دانشگاه علم و فناوری مازندران
وزارت علوم، تحقیقات و فناوری

پایان‌نامه کارشناسی

پیش‌بینی بارش باران

نگارش

امید عقیلی

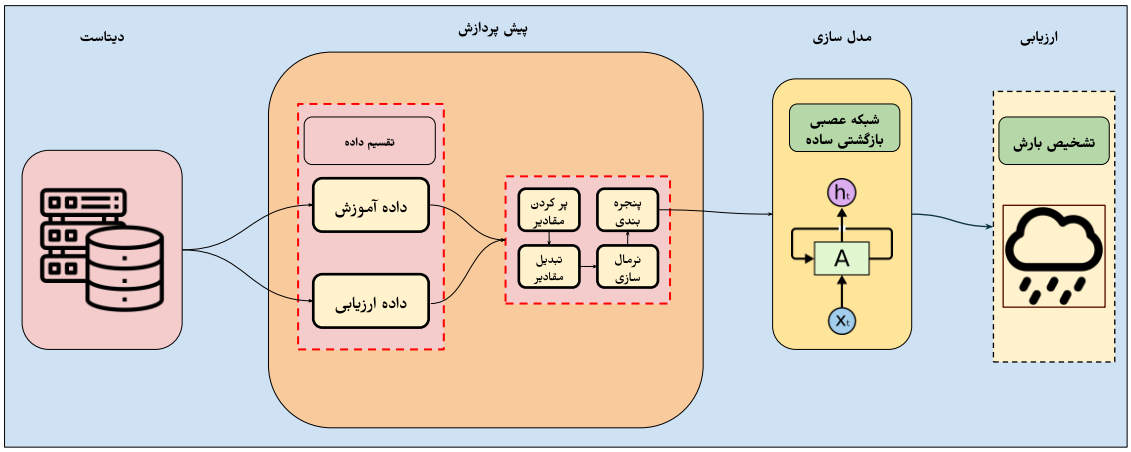
استاد راهنما

دکتر نظری

مرداد ۱۴۰۲

چکیده

کشاورزی نقطه کلیدی برای بقا است. برای کشاورزی، بارندگی بسیار مهم است. این روزها پیش‌بینی بارندگی به یک مشکل بزرگ تبدیل‌شده است. پیش‌بینی بارندگی به مردم آگاهی می‌دهد و از قبل از بارندگی مطلع می‌شوند تا اقدامات احتیاطی خاصی برای محافظت از محصول خود در برابر بارندگی انجام دهند. تکنیک‌های زیادی برای پیش‌بینی بارندگی به وجود آمدند. الگوریتم‌های یادگیری ماشین بیشتر در پیش‌بینی بارندگی مفید هستند.این پایان‌نامه با هدف تشخیص بارندگی روی دیتاست استرالیا با استفاده از مدل RNN ساده انجام‌شده است. در این پژوهش، ابتدا داده‌های هواشناسی استرالیا پیش‌پردازش شده‌اند. سپس مدل RNN ساده با استفاده از این داده‌ها آموزش داده‌شده است. مدل آموزش داده‌شده با داده‌های آزمایشی ارزیابی‌شده و دقت ۹۸٪ در تشخیص بارندگی حاصل‌شده است. این پایان‌نامه نتایج قابل قبولی را در حوزه تشخیص بارندگی روی دیتاست استرالیا با استفاده از مدل RNN ساده ارائه می‌دهد. نتایج حاصل از این پژوهش می‌تواند به محققان و علاقه‌مندان در حوزه هواشناسی و پیش‌بینی آب‌وهوا کمک کند.



واژه‌های کلیدی:

تشخیص بارندگی، RNN ساده، دیتاست بارندگی استرالیا، هواشناسی، پیش‌بینی آب‌وهوا.

فهرست مطالب

صفحه

فهرست جداول

صفحه

فهرست اشکال

صفحه

فصل اول

مقدمه

مقدمه

تشخیص بارندگی و پیش‌بینی آب‌وهوا از اهمیت بسیاری در زمینهٔ‌های مختلف مانند کشاورزی، زراعت، سیستم‌های آبیاری و برنامه‌ریزی شهری برخوردار است. در این زمینه، استرالیا به‌عنوان یکی از مناطقی با آب‌وهوای متغیر و پویا شناخته می‌شود[۱و۲]. بنابراین، توانایی تشخیص صحیح و دقیق بارندگی در استرالیا، اهمیت ویژه‌ای دارد.

