزیابی و نمونه‌گیری را دارا هستند. دسته‌ی دیگری از رویکردهای مدل‌سازی از ظرفیت‌های شبکه‌های عصبی[15] در تقریب توابع استفاده می‌کند و توزیع‌های شرطی مورد نظر را تقریب می‌زنند. در این رویکرد $p_\theta(x|z)$ دارای یک فرم مشخص است که پارامترهای آن توسط یک شبکه عصبی با معماری مناسب تعیین می‌شود. چون این شبکه وظیفه بازتولید نمونه‌های x را دارد به عنوان شبکه کدگشا[16] گفته می‌شود. در این رویکرد چون

[1] Sampling
[2] Marginalization
[3] Intractable
[4] Approximate method
[5] Stochastic approximate
[6] Deterministic approximate
[7] Variational distribution
[8] Jensen's inequality
[9] Evidence lower bound (ELBO)
[10] Kullback-Leibler divergence
[11] Posterior probability distribution
[12] Variational EM
[13] Restricted Boltzmann machine (RBM)
[14] Energy-based models (EBM)
[15] Neural networks
[16] Decoder network

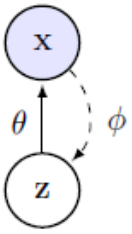
توانایی مدل کردن توزیع احتمال موخر به صورت مستقیم وجود ندارد از یک شبکه بازشناسایی یا کدکننده [1]استفاده می‌شود که وظیفه مدل کردن $q_\phi(z|x)$ را برعهده دارد.

۱-۲- خودرمزنگار تغییراتی

خودرمزنگار تغییراتی یک مدل متغیر پنهان است که برای مدل کردن توزیع‌های احتمال شرطی بیان شده در بخش قبل از شبکه‌های عصبی استفاده می‌کند. قبل از بررسی نوع مدل کردن خود رمزنگار تغییراتی لازم بذکر است که توزیع مولد برای یک مدل متغیر پنهان به صورت $p(z)p(x|z)$ است که به منظور مشخص شدن کامل آن نیازمند هستیم تا هر کدام از این ترم‌ها دقیقا تعریف شود. رابطه توزیع توام و ارتباط آن با توزیع حاشیه‌ای متغیرهای مسئله هیچ محدودیتی از جانب توزیع $p(z)$ را ندارد. به بیان دیگر هر توزیع دلخواهی از $p(z)$ می‌تواند در این مسئله جای داشته باشد اما بهترین نوع توزیع، توزیعی است که به ما اطمینان می‌دهد که مدل کدگشای $p(x|z)$ می‌تواند بر اساس آن نمونه‌های مورد نظر ما را تولید کند. علاوه بر این مسئله، تعریف این ترم‌ها باید به نحوی باشد که فرآیند آموزش را تسهیل کند.

برای خودرمزنگار تغییراتی توزیع لایه‌های مخفی به صورت یک توزیع نرمال مستقل از هم در نظر گرفته شده است. چرایی این مسئله ریشه در این حقیقت دارد که هر توزیع احتمال پیچیده‌ای را می‌توان بر اساس اعمال کردن یک نگاشت قطعی بر روی مجموعه‌ای از نمونه‌های توزیع استاندارد نرمال بدست آورد. به بیان بهتر اگر توانایی تولید نمونه از یک توزیع نرمال استاندارد وجود داشته باشد، هر توزیع دلخواهی را می‌توان بر اساس نگاشت این نمونه‌ها بدست آورد [۵۲]. بر اساس توضیحات ارائه شده انتظار می‌رود، فرآیند تولید نمونه توسط خودرمزنگار تغییراتی از دقت خوبی برخوردار باشد زیرا توزیع لایه‌های کد یک توزیع نرمال است و نگاشت مطرح شده $p(x|z)$ توسط یک شبکه عصبی مدل می‌شود که خاصیت تقریب عمومی[2] را داراست.

تنها مسئله باقی مانده برای آموزش خودرمزنگار تغییراتی وجود یک مکانیزم تولید نمونه برای متغیرهای z است که احتمال $p(x|z)$ تحت نمونه‌های تولید شده مقداری بزرگتر از صفر داشته باشند. به بیان بهتر بر اساس داده‌های موجود مسئله باید فضای مجازی از یک توزیع نرمال استاندارد را انتخاب کرد که توزیع $p(x|z)$ در آن فضا دارای احتمال بزرگتری است. برای این منظور یک شبکه بازشناسایی یا کدکننده در نظر گرفته شده است که بر اساس نمونه‌های ورودی x توزیعی از z را محاسبه می‌کند که بیشترین احتمال تولید x را داشته باشد. شکل ۱ مدل گرافی احتمالاتی[3] خود رمزنگار تغییراتی را نشان می‌دهد.



شکل ۱ – مدل مولد خودرمزنگار تغییراتی [۱۷] – خطوط پررنگ مدل مولد خطوط منقطع مدل استنتاج یا شبکه بازشناسایی را نشان می‌دهد.

برای خودرمزنگار تغییراتی توزیع کدکننده و کدگشا هر دو به صورت نرمال در نظر گرفته شده است که همین مسئله یکی از چالش‌های اساسی و محدود کننده استفاده از خودرمزنگار تغییراتی است. زیرا بر اساس این فرض توزیع‌هایی که دارای مدهای بیشتری باشند قابلیت شناسایی ندارد و مهمتر اینکه کدکننده نمی‌تواند به صورت کامل توزیع احتمال موخر را تقریب بزند. با این حال خودرمزنگار تغییراتی در طیف وسیعی از کاربردها با دقت خوبی به کار گرفته می‌شود. روابط مربوط به آموزش و تولید نمونه در این شبکه به صورت رابه (۶) خواهد بود:

$$\begin{aligned} &q_\phi(z|x) \sim N(\mu_{en}, \sigma_{en}^2) \\ &\rightarrow \begin{cases} \mu_{en} = NN_{en}^{\mu}(x) \\ \log \sigma_{en}^2 = NN_{en}^{\sigma}(x) \end{cases} \qquad (۶) \\ &p_\theta(x|z) \sim N(\mu_{de}, \sigma_{de}^2) \\ &\rightarrow \begin{cases} \mu_{de} = NN_{de}^{\mu}(z) \\ \log \sigma_{de}^2 = NN_{de}^{\sigma}(z) \end{cases} \end{aligned}$$

که در آن هر نمونه z به صورت $z = \mu_{en} + \sigma_{en}.\epsilon$ قابل محاسبه است که ϵ یک توزیع نرمال استاندارد است. این رویکرد حقه دوباره پارامتریزه کردن[4] نام دارد که چون نمونه‌گیری یک عمل گسسته است منجر به ایجاد مشکلات آموزشی با استفاده از گرادیان نزولی خواهد شد که به این ترتیب دیگر مشکلی وجود نخواهد داشت. پارامترهای شبکه نیز با استفاده از حداقل کردن کران پایین مشاهدات بدست می‌آید که جزئیات مربوط به آن در [۱۶] آمده است. نکته مهم درباره آموزش مدل خود رمزنگار تغییراتی این است که هر دو گام مربوط به بیشینه‌سازی امید تغییراتی به صورت همزمان انجام می‌شود و نه به صورت مجزا. این مسئله موجب می‌شود تا تخمین پارامترهای کدگشا به صورت زیربهینه پیدا شود. زیرا آنچه که در بیشینه‌سازی تابع شباهت مشاهدات اثر دارد پارامترهای مربوط به کدگشاست و نه کدکننده. چون این دو مجموعه از پارامترها به صورت همزمان به روزرسانی می‌شوند، پارامترهای کدکننده نیز در بیشینه کردن تابع شباهت اثر خواهد داشت که نتیجه آن فاصله داشتن کران پایین مشاهدات با تابع شباهت مشاهدات است. برای رفع این مشکل، روشی به

[1] Recognition network – Encoder network
[2] General approximation property
[3] Probabilistic graphical model
[4] Reparametrization trick

جهت محدود کردن کران پایین[1] مشاهدات مطرح شد که با وزن‌دهی کردن به نمونه‌های تولید شده توسط کدکننده سعی در نزدیک کردن کران پایین مشاهدات و تابع شباهت نمونه‌ها دارد. مدل بدست آمده از این روش خودرمزنگار وزن‌دهی شده اهمیت[2] نام دارد که جزئیات مربوط به آن در [۵۳] آمده است. یکی دیگر از ایرادهای خودرمزنگار تغییراتی، نوع مدل کردن کدکننده و کدگشا است که هر دو از یک توزیع نرمال پیروی می‌کنند. دلیل این نوع کردن تنها مسئله تسهیل آموزش را در بردارد. دو فاکتور اصلی و اساسی برای آموزش ساده‌تر شدن فرآیند اعمال گرادیان نزولی برای به‌روزرسانی وزن‌ها و نمونه‌گیری از توزیع کدگشا است. چون محاسبه D_{kl} بین دو توزیع نرمال دارای یک فرم مشخص و ساده است، استفاده از این فرض فرآیند آموزش را ساده می‌کند، بهمین دلیل چون طبق خاصیت نگاشت توزیع نرمال استاندارد امکان تغییر توزیع متغیرهای مخفی وجود ندارد و همچنین تغییر توزیع کدکننده منجر به دشواری کدگشا خواهد شد، چارچوب ارائه شده برای سایر توزیع‌های احتمال وجود ندارد. برای رفع این مشکل مدل خودرمزنگار متخاصم[3] ارائه شد که خودرمزنگار تغییراتی را به عنوان یک حالت خاص در خود جای داده است [۵۴].

مباحث مطرح شده برای نوع داده‌های بدون حافظه[4]، بدون هیچ تغییری قابل اعمال است و اصل ابزارها نیز برای این حالات ارائه شده است. به منظور توسعه این ابزار برای هدف شناسایی سیستم‌های غیرخطی، ، تعمیم ابزارهای موجود برای پردازش داده‌های دینامیکی امری ضروری خواهد بود.

۲-۲- مدل‌های متغیر پنهان ترتیبی[5]

عمده اطلاعاتی که در محیط پیرامون ما وجود دارد پیوسته در حال تغییر است. این نوع اطلاعات به لحاظ ریاضی به صورت یک دنباله[6] تعریف می‌شود. دنباله یک بردار معمولی با یک بعد فیزیکی[7] است که این بعد فیزیکی در غالب سیستم‌ها به صورت زمان مدل می‌شود که به آن سری زمانی[8] نیز گفته می‌شود [۱۸]. سری‌های زمانی در مقایسه با سیگنال‌های عادی دارای دو نوع اطلاعات هستند که یک نوع آن مربوط به مقدار عددی اطلاعات و دیگری اطلاعات زمانی[9] سیگنال است. بنابراین روش‌های پردازش این نوع داده‌ها در مقایسه با ابزارهای پردازشی داده‌های ایستا پیچیده‌تر خواهد بود.

از دیدگاه یادگیری ماشین، آنالیز سری زمانی[10] $y_{1:T}$ نیازمند تعیین توزیع توام $p(y_{1:T})$ بر روی تمام گام‌های زمانی[11] است. در حالت کلی امکان مدل کردن این توزیع وجود ندارد زیرا همبستگی بین متغیرها بالا است و معمولا برای مدل کردن آن باید از روابط استقلال شرطی[12] و علیت در زمان[13] استفاده کرد. زیرا اگر فرض کنیم هر متغیر x_t در هر لحظه از زمان یک متغیر دودویی[14] باشد برای آنکه توزیع احتمال را بر روی کل دنباله محاسبه کنیم به تعداد $2^T - 1$ پارامتر آزاد[15] نیاز داریم تا توزیع احتمال به طور کامل مشخص شود که عملیاتی نیست. با استفاده از قانون احتمال شرطی بیز توزیع احتمال روی دنباله را می‌توان به صورت رابطه (۷) نوشت [۱۸]:

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \quad (۷)$$

این نوع تجزیه کردن توزیع توام بر روی گام‌های زمانی منطبق بر علیت در زمان است زیرا هر فاکتور به تمام عوامل گذشته بستگی دارد. در هر صورت همچنان مدل کردن این دنباله کار دشواری است زیرا مجموعه متغیرهایی که در مجموعه‌ی شرط[16] قرار گرفته‌اند با گذر زمان در حال تغییر است. به عنوان مثال اگر فرض کنیم می‌خواهیم این توزیع احتمال را با استفاده از یک شبکه عصبی مدل کنیم، نیازمند یک شبکه خواهیم بود که در هر لحظه از زمان ورودی‌های آن در حال تغییر است که این مسئله امری نشدنی است. بنابراین نیازمند ساده‌سازی و اعمال قیود بیشتری بر روی ساختار زمانی[17] و ارتباط بین گام‌های زمانی متغیرها هستیم. این مسئله در واقع تفاوت بین ابزارهای متنوع که به منظور تحلیل سری‌های زمانی به کار می‌روند را نشان می‌دهد.

