



Improve Anomaly Detection with Deep Learning

Amir Asl Taghivand¹, Ehsan Aminvash²

Abstract

The main reason that data mining has become the focus of attention in the information industry is the availability of large volumes of data and the urgent need to extract useful information and knowledge from this data. In data cleaning operation, the problem of data quality is solved. One of the problems that affects the quality of data is skewed data or abnormal data. These are records whose attribute values are very different from other records. In this research, a method based on deep learning and 14-layer deep neural network on the tensorflow and cross package has been used to diagnose the abnormality and improve its performance. The data set used in this research is a data set with 2% anomalies. The accuracy of the proposed method was 97/08 and the readability and accuracy criteria were 97%. The proposed method was compared with 5 other models based on convolutional neural network and LSTM recursive network. The value of the classification evaluation criteria showed a very good improvement over the proposed method compared to traditional methods and even methods based on deep learning.

Keywords: *Detection of Anomaly, Deep Neural Network, Convulsion Neural Network, Keras*

1. M.sc graduated of software engineering, Bonab Azad University, Bonab, Iran

2. Ph.d student, University of Tabriz, Tabriz, Iran.

Submitted: 19-03-2022

Accepted: 24-07-2022

Corresponding Author: Amir Asl Taghivand

Email: atagivand@gmail.com



بهبود تشخیص برون هشتهای داده‌ها با یادگیری عمیق

امیر اصل تقی‌وند^۱، احسان امین‌وش^۲

چکیده

دلیل اصلی که باعث شد داده کاوی، مورد توجه صنعت اطلاعات قرار بگیرد، مسئله در دسترس بودن حجم وسیعی از داده‌ها و استخراج اطلاعات و دانش سودمند از آن‌ها است. در عملیات پاک‌سازی داده، مشکل کیفیت داده‌ها برطرف می‌شود. یکی از مشکلاتی که بر کیفیت داده‌ها تأثیر می‌گذارد، داده‌های برون هشته هستند. این نمونه‌ها رکوردهایی هستند که مقادیر مشخصه آن‌ها با رکوردهای دیگر بسیار تفاوت دارد. در این تحقیق از یک روش مبتنی بر یادگیری عمیق و شبکه عصبی عمیق ۱۴ لایه‌ای بر روی پکیج تنسورفلو و کراس برای تشخیص برون هشته‌ای و بهبود عملکرد آن استفاده شده است. مجموعه داده مورد استفاده در این تحقیق مجموعه‌ای با ۲ درصد برون هشته‌ای است. میزان صحت روش پیشنهادی مقدار ۹۷/۰۸ را نشان داد و معیارهای بازخوانی و دقت نیز ۹۷ درصد محاسبه شده است. روش پیشنهادی با ۵ مدل دیگر مبتنی بر شبکه عصبی کانولوشن و شبکه بازگشتی LSTM نیز مقایسه شدند. مقدار معیارهای ارزیابی کلاس‌بندها نشان از بهبود بسیار خوب روش پیشنهادی در مقابل روش‌های سنتی و حتی روش‌های مبتنی بر یادگیری عمیق را داده است.

کلمات کلیدی: تشخیص برون هشته‌ای، شبکه عصبی عمیق، شبکه عصبی کانولوشن، کراس

۱. دانش‌آموخته کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه آزاد اسلامی واحد بناب، بناب، ایران.

۲. دانشجوی دکتری، دانشگاه تبریز، تبریز، ایران.

تاریخ دریافت مقاله: ۱۴۰۰/۱۲/۲۸

تاریخ پذیرش نهایی مقاله: ۱۴۰۱/۰۵/۰۲

نویسنده مسئول مقاله: امیر اصل تقی‌وند

Email: atagivand@gmail.com

مقدمه

تحولات چشم‌گیر در حوزه فناوری اطلاعات و ارتباطات در دهه اخیر موجب دگرگونی در بسیاری از زمینه‌ها از جمله تجارت جهانی شده و در نتیجه، فرایندهای بسیاری از فعالیت‌ها از جمله فعالیت‌های تجاری، اقتصادی، بانکداری، گمرکات و ... دستخوش تغییراتی شده است. همچنین پیشرفت در دنیای کامپیوتر و روش‌های هوشمندسازی، سبب شده است تا روش‌های متنوعی برای عملیات مختلف فناوری اطلاعات، برای کارهای گوناگون در نظر گرفته شود. وجود داده‌های متنوع در ارتباطات ماشین به ماشین در بسیاری از محیط‌ها، می‌تواند منجر به افزایش و رشد داده‌ها و اطلاعات شود. لذا در این رشد فزاینده داده‌ای، ممکن است که داده‌های برون هشته نیز وجود داشته باشند که منجر به افزونگی، کاهش پیوندها بین ماشین‌های متصل به اینترنت و همچنین بروز موارد فنی مانند وجود نویز و تخریب داده‌های ارسالی، کاهش نرخ خطای بیت و کاهش گذردهی شود. لذا می‌بایست این داده‌های برون هشته را شناسایی و تحقیقاتی راجع به آن انجام داد.

برای تشخیص داده‌های برون‌هشته در اوایل قرن ۱۹ مطالعاتی انجام شده است و به‌مرور تکنیک‌های مختلفی برای تشخیص آن‌ها توسعه یافته‌اند، برخی از این تکنیک‌ها برای استفاده در یک حوزه کاربردی خاص و برخی دیگر عمومی هستند. معمولاً اکثر الگوریتم‌های تشخیص داده‌های برون‌هشته از معیارهایی برای تعیین برون هشته بودن نقاط داده استفاده می‌کنند، مانند روش‌های مبتنی بر توزیع، مبتنی بر خوشه‌بندی، مبتنی بر فاصله، مبتنی بر چگالی و یا مبتنی بر یادگیری ماشین. هر نقطه داده در طیف داده‌های معمولی تا نویز و در نهایت ناهنجاری قرار می‌گیرد (یانگ و همکاران^۱، ۲۰۱۱)؛ بنابراین برون هشته‌ها امتیاز بیشتری نسبت به نویز خواهند داشت؛ اما این یک عامل تشخیص بین دو تعریف نیست و این علاقه تحلیل‌گر است که تفاوت بین ناهنجاری و نویز را تعریف و تنظیم کند (گوپتا و همکاران^۲، ۲۰۱۴). برخی محققان از اصطلاح برون‌هشته ضعیف و برون‌هشته قوی برای تشخیص بین نویز و ناهنجاری استفاده می‌کنند. برون‌هشته‌ها، الگوهای داده‌ای هستند که با رفتار معمول داده‌ها مطابقت ندارند. داده‌ها دارای دو ناحیه معمولی هستند و بیشتر مشاهدات در این دو ناحیه قرار دارند و نقاط دور از این ناحیه‌ها، همان برون‌هشته‌ها هستند. تشخیص ناهنجاری یکی از مهم‌ترین مراحل پردازش داده‌ها است که در اکثر زمینه‌های داده‌کاوی برای مراحل پیش‌پردازش در نظر گرفته می‌شود (احمد و همکاران^۳، ۲۰۱۶). همیشه باید داده‌هایی که به منظور پردازش وارد یک محیط شبکه‌ای می‌شوند را واری و بازبینی کرد. همیشه این احتمال وجود دارد که داده‌ها دارای مقادیر غیرعادی باشند. موارد غیرعادی شامل مقادیر گم‌شده و مقادیر برون‌هشته هستند (سای و همکاران^۴، ۲۰۱۶). همواره قبل از هرگونه تحلیل آماری بر روی داده‌ها، می‌بایست چاره‌ای در مورد مقادیر برون‌هشته اندیشید. مجموعه داده‌هایی که دارای اندازه‌های غیرعادی یا غیرمعمول در یک متغیر واحد (تک متغیری) یا در ترکیبی از متغیرها (چند متغیری) هستند، برون‌هشته نامیده می‌شوند (سای و همکاران، ۲۰۱۶). داده‌های برون‌هشته اغلب سه یا بیش از سه واحد انحراف معیار از میانگین مربوطه به خودشان، فاصله دارند که به دلیل مشکلات احتمالی در ابزار اندازه‌گیری، شیوه ثبت یا ضبط پاسخ‌ها یا عضویت شرکت‌کنندگان در جامعه‌ای که فرض می‌شود از آن نمونه‌گیری انجام شده است، ناشی می‌شود. وجود داده‌های برون‌هشته می‌تواند ناشی از نتایج تحلیل باشد به‌گونه‌ای که فرض شود از آن نمونه‌گیری انجام شده است. وجود داده‌های برون‌هشته می‌تواند نتایج تحلیل را به شیوه‌ای نامطلوب تحت تأثیر قرار دهد و آن را تحریف کند.

-
1. Yang et al
 2. Gupta et al
 3. Ahmed et al
 4. Tsai et al

پیشینه پژوهش

(مارکو و همکاران^۱، ۲۰۱۴)، یک روش مناسب با RNN^۲ برای مجموعه داده‌های چند متغیره بزرگ ارائه کرده‌اند. در واقع RNN یک نوع MLP است که تعداد نورون‌های ورودی و خروجی یکسانی دارد و دارای سه لایه پنهان است. هدف این روش باز تولید نقاط ورودی در لایه خروجی با حداقل خطای بازسازی، بعد از فشرده‌سازی از طریق لایه‌های پنهان است. (که شامل تعداد گره‌های کمتری از لایه ورودی و خروجی است). این نقاط را می‌توان به عنوان داده‌های برون‌هسته در نظر گرفت. معیار برون‌هسته بودن بر اساس میانگین خطای بازسازی، به عنوان علامت‌گذاری جدید استفاده می‌شود. روش بهبود یافته RNN یک مدل شبکه عصبی auto-associative است که از میانگین مربع خطا به منظور معیار برون‌هسته بودن استفاده می‌کند. این روش برای تشخیص داده برون هشته در داده‌های ارتعاشی برای تشخیص تقلب در واحد مکانیکی به طور هم‌زمان، مورد استفاده قرار گرفته است.