در سال‌های اخیر، با پیشرفت فناوری و گسترش استفاده از شبکه‌های عصبی عمیق، روش‌های مبتنی بر شبکه‌های عصبی برای تشخیص بارندگی و پیش‌بینی آب‌وهوا به‌طور گسترده مورداستفاده قرارگرفته‌اند. یکی از معماری‌های شبکه عصبی که در این حوزه به‌خوبی عملکرد کرده است، مدل RNN (Simple Recurrent Neural Network) است[۳].

هدف اصلی این پایان‌نامه، تشخیص بارندگی روی دیتاست استرالیا با استفاده از مدل RNN ساده است. برای این منظور، از داده‌های استرالیا استفاده‌شده است. با آموزش مدل RNN با استفاده از این داده‌ها، تلاش می‌شود تا دقت بالا در تشخیص بارندگی حاصل شود.

مزیت اصلی مدل RNN در مقایسه با روش‌های سنتی، قابلیت استفاده از اطلاعات گذشته در تصمیم‌گیری است. با در نظر گرفتن اطلاعات گذشته در فرآیند تشخیص بارندگی، مدل RNN قادر به درک الگوهای زمانی و همبستگی‌های موجود در داده‌های هواشناسی استرالیا خواهد بود[۴،۵و۶].

نتایج این پایان‌نامه نه‌تنها می‌تواند به محققان و علاقه‌مندان در حوزه هواشناسی و پیش‌بینی آب‌وهوا کمک کند، بلکه به‌عنوان یک ابزار قابل‌استفاده در تصمیم‌گیری‌های مرتبط با بارندگی در استرالیا مورداستفاده قرار گیرد. همچنین، این مطالعه می‌تواند به‌عنوان یک پایه برای پژوهش‌های آینده در زمینه تشخیص بارندگی و پیش‌بینی آب‌وهوا با استفاده از روش‌های شبکه عصبی عمیق عمل کند.

این پایان‌نامه، مراحل اصلی پژوهش را شامل می‌شود. ابتدا، داده‌های هواشناسی استرالیا پیش‌پردازش می‌شوند. سپس، مدل RNN با استفاده از این داده‌ها آموزش داده می‌شود. نهایتاً، مدل با داده‌های آزمایشی ارزیابی می‌شود و دقت حاصل از آن محاسبه می‌شود.

در فصـــل اول، مقدمه‌ای کلی درباره اهمیت تشـــخیص بارندگی و پیش‌بینی آب‌وهوا ارائه‌شـــده اســـت. همچنین، مدل RNN و دلایل استفاده از آن برای تشخیص بارندگی در بررسی قرارگرفته است.

در فصل دوم، مفاهیم مرتبط با تشخیص بارندگی و ابزارهای استفاده‌شده موردبررسی قرار می‌گیرد.

در فصـل سـوم، معماری مدل RNN و نحوه آموزش آن بر روی‌داده‌های هواشـناسـی اسـترالیا توضـیح داده می‌شود. این فصل شامل جزئیات فنی مدل شامل تعریف لایه‌ها، توابع فعال‌سازی و الگوریتم بهینه‌سازی است.

در فصـل چهارم، نتایج آزمایشـات و ارزیابی مدل بر روی‌داده‌های آزمایشـی ارائه می‌شـود. عملکرد مدل در تشخیص بارندگی و مقایسه آن با روش‌های دیگر موردبررسی قرار می‌گیرد.

در فصـل پنجم، نتایج به‌دسـت‌آمده مورد تحلیل و بحث قرار می‌گیرد. علل عملکرد موفق یا عدم موفقیت مدل بررســـی می‌شـــوند و راهکارهای بهبود کارایی مدل مطرح می‌شـــوند. همچنین جمع‌بندی نتایج و پیشنهادهایی برای پژوهش‌های آینده در این حوزه ارائه می‌شود.

فصل دوم

مفاهیم مرتبط

کارهای پیشین

در این بخش، ابتدا به بررسی تعدادی از مقالات قبلی خواهیم پرداخت که بیشتر از شبکه‌های عصبی بازگشتی برای پیش‌بینی استفاده کرده‌اند، زیرا روش‌های پایه‌ی مورد مقایسه با سامانه پیشنهادی خودمان در این پایان‌نامه، در این دسته از پژوهش‌ها قرار دارند و آشنایی بیشتر با این زیرگروه از روش‌ها، ضروری است. در انتها، مهم‌ترین نقاط قوت و ضعف این پژوهش‌ها را شناسایی خواهیم کرد.