یکی از ساده‌سازی‌های صورت گرفته به منظور تحلیل محدود کردن وابستگی گام‌های زمانی[18] به یکدیگر است. مثلا به جای آنکه فرض کنیم هر گام زمانی به تمام گام‌های زمانی گذشته خود وابسته است، می‌توان فرض کرد که هر گام زمانی به یک طول مشخصی از گام‌های گذشته وابسته است. در این حالت توزیع توام به صورت رابطه (۸) خواهد بود:

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{t-m:t-1}) \quad (۸)$$

این مدل را مدل مارکوف مرتبه m[19] می‌نامند که چند نمونه از آن در شکل ۲ آورده شده است.

[1] ELBO lightening
[2] Importance weighted auto-encoders
[3] Adversarial auto-encoder
[4] Static data
[5] Sequential latent variable models
[6] Sequence
[7] Physical dimension
[8] Time series
[9] Temporal information
[10] Time series analysis
[11] Time steps
[12] Conditional independency
[13] Causal-in-time
[14] Binary variable
[15] Free parameters
[16] Conditioning set
[17] Temporal structure
[18] Temporal dependency
[19] mth-order Markov model

شکل ۲- شکل سمت چپ (a) – شبکه بیزین[1] متناظر با مدل مارکوف مرتبه اول که در آن هر گام زمانی تنها به یک گام گذشته وابسته است. شکل سمت راست (b) – شبکه بیزین متناظر با مدل مارکوف مرتبه دو که در آن هر گام زمانی به دو گام زمانی قبل از خود وابسته است. [۱۸]

اگرچه این روش تا حد زیادی ساده‌سازی را امکان‌پذیر می‌کند، اما بزرگترین مشکل آن تعیین طول حافظه m است. به منظور افزایش قابلیت پیاده‌سازی معمولا این مدل را برای کاربردهای که طول حافظه کوتاه باشد به کار می‌برند زیرا برای کابردهایی که طول حافظه بزرگ نیاز دارند استفاده از این ابزار نیازمند تحلیل همبستگی[2] و یا اطلاعات متقابل[3] برای تعیین طول حافظه است. مدل‌های رگرسیون خودکار[4] که در برخی از کتب به آن‌ها به عنوان یک ابزار تحلیل سری زمانی اشاره می‌شود، از زیر مجموعه‌های مدل‌های مارکوف است [۱۸].

یک چارچوب کلی‌تر و جامع‌تر برای مدل کردن سری‌های زمانی وجود دارد که در واقع ابزارهای موجود همگی به این فرم قابل تبدیل هستند [۱۸]. این چارچوب مدل‌های فضای حالت[5] نام دارد. مدل‌های فضای حالت را می‌توان به نوعی انشقاق زمانی مدل‌های متغیر پنهان معرفی کرد. بر این اساس و طبق تعریف مدل‌های متغیر پنهان، می‌توان گفت مدل‌های فضای فرض می‌کند در هر گام زمانی یک مکانیزم پنهان وجود دارد که تغییرات موجود در متغیرها را مدل می‌کند اما این مکانیزم پنهان باید دارای یک وابستگی زمانی باشد تا بتوان اطلاعات موجود در متغیرها را نیز فرا بگیرد. در ادامه به بررسی این مدل‌ها می‌پردازیم.

۳-۲- مدل‌های فضای حالت

فرض می‌کنیم یک دنباله مشاهدات $x_{1:T}$ وجود دارد که به یک مجموعه دنباله ورودی $u_{1:T}$ وابسته است. در این حالت مسئله مدل کردن توزیع احتمال $p_\theta(x_{1:T}|u_{1:T})$ است. لازم بذکر است که در بخش قبلی فرض بر این بود که مسئله یک مسئله رگرسیون خودکار است و تنها اطلاعاتی که در دست است مربوط به خود سری زمانی $x_{1:T}$ است. فرم ذکر شده در بالا که سری زمانی مد نظر وابستگی به یک سری زمانی خارجی دارد فرم کلی‌تر مسئله مدل کردن سری زمانی است زیرا اگر ورودی مسئله را با گذشته متغیر x_t جایگزین کنیم، مدل رگرسیون خودکار حاصل می‌شود. بنابراین مباحث ذکر شده در ادامه، قابلیت تعمیم دادن به حالتی که ورودی محرک در مسئله وجود ندارد را هم داراست.

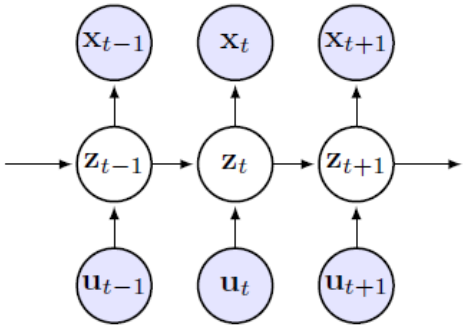
برای مدل کردن توزیع احتمال مورد نظر، مطابق آنچه که در مدل‌های متغیر پنهان بیان شد فرض می‌کنیم یک متغیر پنهان وجود دارد که وابستگی مستقیم به ورودی (یا گذشته خروجی در مدل‌های رگرسیون خودکار) دارد و همچنین به منظور حفظ اطلاعات زمانی به مقدار متغیر پنهان در گام قبل نیز وابسته است. در ادبیات مربوط به مدل‌های فضای حالت، به این متغیر پنهان، متغیر حالت[6] گفته می‌شود. شکل ۳، مدل گرافی احتمالاتی این متغیر را نشان می‌دهد. بنابراین برای مدل کردن $p_\theta(x_{1:T}|u_{1:T})$ خواهیم داشت [۱۸]:

$$p_\theta(x_{1:T}|u_{1:T}) = \int p_\theta(x_{1:T}, z_{1:T}|u_{1:T})dz_{1:T} \quad (۹)$$

که در آن فرض بر این است که متغیرهای حالت پیوسته هستند. در حالت گسسته انتگرال به جمع تبدیل می‌شود. توزیع احتمال توام $p_\theta(x_{1:T}, z_{1:T}|u_{1:T})$ به صورت زیر قابل محاسبه است:

$$p_\theta(x_{1:T}, z_{1:T}|u_{1:T}) = p_\theta(z_1)\prod_{t=1}^{T} p_\theta(x_t|z_t)\prod_{t=2}^{T} p_\theta(z_t|z_{t-1}, u_t) \quad (۱۰)$$

که در آن $p_\theta(x_t|z_t)$ مدل مشاهدات[7] و $p_\theta(z_t|z_{t-1}, u_t)$ مدل انتقال حالت[8] نامیده می‌شود. در برخی منابع مدل مشاهدات به ورودی نیز بستگی دارد ینی به صورت $p_\theta(x_t|u_t, z_t)$ است.



شکل ۳- مدل فضای حالت [۱۷]

[1] Bayesian network (Belief network)
[2] Correlation analysis
[3] Mutual information
[4] Autoregressive models

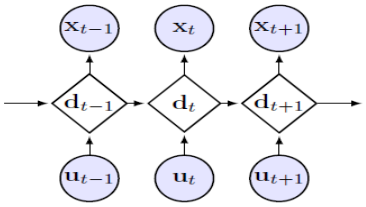
[5] State space models
[6] State variable
[7] Observation model (emission model)
[8] State transition model

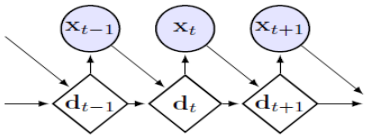
بسته به فرم‌های متفاوتی که برای مدل‌های انتقال حالت و مشاهدات در نظر گرفته می‌شود، مدل‌های فضای حالت متنوعی تولید خواهد شد که در ادامه به آن اشاره می‌شود. نکته مهم در این باره این است که هدف نهایی محاسبه توزیع شرطی دنباله خروجی به شرط ورودی است که این توزیع از طریق حاشیه‌سازی[1] بر روی متغیرهای حالت بدست می‌آید. بنابراین نیازمند محاسبه $p_\theta(z_{1:T}|x_{1:T},u_{1:T})$ خواهیم بود. در حالت کلی محاسبه این توزیع، و یا بیان بهتر استنتاج دقیق[2]، امکان‌پذیر نیست و تنها در شرایط خاص مسئله دارای جواب دقیق است [۱۷] و [۵۵]. به عنوان مثال اگر فرض کنیم مدل مشاهدات و مدل انتقال حالت در دسترس باشد و ارتباط بین متغیرها نیز خطی باشد، در شرایطی که هر دو توزیع احتمال انتقال حالت و مشاهدات گوسی باشد، مسئله استنتاج درباره متغیرهای حالت دارای یک فرم بسته خواهد بود که این فرم بسته به فیلتر کالمن[3] مشهور است [۱۷]. در شرایطی که مدل ارتباطی بین متغیرهای حالت و متغیر خروجی به صورت یک تابع غیرخطی مشتق پذیر باشد، فیلتر کالمن به صورت تقریبی قابل استفاده را دارد که ابزار مورد نظر تحت عنوان‌های فیلتر کالمن خنثی[4] و یا توسعه یافته[5] مورد استفاده قرار می‌گیرند. برای طیف وسیعی از ابزارها مسئله دارای جواب دقیق نیست و باید از روش‌های تقریبی[6] مانند روش‌های تغییراتی و یا نمونه‌گیری استفاده کرد. مهم‌تر اینکه فرض در دسترس بودن مدل فرض درستی نیست زیرا در طیف وسیعی از کابردها مدل ارتباطی بین متغیرها در دست نیست.

جزئیات مربوط به مدل‌های خطی و مدل‌های غیرخطی که در آن استنتاج دقیق امکان‌پذیر است در [۵۶]، [۱۷] و [۱۸] آورده شده است. ظرفیت استفاده از این مدل‌ها در این نکته نهفته است که مدل‌های انتقال حالت و مدل مشاهدات توانایی تقریب زدن توسط هر تقریب زنی که باعث بهبود دقت در مسئله شود را داراست. در این راستا دسته‌ای از مدل‌ها در حال ظهور هستند که از ظرفیت شبکه‌های عصبی عمیق برای مدل کردن حالت و مشاهدات استفاده می‌کند. به این دسته از مدل‌ها، مدل‌های فضای حالت عمیق[7] گفته می‌شود. پر واضح است که در این مدل‌ها امکان استنتاج دقیق وجود ندارد و باید از روش‌های تقریبی برای استنتاج و آموزش آن استفاده کرد. در ادامه شبکه‌های عصبی بازگشتی[8] به عنوان یک مدل مشهور فضای حالت مورد بررسی قرار می‌گیرد که از ظرفیت‌های مدل‌سازی آن در برخی از اقدامات این پژوهش استفاده شده است.

۴-۲- شبکه‌های عصبی بازگشتی

شبکه‌های عصبی بازگشتی معماری خاصی از شبکه‌های عصبی هستند که توانایی مدل کردن دنباله با طول متغیر[9] را توسط نگاشت‌های قطعی[10] دارد [۵۵] و [۱۷]. شکل ۴ شبکه عصبی بازگشتی را برای دو حالت بدون ورودی و با ورودی نشان می‌دهد.