(لیو و فرن^۳، ۲۰۱۲) روش CISO^۴ را پیشنهاد کرده‌اند. در این روش از نمونه‌های برجسب‌دار مشکوک که با روش‌های بدون ناظر شناسایی شده‌اند، استفاده می‌شود. در واقع، مجموعه به صورت برجسب‌دار تعریف می‌شود؛ در حالی که قبلاً مجموعه برجسب‌گذاری شده، معادل مجموعه آموزش بوده است. بر اساس این ایده، الگوریتم CISO، (بر اساس پاسخ به این سؤال که چگونه می‌توان تشخیص داد که مجموعه، برجسب‌دار است؛ به طوری که تقریباً شامل تمام داده‌های برون‌هسته و حداقل درون رسته‌ها باشد؟) ارائه شده است. ابتدا نمونه‌ها به وسیله الگوریتم بدون ناظر در مخزن بر اساس تشخیص داده برون‌هسته طبقه‌بندی می‌شوند، سپس نمونه‌های مشکوک به صورت دستی انتخاب و برجسب زده می‌شوند و نمونه‌های باقیمانده برجسب درون‌هسته دریافت می‌کنند. در نتیجه تمام نمونه‌ها، در مخزن برجسب دارند و در مجموعه آزمایش استفاده می‌شوند. یک نقطه برش خوب باید مجموعه مشکوک حاوی تمام داده‌های برون‌هسته و تعداد کمی درون‌هسته را تولید کند، بنابراین تمام داده‌های برون‌هسته می‌توانند توسط متخصص، با کمترین تلاش ممکن تأیید شوند. ممکن است کاربر بخواهد روی اعتبار برجسب زدن کنترل داشته باشد؛ بنابراین روش Bag at a CISO پیشنهاد می‌شود. در این روش، داده‌های بدون برجسب ارائه شده، به عنوان مجموعه داده اصلی برای ساخت مجموعه آزمایش با توزیع یکان انتخاب می‌شوند و هدف به حداقل رساندن برجسب زدن دستی است.

(آلام و همکاران^۵، ۲۰۱۲) روش HAC^۶ را پیشنهاد کرده‌اند. در روش پیشنهادی آن‌ها، ازدحام ذرات طی مراحل مختلف برای شناسایی نقاط دور افتاده و خوشه‌های طبیعی مورد استنتاج قرار می‌گیرند و از یک روش عملی برای تولید سلسله‌مراتب خوشه‌ها استفاده می‌شود. هر سطح سلسله به عنوان نسلی از ازدحام عمل می‌کند. نسل اول کل ازدحام را شامل می‌شود. هر ذره از ازدحام به عنوان یک مرکز خوشه است، سپس، ازدحام به وسیله ادغام دو خوشه در هر نسل موفق به سمت تبدیل شدن به یک خوشه پیش می‌رود. مزیت این روش این است که ذرات در بهینه محلی به دام نمی‌افتند. در این روش، از رفتار جمعی ذرات برای خودسازمان‌دهی موقعیت آن‌ها با توجه به پیکربندی خوشه‌بندی صحیح نمایش داده شده، توسط داده‌ها استفاده می‌شود.

(باسرس و همکاران^۷، ۲۰۱۴) در این تحقیق روشی را با استفاده از رویکرد نقص‌گرا^۸ و چارچوب ریاضی و مجموعه‌ای از ابزار هدف، به منظور انجام آزمایش‌هایی برای تشخیص داده برون‌هسته و تعیین پارامترها ارائه کردند. در این روش آزمایش‌هایی انتخاب می‌شوند که اگر همه آن‌ها با محدودیت‌های مشخص اجرا شوند، شکست‌های یکنواخت، شناسایی می‌شوند. در نهایت

-
1. Marco et al
 2. Recurrent Neural Networks
 3. Liu and Fern
 4. Constructing Training set by Identification Suspected Outliers
 5. Alam et al
 6. Hierarchical Agglomerative Clustering
 7. Bossers et al
 8. Defect-oriented

کاهش عملکرد حاصل به عنوان ابزار شکست، می‌تواند نشانه‌ای از قابلیت اطمینان پایین باشد. بخش کلیدی مدل صفحه‌نمایش برون‌هسته نامیده می‌شود. صفحه‌نمایش برون‌هسته به عنوان ترکیبی از روش‌های تشخیص تعریف شده است و آزمایش و تنظیمات پارامترها روی آن اعمال می‌شود. این بدان معنا است که اگر یک روش برای شناسایی داده برون‌هسته بر روی آزمایش‌های مختلفی اعمال شود، همه این‌ها به عنوان صفحه‌نمایش در این مدل در نظر گرفته می‌شوند. از این رو، استفاده از روش‌های اکتشافی راه‌حل که قادر به مقابله با تعداد زیادی از صفحه‌نمایش برون‌هسته هستند، ضروری است. در این مدل تنها نتیجه روش تشخیص داده برون‌هسته در نظر گرفته شده است و از آنجایی که می‌تواند هر روش تشخیص، داده برون‌هسته را کنترل کند، هر کاربر این امکان را دارد که از روش مورد علاقه خود استفاده کند. یکی دیگر از اجزای مهم مدل، مجموعه هدف است. مجموعه هدف، زیرمجموعه‌ای از ابزارهایی است که باید توسط روش‌های تشخیص داده برون‌هسته، شناسایی شوند. این مجموعه هدف برای هدایت فرایند انتخاب آزمایش ضروری است. همچنین مجموعه پاداش معرفی می‌شود.

در تحقیق (کیوو و همکاران^۱، ۲۰۱۸) از یک روش تشخیص داده‌های برون‌هسته بر پایه شبکه عصبی کانولوشن و شبکه LSTM با Auto encoder ها استفاده شده است. این تحقیق بر پایه سری‌های زمانی سعی در تشخیص داده‌هایی دارد که از ثابت بودن مقدار سری زمانی تبعیت نمی‌کنند.

(زناتی و همکاران^۲، ۲۰۱۸) برای تشخیص داده‌های برون‌هسته از یک روش بر پایه شبکه‌های مقابله‌ای^۳ استفاده کرده است. این شبکه‌ها، توانایی تشخیص داده‌های برون‌هسته در دنیای واقعی را دارند، زناتی و همکارانش در این تحقیق، بر روی تصاویر کار کردند و سعی در تشخیص ناهنجاری از روی عکس برآمدند.

با توجه به این که مبحث شناسایی داده‌های برون‌هسته در محیط اینترنت اشیا به‌تازگی در جوامع علمی ظهور کرده است، لذا تحقیقات زیادی در این حوزه انجام نشده است. همچنین تا جایی که تحقیق شد، هیچ مطالعه داخلی در ایران بر روی این موضوع تا زمان نگارش این تحقیق، صورت نپذیرفته است.

در پژوهش (سوزا و همکاران^۴، ۲۰۱۵) به شناسایی داده‌های برون‌هسته در محیط اینترنت اشیا به کمک روش خوشه‌بندی K-Means مبتنی بر مه‌داده^۵ پرداخته شده است. شبیه‌سازی این تحقیق در محیط نرم‌افزاری NS-2 انجام گرفته است که معیار ارزیابی آن نیز AUC بوده است که نتایج قابل توجهی را نشان می‌دهد.

(در یان و همکاران^۶، ۲۰۱۷) مه‌داده‌ها در محیط اینترنت اشیا و مسائلی و مشکلاتی که به همراه داشته است، مورد بحث واقع شده است. یکی از مهم‌ترین بخش‌های این تحقیق، اشاره به وجود داده‌های برون‌هسته در ارسال و دریافت داده‌ها و ذخیره‌سازی بر روی پایگاه‌های داده‌ای است که افزونگی را افزایش می‌دهد و همین‌طور باعث کاهش نرخ خطای بیت و گذردهی می‌شود. (کاراکوچ و همکاران^۷، ۲۰۱۶) نیز یک مطالعه کلی بر روی کیفیت داده‌ها در استفاده از اینترنت اشیا و نوع داده‌های ارسالی و دریافتی از طریق اشیا، انجام دادند. کیفیت داده از این نظر دارای اهمیت است که اگر در آن داده برون‌هسته‌ای رخ دهد، منجر به ایجاد افزونگی و کاهش پیوندهای بین گره‌های موجود در بستر شبکه اینترنت اشیا می‌شود. در یک تحقیق مروری دیگر که توسط (مهدوی‌نژاد و همکاران^۸، ۲۰۱۷) ارائه شده است، استفاده از روش‌های یادگیری ماشین در تحلیل داده‌های اینترنت اشیا، استفاده از انواع شبکه‌های عصبی، روش‌های تکاملی و هوش ازدحامی و همین‌طور درخت‌های تصمیم به همراه روش‌های آشوب‌ناک، مدنظر واقع شده است. به‌صورت کلی، دسته‌بندی روش‌های تحلیل داده‌ها در این تحقیق با روش‌های مختلف انجام گرفته است که شامل طبقه‌بندی، خوشه‌بندی و رگرسیون است که دسته‌های مختلف الگوریتمی مانند بیزین^۹، نزدیک‌ترین

-
1. Kieu et al
 2. Zenati et al
 3. Generative adversarial networks
 4. Souza et al
 5. Big Data
 6. Yan et al
 7. Karkouch et al
 8. Mahdavejad et al
 9. Bayesian

همسایه K^1 ، ماشین بردار پشتیبان^۲، الگوریتم ژنتیک^۳ و شبکه‌های عصبی در آن قرار دارند. (شادرو و همکاران^۴، ۲۰۱۵) نیز مطالعه‌ای سیستماتیک بر روی پردازش داده‌ها به‌خصوص مه‌داده‌ها و شناسایی داده‌های برون‌هسته در محیط اینترنت اشیا انجام داده‌اند. استفاده از الگوریتم‌های داده‌کاوی و ماشین یادگیری به پیش‌بینی سگته مغزی در بیماران همودیالیزی مورد پژوهش قرار گرفته است (رضاپور، ۱۳۹۹). روش پژوهش تحقیق حاضر، استفاده از الگوریتم‌های با ناظر داده‌کاوی و با محوریت درخت تصمیم بوده است. نتایج نشان داد ریسک ابتلا به سگته مغزی در بیماران، دسترسی عروق آن‌ها با ایجاد کاتتر قبل از فیستول صورت گرفته تا ۸۴/۲۱ درصد بوده است. (دانشفراز و همکاران^۵، ۲۰۲۱) با استفاده از شبکه عصبی و روش ماشین‌بردار پشتیبان به بررسی و پیش‌بینی پارامترهای هیدرولیکی جریان سیال در دراپ‌های قائم و همچنین مقطع انقباض یافته پرداخته‌اند. نتایج نشان از پیش‌بینی دقیق این پارامتر دارند.