مقاله [۱۱] چندین مدل پیشرفته هوش مصنوعی (AI) یعنی سیستم استنتاج فازی مبتنی بر شبکه تطبیقی بهینه‌سازی شده با بهینه‌سازی ازدحام ذرات (PSOANFIS)، شبکه‌های عصبی مصنوعی (ANN) و ماشین‌های بردار پشتیبانی (SVM) را برای پیش‌بینی بارندگی روزانه استفاده کرده است. برای این منظور پارامترهای متغیر هواشناسی مانند حداکثر دما، حداقل دما، سرعت باد، رطوبت نسبی و تابش خورشیدی جمع‌آوری و به‌عنوان پارامترهای ورودی و بارندگی روزانه به‌عنوان پارامتر خروجی در مدل‌ها استفاده‌شده‌اند. نتایج نشان داده که تمام مدل‌های هوش مصنوعی پیش‌بینی‌های معقولی از بارش روزانه ارائه می‌کنند، اما SVM بهترین روش برای پیش‌بینی بارندگی است.

مقاله [۱۲] یک تحلیل مقایسه‌ای با استفاده از مدل‌های تخمین بارندگی ساده‌شده بر اساس الگوریتم‌های یادگیری ماشین معمولی و معماری‌های یادگیری عمیق ارائه می‌کند که برای این کاربردهای پایین‌دستی کارآمد هستند. مدل‌های مبتنی بر LSTM، Stacked-LSTM، شبکه‌های دو جهته-LSTM، XGBoost، و مجموعه‌ای از رگرسیون تقویت‌کننده گرادیان، رگرسیون بردار پشتیبان خطی، و رگرسیون Extra-trees در کار پیش‌بینی حجم بارندگی ساعتی با استفاده از داده‌های سری زمانی مقایسه شدند. در بین تمام مدل‌های آزمایش‌شده، شبکه Stacked-LSTM با دو لایه پنهان و شبکه Bidirectional-LSTM بهترین عملکرد را داشتند. این نشان می‌دهد که مدل‌های مبتنی بر شبکه‌های LSTM با لایه‌های پنهان کمتر عملکرد بهتری برای این رویکرد دارند. نشان‌دهنده توانایی آن برای استفاده به‌عنوان رویکردی برای برنامه‌های کاربردی پیش‌بینی بارش ازنظر بودجه است.

مقاله [۱۳] مدل ARIMA و شبکه‌های عصبی را مقایسه می‌کند و به مؤثر بودن شبکه‌های عصبی در تشخیص بارندگی اشاره دارد.

مقاله [۱۴] عملکرد ۸ روش آماری و یادگیری ماشینی را که توسط الگوهای سینوپتیک جوی هدایت می‌شوند، برای پیش‌بینی بلندمدت بارش روزانه در یک آب‌وهوای نیمه‌خشک ارزیابی می‌کند. نتایج کار نشان می‌دهد که عملکرد اکثر مدل‌های یادگیری ماشینی نسبت به فراپارامترهای انتخابی بسیار حساس است. شبکه‌های عصبی برای پیش‌بینی وقوع و شدت بارندگی بهترین عملکرد را دارند. همه روش‌ها واریانس سری‌های مشاهده‌شده را در مقیاس‌های زمانی روزانه دست‌کم می‌گیرند. مدل‌های خطی تعمیم‌یافته با استفاده از خطاهای توزیع‌شده گاما برای پیش‌بینی شدت بارندگی بهترین عملکرد را دارند، بااین‌حال، عملکرد آن‌ها کاربردهای عملی آن را محدود می‌کند. نتایج به‌طور قابل‌توجهی در تجمعات زمانی بزرگ‌تر (ماهانه یا سالانه) بهبود می‌یابد و روش‌های آماری و یادگیری ماشینی را برای مطالعات منابع آب ارزشمندتر می‌کند.

درنهایت با بررسی مقالات فوق تصمیم گرفتیم از شبکه‌های عصبی بازگشتی استفاده کنیم. همچنین دیدیم که نیاز به حفظ دنباله به تعداد زیاد نیز نداریم بنابراین بر این شدیم از مدل ساده شبکه‌های عصبی بازگشتی استفاده کنیم. مزیت اصلی مدل RNN در مقایسه با روش‌های سنتی، قابلیت استفاده از اطلاعات گذشته در تصمیم‌گیری است. با در نظر گرفتن اطلاعات گذشته در فرآیند تشخیص بارندگی، مدل RNN قادر به درک الگوهای زمانی و همبستگی‌های موجود در داده‌های هواشناسی استرالیا خواهد بود[۴،۵و۶].