شکل۴- شبکه عصبی بازگشتی – شکل سمت راست مربوط به حالتی است که ورودی محرک وجود دارد و شکل سمت راست مربوط به حالتی که ورودی محرک وجود ندارد. هر دو شماتیک یکسان هستند اگر در شماتیک سمت راست به جای ورودی محرک، خروجی لحظه قبل قرار بگیرد [۱۷].

متغیرهای حالت در این مدل توسط یک نگاشت غیرخطی به صورت زیر محاسبه می‌شوند [۱۳]:

$d_t = f_\theta(d_{t-1}, u_t; \theta)$	(۱۱)

انشقاق‌های متفاوتی از شبکه‌های عصبی بازگشتی وجود دارد که هر یک بسته به نوع کابرد و در تقویت توانایی شبکه‌های عصبی بازگشتی به منظور مدل کردن حافظه توسعه پیدا کرده‌اند [۵۷]، [۵۸] و [۵۹]. تفاوت این ابزارها تنها در نحوه محاسبه متغیرها حالت است و در ماهیت یکسان هستند به این معنا که تمام آن‌ها یک نگاشت قطعی را مدل می‌کنند.

معماری حافظه بلند کوتاه مدت[11] و واحد بازگشتی دروازه‌ای[12] از جمله این مدل‌ها هستند.

شبکه عصبی بازگشتی را می‌توان به صورت یک مدل فضای حالتی دید که توزیع احتمال انتقال حالت تابع دلتای دیراک است یعنی $p_\theta(d_t|d_{t-1},u_t) = \delta\big(d_t - f_\theta(d_{t-1},u_t)\big)$. مدل مشاهدات نیز در کابردهای معمول به صورت یک نگاشت قطعی در نظر گرفته می‌شود، اما در برخی کاربردهای تولید متن مدل مشاهدات می‌تواند به صورت احتمالاتی در نظر گرفته شود [۶۰].

[1] Marginalizing out
[2] Exact inference
[3] Kalman filter
[4] Unscented Kalman filter
[5] Extended Kalman filter
[6] Approximation method
[7] Deep state space models
[8] Recurrent neural network
[9] Variable length sequence
[10] Deterministic maps
[11] Long short-term memory (LSTM)
[12] Gated recurrent unit (GRU)

۵-۲- استنتاج در مدل‌های فضای حالت غیرخطی

امکان استنتاج و آموزش مدل‌های زمانی در شرایط خاصی اعم از خطی بودن مدل‌ها و یا گوسی بودن توزیع‌های شرطی به صورت دقیق وجود دارد حال آنکه برای شرایطی که این فرضیات صادق نباشد، امکان یافتن پاسخ به صورت دقیق نیست و باید از روش‌های تقریبی استفاده کرد. بنابراین گام نخست، مدل کردن توزیع‌های مورد اشاره است. مطابق آنچه که در بخش مربوط به متغیرهای پنهان ایستا عنوان شد، مدل کردن توزیع‌های شرطی (فرض بر این است مدلی در دست نیست) می‌تواند بر مبنای مدل‌های مبتنی بر انرژی باشد و یا از ظرفیت شبکه‌های عصبی استفاده شود. این نوع مدل‌ها نیز از این قاعده مستثنی نیستند. برای این منظور می‌توان توزیع انتقال حالت و مشاهدات را به صورت یک فرم مشخص در نظر گرفت و سپس از ظرفیت شبکه‌های عصبی برای تخمین پارامتر استفاده کرد [۶۰]. روابط ریاضی مربوطه به صورت زیر خواهد بود:

$$\begin{aligned} &p_\theta(z_t|z_{t-1},u_t)\sim N(\mu_{tr},\Sigma_{tr})\\ &\rightarrow\begin{cases}\mu_{tr}=NN_{tr}^{\mu}(z_{t-1},u_t;\theta)\\ \log\Sigma_{tr}=NN_{tr}^{\sigma}(z_{t-1},u_t;\theta)\end{cases}\\ &p_\theta(x_t|z_t)\sim N(\mu_{em},\Sigma_{em})\\ &\rightarrow\begin{cases}\mu_{em}=NN_{em}^{\mu}(z_t;\theta)\\ \log\Sigma_{em}=NN_{em}^{\sigma}(z_t;\theta)\end{cases}\end{aligned} \quad (۱۲)$$

به منظور آموزش پارامترهای شبکه، تابع بیشینه شباهت را به صورت زیر می‌نویسم و با استفاده از روابط نوشته شده برای مدل متغیرهای پنهان ایستا در بخش قبلی، کران پایین تابع شباهت را محاسبه می‌کنیم [۵۵]:

$$\begin{aligned}&\log p_\theta(x_{1:T}|u_{1:T})\\ &\geq\int q_\phi(z_{1:T}|x_{1:T},u_{1:T})\log\frac{p_\theta(x_{1:T},z_{1:T}|\,u_{1:T})}{q_\phi(z_{1:T}|x_{1:T},u_{1:T})}dz_{1:T}\\ &=E_{q_\phi}\left(\log\frac{p_\theta(x_{1:T},z_{1:T}|\,u_{1:T})}{q_\phi(z_{1:T}|x_{1:T},u_{1:T})}\right)\end{aligned}$$

گام بعدی تعیین ساختار توزیع تغییراتی است که هیچ محدودیتی برای آن وجود ندارد اما دقت روش‌های تقریبی به شدت به نوع انتخاب این توزیع بستگی دارد. تنها نکته‌ای که باید برای تعریف آن لحاظ کرد مسئله تسهیل فرآیند آموزش مدل است. بهترین انتخاب برای توزیع تغییراتی در روش‌های بهینه‌سازی تغییراتی استفاده از ساختار خود مسئله و تحمیل کردن ساختار مسئله به این توزیع است. به بیان بهتر برای آنکه بهترین انتخاب را برای توزیع تغییراتی داشته باشیم لازم است $p_\theta(z_{1:T}|x_{1:T},u_{1:T})$ را تعیین کنیم زیرا توزیع تغییراتی مورد نظر، تقریبی از این توزیع احتمال موخر را بدست می‌دهد (فرآیند کمینه‌سازی کران پایین مشاهدات در جایی حداقل است که توزیع تغییراتی برابر با توزیع احتمال موخر باشد). بنابراین خواهیم داشت:

$$\begin{aligned}&p_\theta(z_{1:T}|x_{1:T},u_{1:T},z_0)\\ &=p_\theta(z_{2:T}|z_1,x_{1:T},u_{1:T},z_0)p_\theta(z_1|x_{1:T},u_{1:T},z_0)\end{aligned}$$

که در آن $p_\theta(z_{2:T}|z_1,x_{1:T},u_{1:T})$ را بر اساس قاعده استقلال شرطی می‌توان به صورت $p_\theta(z_{2:T}|z_1,x_{2:T},u_{2:T})$ نوشت. با ادامه دادن این روند برای تمام گام‌های زمانی خواهیم داشت:

$$p_\theta(z_{1:T}|x_{1:T},u_{1:T},z_0)=\prod_{t=1}^{T}p_\theta(z_t|z_{t-1},x_{t:T},u_{t:T})$$

بنابراین ساختار توزیع تغییراتی نیز مطابق با همین الگو به صورت زیر قابل تجزیه خواهد بود:

$$\begin{aligned}&q_\phi(z_{1:T}|x_{1:T},u_{1:T},z_0)\\ &=\prod_{t=1}^{T}q_\phi(z_t|z_{t-1},x_{t:T},u_{t:T})\end{aligned} \quad (۱۳)$$

کران بالای مشاهدات نیز به صورت زیر خواهد بود:

$$\begin{aligned}&\log p_\theta(x_{1:T}|u_{1:T})=E_{q_\phi}\left(\log\frac{p_\theta(x_{1:T},z_{1:T}|\,u_{1:T})}{q_\phi(z_{1:T}|x_{1:T},u_{1:T})}\right)\\ &=\sum_{t=1}^{T}E_{q_\phi^*(z_{t-1})}\left(E_{q_\phi(z_t|z_{t-1},x_{t:T},u_{t:T})}(\log p_\theta(x_t|z_t)\right.\\ &\left.-D_{kl}\left(q_\phi(z_t|z_{t-1},x_{t:T},u_{t:T})\middle|p_\theta(z_t|z_{t-1},u_t)\right)\right)\end{aligned}$$

که در آن $q_\phi^*(z_{t-1})$ توزیع حاشیه‌ای توزیع تغییراتی حول متغیر z_{t-1} است. واضحا رابطه بدست آمده مشابه با رابطه بدست آمده برای خودرمزنگار تغییراتی است با این تفاوت که توزیع پیشین[1] با زمان در حال تغییر کردن است.

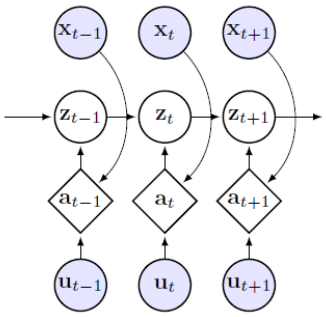
پس از مشخص شدن تابع هزینه، نوبت به مدل کردن شبکه بازشناسایی یا کدکننده می‌رسد. همانطور که مشخص است، مدل کردن شبکه کدکننده به سادگی امکان پذیر نیست زیرا به مقادیر آینده ورودی و خروجی بستگی دارد. برای رفع این مشکل دو روش در حالت کلی وجود دارد. در روش اول می‌توان فرض کرد که کل دنباله در اختیار نیست و داده‌ها با گذر زمان در اختیار مدل قرار می‌گیرند. این رویکرد فیلترینگ نام دارد و چون در آن از شبکه عصبی استفاده می‌شود به آن فیلتر کالمن عمیق[2] گفته می‌شود [۱۷]، [۵۵] و [۶۱]. بر اساس این رویکرد یک متغیر کمکی a_t تعریف می‌شود که وابستگی مستقیم به ورودی و خروجی در زمان یکسان دارد و متغیر حالت تابعی از این متغیر کمکی و متغیر پیشین حالت خواهد بود [۶۱]. به بیان ریاضی خواهیم داشت:

$$\begin{aligned}&q_\phi(z_{1:T}|x_{1:T},u_{1:T},z_0)\\ &=\prod_{t=1}^{T}q_\phi(z_t|z_{t-1},a_t),\\ &a_t=f_\lambda(x_t,u_t)\end{aligned} \quad (۱۴)$$

مدل گرافی متناظر با این مدل به صورت شکل ۵ خواهد بود.

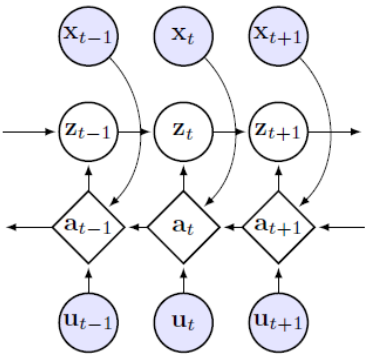
[1] Prior distribution

[2] Deep Kalman Filter (DKF)



شکل۵ کالمن فیلتر عمیق به منظور رفع مشکل وابستگی متغیرهای حالت به مقادیر آینده ورودی و خروجی در شبکه بازشناسایی توسعه داده شده [۱۷]

رویکرد غالب دیگری نیز وجود دارد که از یک شبکه عصبی بازگشتی معکوس[1] استفاده می‌کند. بر اساس این رویکرد فرض بر این است که دنباله ورودی و خروجی به طور کامل در اختیار است و شبکه با مقدار نهایی این دنباله‌ها کار خود را آغاز می‌کند و خلاصه‌ای از مقادیر آینده را در اختیار شبکه بازشناسایی قرار می‌دهد. اگرچه دقت این رویکرد از رویکرد فیلتر کردن بالاتر است، بار محاسباتی آن نیز بالاتر است زیرا امکان پردازش موازی وجود ندارد و تا به طور کامل کل دنباله در جهت عکس پردازش نشود امکان تولید نمونه‌های بعدی برای متغیرهای حالت وجود ندارد. این رویکرد هموارساز کالمن عمیق[2] نامیده می‌شود که در شکل۶ به نمایش در آمده است.