شناسایی داده‌های برون‌هسته از مجموعه مه‌داده‌ها، نیاز به روش‌هایی سریع و با خطای کمتر دارد. از این رو، نیاز است تا یک ارزیابی مبتنی بر شبکه عصبی عمیق با دقت خوب برای داده‌های برون‌هسته به کار رود. لذا استفاده از روش‌های خانواده شبکه‌های عصبی، جز روش‌هایی است که می‌توان با سرعت بالا، آموزش و آزمون داده با هدف شناسایی داده‌های برون‌هسته را انجام داد. نوآوری این تحقیق در آن است که بتوان داده‌های برون‌هسته را با استفاده از یادگیری عمیق و شبکه عصبی چندین لایه‌ای، شناسایی کرد تا کاهش خطا در این شبکه برای عملیات مختلف و نظارت، تسهیل شود. هدف کلی از انجام این تحقیق، ارائه یک روش مؤثر با دقتی بالا جهت شناسایی داده‌های برون‌هسته، استفاده از یادگیری عمیق در شناسایی داده‌های برون‌هسته و بهبود نویز، ارسال و دریافت داده‌ها با حذف داده‌های نامربوط است. همچنین استفاده از روش‌های شبکه عصبی در شناسایی، تشخیص، درمان و بهبود بیماران می‌تواند در درمان بیماری‌ها سرعت ایجاد کرده و از نظر آماری نیز می‌تواند نتایج دقیق‌تر و سریع‌تری را ارائه دهد.

روش پژوهش

شبکه عصبی عمیق

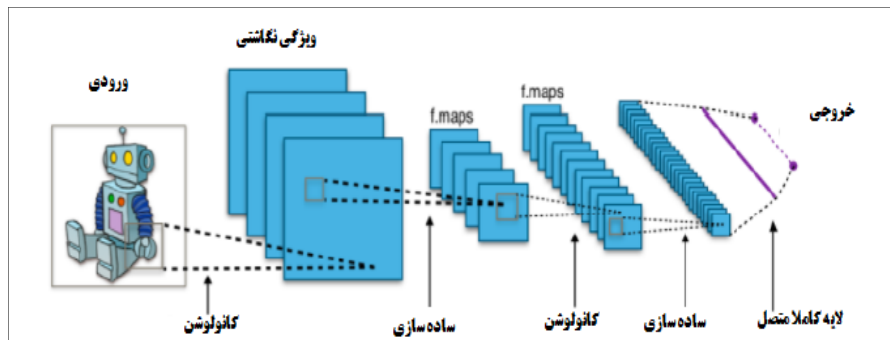
در این قسمت از پژوهش حاضر به بررسی شبکه عصبی مورد استفاده می‌پردازیم:

شبکه عصبی کانولوشن (CNN)^۶

شبکه عصبی کانولوشن یک FNNs^۷ عمیق است که سلسله مراتبی از ویژگی‌های ثابت را به طور خودکار یاد می‌گیرد. دارای ویژگی‌ها و خاصیت‌های متمایزی هستند، فیلتر در شبکه عصبی کانولوشن برای تبدیل‌های یکسان در مکان‌های مختلف استفاده می‌شود. شبکه کانولوشن اغلب برای پردازش داده‌هایی استفاده می‌شود که یک ارتباط فضایی و یا شبکه مانند دارند. (سرمانت و همکاران^۸، ۲۰۱۱)، (لسون و همکاران^۹، ۱۹۹۴). کانولوشن از یک لایه ورودی و یک لایه خروجی با چند لایه پنهان در میان آن‌ها تشکیل شده است که در آن لایه‌های پنهان می‌توانند به لایه drop-out کاملاً متصل^{۱۰} شوند (ماساکازو و همکاران^{۱۱}، ۲۰۰۳). در شبکه‌های کانولوشن عمیق، عملیات Conv انجام می‌پذیرد و هر عملیات Conv یک هسته دارد. هسته یا فیلتر، یک ماتریس کوچک‌تر از تصویر اصلی است. در طی عملیات کانولوشن، فیلتر روی تصویر حرکت می‌کند تا یک ویژگی خاص را کشف کند. پس از استفاده از کانولوشن بر روی تصویر ورودی با استفاده از یک فیلتر، خروجی در قالب یک

1. K-Nearest Neighbor
2. Support Vector Machine (SVM)
3. Genetic Algorithm (GA)
4. Shadroo et al
5. Daneshfaraz et al
6. Convolutional Neural Network
7. Feed Forward Neural Network
8. Sermanet et al
9. Lecun et al
10. Fully connected
11. Masakazu et al

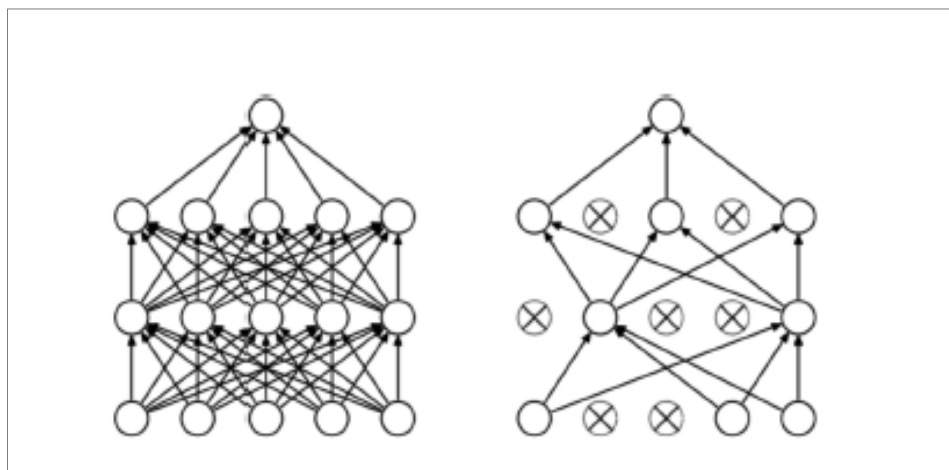
نقشه ویژگی تولید می‌شود. اندازه نقشه ویژگی کوچک‌تر از ورودی است؛ زیرا فیلتر باید در ورودی باشد. برای حفظ ابعاد، می‌توان از پدینگ استفاده کرد، یعنی ورودی را با محفظه احاطه کرد. پدینگ می‌تواند صفر یا مقادیر لبه باشد. پدینگ در CNNs برای حفظ اندازه نقشه‌های ویژگی استفاده می‌شود (زوو^۱، ۲۰۱۸). شکل ۱، یک مثال گرافیکی از شبکه CNN است.



شکل ۱: مرور کلی شبکه عصبی کانولوشن (CNN)

لایه dropout

Dropout به این معنی است که برخی از واحدها (پنهان و قابل مشاهده) در شبکه عصبی به طور موقت از بین می‌روند، یعنی خارج شبکه هستند. واحدهای رها شده Dropout در انتقال و ارسال پیشین شرکت نمی‌کنند. این باعث می‌شود که شبکه عصبی هر بار که یک ورودی ارائه می‌شود، معماری متفاوتی را انتخاب کند. Dropout ترکیب سازگاری پیچیده واحدها در شبکه عصبی را کاهش می‌دهد، از آنجا که یک واحد نمی‌تواند به طور کامل بر واحدهای خاص دیگر تکیه کند، ممکن است حذف شوند و این باعث قوی‌تر شدن شبکه می‌شود. همان‌طور که در شکل ۲ مشخص است، انتخاب واحدها برای Dropout تصادفی است و در ساده‌ترین حالت، بدان معنی است که هر واحد با یک احتمال ثابت p در شبکه حفظ می‌شود. معمولاً احتمال p به مقدار 0.5 تنظیم می‌شود؛ زیرا به نظر می‌رسد این مقدار نزدیک به بهینه است و برای انواع شبکه‌ها و مشکلات بینایی کامپیوتر استفاده می‌شود.

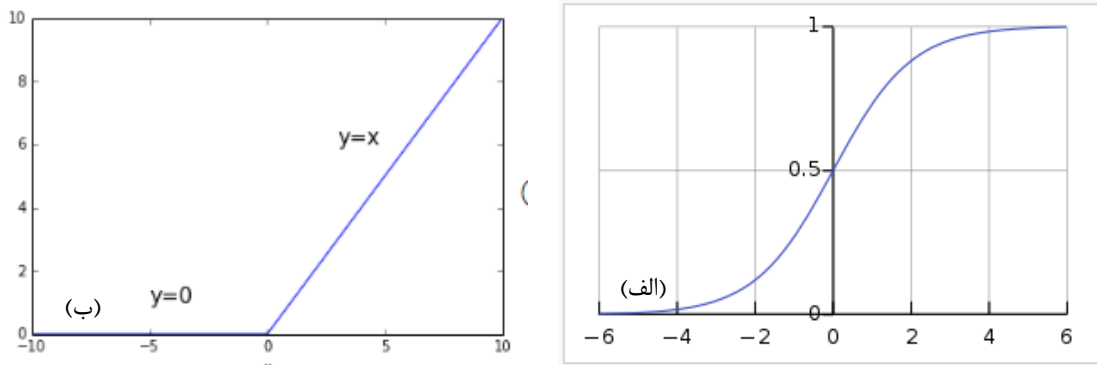


شکل ۲: شکل شماتیک dropout

شبکه عصبی در لایه پنهان خود از یک تابع غیرخطی مانند تابع سیگموئید^۱ و تابع Relu که در روابط ۱ و ۲ نشان داده شده است، استفاده می کند. همچنین شکل های ۳ (الف و ب) به ترتیب نمودار توابع سیگموئید و Rule را نشان می دهد.