مفاهیم مرتبط

این فصل به ارائه مفاهیم مرتبط با مسئله تشخیص بارندگی می‌پردازد. هدف از این قسمت، آشنایی نظری با روش‌ها و الگوریتم‌هایی است که در معماری سامانه پیشنهادی و روش‌های پایه‌ای که در انتها برای مقایسه با آن انتخاب می‌شوند و دانستن آن‌ها برای فهم روند طی شده در این پایان‌نامه ضروری است.

۲-۱- داده‌های سری زمانی

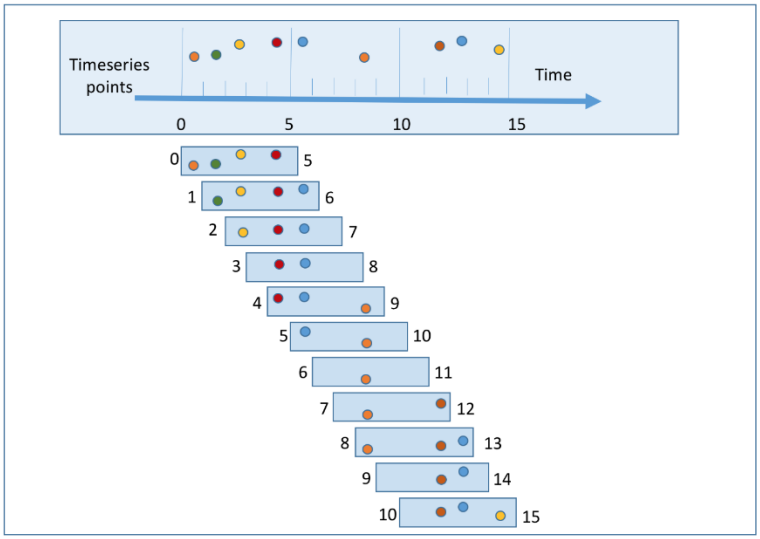
داده‌های سری زمانی به داده‌هایی اطلاق می‌شود که به‌صورت مرتب در طول زمان جمع‌آوری‌شده‌اند. در این نوع داده‌ها، مشاهده‌ها به ترتیب زمانی قرار دارند و بین هر دو مشاهده ممکن است فاصله‌های زمانی ثابت یا غیرثابت وجود داشته باشد. به‌عنوان‌مثال، داده‌های سری زمانی می‌توانند شامل اطلاعات روزانه درباره درجه حرارت، میزان بارش، قیمت سهام یا تعداد فروش یک محصول در بازار باشند. داده‌های سری زمانی معمولاً به دلیل وجود الگوها، تغییرات موسمی و وابستگی‌های زمانی، تحلیل و پیش‌بینی آن‌ها می‌تواند اطلاعات مفیدی را در اختیار ما قرار دهد[۷].

شکل ۱- داده‌های سری زمانی

۲-۲- پنجره‌بندی

پنجره‌بندی یک روش مهم در پردازش داده‌های سری زمانی است که برای تجزیه‌وتحلیل و پیش‌بینی این نوع داده‌ها استفاده می‌شود. هدف از پنجره‌بندی، تقسیم داده‌های سری زمانی به بخش‌های کوچک‌تر و قابل مدل‌سازی تر است. در این روش، با انتخاب یک پنجره زمانی مشخص، بخشی از داده‌های سری زمانی را انتخاب می‌کنیم و آن را به‌عنوان ورودی برای مدل استفاده می‌کنیم. این پنجره زمانی معمولاً به‌صورت متحرک در طول داده حرکت می‌کند، به این معنی که پس از استفاده از یک پنجره، به پنجره بعدی منتقل می‌شویم و همین روند تا انتهای داده ادامه می‌یابد[۸].

مزیت استفاده از پنجره‌بندی در تحلیل داده‌های سری زمانی این است که می‌توانیم وابستگی‌های زمانی و الگوهای موجود در داده را به‌خوبی مدل کنیم. با استفاده از پنجره‌بندی، می‌توانیم اطلاعات قبلی و بعدی از هر نقطه داده را در نظر بگیریم و با ایجاد الگوهای زمانی مختلف، تغییرات پیچیده را تحلیل کنیم. همچنین، با تغییر اندازه پنجره می‌توانیم تجزیه‌وتحلیل دقیق‌تری از داده‌ها داشته باشیم[۹].