شکل ۶- هموارساز کالمن عمیق به منظور خلاصه‌سازی گام‌های زمانی آینده برای شبکه بازشناسایی [۱۷]

روابط مربوط به این شبکه نیز به صورت زیر خواهد بود [۶۱]:

$$q_\phi(z_{1:T}|x_{1:T},u_{1:T},z_0) = \prod_{t=1}^{T} q_\phi(z_t|z_{t-1},a_t)\,, \quad a_t = f_\lambda(a_{t+1},[x_t,u_t]) \tag{۱۵}$$

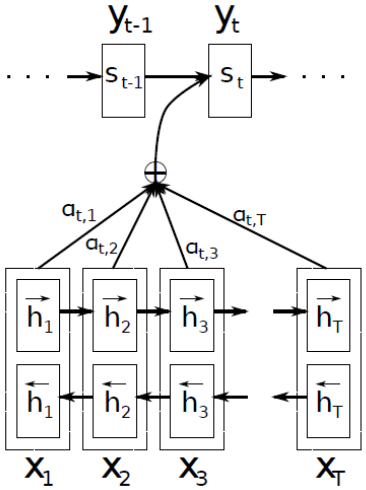
نتیجتا شبکه بازشناسایی به صورت زیر مدل خواهد شد:

$$q_\phi(z_t|z_{t-1},a_t)\sim N(\mu_{en},\Sigma_{en}) \rightarrow \begin{cases} \mu_{en} = NN^{\mu}_{en}(z_{t-1},a_t;\phi) \\ \log\Sigma_{en} = NN^{\sigma}_{tr}(z_{t-1},a_t;\phi) \end{cases} \tag{۱۶}$$

همانطور که مشاهده شد، مدل فضای حالت غیرخطی دارای یک فرآیند آموزش قابل اجرا است اما دارای ایراداتی نیز هست. اولین ایراد مربوط به بار پردازشی بالا این مدل است که در آن یک شبکه عصبی بازگشتی مورد استفاده قرار می‌گیرد و در شرایطی که مدل نیازمند عمق بالای دینامیک[3] باشد، مدت زمان زیادی طول می‌کشد تا کل دنباله ورودی و خروجی پردازش شود. این بار پردازشی در شرایطی که در شبکه بازشناسایی از یک شبکه عصبی معکوس استفاده کنیم دو چندان خواهد شد. ایراد مهم بعدی ناکارآمد بودن این مدل در شرایطی است که ابعاد مسئله بالا باشد. برای حل مشکل اول و دوم که عدم توانایی پردازش دنباله‌های با عمق بالا و نیازمند تحلیل دستی برای تعیین عمق است می‌توان از ترنسفرمرها[4] و مکانیزم توجه[5] استفاده کرد که قادر است در یک فرآیند انتها به انتها و با استفاده از مکانیزم توجه متغیرهای حالت را به صورت یکجا محاسبه کند. بنابراین این امکان وجود دارد که در این مدل‌ها، به‌جای استفاده از شبکه‌های عصبی عمیق از ترنسفرمرها استفاده کرد.

۶-۲-مکانیزم توجه[6]

مدل‌هایی که تاکنون بررسی شد، برای فراگرفتن اطلاعات زمانی از شبکه‌های عصبی بازگشتی استفاده می‌کنند. بزرگترین ایراد این شبکه‌ها در ساختار کدکننده-کدگشا است که لایه میانی این مدل‌ها وظیفه خلاصه کردن اطلاعات مربوط به دنباله‌ها را به تنهایی برعهده دارد. در شرایطی که طول دنباله‌ها بزرگ باشد، این لایه توانایی خلاصه‌سازی به صورت صحیح را نخواهد داشت [۶۲].



شکل ۷ – ساختار کدکننده – کدگشا در حضور مکانیزم توجه [۶۲]

[1] Reversed RNN
[2] Deep Kalman Smoother (DKS)
[3] Large dynamic range
[4] Transformers
[5] Attention mechanism
[6] Attention mechanism

برای مدل کردن این لایه از یک شبکه عصبی بازگشتی استفاده می‌شود که گام آخر مربوط به متغیر حالت شبکه به عنوان بردار زمینه[1] در نظر گرفته می‌شود. به منظور رفع این مشکل در [۲۹] یک فرآیندی تحت عنوان مکانیزم-توجه معرفی شده است که بردار زمینه را بر اساس میانگین وزن‌دار متغیرهای حالت شبکه عصبی در تمام گام‌های زمانی محاسبه می‌کند. شکل ۷ ساختار مربوط به این مدل را نشان می‌دهد. بر اساس این ساختار، کدکننده ابتدا اطلاعات زمانی مربوط به دنباله ورودی را استخراج می‌کند و برای هر گام از اطلاعات زمانی استخراج شده توسط کدکننده یک مقدار اهمیت برای هر متغیر حالت شبکه عصبی بازگشتی کدکننده محاسبه می‌کند تا گام بعدی متغیر حالت را برای شبکه عصبی بازگشتی به کارگرفته شده برای کدگشا محاسبه کند [۶۲]. این مقدار اهمیت توسط یک شبکه عصبی مدل می‌شود که در ادبیات یادگیری ماشین به آن توجه جمع‌شونده[2] نیز گفته می‌شود. مقادیر اهمیت مطابق با رابطه (۱۷) ساخته می‌شوند.

$$e_{ij} = NN(s_{i-1}, h_j), \alpha_{ij} \quad (۱۷)$$
$$= \frac{\exp(e_{ij})}{\sum_j \exp(e_{ij})}$$

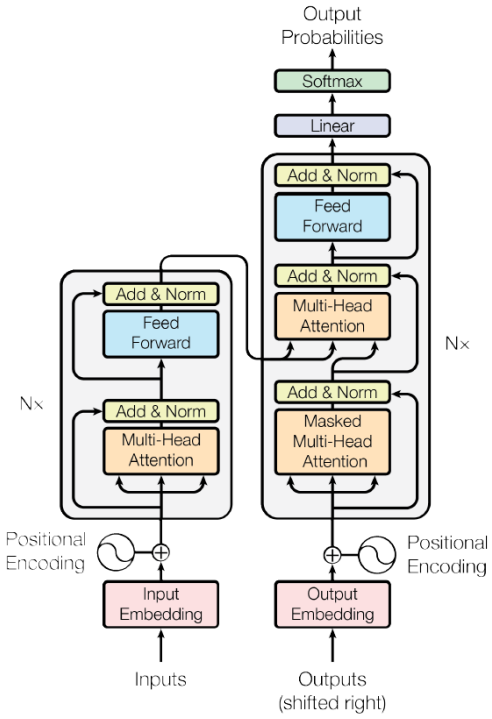
که در آن e_{ij} شباهت بین حالت iام کدگشا با حالت jام کدکننده را محاسبه می‌کند. بردار زمینه در این حالت به صورت رابطه (۱۸) مدل می‌شود.

$$c_i = \sum_j \alpha_{ij} h_j \quad (۱۸)$$

پس از محاسبه این بردار، حالت بعدی کدگشا با استفاده از یک شبکه عصبی بازگشتی و در شرایطی که بردار زمینه و مقدار واقعی خروجی در گام قبلی، به عنوان ورودی شبکه لحاظ می‌شوند. این مسئله مشکل عدم توانایی شبکه‌های عصبی بازگشتی برای خلاصه کردن دنباله ورودی در یک متغیر با بعد محدود را مرتفع می‌کنند. اما دارای یک چالش بزرگ است که مربوط به عدم توانایی شبکه‌های عصبی بازگشتی در مدل کردن دنباله به صورت موازی هستند. شبکه‌های عصبی بازگشتی، دنباله ورودی را به صورت ترتیبی پردازش می‌کنند. برای حل این مشکل [۶۳] یک مکانیزمی به نام خود-توجه[3] را معرفی می‌کند که زمینه حذف کردن لایه‌های بازگشتی در شبکه‌های عصبی را فراهم می‌کند.

۸-۲- مکانیزم خود-توجه و ترنسفرمرها

ساختار کدکننده-کدگشا که پیش‌تر دارای از دو شبکه عصبی بازگشتی تشکیل می‌شد، توسط مولفین [۶۳] با شبکه‌ای تحت عنوان ترنسفرمرها جابجا شد که موجبات مدل‌سازی‌های گسترده در حوزه پردازش زبان طبیعی[4] را فراهم کرد. بخش اصلی ترنسفرمرها که در شکل۸ به نمایش درآمده است که مکانیزم خود-توجه است. بر اساس این مکانیزم، بازنمایی که برای هر کلمه در وظیفه ماشین ترجمه استخراج می‌شود، حساسیتی برای ترتیب کلمات حاضر در یک دنباله متنی ورودی قائل نیست. به بیان بهتر، بازنمایی هر کلمه تنها به حضور کلمات حساس است. لازم بذکر است امروزه استفاده از ترنسفرمرها صرفا معطوف به حوزه پردازش زبان طبیعی و مدل‌سازی داده‌های ترتیبی نیست و در حوزه بینایی ماشین نیز از این ابزار استفاده می‌شود [۶۴]، [۶۵]، [۶۶] و [۶۷].



شکل ۸- ساختار کدکننده –کدگشا در حضور مکانیزم خودتوجه – ترنسفرمر [۶۳]

بر اساس این مدل، ابتدا بازنمایی‌هایی که از هر کلمه استخراج می‌شود، از یک مکانیزم توجه عبور می‌کند که این مکانیزم یک رابطه ساده محاسبه ضرب داخلی مقیاس شده[5] است. ریشه‌ی اصلی این مکانیزم مربوط به سیستم‌های بازیابی اطلاعات است [۶۳]. در یک سیستم بازیابی اطلاعات[6] زمانی که کاربر یک عبارت جستجو[7] را وارد می‌کند، سیستم بر اساس اطلاعاتی که در پایگاه داده خود دارد به هر مقدار[8] یک کلید[9] اختصاص می‌دهد و بر اساس شباهتی[10] که بین عبارت جستجو و کلید وجود دارد، مقدار متناظر را برمی‌گرداند. بر این اساس، کارکرد مدل بر

[1] Context vector
[2] Additive attention
[3] Self-attention
[4] Natural language processing (NLP)
[5] Scaled dot-product
[6] Information retrieval system
[7] Query
[8] Value
[9] Key
[10] Similarity

محاسبه سه ویژگی جستجو، مقدار و کلید استوار است که به نسبت شباهتی که بین ویژگی‌های جستجو و کلید وجود دارد، یک نگاشت مناسب با مقدار را برمی‌گرداند. ویژگی‌های مقدار، کلید و جستجو هر سه از یک تبدیل خطی بدست می‌آیند و سپس مطابق با رابطه()، مقدار اهمیت هر کلمه مشخص می‌شود.

$$O = softmax\left(\frac{Q^T K}{\sqrt{d}}\right)V \qquad (۱۸)$$

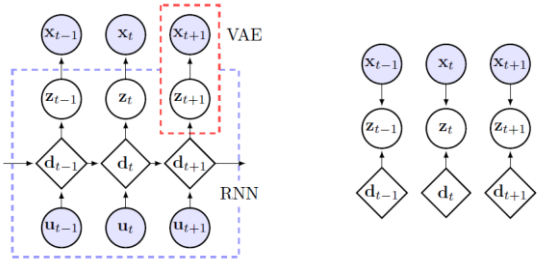
نکته مهم درباره ترنسفرمرها این است که هیچ لایه بازگشتی وجود ندارد و مدل از طریق بردار کدکننده موقعیت[1]، اطلاعات زمانی را فرا می‌گیرد. در [۶۳] از توابع مثلثاتی برای کد کردن موقعیت استفاده شده است. رویکردهای متفاوتی برای محاسبه کدکننده موقعیت وجود دارد که در [۶۸] آورده شده است. روند مشابه با روند معرفی شده برای کدکننده در کدگشا اتفاق می‌افتد با این تفاوت که چون قصد داریم مقدار آینده خروجی را پیش‌بینی کنیم، از مکانیزم خود-توجه پوشش داده شده برای کدگشا استفاده می‌شود تا مقادیر آینده خروجی در مدل‌سازی لحاظ نگردد. جزئیات مربوط به ترنسفرمرها در [۶۳] آمده است.