$$S(t) = \frac{1}{1 + e^t} \tag{۱}$$

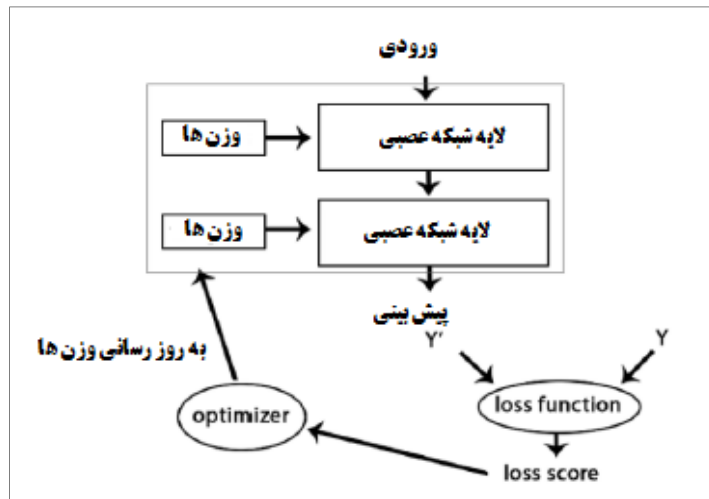
$$f(x) = \max(0, x) \tag{۲}$$



شکل ۳: نمودار توابع (الف) سیگموئید (ب) Rule

روند آموزش

فرایند یادگیری عمیق، بر اساس مرحله آموزش است که پایه و اساس آن محسوب می شود. هدف از آموزش، این است که مدل را با آموزش نمونه هایی از مجموعه داده ها به دست آوریم. پس از آموزش، ما می خواهیم با آزمایش مدل بر روی داده های غیرآموزشی، چیزی که مدل آموخته است را اعتبارسنجی و ارزیابی کنیم. پیش بینی یک کلاس صحیح از داده های آزمایش، نشانگر خوب بودن مدل است. در مدل پیشنهادی حتماً باید از آموخته هایمان برای اعتبارسنجی بهره ببریم. کلیات روند آموزش در روش پیشنهادی همانند شکل ۴ است.



شکل ۴: مرور کلی بر روند آموزش

1. Sigmoid function

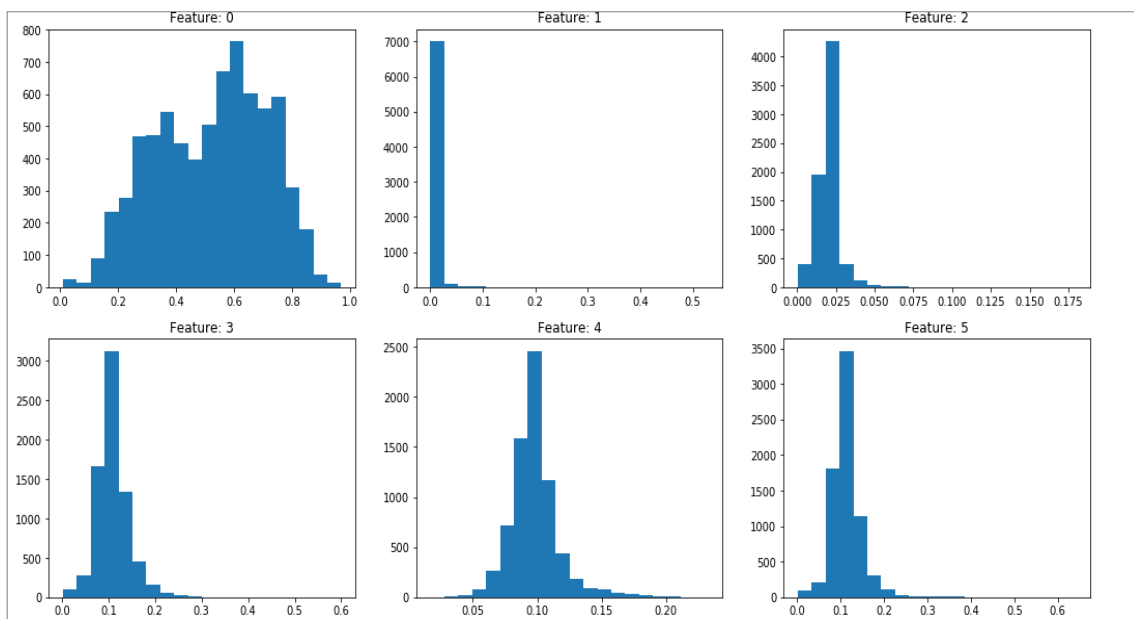
تابع از دست رفتگی یا تابع زیان

در یادگیری عمیق، هدف از آموزش، افزایش عملکرد کلی مدل، با توجه به مجموعه آزمون تعریف شده است. بهینه‌سازی می‌تواند برای اندازه‌گیری عملکرد دشوار باشد؛ به همین دلیل تابع از دست رفتگی^۱ با امید به اینکه عملکرد کلی مدل را بهبود بخشد، کاهش پیدا می‌کند. تابع زیان مقدار کمیت مدل را از مقادیر واقعی تعیین می‌کند. توابع زیان زیادی وجود دارد که می‌توانند برای متغیرهای هدف خاص استفاده شوند. Cross entropy احتمالاً محبوب‌ترین است. رابطه کلی Cross entropy را می‌توان به صورت زیر در نظر گرفت.

$$H(p, g) = - \sum_x P(x) \log_p(P(x)) \quad (۳)$$

مجموعه داده

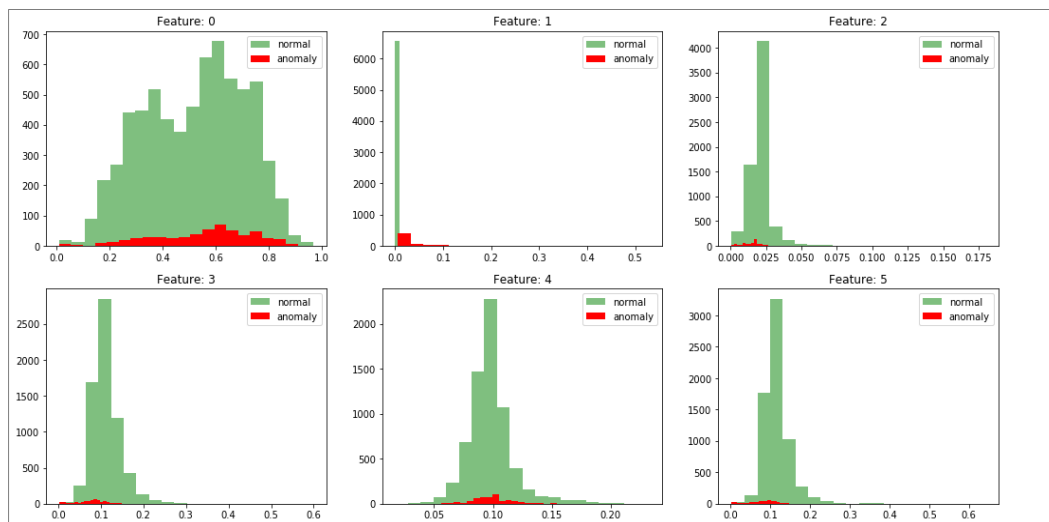
مجموعه داده استفاده شده در پژوهش حاضر، مجموعه داده پزشکی کم‌کاری تیروئید^۲ است که یکی از مجموعه داده‌های معروف برای تشخیص برون‌هسته‌ای است. این مجموعه داده با مقدار برون‌هسته‌ای ۲ درصد در آدرس^۳ قابل دسترسی است. این مجموعه داده شامل سه کلاس شامل کلاس عادی، عملکرد فوق‌العاده و عملکرد غیرطبیعی (برون‌هسته‌ای) برای آزمایش تشخیص برون‌هسته‌ای است. قسمتی از کلاس‌های این مجموعه داده در شکل ۵ قابل مشاهده است.



شکل ۵: قسمتی از کلاس‌های مجموعه داده تشخیص برون‌هسته‌ای کم‌کاری تیروئید

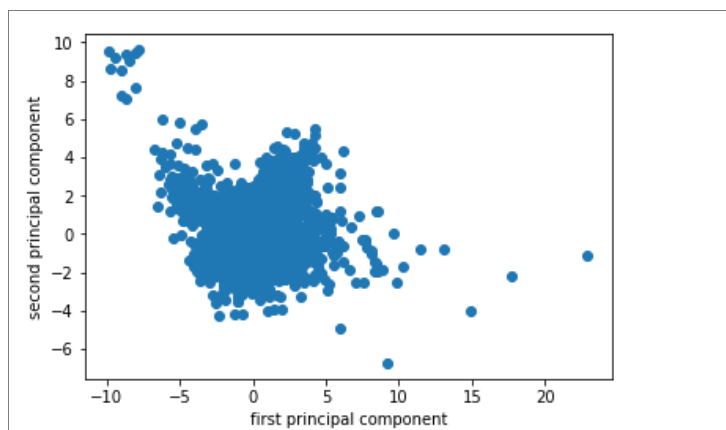
در شکل ۶ قسمتی از کلاس‌های مجموعه داده تشخیص برون‌هسته‌ای کم‌کاری تیروئید دیده می‌شود.