شکل ۲- پنجره‌بندی

۲-۳- شبکه‌های عصبی بازگشتی ساده (Simple Recurrent Neural Networks - SRNN)

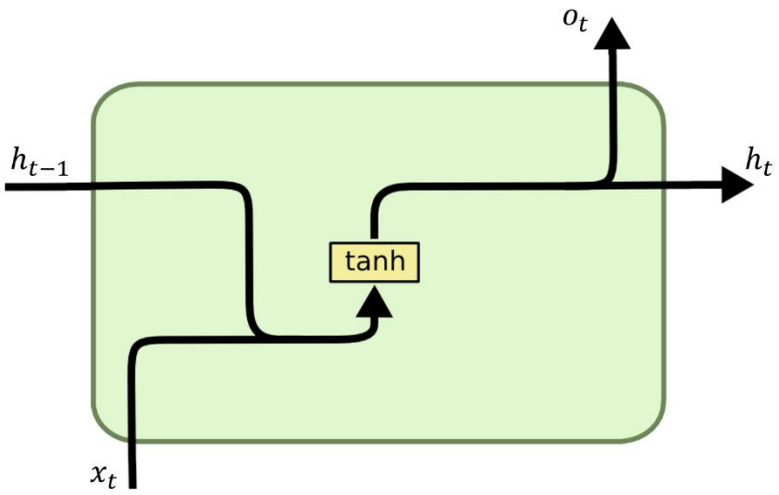
شبکه‌های عصبی بازگشتی ساده (SRNN) یک نوع از شبکه‌های عصبی مصنوعی هستند که برای پردازش داده‌های سری زمانی و دارای وابستگی‌های زمانی استفاده می‌شوند. SRNNها، قابلیت ذخیره و استفاده از اطلاعات گذشته را دارند و این امکان را به ما می‌دهند تا الگوهای زمانی پیچیده را در داده‌های سری زمانی شناسایی کنیم[۵].

ساختار SRNN شامل یک لایه بازگشتی است که وظیفه تکرار اطلاعات در زمان را بر عهده دارد. هر نود در این لایه، یک واحد حافظه را در خود دارد که می‌تواند اطلاعات مربوط به تمام تاریخچه ورودی‌های قبلی را در خود ذخیره کند. این حافظه به‌عنوان حالت (state) مخفی شبکه شناخته می‌شود[۵].

نحوه عملکرد SRNN به این صورت است که در هر مرحله زمانی، ورودی جدید و حالت مخفی قبلی به‌عنوان ورودی به لایه بازگشتی داده می‌شوند. لایه بازگشتی این اطلاعات را در حافظه خود ذخیره کرده و خروجی مربوط به آن محاسبه می‌کند. سپس، این خروجی به‌عنوان خروجی شبکه در این مرحله استفاده می‌شود و همچنین به‌عنوان حالت مخفی ورودی برای مرحله بعدی مورداستفاده قرار می‌گیرد. این فرآیند به‌صورت تکراری ادامه می‌یابد تا به انتهای داده برسیم. SRNN با استفاده از وزن‌های قابل‌آموزش، تلاش می‌کند الگوهای زمانی و وابستگی‌های زمانی را در داده‌های سری زمانی شناسایی کند و پیش‌بینی کند. در فرآیند آموزش، با مقایسه خروجی مدل با خروجی مطلوب، وزن‌ها به‌گونه‌ای به‌روزرسانی می‌شوند که عملکرد و پیش‌بینی مدل بهبود یابد[۱۰].

استفاده از SRNN در حوزه‌های مختلف بسیار گسترده است. این شبکه‌ها در تشخیص الفبا، تشخیص گفتار، ترجمه ماشینی، تحلیل متن، پیش‌بینی بازارهای مالی و بسیاری از مسائل دیگر به کار می‌روی شده‌اند. SRNN به دلیل قابلیتشان در مدل‌سازی وابستگی‌های زمانی، مورداستفاده گسترده قرار می‌گیرند[۵].