۳- مروری بر ساختارهای مدل‌های دینامیکی عمیق در شناسایی سیستم‌ها

هیچ‌یک ابزارهایی که از حوزه یادگیری ماشین برای توسعه روش‌های شناسایی غیرخطی به کار گرفته شده است، توسط جوامع مهندسی کنترل توسعه نیافته است و عمده این روش‌ها برای منظور شناسایی گفتار و پردازش متن توسعه یافته است [۱۷]. برخی از این پژوهش‌ها از خودرمزنگار تغییراتی و یا ترکیبی از خودرمزنگار تغییراتی و شبکه‌های عصبی بازگشتی به منظور یک ابزار برای شناسایی فضای حالت غیرخطی و یا خطی استفاده کرده‌اند [۳۲]، [۳۳] ، [۳۶] ، [۳۴]، [۶۹]، [۷۰] و [۷۱]. چون در بخش قبل، خودرمزنگار تغییراتی معرفی شد، در این بخش ابزارهایی که از ترکیب خودرمزنگار تغییراتی و شبکه‌های عصبی بازگشتی به‌وجود آمده‌اند، معرفی خواهد شد.

۳-۱- خودرمزنگار تغییراتی – شبکه عصبی بازگشتی[2]

خود رمزنگار تغییراتی – شبکه عصبی بازگشتی که در [۱۷] معرفی شده است با هدف اولیه تولید متن و گفتار[3] توسعه داده شده است که قادر است از دو منظر یک مدل مولد گفتار[4] تولید کند. بر اساس این مدل، گفتار هر شخص دارای دو منبع اطلاعات است. منبع اول مربوط به مشخصات زبانی گوینده است که بین تمام گوینده‌های آن زبان یکسان است و منبع دوم مربوط به تغییراتی است که از هر فرد به فرد دیگر متفاوت است. از دیدگاه ریاضی می‌توان این مسئله را به این صورت عنوان کرد که مدل کردن سیگنال گفتار نیازمند دو نوع نگاشت غیرخطی است که نگاشت اول یک نگاشت معین و نگاشت دوم یک نگاشت تصادفی است. با همین انگیزه می‌توان مسئله بیان شده را به حوزه شناسایی سیستم‌های غیرخطی نیز تعمیم داد چرا که هر دنباله ورودی و خروجی دارای اشتراکاتی هستند که مربوط به رابطه بین ورودی و خروجی است و از طرفی دارای تغییراتی نیز هستند که برای هر دنباله ورودی و خروجی منحصر بفرد است (با این فرض که این تغییرات به صورت ایستا مدل می‌شود و نه غیر ایستا که خود این مطلب یکی از ایرادات این معماری است زیرا فرض می‌کند که متغیرهای تصادفی هیچ ارتباطی با هم ندارند). این مسئله برای شناسایی سیستم‌های غیرخطی در [۳۳] بررسی شده است. شکل ۹ مدل گرافی احتمالاتی این شبکه را برای هر دو شبکه مولد و شبکه بازشناسایی به تصویر کشیده است.



شکل ۹ – مدل گرافی احتمالاتی مربوط به ساختار VAE-RNN[2] [۱۷]

در این معماری ابتدا به کمک یک شبکه عصبی بازگشتی، یک بازنمایی از داده‌های ورودی استخراج می‌شود که این بازنمایی مربوط به اطلاعات زمانی متغیرهای ورودی است. سپس بازنمایی استخراج شده به عنوان ورودی یک خود رمزنگار تغییراتی عمل می‌کند و بخش کدگشای مدل سعی می‌کند اطلاعات مربوط به خروجی را از روی اطلاعات کد شده ورودی، تولید کند. مزیت این رویکرد استخراج هر دو مجموعه ویژگی مرتبط با تغییرات متغیرهای ورودی و ویژگی‌های قطعی منحصر به‌فرد سیگنال ورودی است که توانایی مدل کردن توزیع شرطی $p_\theta(x_{1:T}|u_{1:T})$ داراست. توزیع توام برای مدل مولد به صورت زیر خواهد بود:

$$p_\theta(x_{1:T}, d_{1:T}, z_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t|d_t)p_\theta(d_t|d_{t-1}, u_t)p_\theta(x_t|z_t) \qquad (۱۹)$$

که در آن $p_\theta(d_t|d_{t-1}, u_t) = \delta(d_t - \tilde{d}_t), \tilde{d}_t = f_\theta(d_{t-1}, u_t)$ است که با توجه به خاصیت غربالی تابع ضربه خواهیم داشت:

[1] Positional encoding
[2] VAE-RNN
[3] Speech
[4] Speech generative model

$$p_\theta(x_{1:T}, z_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t|\tilde{d}_t)p_\theta(x_t|z_t) \quad (۲۰)$$

به منظور آموزش پارامترهای مدل مشابه حالتی که برای فضای حالت غیرخطی دیدیم، کران پایین مشاهدات را به صورت زیر می‌نویسیم:

$$\log p_\theta(x_{1:T}|u_{1:T}, d_0) \geq E_q\left(\log \frac{p_\theta(x_{1:T}, z_{1:T}|u_{1:T}, d_0)}{q_\phi(z_{1:T}|x_{1:T}, u_{1:T}, d_0)}\right) = F(\theta, \phi) \quad (۲۱)$$

که در آن توزیع تغییراتی، بر اساس ساختار واقعی توزیع احتمال موخر به صورت رابطه (۲۲) قابل تجزیه است:

$$q_\phi(z_{1:T}|x_{1:T}, u_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|x_t, \tilde{d}_t) \quad (۲۲)$$

با این تعریف کران پایین مشاهدات به صورت زیر خواهد بود:

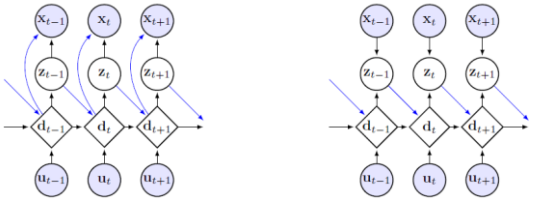
$$F(\theta, \phi) = \sum_{t=1}^{T} E_{q_\phi}[\log p_\theta(x_t|z_t)] - D_{kl}\left(q_\phi(z_t|x_t, \tilde{d}_t)\middle|p_\theta(z_t|\tilde{d}_t)\right) \quad (۲۳)$$

تابع هزینه مناسب برای آموزش این مدل، مشخصا شباهت نزدیک با تابع هزینه و مکانیزم آموزش خود رمزنگار تغییراتی دارد که نتیجتا به سادگی و با استفاده از الگوریتم گرادیان نزولی قابل بهینه‌سازی است. تمامی ترم‌های توزیع شرطی موجود در این مدل از یک توزیع گوسی پیروی می‌کند که پارامترهای آن توسط یک شبکه عصبی تعیین می‌شود.

بزرگترین ایراد مدل توسعه داده شده در این است که خروجی مدل تنها تابعی از بخش تصادفی ویژگی‌های ورودی است. در حالی که هر سیگنال زمانی را می‌توان به دو مولفه تصادفی و قطعی تجزیه کرد. بنابراین لازم است که یک ارتباط مستقیم از ویژگی‌های قطعی به خروجی مدل وجود داشته باشد. ساختار پیدا شده از این طریق شبکه عصبی بازگشتی تغییراتی[1] نامیده می‌شود که در ادامه مورد بررسی قرار می‌گیرد.

۲-۳- شبکه عصبی بازگشتی تغییراتی

یکی از ایرادات ساختار خودرمزنگار تغییراتی- شبکه عصبی بازگشتی تولید خروجی تنها بر اساس ویژگی‌های تصادفی است حال آنکه خروجی از ویژگی‌های قطعی نیز تاثیر می‌پذیرد [۱۷] و [۵۵]. بنابراین لازم است تا هر دو مورد ذکر شده در ساختار بررسی شده در بخش ۱-۳ نیز لحاظ شود. تغییرات اعمال شده بر روی ساختار خودرمزنگار تغییراتی – شبکه عصبی بازگشتی منجر به تولد یک مدل فضای حالت به‌نام شبکه عصبی بازگشتی تغییراتی می‌شود [۷۲] که شبکه‌های مولد و بازشناسایی آن مطابق شکل ۱۰ خواهد بود.



شکل ۱۰ – شبکه‌های مولد و بازشناسایی مدل عصبی بازگشتی تغییراتی [۱۷]

مطابق با این ساختار، ویژگی‌های قطعی محاسبه شده توسط شکل ۲-۸ تابعی از ویژگی‌های تصادفی نیز بوده و خروجی نیز بر اساس ویژگی‌های تصادفی و قطعی ساخته می‌شود. از این ساختار در [۱۸] به منظور شناسایی سیستم‌های غیرخطی استفاده شده است.

روابط مربوط به شبکه مولد به صورت زیر خواهد بود:

$$p_\theta(x_{1:T}, z_{1:T}, d_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t|z_t, d_t)p_\theta(d_t|z_{t-1}, d_{t-1}, u_t)p_\theta(z_t|d_t) \quad (۲۴)$$

که در آن $p_\theta(d_t|d_{t-1}, z_{t-1}\, u_t) = \delta(d_t - \tilde{d}_t), \tilde{d}_t = f_\theta(d_{t-1}, z_{t-1}, u_t)$ خواهد بود. ممکن است این شائبه ایجاد شود که چون متغیرهای حالت شبکه عصبی بازگشتی به z_{t-1} وابسته است دیگر ماهیت قطعی بودن خود را از دست می‌دهد و دارای عدم قطعیت نیست. اما این مطلب درست نیست زیرا وابستگی به z_{t-1}ای که به لحاظ زمانی رخ داده است به صورت قطعی است که از گام پیشین زمانی نمونه‌گیری شده است. با توجه به خاصیت غربالی تابع ضربه خواهیم داشت:

$$p_\theta(x_{1:T}, z_{1:T}, d_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t|z_t, \tilde{d}_t)p_\theta(z_t|\tilde{d}_t) \quad (۲۵)$$

به منظور آموزش پارامترهای مدل مشابه حالت قبل، کران پایین مشاهدات را به صورت زیر می‌نویسیم:

$$\log p_\theta(x_{1:T}|u_{1:T}, d_0) \geq E_q\left(\log \frac{p_\theta(x_{1:T}, z_{1:T}|u_{1:T}, d_0)}{q_\phi(z_{1:T}|x_{1:T}, u_{1:T}, d_0)}\right) = F(\theta, \phi)$$

که در آن توزیع تغییراتی، مطابق آنچه که در بخش مربوط به مدل متغیرهای پنهان گفته شود بر اساس ساختار واقعی توزیع احتمال موخر به صورت زیر قابل تجزیه است:

$$q_\phi(z_{1:T}|x_{1:T}, u_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|x_t, \tilde{d}_t) \quad (۲۶)$$

با این تعریف کران پایین مشاهدات به صورت زیر خواهد بود:

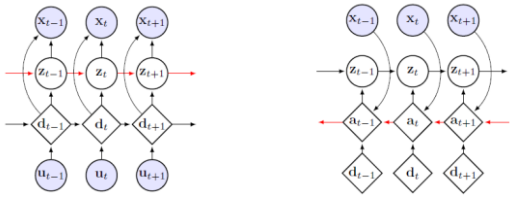
[1] Variational RNN

$$F(\theta,\phi) \quad (۲۷)$$
$$= \sum_{t=1}^{T} E_{q_\phi}[\log p_\theta(x_t|z_t,d_t)]$$
$$- D_{kl}\left(q_\phi(z_t|x_t,\tilde{d}_t)\middle|p_\theta(z_t|\tilde{d}_t)\right)$$

تابع هزینه مناسب برای آموزش این مدل، مشخصا شباهت نزدیک با تابع هزینه و مکانیزم آموزش خود رمزنگار تغییراتی دارد که نتیجتا به سادگی و با استفاده از الگوریتم گرادیان نزولی قابل بهینه‌سازی است. تفاوت تابع هزینه این ساختار با ساختار قبل، وابستگی خروجی به ویژگی‌های تصادفی استخراج شده است که برای ساخت متغیرهای حالت شبکه عصبی بازگشتی نیز لازم است تا متغیرهای تصادفی مقداردهی اولیه داشته باشند. تمامی ترم‌های توزیع شرطی موجود در این معادله از یک توزیع گوسی پیروی می‌کند که پارامترهای آن توسط یک شبکه عصبی تعیین می‌شود. ایراد اساسی در این ساختار، این است که مدل قادر به فراگرفتن عدم قطعیت‌های مدلسازی به صورت کامل نیست. برای حل این مشکل ساختار جدیدی به نام شبکه عصبی بازگشتی تصادفی[1] پیشنهاد شد که متغیرهای تصادفی یک مدل مارکوف مرتبه اول را مدل می‌کنند [۷۳].. مدل مشابه دیگری نیز با عنوان شبکه تصادفی بازگشتی[2] در [۷۴] معرفی شده است که به منظور اغنای ویژگی‌های استخراج شده در یک شبکه عصبی بازگشتی به کارگرفته شده است.