1. Loss function
2. Hypothyroidism
3. http://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/semantic/Anthyroid/Anthyroid_02_v01.html



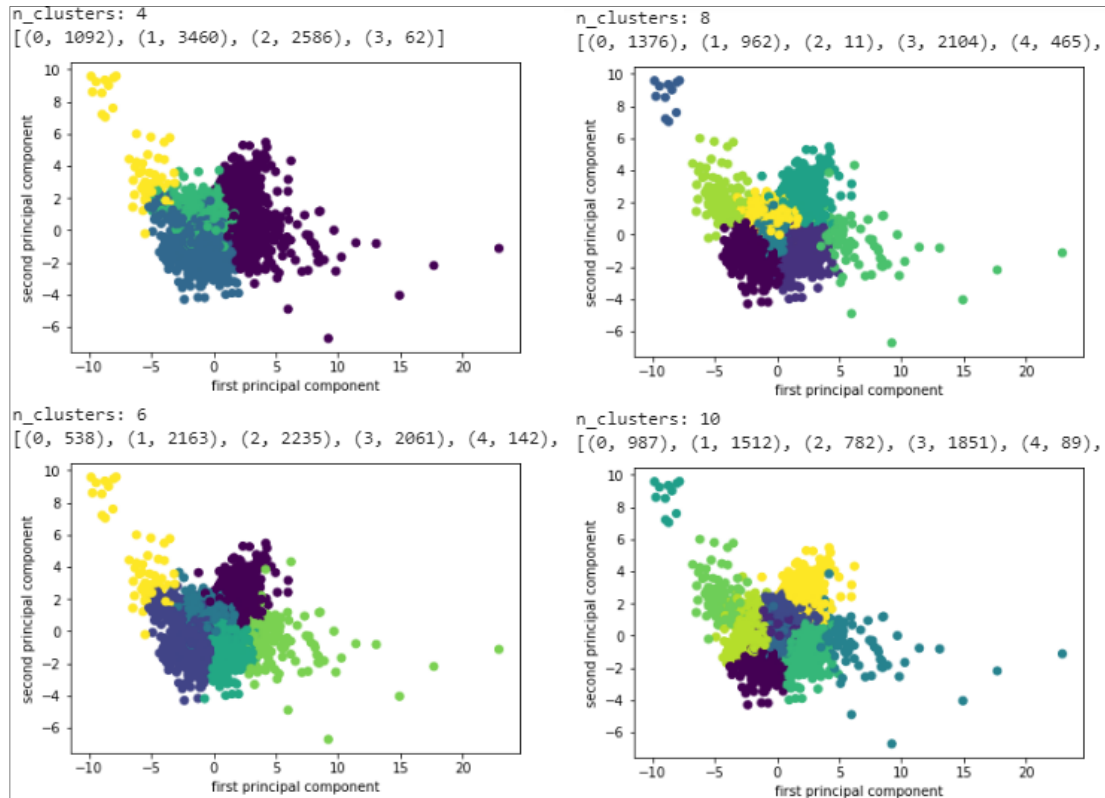
شکل ۶: نمایشی از ویژگی‌های مجموعه داده و برون‌هشته‌ای موجود

در حالت کلی مجموعه داده شامل ۷۲۰۰ سطر با ۳۳۰ ویژگی است. از این بین برای آموزش، می‌توان کلاس برون‌هشته‌ای را به عنوان y و بقیه را به عنوان X به شبکه عصبی عمیق یاد داد. اگر از روش تحلیل مؤلفه اصلی^۱ استفاده کنیم، پردازش داده‌ها آسان‌تر خواهد بود. می‌دانیم روش تحلیل مؤلفه یک تبدیل برداری برای کاهش ابعاد در مجموعه داده‌ها است. با توجه به ویژگی‌های بسیار زیاد این مجموعه داده از این روش برای کاهش ابعاد استفاده می‌کنیم. شکل ۷ کاهش ابعاد یافته مجموعه داده را نشان می‌دهد.



شکل ۷: کاهش یافته مجموعه داده در فضای برداری

می‌توان از تکنیک خوشه‌بندی برای خوشه‌سازی مجموعه داده استفاده کرد. شکل ۸ کلیات خوشه‌بندی با تعداد خوشه‌های ورودی برای مجموعه داده را نشان می‌دهد.



شکل ۸: خوشه‌سازی با تعداد خوشه‌ها در مجموعه داده تشخیص برون‌هشته‌ای

مدل پیشنهادی

کلیات کار، استفاده از شبکه عصبی عمیق (DNN) برای تشخیص داده‌های برون‌هشته‌ای است. همان‌طور که قبلاً ذکر شد، شبکه‌های عصبی نیازی به پیش‌پردازش‌های زیادی ندارند؛ زیرا ذاتاً هر لایه در شبکه‌های عصبی از لایه‌های قبلی یاد می‌گیرد و با استفاده از بازگشت خطاها به عقب، این خطاهای یادگیری را کاهش می‌دهد. مجموعه داده آموزشی و آزمایشی در مدل پیشنهادی بارگذاری می‌شوند، سپس به یک مدل ۱۴ لایه‌ای از شبکه عصبی عمیق وارد می‌شوند. پیاده‌سازی بر اساس تنسورفلو با پشتیبانی^۱ کراس بر روی ابزار گوگل کولب است. خلاصه‌ای از مدل پیشنهادی در شکل ۹ نمایش داده شده است. شبکه‌های عصبی، سیستم‌ها و روش‌های محاسباتی جدیدی برای یادگیری ماشین، نمایش دانش و در نهایت استفاده از دانش به دست آمده برای پیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده هستند. ایده اصلی چنین شبکه‌هایی از نحوه کارکرد سیستم عصبی زیستی، برای پردازش داده‌ها و اطلاعات برای یادگیری و ایجاد دانش الهام گرفته شده است. عنصر کلیدی این ایده، ایجاد ساختارهای جدید برای سیستم پردازش اطلاعات است. این سیستم از تعداد زیادی عناصر پردازشی فوق‌العاده به هم پیوسته با نام نورون تشکیل شده است که برای حل یک مسئله با هم هماهنگ عمل می‌کنند و اطلاعات را توسط سیناپس‌ها (ارتباطات الکترومغناطیسی) منتقل می‌کنند. در این شبکه‌ها، اگر سلولی آسیب ببیند، بقیه سلول‌ها می‌توانند فقدان آن را جبران کرده و در بازسازی آن سهیم باشند. این شبکه‌ها قادر به یادگیری هستند. برای مثال، با اعمال سوزش روی سلول‌های عصبی لامسه، سلول‌ها یاد می‌گیرند که به طرف جسم داغ نروند و با این الگوریتم سیستم می‌آموزد که خطای خود را اصلاح کند. یادگیری در این سیستم‌ها به صورت تطبیقی انجام می‌شود، یعنی با استفاده از مثال‌ها، وزن سیناپس‌ها به گونه‌ای تغییر می‌کند که با دادن ورودی‌های جدید، سیستم پاسخ درستی تولید کند. در مسائل پیچیده‌تر که نیاز به طبقه‌بندی غیرخطی دارند، از شبکه‌های عصبی چندلایه استفاده می‌کنند. خروجی‌های هر لایه ورودی، لایه بعدی می‌شوند. در این مدل با وارد

کردن یک سری داده‌های برون‌هشته که شامل بیماری تیروئید است، یادگیری عمیق که یادگیری ماشینی نیز نام دارد، داده‌ها را شناسایی کرده و آن‌ها را انتخاب می‌کند. در این حالت علائم بیماری تیروئید را مشخص کرده و سپس با یادگیری عمیق ماشینی شروع به بررسی داده‌های برون‌هشته می‌کنیم که اولین علائمی که نرم‌افزار مشخص می‌کند و دارای پاسخ مثبت باشد، علائم دیگر را بررسی کرده و در نهایت در صورت مثبت بودن پاسخ‌ها بیماری شخص تشخیص و سپس شروع به درمان آن می‌کند.

Layer (type)	Output Shape	Param #
dense_52 (Dense)	(None, 6)	1986
dense_53 (Dense)	(None, 64)	448
dense_54 (Dense)	(None, 128)	8320
dropout_21 (Dropout)	(None, 128)	0
dense_55 (Dense)	(None, 64)	8256
dropout_22 (Dropout)	(None, 64)	0
dense_56 (Dense)	(None, 32)	2080
dropout_23 (Dropout)	(None, 32)	0
dense_57 (Dense)	(None, 16)	528
dropout_24 (Dropout)	(None, 16)	0
dense_58 (Dense)	(None, 8)	136
dropout_25 (Dropout)	(None, 8)	0
dense_59 (Dense)	(None, 6)	54
dense_60 (Dense)	(None, 1)	7

شکل ۹: خلاصه‌ای از مدل پیشنهادی پیاده‌سازی شده بر روی تنسورفلو با پشتیبانی کراس

در این تحقیق، با توجه به پیاده‌سازی که تاکنون صورت گرفته است؛ از یک شبکه عصبی عمیق ۱۴ لایه استفاده شده است. ترکیب لایه‌ها در این شبکه ۱۴ لایه به این صورت است: لایه ورودی، سه لایه پشت سرهم شبکه عصبی DNN، لایه رها کردن^۱، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN و لایه سیگموئید برای انتخاب کلاس متناسب استفاده شده است. با توجه به محاسبات بسیار زیاد از GPU بهره برده شده است.

1. Drop-out

لایه‌های مدل پیشنهادی

لایه اول تا لایه سوم: لایه DNN

لایه اول تا سوم در مدل پیشنهادی یک شبکه عصبی عمیق با ورودی ۳۳۰ (به تعداد ویژگی‌های مجموعه داده) و تعداد کل پارامترهای لایه‌های اول ۱۹۸۶، ۴۴۸، ۸۳۲۰ به دلیل شکل خروجی‌های این لایه‌ها است. تابع فعال‌سازی، یکی از بهترین تابع‌های آزمایش شده برای شبکه عصبی است و از تابع غیرخطی Relu استفاده شده است.

لایه چهارم: لایه رها کردن

لایه رها کردن برای جلوگیری از انتقال نویزهای تصادفی در مدل شبکه عصبی استفاده می‌شود؛ یعنی شبکه عصبی در آموزش اطلاعات بسیار خوب عمل می‌کند و به نوبه خود، نوسانات تصادفی در آن داده‌ها را نیز یاد می‌گیرد، به عبارت دیگر، شبکه عصبی دقت بسیار بالایی در داده‌های آموزشی دارد؛ اما دقت پیش‌بینی در واقعیت کاهش می‌یابد، این نویزهای یاد گرفته شده، می‌توانند در تفسیر نادرست داده‌های واقعی مؤثر بوده و اشتباه شبکه را افزایش دهد. تکرار این نویزها می‌تواند در هنگام عبور داده‌ها از شبکه نیز مؤثر بوده و به طور کامل در فرایند یادگیری مشارکت داشته باشند، از آنجایی که وزن‌دهی نورون‌ها و تغییرات آن‌ها به نورون‌های همسایه نیز تأثیرگذار است، ادامه این کار باعث اختلال کلی بر روی داده‌های پیش‌بینی می‌شود. Dropout تکنیکی است که از «over-fit» شدن شبکه جلوگیری می‌کند. همان‌طور که از اسمش پیداست، در طول یادگیری، برخی از نورون‌ها به صورت تصادفی رها می‌شوند. این بدان معنی است که یادگیری روی معماری‌های مختلف با مجموعه نورون‌های متفاوت صورت می‌گیرد. در مدل پیشنهادی، پس از اولین لایه maxpool از یک لایه Drop-out با مقدار ۰/۲ برای حذف ورودی لایه DNN با احتمال ۰/۲ و ادامه یادگیری استفاده شده است. این لایه تأثیری بر ورودی ندارد، فقط برای جلوگیری از overfit و یادگیری بهتر استفاده می‌شود.