در اینجا یک نمونه ساده از ساختار یک SRNN را مشاهده می‌کنید:



شکل ۳- ساختار RNN ساده

در این ساختار، x(t) ورودی در زمان t را نشان می‌دهد. h(t) حالت مخفی در زمان t است که اطلاعات گذشته را نگه می‌دارد. علامت SRNN، لایه بازگشتی را نمایش می‌دهد. سپس، خروجی y(t) در زمان t بر اساس ورودی و حالت مخفی محاسبه می‌شود.

معادلات مربوط به یک SRNN ساده به‌صورت زیر است:

$$h(t) = f(Ux(t) + Wh(t-1))$$
$$y(t) = g(Vh(t))$$

در این معادلات، U و V وزن‌های مربوط به ورودی و خروجی هستند. W وزن مربوط به حالت مخفی است. تابع f و g توابع فعال‌سازی هستند که معمولاً توابع غیرخطی مانند تانژانت هایپربلیک یا سیگموید استفاده می‌شوند.

در اینجا، ورودی x(t) و حالت مخفی قبلی h(t-1) ترکیب می‌شوند و توسط تابع فعال‌سازی f پردازش می‌شوند تا حالت مخفی جدید h(t) تولید شود. سپس، با استفاده از وزن‌های V و تابع فعال‌سازی g، خروجی y(t) محاسبه می‌شود.

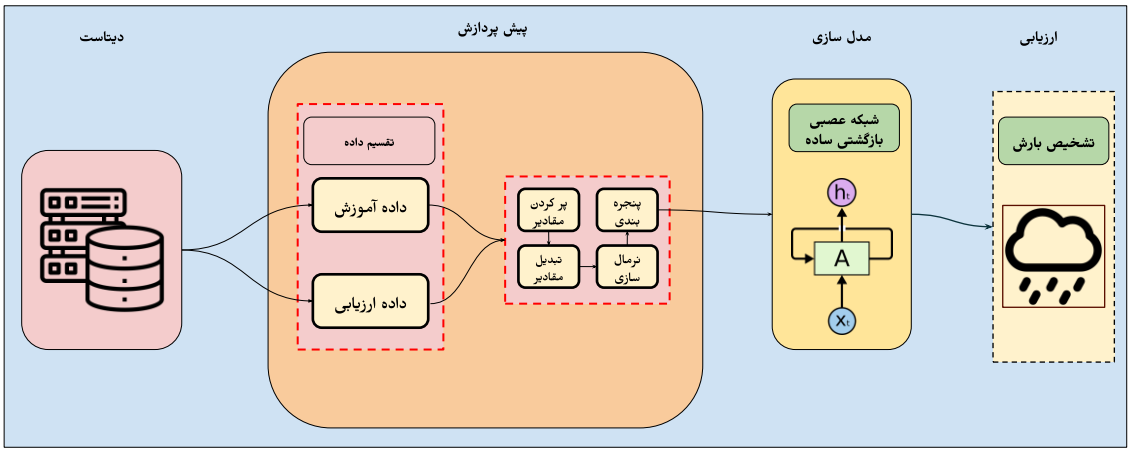
با استفاده از عملیات تکراری این مدل بر روی دنباله‌های طولانی از داده‌های سری زمانی، می‌توان الگوهای زمانی پیچیده را شناسایی و پیش‌بینی کرد. این قابلیت شبکه‌های عصبی بازگشتی ساده را برای مسائلی مانند ترجمه ماشینی، تشخیص گفتار و پیش‌بینی سری زمانی بسیار ارزشمند می‌کند.

فصل سوم

پیش‌بینی بارش باران

معماری سامانه پیشنهادی

در این فصل، سامانه پیشنهادی برای تشخیص بارندگی شرح داده می‌شود. شکل زیر معماری کلی سیستم را نشان می‌دهد.



شکل ۴- معماری سیستم پیشنهادی

ورودی سامانه، ویژگی‌های آب‌وهوایی هستند که به شکل خام به سامانه داده می‌شوند. خروجی سامانه، بارش یا عدم بارش در آن روز را مشخص می‌کند. این سامانه بعد از انجام پیش‌پردازش‌های لازم روی‌داده‌های خام، یک مدل می‌سازد که وظیفه‌اش دسته‌بندی داده‌ها به دو گروه بارش و عدم بارش است. در ادامه جزییات زیرسامانه‌های این معماری تبیین می‌شوند.