۳-۳- شبکه عصبی بازگشتی تصادفی

به منظور ایجاد یکنواختی واریانس بین متغیرهای تصادفی تعریف شده در ساختارهای ارائه شده نیاز به وجود ارتباط بین این ویژگی‌های تصادفی وجود دارد [۵۵]. در ارتباط غیرمستقیم هر ویژگی تصادفی استخراج شده بر روی ویژگی قطعی در گام بعدی اثر می‌گذارد در حالی که در ارتباط مستقیم، هر ویژگی تصادفی به صورت مستقیم بر روی ویژگی تصادفی بعد از خود اثرگذار است [۷۳]. شکل ۱۱ ساختار شبکه عصبی بازگشتی تصادفی را به نمایش کشیده است.



شکل ۱۱ – شبکه‌های مولد و بازشناسایی ساختار عصبی بازگشتی تصادفی [۱۷]

روابط مربوط به شبکه مولد برای این ساختار به قرار زیر است:

$$p_\theta(x_{1:T},z_{1:T},d_{1:T}|u_{1:T}) \quad (۲۸)$$
$$= \prod_{t=1}^{T} p_\theta(x_t|z_t,d_t)p_\theta(d_t|d_{t-1},u_t)p_\theta(z_t|d_t,z_{t-1})$$

که در آن $p_\theta(d_t|d_{t-1},u_t)=\delta(d_t-\tilde{d}_t),\tilde{d}_t=f_\theta(d_{t-1},u_t)$ است که با توجه به خاصیت غربالی تابع ضربه خواهیم داشت:

$$p_\theta(x_{1:T},z_{1:T}|u_{1:T}) \quad (۲۹)$$
$$= \prod_{t=1}^{T} p_\theta(z_t|z_{t-1},\tilde{d}_t)p_\theta(x_t|z_t,\tilde{d}_t)$$

به منظور آموزش پارامترهای مدل مشابه حالت قبل، کران پایین مشاهدات را به صورت زیر می‌نویسیم:

رابطه
$$\log p_\theta(x_{1:T}|u_{1:T},d_0) \quad (۳۰)$$
$$\geq E_q\left(\log\frac{p_\theta(x_{1:T},z_{1:T}|u_{1:T},d_0)}{q_\phi(z_{1:T}|x_{1:T},u_{1:T},d_0)}\right)$$
$$= F(\theta,\phi)$$

به منظور آموزش شبکه نیاز است تا یک توزیع تغییراتی مناسب انتخاب شود که مانند بخش‌های قبلی از ساختار توزیع احتمال موخر به صورت مستقیم استفاده می‌کنیم. نکته مهم درباره این ساختار این است که چون وابستگی بین متغیرهای تصادفی وجود دارد، به منظور تولید نمونه‌های تصادفی نیازمند پردازش گام‌های زمانی آینده خواهیم بود. به بیان ریاضی خواهیم داشت:

$$p_\theta(z_{1:T}|x_{1:T},u_{1:T},d_0) \quad (۳۱)$$
$$= \prod_{t=1}^{T} p_\theta(z_t|z_{t-1},d_{t:T})$$

در نتیجه توزیع تغییراتی نیز دارای ساختار مشابه به صورت زیر خواهد بود:

$$q_\phi(z_{1:T}|x_{1:T},u_{1:T},d_0) \quad (۳۲)$$
$$= \prod_{t=1}^{T} q_\phi(z_t|z_{t-1},\tilde{d}_{t:T})$$

که همانند مدل‌های فضای حالت غیرخطی برای حل مشکل وابستگی به آینده از روش هموارساز کالمن عمیق باید استفاده کرد. با این تعریف کران پایین مشاهدات مطابق رابطه زیر خواهد بود:

$$F(\theta,\phi) \quad (۳۳)$$
$$= \sum_{t=1}^{T} E_{q_\phi}[\log p_\theta(x_t|z_t,\tilde{d}_t)]$$
$$- D_{kl}\left(q_\phi(z_t|z_{t-1},\tilde{d}_{t:T})\middle|p_\theta(z_t|z_{t-1},\tilde{d}_t)\right)$$

که تمامی ترم‌های توزیع شرطی موجود در این معادله از یک توزیع گوسی پیروی می‌کند که پارامترهای آن توسط یک شبکه عصبی تعیین می‌شود. ساختار دیگری در همین راستا با نامی مشابه وجود دارد که عملکرد متفاوتی دارد. بر اساس این ساختار یک مکانیزم پنهان وجود دارد

[1] Stochastic RNN

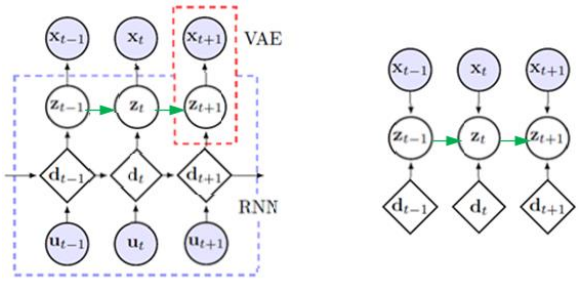
[2] Stochastic recurrent network

که این مکانیزم به نگاشت معین دنباله ورودی کمک می‌کند. این ساختار شبکه تصادفی بازگشتی نامیده شده است که جزئیات مربوط به آن در [۷۴] آورده شده است. در مقایسه با مدل ارائه شده، بدلیل وابستگی بین ویژگی-های قطعی و ویژگی‌های تصادفی در همان گام زمانی، ویژگی‌های قطعی دیگر ماهیت قطعی ندارند و دارای یک توزیع احتمال هستند. فرآیند آموزش در این ساختار همانند آنچه که در ساختارهای قبلی ارائه شده است از روش گرادیان نزولی تغییراتی بیز[1] قابل استفاده می‌کند.

در تمام ساختارهای ارائه شده ابتدا مدل ویژگی‌های زمانی داده‌های ورودی را فرا می‌گیرد سپس با استفاده از یک ساختار مشابه خودرمزنگار تغییراتی یک بازنمایی بعد پایین[2] از داده‌ها را برای مدل کردن توزیع احتمال محاسبه می‌کند. رویکرد عکس این ساختار نیز وجود دارد که عمدتا برای تحلیل داده‌های ویدیویی به کار گرفته می‌شود. در این ساختار، مدل سعی می‌کند ابتدا یک نگاشت در ابعاد پایین از داده‌ها محاسبه کند و سپس اطلاعات زمانی را فرا بگیرد که با نام خودرمزنگار تغییراتی کالمن[3] شناخته می‌شود [۷۵]. این ساختار به منظور تشخیص و ردیابی اشیاء موجود در ویدیو[4] ارائه شده است، از یک خودرمزنگار تغییراتی پیچشی[5] استفاده شده است تا هر فریم تصویر به یک فضای با بعد پایین نگاشت داده شود. سپس هر ویژگی محاسبه شده به یک مدل فضای حالت خطی گوسی[6] داده می‌شود که پارامترهای آن توسط ترکیب خطی یک بانک از پارامترهای مدل خطی محاسبه می‌شود. چون در این حالت، نگاشت ویژگی‌های هر فریم ورودی و خروجی مشخص است، متغیرهای مخفی موجود در این مدل توسط فیلتر کالمن خطی تخمین زده می‌شود (چون ورودی و خروجی توسط خود رمزنگار تغییراتی محاسبه می‌شود). کل ساختار کران پایین مشاهدات با روش گرادیان تصادفی تغییراتی بیز آموزش می‌بینند. از میان ساختارهای موجود، تنها این ساختار در حوزه شناسایی سیستم‌های غیرخطی ورود نکرده است حال آنکه می‌تواند برای شناسایی سیستم‌های چندورودی-چندخروجی[7] به کار گرفته شود . جزئیات مربوط به این ساختار در [۷۵] آورده شده است.

٤-فضای حالت عمیق ساختار یافته و شبیه‌سازی مقایسه‌ای

بر اساس مطلبی که در [۵۵] و [۱۷] بیان شده است، برای یک مدل مولد دنباله نیاز است تا عدم قطعیت به صورت ایستا مدل شود. به بیان بهتر، باید مدل‌سازی لایه مربوط به نگاشت تصادفی، از گام‌های زمانی پیش از خود نیز اثر بپذیرد تا اطمینان از اینکه سطح مشخصی از عدم قطعیت در مدل‌سازی لحاظ شده است حاصل گردد. این موضوع دقیقا مطابق با مدل ارائه شده فضای حالت در [۱۸] است با این تفاوت که ساختار ساختار شبکه عصبی بازگشتی – خودرمزنگار تغییراتی را دستخوش تغییر کرده است. به بیان بهتر، لایه پنهان دارای ساختار است و از یک فرآیند مارکوف مرتبه اول مطابق شکل ۱۲، پیروی می‌کند.



شکل ۱۲ – مدل شبکه عصبی بازگشتی – خودرمزنگار تغییراتی ساختار یافته

در مقایسه با مدل ارائه شده در شکل ۱۱، ارتباط بین متغیرهای پنهان با رنگ سبز به ساختار اضافه شده است. دلیل تغییر دادن این ساختار، حفظ اطلاعات مربوط به بخش نگاشت تصادفی است که در هر گام زمانی باید لحاظ شود. همچنین این ساختار تطبیق‌پذیری بیش‌تری با فرم فضای حالت سیستم‌های کنترلی دارد که در آن، متغیرهای حالت در هر گام زمانی تابعی معین از متغیرهای حالت در زمان‌های پیشین است که تحت تحریک عدم قطعیت گوسی جمع‌شوند قرار گرفته است. چون هر متغیر حالت تحت تاثیر عدم قطعیت است، این اثر به صورت غیرخطی باید به حالت بعدی منتقل شود. دلایل مطرح شده برای تغییر ساختار این مدل را می‌توان عینا برای شبکه عصبی تصادفی نیز به کار برد.نکته بسیار مهم در تغییر ساختار به شکل بیان شده این است که فرآیند آموزش مدل نیازمند استنتاج درباره گام‌های زمانی آینده نیز خواهد بود که این ارتباط به دلیل مرتبط بودن متغیرهای پنهان به یکدیگر است. برای رفع این مشکل از هموار ساز کالمن عمیق استفاده شده است.

٤-١-شبیه‌سازی

در این بخش نتایج حاصل از شبیه‌سازی ساختارهای معرفی شده و ساختار تغییر یافته را با یکدیگر مقایسه خواهیم کرد. برای این منظور از سه سیستم برای شبیه‌سازی استفاده شده است که ابتدا در ادامه معرفی خواهند شد.

٤-١-١- سیستم گوسی خطی[8]

رابطه (۳۴) به عنوان اولین سیستم محک که در [۳۳] معرفی شده است مورد ارزیابی قرار می‌گیرد.