لایه پنجم تا لایه سیزدهم: لایه DNN و رها کردن پشت سرهم

از لایه پنجم تا لایه سیزدهم با مقادیر متفاوت، لایه DNN و لایه رها کردن پشت سرهم تکرار می‌شوند، البته مقادیر پارامترهای خروجی و ورودی لایه‌های DNN با لایه‌های قبل متفاوت است. برای تمامی مقادیر لایه رها کردن مقدار ۰/۲ در نظر گرفته شده است، فقط در لایه هشتم با توجه به افزایش پارامترها، نرخ لایه رها کردن نصف در نظر گرفته شده است تا نصف یاد گرفته شده‌ها تا یادگیری بهتری داشته باشیم.

لایه کاملاً متصل آخر

یک لایه کاملاً متصل^۱ برخلاف یک لایه DNN، لایه‌ای است که به تمام نورون‌های لایه‌های قبلی متصل است، وظیفه اصلی لایه fc ترکیب ویژگی محلی در لایه پایین با ویژگی محلی در لایه‌های بالا است. لایه کاملاً متصل ورودی، لایه قبلی را به عنوان یک پارامتر، تعداد کلاس‌ها برای پیش‌بینی (۳ کلاس خروجی) و یک تابع فعال در شکل یک تابع سیگموئید (با توجه به دسته‌بندی سه کلاسه) را بر روی شبکه اعمال می‌کند. این لایه، لایه‌های ورودی را ترکیب کرده و پس از محاسبه کلاس جمله، آن را به خروجی هدایت می‌کند.

لایه آخر در مدل پیشنهادی، یک لایه کاملاً متصل برای مشخص کردن کلاس خروجی، مدل پیشنهادی و انتخاب برون‌هسته‌ای از ۳ کلاس ورودی است. این لایه ورودی‌های کلی یاد گرفته شده در لایه‌های قبل را جمع کرده و به کلاس متناسب آن اختصاص می‌دهد. در مدل پیشنهادی بهینه‌ساز مورد استفاده بهینه‌ساز Adam بود. جدول ۱ کلیات و نحوه عملکرد روش پیشنهادی را نشان می‌دهد.

جدول ۱: نحوه عملکرد روش پیشنهادی تحقیق حاضر

گامها و ورودی و خروجی	پارامترهای ورودی و خروجی
ورودی	داده‌های مجموعه داده تشخیص برون هشته‌ای کم کاری تیروئید (تعداد کل برون هشته‌ای ۰/۲)
خروجی	طبقه‌بندی برای تشخیص برون هشته‌ای
گام ۱	بارگذاری کل شکل‌های مجموعه داده
گام ۲	مجموعه آموزشی شامل ۵۷۶۰ با ۳۳۰ ویژگی در سه کلاس
گام ۳	مجموعه داده آزمایشی شامل ۱۴۴۰ بدون برچسب برون هشته‌ای
گام ۴	لایه DNN
گام ۵	لایه DNN
گام ۶	لایه DNN
گام ۷	لایه رها کردن با مقدار ۰/۲
گام ۸	لایه DNN
گام ۹	لایه رها کردن با مقدار ۰/۲
گام ۱۰	لایه DNN
گام ۱۱	لایه رها کردن با مقدار ۰/۵
گام ۱۲	لایه DNN
گام ۱۳	لایه رها کردن با مقدار ۰/۲
گام ۱۴	لایه DNN
گام ۱۵	لایه رها کردن با مقدار ۰/۲
گام ۱۶	لایه DNN
گام ۱۷	لایه رها کردن با مقدار ۰/۲
گام ۱۸	لایه آخر dense برای تشخیص علامت با فعال‌ساز Sigmoid
گام ۱۹	تشخیص برون هشته‌ای از روی مجموعه داده ورودی
گام ۲۰	پایان

ابزارهای مورد استفاده برای پیاده‌سازی

1. Jupyter notebook
2. Google-collab
3. Tensorflow
4. Keras
5. Pipeline
6. Python 3
7. GPU-TPU
8. Matplotlib

Jupyter Notebook به عنوان محیط توسعه برای این آزمایش استفاده شد. زبان برنامه‌نویسی، پایتون است. از کتابخانه یادگیری عمیق Tensorflow استفاده شده است. تنسورفلو، یک کتابخانه نرم‌افزاری متن‌باز برای یادگیری ماشین در انواع مختلف وظایف مفهومی و زبان است که در حال حاضر توسط ۵۰ تیم تحقیقاتی و محصولات مختلف گوگل از جمله تشخیص گفتار، جیمیل، گوگل فوتوز و جستجو استفاده می‌شود که بسیاری از آن‌ها سابقاً از دیست‌بلیف استفاده کرده بودند. در این پایان‌نامه از ابزار

رایگان در دسترس عموم قرار گرفته و امکان موازی‌سازی بر روی پردازنده های گرافیکی را فراهم می‌کند، استفاده شده است. به صورت Jupyter notebook و پکیج Tensorflow در آن نصب شده است و به صورت

نتایج و بحث

تمام تحقیقات ارائه شده توسط محققان در زمینه های مختلف، در نهایت باید با یک سری از معیارها ارزیابی شوند. به منظور ارزیابی مدل پیشنهادی از معیارهای ارزیابی: درستی، مقدار تابع زیان و پیش بینی های سایر مدل ها استفاده شده است که در این فصل این معیارها بررسی شده و نتایج روش پیشنهادی بر اساس آن ها مورد ارزیابی قرار می‌گیرد.

معیارهای ارزیابی

در شبکه های عصبی عمیق با استفاده از توابع غیرخطی و لایه های لازم، کل شبکه آموزش داده می‌شود. معیار صحت^۱ بیان می‌کند که چند درصد از این داده ها، درست طبقه بندی شده اند و روابط ارزیابی در معادلات ۴ و ۵ ارائه شده است.

جدول ۲: نحوه به دست آوردن معیارهای ارزیابی

وضعیت واقعی \ وضعیت پیش بینی شده	True	False
	True-Positive	False-Positive
Positive	True-Positive	False-Positive
Negative	False-Negative	True-Negative

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (۴)$$

$$Precision \text{ and } Recall = \frac{TP_i}{TP_i + FP_i} \quad (۵)$$

تجزیه و تحلیل یافته ها

برای ارزیابی روش پیشنهادی کل مجموعه داده ورودی بر روی گوگل کولب و در فضای گوگل درایو بارگذاری می‌شود. علاوه بر مدل پیشنهادی، پنج مدل دیگر بر روی زبان پایتون و کراس آزمایش شده است. در هر شش مدل تعداد اجرا ۵۰ تکرار با شرایط یکسان خواهد بود تا مدل پیشنهادی با ۵ مدل دیگر مورد ارزیابی و بررسی قرار گرفته است. جدول ۳ پنج مدل متشکل از شبکه عصبی کانولوشن و شبکه بازگشتی LSTM و یک مدل از شبکه عمیق را با مدل پیشنهادی مقایسه کرده است. در هر ستون، یک مدل با تعداد لایه ها و ورودی های متناظرشان نمایش داده شده است. در اولین و دومین مدل مقایسه ای، از یک شبکه کانولوشنی ۱۱ لایه ای استفاده کردیم، در مدل اول ۳۲ فیلتر کانولوشن یک بعدی با maxpool و dropout های پشت سرهم و سپس لایه های کاملاً متصل داریم. مقدار صحت (درستی) مدل اول مقدار ۹۳/۰۵ را نشان می‌دهد، این مقدار درستی قابل قبول است. با این حال مقدار تابع زیان این مدل مقدار ۱/۱۸۷۷ را نشان می‌دهد که مقدار بالایی است، دلیل این مقدار را می‌توان چیدمان نادرست لایه ها دانست.

1. Accuracy

در مورد مدل کانولوشن دوم، چیدمان کمی تغییر پیدا کرده است و از ۲ لایه فیلتر کانولوشن ۳۲ تایی با هسته سه در سه بهره برده ایم. مشاهده می شود، صحت مقدار ۹۲/۶۳ را نشان داد و مقدار زیان همان مقدار ماند. می توان نتیجه گرفت که افزایش فیلتر کانولوشن لزوماً باعث بهبود صحت و درستی نخواهد شد.

در مورد مدل کانولوشن سوم، لایه ها کردن به طور مناسبی بین لایه فیلتر کانولوشن و لایه maxpooling استفاده شده است، در این مدل از سه لایه فیلتر کانولوشن استفاده شده است، مشاهده می شود، مقدار دقت در این مدل از دو مدل قبلی بالاتر شد و مقدار زیان بسیار کاهش یافت و به مقدار ۰/۲ تقریباً رسید که بسیار قابل قبول تر از دو مدل دیگر کانولوشن است. با این حال با مقایسه ۳ مدل کانولوشن و مدل شبکه DNN مشاهده می شود که برای این مسئله خاص، استفاده از کانولوشن مناسب به نظر نمی رسد. مدل های کانولوشن برای تصاویر بسیار خوب عملکرد دارند.