۳-۱- زیرسامانه پیش‌پردازش

بخش ابتدایی معماری سامانه پیشنهادی از زیر بخش پیش‌پردازش داده‌ها تشکیل‌شده است. هدف از طراحی این قسمت، پردازش داده‌های خام است. بخش ۳-۱-۱ به پاک‌سازی و نرمال‌سازی داده‌های خام اختصاص دارد. سپس در بخش ۳-۱-۲ نحوه جداسازی داده آموزش و آزمایش را بررسی می‌کنیم.

لازم به ذکر است که سامانه پیش‌پردازش و به‌طورکلی معماری پیشنهادی به نحوی طراحی‌شده است که آمادگی پذیرش انواع مختلف ویژگی‌های آب‌وهوایی با مشخصات گوناگون را داشته باشد. بدین‌جهت، کلیه‌ی گام‌های سامانه پیش‌پردازش قابل‌اعمال روی‌داده‌های هر نوع بارندگی را دارد.

۳-۱-۱- پاک‌سازی و نرمال‌سازی

پاک‌سازی داده‌ها، فرآیند شناسایی حذف یا تصحیح سوابق اشکال‌دار یا نادرست از یک مجموعه دادگان است و به شناسایی قسمت‌های ناقص، نادرست یا نامربوط از داده‌ها و سپس جایگزینی، اصلاح یا پاک کردن این نوع داده‌ها می‌پردازد.

در روش پیشنهادی، ابتدا مجموعه دادگان ازلحاظ داده‌های نویزی و تکراری بررسی‌شده و در صورت وجود، این نوع داده‌ها حذف می‌شوند. سپس به دلیل توالی‌دار بودن داده‌ها ویژگی زمان (ترتیب) حذف می‌شود، زیرا زمان فقط در بحث توالی و ترتیب برای ما اهمیت دارد و به‌تنهایی ویژگی مؤثر در تشخیص بارندگی نیست. بااین‌وجود، اگر این ویژگی به همراه تاریخ و ساعت بود، می‌توانست دارای اهمیت باشد.

سپس به نرمال‌سازی داده‌ها می‌پردازیم. نرمال‌سازی داده‌ها، روشی است که برای استانداردسازی دامنه ویژگی‌های داده استفاده می‌شود. ازآنجایی‌که دامنه مقادیر داده‌ها ممکن است بسیار متفاوت باشد، این‌یک مرحله ضروری در پیش‌پردازش داده‌ها در حین استفاده از الگوریتم‌های یادگیری ماشین است، زیرا الگوریتم‌های یادگیری ماشین فقط اعداد را می‌بینند و اگر تفاوت زیادی در دامنه وجود داشته باشد، ممکن است این فرض اساسی را ایجاد کند که اعداد با محدوده بالاتر به‌نوعی برتری دارند. در این پژوهش از نرمال‌سازی استاندارد استفاده می‌کنیم. در یادگیری ماشین، می‌توان از نرمال‌سازی استاندارد برای تغییر اندازه توزیع مقادیر استفاده کرد به‌طوری‌که میانگین مقادیر ۰ و انحراف استاندارد ۱ باشد، به‌عبارتی‌دیگر

نرمال‌سازی استاندارد، میانگین را حذف می‌کند و داده‌ها را به واریانس واحد مقیاس می‌کند. برای نرمال‌سازی استاندارد از رابطه زیر بهره برده می‌شود:

$$z = \frac{(x - \mu)}{\sigma}$$

که μ میانگین

$$\mu = \frac{1}{N} \sum_{i=1}^{N}(x_i)$$

و σ انحراف معیار است:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N}(x_i - \mu)^2}$$

۳-۱-۲- جداسازی داده

در این بخش، قبل از ورود به فرآیند مدل‌سازی، ما به جداسازی داده آموزش و آزمایش می‌پردازیم، زیرا این کار باید قبل از مدل‌سازی انجام شود تا از نشت داده جلوگیری شود