[1] Stochastic Gradient Variational Bayes (SGVB)
[2] Low dimensional manifold
[3] Kalman Variational Auto-encoder (KVAE)
[4] Video object detection and tracking
[5] Convolutional VAE
[6] Linear Gaussian state space mode (LGSSM)
[7] Multivariate systems
[8] Linear Gaussian system

$$x_{k+1} = \begin{bmatrix} 0.7 & 0.8 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} -1 \\ 0.1 \end{bmatrix} u_k + w_k \quad (۳۴)$$

$$y_k = [1 \quad 0] x_k + v_k$$

که در آن، w_k و v_k به ترتیب نویز فرآیند و نویز اندازه‌گیری هستند. در شبیه‌سازی‌های مربوط به این سیستم، نویزهای فرآیند و اندازه‌گیری، هر دو از یک توزیع گوسی با میانگین صفر پیروی می‌کند. واریانس نویز اندازه‌گیری برابر با ۱ و واریانس نویز فرآیند برابر با ۰.۵ در نظر گرفته شده است. به منظور شناسایی سیستم، برای داده‌های آموزش و اعتبارسنجی[1] از یک ورودی که به صورت یکنواخت که در بازه‌ی ۲.۵- تا ۲.۵ توزیع شده‌اند، استفاده شده است. هر دو مجموعه از نمونه‌های آموزش و اعتبارسنجی، بعد از ۵۰ بار اجرا و اعمال عملگر میانگین مورد استفاده قرار گرفته‌اند. برای داده‌های تست نیز از ورودی $u_k = \sin\frac{2k\pi}{10} + \sin\frac{2k\pi}{5}$ استفاده شده است. در مجموعه ۶۰۰۰ داده آموزش و ۲۰۰۰ داده اعتبارسنجی و تست مورد استفاده قرار گرفته است.

۲-۱-۴- سیستم محک Narendra-Li[2]

این سیستم برای اولین بار توسط Narendra معرفی شده است که یک سیستم غیرخطی فرضی است و نمونه فیزیکی آن در دنیای واقعی ساخته نشده است. این سیستم به عنوان دومین سیستم محک برای ارزیابی ساختار ارائه شده در [۳۳] که در حوزه شناسایی سیستم‌ها در [۳۳] به کار گرفته شده است، مورد ارزیابی قرار گرفته است. معادلات دینامیکی و خروجی این سیستم مطابق رابطه (۳۵) و (۳۶) است.

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} \left(\frac{x_k^1}{1+(x_k^1)^2}\right)\sin x_k^2 \\ x_k^2\cos x_k^2 + x_k^1 \exp\left(-\frac{(x_k^1)^2+(x_k^2)^2}{8}\right) + \frac{u_k^3}{1+u_k^2+0.5\cos(x_k^1+x_k^2)} \end{bmatrix} \quad (۳۵)$$

$$y_k = \frac{x_k^1}{1+0.5\sin x_k^2} + \frac{x_k^2}{1+0.5\sin x_k^1} + e_k \quad (۳۶)$$

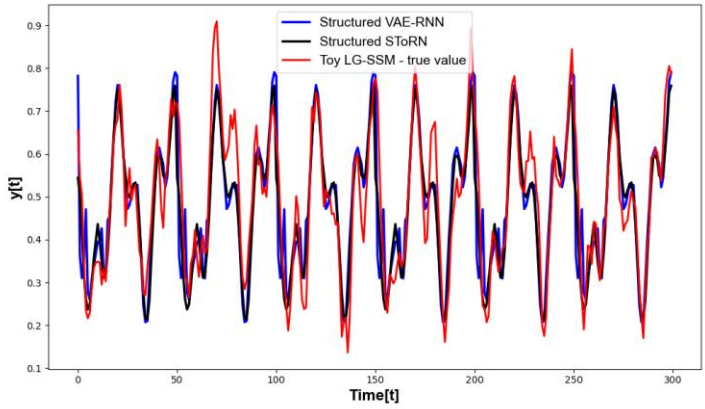
که در آن e_k نویز اندازه‌گیری است. لازم بذکر است که در نسخه اصلی این سیستم محک نویز اندازه‌گیری لحاظ نشده است و نویز اندازه‌گیری به منظور افزایش پیچیدگی به سیستم اضافه شده است. در شبیه‌سازی‌های این بخش، این نویز از یک توزیع گوسی با میانگین ۰ و واریانس ۰.۵ پیروی می‌کند.

۳-۱-٤- سیستم محک Wiener-Hammerstein[3]

فرآیند Wiener-Hammerstein سومین سیستم محکی است که به منظور نمایش کارآیی ساختارهای معرفی شده مورد بررسی قرار گرفته است. داده‌های مربوط به این فرآیند که دارای نویز فرآیند است توسط پژوهشگران دانشگاه آیندهوون جمع‌آوری شده است و در وبگاه مربوط به سیستم‌های محک غیرخطی منتشر شده است.

٤-۱-٤- نتایج

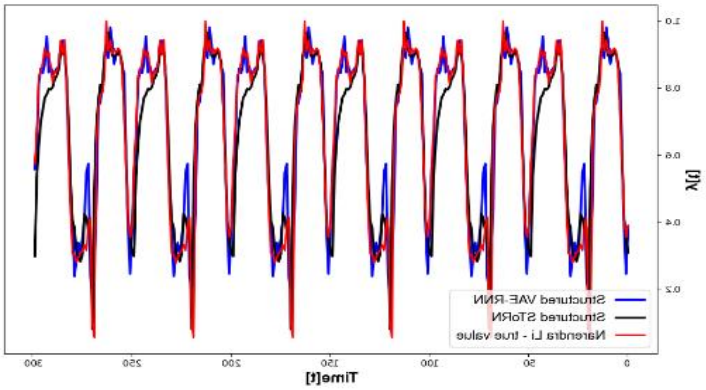
در این بخش نتایج مربوط به شناسایی سیستم‌های معیار معرفی شده در بخش قبل ارائه خواهد شد. کلیه شبیه‌سازی‌های صورت گرفته با استفاده از قاعده مونت کارلو[4] صورت گرفته است. برای این منظور به ازای هر ورودی، از خروجی کدکننده ۱۰۰ نمونه استخراج شده است و خروجی کدگشا میانگین نگاشتی از این ۱۰۰ نمونه است. شکل ۱۳ خروجی مدل را برای سیستم خطی گوسی نشان می‌دهد.



شکل ۱۳- عملکرد ساختارهای Structured STORN و Structured VAE-RNN برای سیستم خطی گوسی

[1] Validation
[2] Narendra-Li benchmark
[3] Wiener-Hammerstein benchmark
[4] Monte Carlo simulation

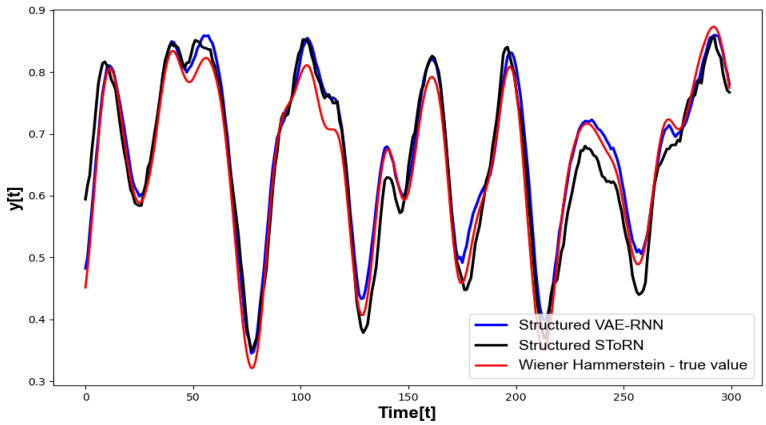
ابعاد لایه پنهان شبکه عصبی بازگشتی برای این شبیه‌سازی برابر با ۸۰ و ابعاد لایه تصادفی برابر با ۱۰ در نظر گرفته شده است. همچنین از یک لایه بازگشتی برای مدل‌سازی استفاده شده است. طول دنباله ورودی نیز برابر با ۶۰ لحاظ شده است. مقدار موثر خطا[1] و منفی لگاریتم تابع بیشینه شباهت[2] $\frac{1}{T}\sum -\log p(y_t|\mu_{de},\Sigma_{de})$ نیز برای خودرمزنگار تغییراتی – شبکه عصبی ساختار یافته به ترتیب برابر با ۱.۰۴ و ۰.۹۹ محاسبه شده است که این اعداد برای شبکه تصادفی بازگشتی ساختار یافته برابر با ۰.۹۴ و ۰.۹۳ است. شکل ۱۴ خروجی مدل‌های ارائه شده برای سیستم محک Narendra-Li را نشان می‌دهد.



شکل ۱۴ – عملکرد ساختارهای Structured STORN و Structured VAE-RNN برای سیستم Narendra-Li

مدل در این حالت ساختاری مشابه با حالت قبل دارد و تنها تفاوت آن طول دنباله ورودی است که برای این سیستم برابر با ۵۰ در نظر گرفته شده است. همچنین مقدار موثر خطا و تابع بیشینه شباهت برای ساختار خودرمزنگار تغییراتی – شبکه عصبی بازگشتی ساختار یافته به ترتیب برابر با ۰.۷۶ و ۰.۹۲، و برای ساختار شبکه عصبی تصادفی ساختاریافته برابر با ۱.۰۱ و ۰.۹۶ محاسبه شده است. به عبارتی برای این سیستم عملکرد مدل خودرمزنگارتغییراتی- شبکه عصبی بازگشتی ساختار یافته بهتر است.

شکل۱۵ نیز خروجی مدل‌های ارائه شده برای سیستم معیار Wiener-Hammerstein را نشان می‌دهد. برای این حالت نیز از ساختار مشابه با دو شبیه‌سازی قبلی استفاده شده است با این تفاوت که برای این سیستم طول دنباله ورودی برابر با ۱۵۰ در نظر گرفته شد. مقدار موثر خطا و تابع بیشینه شباهت برای ساختار خودرمزنگار تغییراتی – شبکه عصبی بازگشتی ساختار یافته به ترتیب برابر با ۰.۶۳ و ۰.۹۱ و برای ساختار شبکه عصبی بازگشتی تصادفی ساختار یافته برابر با ۰.۷۴ و ۰.۹۲ محاسبه شده است.



شکل ۱۵ – عملکرد ساختارهای Structured STORN و Structured VAE-RNN برای سیستم Wiener-Hammerstein

[1] RMS

[2] Negative log-likelihood

خلاصه نتایج حاصل شده در جدول ۱ آمده است. نتایج مرتبط به ساختارهای خودرمزنگار تغییراتی – شبکه عصبی بازگشتی، شبکه عصبی بازگشتی تصادفی و شبکه عصبی بازگشتی تغییراتی از مرجع [۳۳] آورده شده است.

جدول ۴ - ۱- خلاصه نتایج حاصل شده برای دو ساختار ارائه شده در مقایسه با ساختارهای موجود

مدل	مقدار موثر خطا (RMS)			منفی لگاریتم تابع شباهت (NLL)		
	Toy-LGSSM	Narendra-Li	Wiener-Hammerstein	Toy-LGSSM	Narendra-Li	Wiener-Hammerstein
VRNN [۳۳]	۱/۴۸	۰.۸۹	۰.۸۶	۱.۸۲	۱.۳۱	۱.۳۳
VAE-RNN [۳۳]	۱/۵۶	۰.۸۴	۰.۷۴	۱.۹۵	۱.۳۴	۱.۲۹
STORN [۳۳]	۱/۴۳	۰.۶۴	۰.۷۴	۱.۷۹	۱.۲۰	۱.۰۱
S-VAE-RNN	۱/۰۴	۰.۷۶	۰.۶۳	۰.۹۳	۰.۹۱	۰.۹۱
S-STORN	۰/۹۴	۱.۰۱	۰.۷۴	۰.۹۲	۰.۹۲	۰.۹۲

ساختارهایی که در پژوهش [۳۳] برای شبیه‌سازی انتخاب شده است از جستجوی مشبک[1] استفاده کرده است حال آنکه برای ساختارهای ارائه شده در این بخش، ساختار شبکه را ثابت در نظر گرفتیم تا بتوانیم دو ساختار را با هم مقایسه کنیم. همچنین برای سیستم Wiener-Hammerstein در مرجع [۳۳] از یک شبکه عصبی بازگشتی ۳ لایه استفاده شده است که به منظور مقایسه ساختارها، در این پژوهش از یک لایه بازگشتی استفاده شده است. بهمین دلیل نتایج ارائه شده در جدول ۱ با نتایج ارائه شده در [۳۳] متفاوت است.