مدل چهارم، استفاده از یک مدل شبکه بازگشتی LSTM است، این مدل بهتر از مدل شبکه کانولوشن برای تشخیص برون هشته ای کار می کند، تعداد لایه های این مدل به مراتب از مدل های دیگر کمتر است، مشاهده شد که بالا بردن لایه ها الزاماً باعث بهبود درستی نخواهد شد، در مدل LSTM بعد از یک لایه تعبیه شده از لایه های Desne و رها کردن استفاده شده است، مشاهده می شود، این مدل به نسبت تمامی مدل ها صحت کمتری دارد. البته این امر را می توان از بازگشتی بودن این مدل نیز دانست، این شبکه ها لزوماً و بهتر کار می کنند. مقدار درستی این مدل ۹۱/۶ است.

مدل پیشنهادی همان طور که در فصل سوم نیز توضیح داده شد، از یک شبکه عمیق DNN ۱۴ لایه ای با ۴ لایه شبکه DNN با مقادیر متفاوت و لایه رها کردن برای جلوگیری از بیش برآزش استفاده کرده است، تعداد فیلترهای کانولوشن رفته رفته کاهش پیدا می کند و با لایه رها کردن کنترل می شود، فعال ساز سیگموئید و لایه کاملاً متصل بعد از تابع هم سطح ساز به خوبی با لایه های بعدی کامل تر شده است، به طوری که حتی با مقایسه مقدار صحت از نزدیک ترین مقدار صحت فاصله ۳/۹۹۵ (مدل کانولوشن دوم) را مشاهده می کنیم، از نظر کاهش مقدار تابع زیان، این مدل به مقدار تابع زیان ۰/۰۳۸۵ رسیده است، یعنی نه تنها توانسته است صحت بیشتری داشته باشد، بلکه تا جایی که امکان داشت تابع زیانش بسیار کمتر از مدل ها، حتی با کانولوشن های بیشتر است. در ادامه به بررسی نمودارهای تابع خطا خواهیم پرداخت. دقت شود، صحت مدل پیشنهادی به مقدار ۹۷/۰۸ رسیده است یعنی عددی نزدیک به ۱۰۰ درصد که برای یک مدل دارای تشخیص بسیار خوبی است.

جدول ۳: مقایسه عملکرد روش پیشنهادی با سه مدل شبکه عصبی دیگر

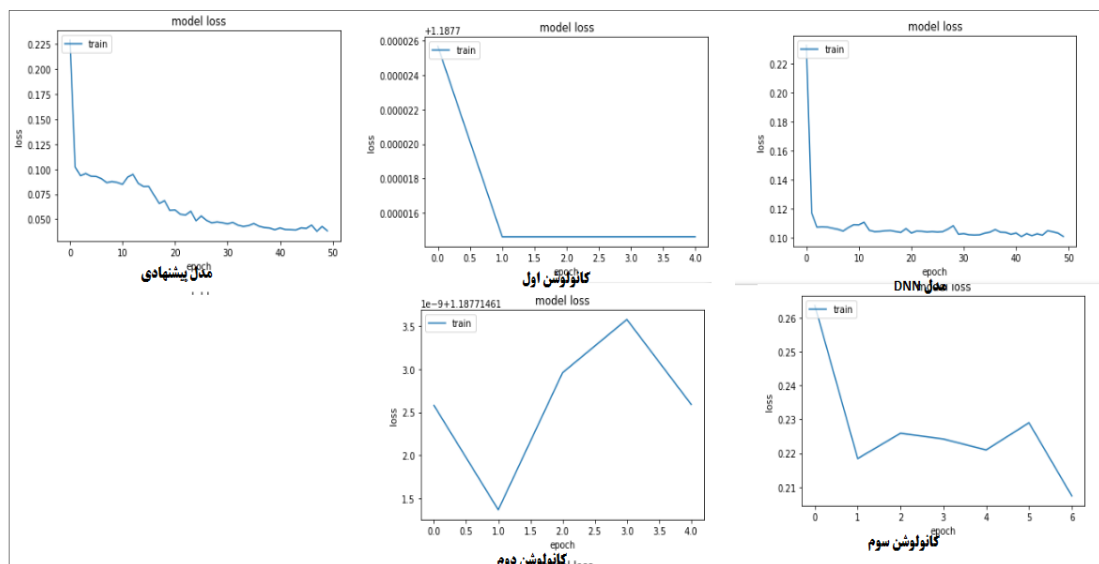
لایه ها	مدل کانولوشن اول	مدل کانولوشن دوم	مدل کانولوشن سوم	مدل LSTM	مدل DNN اول	مدل پیشنهادی
لایه اول	Embedding	Embedding	Embedding	Embedding	Dense (60)	Dense (6)
لایه دوم	Dropout (0/1)	Dropout (0/1)	Dropout (0/1)	Dropout (0/1)	Dense (64)	Dense (64)
لایه سوم	32conv (2*2)	32conv (3*3)	32conv (3*3)	LSTM (128)	Dense (32)	Dense (128)
لایه چهارم	Maxpool (2*2)	Maxpool (2*2)	Dropout (0/5)	Dense (100)	Dense (64)	Dropout (0/2)
لایه پنجم	Dense (100)	32conv (2*2)	32Conv (3*3)	Dropout (0/2)	Dropout (0/2)	Dense (64)
لایه ششم	Dropout (0/1)	Maxpool (2*2)	Dropout (0/5)	Dense (10)	Dense (32)	Dropout (0/2)
لایه هفتم	Dropout (0/1)	Dense (100)	Maxpool (2*2)	Dropout (0/5)	Dense (16)	Dense (32)
لایه هشتم	Maxpool (2*2)	Dropout (0/1)	32conv (3*3)	Dense (1)	Dense (8)	Dropout (0/5)
لایه نهم	Dense (10)	Dense (100)	Dropout (0/5)	Fully-Connected	Dropout (0/2)	Dense (16)
لایه دهم	Dense (1)	Dense (1)	Maxpool (2*2)	-	Dense (6)	Dropout (0/2)
لایه یازدهم	Fully-Connected	Fully-Connected	Dropout (0/5)	-	Dense (1)	Dense (8)
لایه دوازدهم	-	-	Dense (1)	-	Fully-Connected	Dropout (0/2)

Dense (6)	-	-	Fully-Connected	-	-	لایه سیزدهم
Fully-Connected	-	-	-	-	-	لایه چهاردهم
۹۷/۰۸	۹۲/۸۱	۹۱/۶	۹۳/۱۲۵	۹۲/۶۳	۹۳/۰۵	درستی
۰/۰۳۸۵	۰/۱۰۰۷	Nan	۰/۲۰۴۷	۱/۱۸۷۷	۱/۱۸۷۷	مقدار تابع زیان

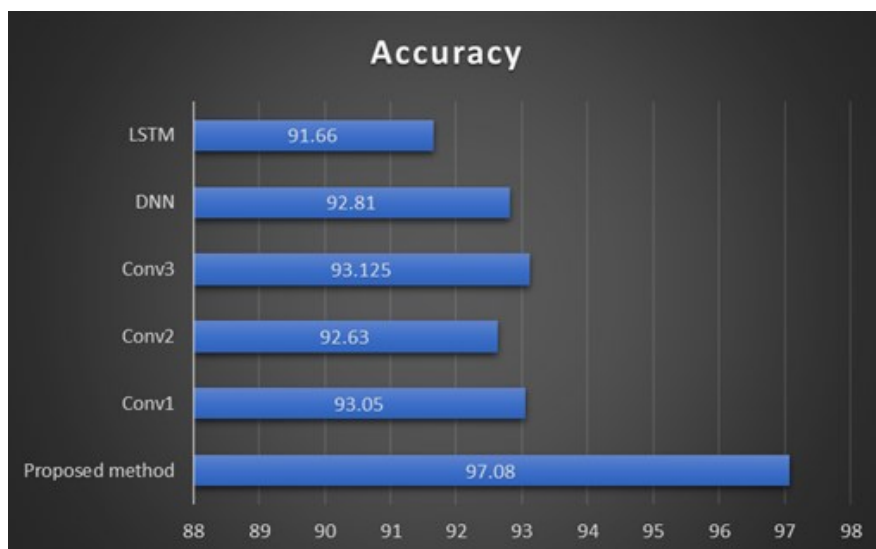
با مقایسه کلیات تابع زیان در شکل ۱۰، همان‌طور که در نتایج هم مشاهده شد، مدل کانولوشن اول با شیب ثابت شونده‌ای مقدار تابع زیان را کاهش داده ولی بعد از چندین تکرار یک مقدار کاملاً ثابت شده است، دلیل این ثابت شدن را پیش‌برازش می‌توان نام برد. استفاده درست از تابع رها کردن می‌توانست احتمالاً تابع زیان را از خط ثابت بودن خارج کند. در مورد مدل کانولوشن دوم، این مدل ابتدا با یک شیب خوب تابع زیان را کاهش می‌دهد ولی به دلیل چیدمان نادرست، لایه‌های تابع زیان دوباره افزایش را نشان می‌دهد که اصلاً قابل قبول نیست.

مدل کانولوشن سوم، باز حرکت شیب نمودار افزایش کاهش را نشان می‌دهد، البته با شیب بهتر، دلیل بهبود عملکرد به نسبت مدل قبل را استفاده درست از لایه maxpool و لایه رها کردن می‌توان نام برد، با این حال فیلترهای کانولوشن در این مسئله خاص عملکرد به نسبت بدتری در برابر مدل شبکه DNN داشتند. نزدیک‌ترین مدل به مدل پیشنهادی، مدل DNN با چیدمان متفاوت‌تر بود، مشاهده می‌شود، تابع زیان این مدل توانسته است با شیب بسیار خوبی تابع زیان را کاهش دهد و تقریباً به مقدار ثابتی در حدود ۰/۱ برساند که به نسبت مدل‌های دیگر از همه بهتر است. این مدل راهگشای استفاده از چیدمان‌های دیگر بر پایه شبکه DNN را نشان می‌دهد.