۵- نتیجه‌گیری

بدلیل شباهت فرآیندی که در شناسایی سیستم‌ها با مدل‌سازی در حوزه یادگیری ماشین دارد، می‌توان از ساختارها و الگوریتم‌هایی که در حوزه یادگیری ماشین توسعه یافته‌اند، برای هدف شناسایی سیستم‌های غیرخطی استفاده کرد. این انتقال ابزار از حوزه یادگیری ماشین به حوزه شناسایی سیستم‌ها، نه تنها ابزارهای موجود برای مدل‌سازی را توسعه می‌دهد بلکه در برخی موارد می‌تواند از ابزارهای موجود بهتر عمل کند. در این مقاله مروری، ضمن معرفی ابزارها و بررسی مطالعات پژوهشی صورت گرفته در این راستا، مدل‌های متغیر پنهان به عنوان ابزاری با ظرفیت بالا، به منظور مدل‌سازی و شناسایی فضای حالت سیستم‌های غیرخطی معرفی شد. مدل‌های متغیر پنهان از خانواده مدل‌های مولد هستند که یک نگاشت تصادفی از داده‌های ورودی محاسبه می‌کند که تغییرات موجود در داده‌ها را، بر خلاف نگاشت‌های معین، لحاظ می‌کند. در پایان با اعمال تغییر جزئی در ساختار مدل‌های متغیر پنهان به منظور تطبیق‌پذیری بیش‌تر با مفهوم متغیر حالت در سیستم‌های کنترل، مدل ساختار یافته متغیر پنهان معرفی شد. نتایج حاصل از شبیه‌سازی نشان می‌دهد که بهره‌وری این مدل‌ها قابلیت مقایسه با مدل‌های سنتی حوزه شناسایی سیستم‌ها را داراست، در برخی موارد نتایج بهتر و در برخی موارد نتایج نزدیک گزارش شده است.

٦- مراجع

[1] L. Ljung, “System identification, Theory for the user.” System science series, Prentice Hall, Upper Saddle River, NJ, USA, Second edition, 1999.

[2] Zadeh, L. "On the identification problem." *IRE Transactions on Circuit Theory* 3.4 (1956): 277-281.

[3] Nelles, Oliver. "Nonlinear system identification, from classical approach to neural networks, fuzzy systems, and Gaussian process". Springer, Berlin, Heidelberg, 2020.

[4] T. Soderstrom, P. Stocia, “System identification”, Prentice-Hall, Inc, 1988.

[5] Schoukens, Johan, and Lennart Ljung. "Nonlinear system identification: A user-oriented road map." *IEEE Control Systems Magazine* 39.6 (2019): 28-99.

[6] Billings, Stephen A. "Identification of nonlinear systems–a survey." *IEE Proceedings D (Control Theory and Applications)*. Vol. 127. No. 6. IET Digital Library, 1980.

[7] Zheng, Qingsheng, and Evanghelos Zafiriou. "Nonlinear system identification for control using Volterra-Laguerre expansion." *Proceedings of 1995 American Control Conference-ACC'95*. Vol. 3. IEEE, 1995.

[8] Korenberg, Michael J., and Ian W. Hunter. "The identification of nonlinear biological systems:

[1] Grid search

Wiener kernel approaches." *Annals of Biomedical Engineering* 18 (1990): 629-654.

[9] Schoukens, Maarten, and Koen Tiels. "Identification of block-oriented nonlinear systems starting from linear approximations: A survey." *Automatica* 85 (2017): 272-292.

[10] Chiuso, Alessandro, and Gianluigi Pillonetto. "System identification: A machine learning perspective." *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019): 281-304.

[11] Pillonetto, Gianluigi, et al. "Kernel methods in system identification, machine learning and function estimation: A survey." *Automatica* 50.3 (2014): 657-682.

[12] Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. No. 4. New York: springer, 2006.

[13] Heaton, Jeff. Ian goodfellow, yoshua bengio, and aaron courville: “Deep learning." (2018): 305-307.

[14] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013): 1798-1828.

[15] Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

[16] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv: 1312.6114* (2013).

[17] Fraccaro, Marco. "Deep latent variable models for sequential data." *English. PhD thesis* DTU University (2018).

[18] Barber, David. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[19] Koller, Daphne, and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[20] Hinton, Geoffrey E. "A practical guide to training restricted Boltzmann machines." *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 599-619.

[21] Fischer, Asja, and Christian Igel. "An introduction to restricted Boltzmann machines." *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings 17*. Springer Berlin Heidelberg, 2012.

[22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In CVPR, 2006.

[23] Gianluigi Pillonetto, Aleksandr Aravkin, Daniel Gedon, Lennart Ljung, Antônio H. Ribeiro, Thomas B. Schön, Deep network for system identification, arXiv, 2023.

[24] Ljung, Lennart, et al. "Deep learning and system identification." *IFAC-PapersOnLine* 53.2 (2020): 1175-1181.

[25] M. Forgione and D. Piga. Dynonet: A neural network architecture for learning dynamical systems. Int. J. Adapt. Control Signal Process., 35(4):612–626, 2021.

[26] J. Hendriks, F.K. Gustafsson, A.H. Ribeiro, A. Wills, and T.B. Schon. Deep energy-based NARX models. In Proceedings of the 19th IFAC Symposium on System Identification (SYSID), 2021.

[27] LeCun, Yann, et al. "Energy-based models in document recognition and computer vision." *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Vol. 1. IEEE, 2007.

[28] Andersson, Carl, et al. "Deep convolutional networks in system identification." *2019 IEEE 58th conference on decision and control (CDC)*. IEEE, 2019.

[29] Lea, Colin, et al. "Temporal convolutional networks: A unified approach to action segmentation." *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer International Publishing, 2016.

[30] R. Calandra, J. Peters, C. Rasmussen, and M.P. Deisenroth. Manifold gaussian processes for regression. In 2016 International Joint Conference on Neural Networks (IJCNN), pages 3338–3345,2016.

[31] Y. Cho and L. Saul. Kernel methods for deep learning. In Advances in Neural Information Processing Systems, volume 22, 2009.

[32] Nagel, Tobias, and Marco F. Huber. "Autoencoder-inspired Identification of LTI systems." *2021 European Control Conference (ECC)*. IEEE, 2021.

[33] D. Gedon, N. Wahlstr¨om, T.B. Sch¨on, and L. Ljung. Deep state space models for nonlinear system identification. In Proceedings of the 19th IFAC Symposium on System Identification (SYSID), 2021.

[34] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data, 2017.

[35] M. Watter, J. Springenberg, J. Tobias, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Volume 2, pages 2746–2754, Cambridge, MA, USA, 2015. MIT Press.

[36] Rangapuram, Syama Sundar, et al. "Deep state space models for time series forecasting." *Advances in neural information processing systems* 31 (2018).

[37] Courts, Jarrad, et al. "Variational state and parameter estimation." *IFAC-PapersOnLine* 54.7 (2021): 732-737.

[38] Courts, Jarrad, et al. "Variational System Identification for Nonlinear State-Space Models." *arXiv preprint arXiv: 2012.05072* (2020).

[39] Menghani, Gaurav. "Efficient deep learning: A survey on making deep learning models smaller, faster, and better." *ACM Computing Surveys* 55.12 (2023): 1-37.

[40] Xu, Canwen, and Julian McAuley. "A survey on model compression and acceleration for pretrained language models." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 9. 2023.

[41] Choudhary, Tejalal, et al. "A comprehensive survey on model compression and acceleration." *Artificial Intelligence Review* 53 (2020): 5113-5155.

[42] Cheng, Yu, et al. "A survey of model compression and acceleration for deep neural networks." *arXiv preprint arXiv:1710.09282* (2017).

[43] Cheng, Yu, et al. "Model compression and acceleration for deep neural networks: The principles, progress, and challenges." *IEEE Signal Processing Magazine* 35.1 (2018): 126-136.

[44] Liang, Tailin, et al. "Pruning and quantization for deep neural network acceleration: A survey." *Neurocomputing* 461 (2021): 370-403.

[45] Rokh, Babak, Ali Azarpeyvand, and Alireza Khanteymoori. "A comprehensive survey on model quantization for deep neural networks." *arXiv preprint arXiv: 2205.07877* (2022).

[46] Gholami, Amir, et al. "A survey of quantization methods for efficient neural network inference." *Low-Power Computer Vision*. Chapman and Hall/CRC, 2022. 291-326.

[47] Liang, Tailin, et al. "Pruning and quantization for deep neural network acceleration: A survey." *Neurocomputing* 461 (2021): 370-403.

[48] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." *arXiv preprint arXiv: 1803.03635* (2018).

[49] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv: 1503.02531* (2015).

[50] Gou, Jianping, et al. "Knowledge distillation: A survey." *International Journal of Computer Vision* 129 (2021): 1789-1819.

[51] Hastie, Trevor, et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. New York: springer, 2009.

[52] Doersch, Carl. "Tutorial on variational autoencoders." *arXiv preprint arXiv: 1606.05908* (2016).

[53] Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." *arXiv preprint arXiv: 1509.00519* (2015) [40] Barber, David, A. Taylan Cemgil, and Silvia Chiappa, eds. *Bayesian time series models*. Cambridge University Press, 2011.

[54] Makhzani, Alireza, et al. "Adversarial autoencoders." *arXiv preprint arXiv: 1511.05644* (2015). [39] Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." *arXiv preprint arXiv: 1509.00519* (2015)

[55] Girin, Laurent, et al. "Dynamical variational autoencoders: A comprehensive review." *arXiv preprint arXiv:2008.12595* (2020).

[56] Beal, Matthew James. *Variational algorithms for approximate Bayesian inference*. University of London, University College London (United Kingdom), 2003.

[57] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[58] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

[59] Su, Yuanhang, and C-C. Jay Kuo. "Recurrent neural networks and their memory behavior: a survey." *APSIPA Transactions on Signal and Information Processing* 11.1 (2022).

[60] Graves, Alex. "Generating sequences with recurrent neural networks." *arXiv preprint arXiv:1308.0850* (2013).

[61] Krishnan, Rahul G., Uri Shalit, and David Sontag. "Deep kalman filters." *arXiv preprint arXiv: 1511.05121* (2015).

[62] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv: 1409.0473* (2014).

[63] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[64] Han, Kai, et al. "A survey on vision transformer." *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022): 87-110.

[65] Khan, Salman, et al. "Transformers in vision: A survey." *ACM computing surveys (CSUR)* 54.10s (2022): 1-41.

[66] Liu, Yang, et al. "A survey of visual transformers." *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[67] Lin, Tianyang, et al. "A survey of transformers." *AI Open* (2022).

[68] Rosendahl, Jan, et al. "Analysis of positional encodings for neural machine translation." *Proceedings of the 16th International Conference on Spoken Language Translation*. 2019.

[69] Beintema, Gerben I., Maarten Schoukens, and Roland Tóth. "Deep subspace encoders for nonlinear system identification." *Automatica* 156 (2023): 111210.

[70] Masti, Daniele, and Alberto Bemporad. "Learning nonlinear state–space models using autoencoders." *Automatica* 129 (2021): 109666.

[71] Lopez, Ryan, and Paul J. Atzberger. "Variational autoencoders for learning nonlinear dynamics of physical systems." *arXiv preprint arXiv:2012.03448* (2020).

[72] Chung, Junyoung, et al. "A recurrent latent variable model for sequential data." *Advances in neural information processing systems* 28 (2015).

[73] Bayer, Justin, and Christian Osendorfer. "Learning stochastic recurrent networks." *arXiv preprint arXiv: 1411.7610* (2014).

[74] Fraccaro, Marco, et al. "Sequential neural models with stochastic layers." *Advances in neural information processing systems* 29 (2016).

[75] Fraccaro, Marco, et al. "A disentangled recognition and nonlinear dynamics model for unsupervised learning." *Advances in neural information processing systems* 30 (2017).