در مورد استفاده از لایه رها کردن و لایه Dense به‌صورت متوالی به عنوان کنترل‌کننده یادگیری عملکرد عالی این مدل را نشان می‌دهد، مقدار تابع زیان با شیب خوبی کاهش پیدا کرده است و این کاهش با ترتیب توالی بسیار مناسب برخلاف مدل DNN دیگر ادامه‌دار است و به مقدار ۰/۰۳ در ۵۰ تکرار می‌رسد و درستی را تا ۹۷ درصد بالا می‌برد. یک نتیجه خوب که می‌توان گرفت، این است که همیشه استفاده از شبکه کانولوشن یا شبکه بازگشتی LSTM و یا لایه‌های بیشتر الزاماً باعث بهبود صحت و کاهش تابع زیان نخواهد شد. در شکل ۱۱، ارزیابی میزان درستی روش پیشنهادی با سایر مدل‌ها نشان داده شده است.



شکل ۱۰: مقایسه ۵ مدل با مدل پیشنهادی در تابع زیان



شکل ۱۱: ارزیابی میزان درستی روش پیشنهادی با سایر مدل‌ها

نتایج حاصل از ارزیابی روش پیشنهادی که با روش‌های دیگر که در شکل ۱۱ نمایش داده شده است، به این صورت است که در روش پیشنهادی استفاده از یادگیری عمیق برای تشخیص برون هشته‌ای، شبکه عصبی عمیق پیشنهادی، مقدار معیار دقت ۰/۹۷، مقدار معیار بازخوانی ۰/۹۷ و مقدار معیار درستی ۰/۹۷۰۸ است. در مدل کانولوشن اول مقدار معیار دقت ۰/۸۷، مقدار معیار بازخوانی ۰/۹۳ و مقدار معیار درستی ۰/۹۳۰۵ است. در مدل کانولوشن دوم مقدار معیار دقت ۰/۸۷، مقدار معیار بازخوانی ۰/۹۱ و مقدار معیار درستی ۰/۹۲۶۳ است. در مدل کانولوشن سوم مقدار معیار دقت ۰/۹۱، مقدار معیار بازخوانی ۰/۹۳ و مقدار معیار درستی ۰/۹۳۱۲۵ است. برای شبکه عصبی LSTM مقدار دقت ۰/۸۴، میزان بازخوانی ۰/۹۲ میزان درستی عدد ۰/۹۱۶ را نشان می‌دهد و برای مدل شبکه عصبی DNN با لایه‌های کمتر و چیدمان متفاوت مقدار دقت ۸۴ و بازخوانی و صحت به ترتیب ۹۲ و ۹۲/۸۱ را نشان می‌دهد.

در جدول ۴ نتایج حاصل از معیارهای ارزیابی روش پیشنهادی و سایر معیارهای مربوط به مطالعات سایر محققین جمع‌آوری و نشان داده شده است.

جدول ۴: نتایج مقایسه خروجی روش پیشنهادی با نتایج روش‌های سایر محققین

F1-Score	درستی	بازخوانی	دقت	الگوریتم
۰/۹۷	۰/۹۷۰۸	۰/۹۷	۰/۹۷	روش پیشنهادی (تحقیق حاضر)
۰/۹۰	۰/۹۳۰۵	۰/۹۳	۰/۸۷	مدل کانولوشن اول
۰/۹۰	۰/۹۲۶۳	۰/۹۱	۰/۸۷	مدل کانولوشن دوم
۰/۹۲	۰/۹۳۱۲۵	۰/۹۲	۰/۹۱	مدل کانولوشن سوم
۰/۹۲	۰/۹۱۶۶	۰/۹۲	۰/۸۴	LSTM
۰/۸۸	۰/۹۲۸۱	۰/۹۲	۰/۸۴	مدل DNN

نتیجه‌گیری

در این تحقیق، برای پیاده‌سازی از یک شبکه عصبی عمیق ۱۴ لایه استفاده شده است. ترکیب لایه‌ها در این شبکه ۱۴ لایه به این صورت است: لایه ورودی، سه لایه پشت سرهم شبکه عصبی DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه DNN، لایه رها کردن، لایه سیگموئید برای انتخاب کلاس متناسب استفاده شده است. با توجه به محاسبات بسیار زیاد از GPU بهره برده شده است. لایه اول تا سوم در مدل پیشنهادی یک شبکه

عصبی عمیق با ورودی ۳۳۰ (به تعداد ویژگی‌های مجموعه داده) و تعداد کل پارامترها لایه‌های اول ۱۹۸۶، ۴۴۸، ۸۳۲۰ به دلیل شکل خروجی‌های این لایه‌ها است. تابع فعال‌سازی یکی از بهترین تابع‌های آزمایش شده برای شبکه عصبی است و از تابع غیرخطی Relu استفاده شده است. در مدل پیشنهادی بعد از لایه maxpool اول از لایه Drop-out با مقدار ۰/۲ استفاده شده است تا با احتمال ۰/۲ ورودی لایه DNN حذف و یادگیری ادامه پیدا کند.

این لایه تأثیری بر ورودی ندارد فقط برای جلوگیری از overfit و یادگیری بهتر استفاده می‌شود. از لایه پنجم تا لایه سیزدهم با مقادیر متفاوت لایه DNN و لایه رها کردن پشت سرهم تکرار می‌شوند، البته مقادیر پارامترهای خروجی و ورودی لایه‌های DNN با لایه‌های قبل متفاوت است. برای تمامی مقادیر لایه رها کردن مقدار ۰/۲ در نظر گرفته شده است، فقط در لایه هشتم با توجه به افزایش پارامترها، نرخ لایه رها کردن نصف در نظر گرفته شده است تا نصف یاد گرفته شده‌ها، فراموش شده یادگیری بهتری داشته باشیم. لایه آخر در مدل پیشنهادی، یک لایه کاملاً متصل برای مشخص کردن کلاس خروجی، مدل پیشنهادی، انتخاب برون‌هشتمی از ۳ کلاس ورودی است. این لایه ورودی‌های کلی یاد گرفته شده در لایه‌های قبل را جمع کرده و به کلاس متناسب آن اختصاص می‌دهد. در مدل پیشنهادی بهینه‌ساز مورد استفاده بهینه‌ساز Adam بود. مجموعه داده استفاده شده در این پژوهش، مجموعه داده پزشکی کم‌کاری تیروئید است که یکی از مجموعه داده‌های معروف برای تشخیص برون‌هشتمی است. این مجموعه داده شامل ۳ کلاس شامل کلاس عادی، عملکرد فوق‌العاده و عملکرد غیرطبیعی (برون‌هشتمی) برای آزمایش تشخیص برون‌هشتمی است. نتایج حاصل مقادیر ارزیابی و دقت، بیانگر عملکرد خوب و مناسب روش پیشنهادی در مقایسه با سایر روش‌ها است.

منابع

- Ahmed M, Mahmood A N & Hu J. 2016. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19-31.
- Alam S, Dobbie G, Riddle P, and Naeem A. 2010. A swarm intelligence based clustering approach for outlier detection. *IEEE Press*, 978(1), 4244-8126.
- Bossers H, Hurink J, and Smit G. 2014. Selection of tests for outlier detection. *IEEE 31st VLSI Test Symposium (VTS)*.
- Daneshfaraz R, Aminvash E, Ghaderi A, Abraham J, Bagherzadeh M. 2021a. SVM Performance for Predicting the Effect of Horizontal Screen Diameters on the Hydraulic Parameters of a Vertical Drop. *Applied science*. 11, 4238. <https://doi.org/10.3390/app11094238>.
- Daneshfaraz R. Aminvash E, Mirzaei R, Abraham J. 2021b. Predicting the energy dissipation of a rough sudden expansion rectangular stilling basins using the SVM algorithm. *Journal of Applied Research in Water and Wastewater*, 11, 4238. <https://doi.org/10.3390/app11094238>.
- Gupta M, Gao J, Aggrawal C & Jiawei H. 2014. *Outlier detection for temporal data*. Morgan and Claypool Publishers. The U.S.
- Kieu T, Yang B & Jensen C. S. 2018. Outlier detection for multidimensional time series using deep neural networks. In 2018 19th IEEE International Conference on Mobile Data Management (MDM) (pp. 125-134). IEEE.
- LeCun Y, Bottou L, Bengio Y & Haffner P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Liu L. and Fern Z. 2012. Constructing training sets for outlier detection.
- Marco A, Pimental N, Clifton D, Clifton L & Tarassenko L. 2014. A review of novelty detection. Retrieved from www.elsevier.com/locate/sigpro, page 215-249.
- Masakazu M, Mori K, Mitari Y, Kaneda Y. 2003. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *International Journal of Science and Research(IJSR)*.
- Rezapour M, Asadi R, Marghoob B. 2021. Machine Learning Algorithms as new screening framework for recommendation of Appropriate Vascular Access and Stroke Reduction. *Int J Hosp Res*, 10 (3).
- Rezapour M. 2021. Predicting Stroke in Hemodialysis Patients Using Data Mining. *Digital Transformation*, 1(1), 45-57. doi: 10.22034/dtj.2021.250451.1001
- Sermanet P, LeCun Y. 2011. Traffic sign recognition with multi-scale Convolutional Networks. In *The 2011 International Joint Conference on Neural Networks*; 2809–2813.

- Shadroo Sh, Rahmani A M. 2018. Systematic survey of big Data and Data Mining in Internet of Things. Computer Networks, Accepted Manuscript, 70 pages.
- Souza A M & Amazonas J. R. 2015. An outlier detect algorithm using big data processing and internet of things architecture. *Procedia Computer Science*, 52, 1010-1015.
- Tsai C F, Chang Fu-Yu. 2016. Combining instance selection for better missing value imputation, *Journal of Systems and Software*, 122, 63-71.
- Yan Z, Liu J, Yang L T & Chawla N. 2018. Big data fusion in Internet of Things. *Information fusion*, 100(40), 32-33.
- Yang S, Liu W. 2011. Anomaly detection on collective moving patterns: A hidden markov model based solution". *IEEE International Conference of Things, and Cyber, Physical and Computing*.
- Zenati H, Foo C S, Lecouat B, Manek G & Chandrasekhar V R. (2018). Efficient gan-based anomaly detection. Institute for Infocomm Research, Singapore .School of Computer Science, Nanyang Technological University.
- Zhou Q. 2018. Bidirectional long short-term memory network for proto-object representation (Doctoral dissertation).