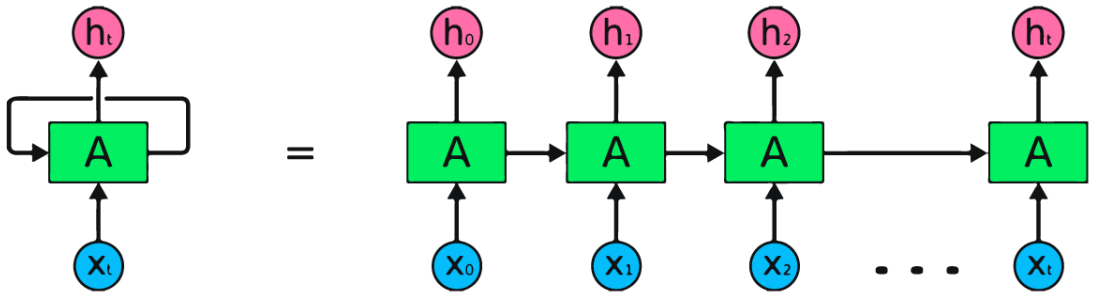
برای جداسازی چون ترتیب دارای اهمیت است ما صرفاً ۳ سال انتهایی داده را به‌عنوان داده آزمایش در نظر می‌گیریم که مدل همانند دنیای واقعی داده‌های اخیر را به ترتیب دیده، به‌روزرسانی شده، الگوهایی که منسوخ‌شده‌اند را فراموش کرده و حال به پیش‌بینی داده‌های جدید بپردازد.

۳-۲- تشخیص بارندگی با توجه به توالی

در این بخش به‌تفصیل سامانه تشخیص بارش با توجه به توالی می‌پردازیم و مدل‌سازی این سامانه را تشریح می‌کنیم و در انتها کل چارچوب را یکپارچه توصیف می‌کنیم.

۳-۳-۱- مدل‌سازی با توجه به توالی

برای مدل‌سازی در این بخش، از مدل شبکه‌های عصبی بازگشتی استفاده خواهیم کرد. شبکه‌های عصبی بازگشتی می‌توانند انواع وابستگی‌ها را به‌ویژه در مسائل پیش‌بینی توالی یاد بگیرند. همان‌گونه که در شکل مشخص‌شده است، برای تشخیص بارش در روز از پنجره‌ای با اندازه ثابت بهره برده می‌شود که ارتباط بارش در روز فعلی، با بارش در روزهای قبلی‌اش در نظر گرفته می‌شود.



شکل ۵- یک لایه شبکه عصبی بازگشتی

۳-۳-۲- تحلیل حساسیت فراپارامترها

با پایان توضیح عملکرد کل سامانه تشخیص بارش با توجه به توالی، لازم است در مورد نحوه‌ی تعیین معماری‌های شبکه‌های عصبی بازگشتی و فراپارامترهای آن‌ها هم توضیحاتی داده شود. بدین منظور، برای هر فراپارامتر تعداد مختلفی متغیر کاندید شده و سامانه به ازای این متغیرها اجراشده و بهترین این متغیرها بر اساس عملکرد نهایی سامانه روی مجموعه دادگان با استفاده از معیارهای ارزیابی که در فصل بعد با

آن‌ها آشنا می‌شویم، انتخاب‌شده‌اند. قبل از ادامه دادن بحث به‌صورت خلاصه روند تحلیل حساسیت فراپارامترها را شرح می‌دهیم:

۱. **معماری شبکه:**

برای تعیین معماری شبکه‌ها، تعداد مختلفی از لایه‌ها آزمایش‌شده‌اند. نتیجه به‌دست‌آمده، نشان داد که یک لایه با شبکه عصبی بازگشتی ساده بهترین نتیجه را برمی‌گرداند.

۲. **توابع فعال‌ساز:** برای انتخاب بهترین تابع فعال‌ساز، توابعی مانند سیگموید، تابع هذلولوی تانژانت، واحد یکسو شده‌ی خطی[۱]، واحد یکسو شده‌ی خطی با نشت[۲] و سافتمکس[۳] بررسی‌شده‌اند که بهترین عملکرد توسط تابع فعال‌ساز خطی به دست آمد.

۳. **بهینه‌ساز:**

برای انتخاب بهترین بهینه‌ساز نیز روش‌های مختلفی مانند آدام، گرادیان کاهشی، پس انتشار مبتنی بر ریشه میانگین مربعات[۴]، دلتای تطبیقی[۵] و گرادیان تطبیقی[۶] بررسی‌شده‌اند که بهترین عملکرد توسط بهینه‌ساز آدام به دست آمد.

۴. **تابع زیان:**

با توجه به مسئله با ناظر بودن مدل و دودویی بودن برچسب‌های مجموعه دادگان، تابع زیان آنتروپی متقاطع دودویی استفاده شد و از بررسی توابع زیان دیگر پرهیز گردید.

۵. **اندازه پنجره:**

برای در نظر گرفتن ارتباط بارندگی روز فعلی با بارندگی روزهای قبلی، از پنجره‌بندی استفاده شد و پس از بررسی اندازه‌های مختلف، این نتیجه حاصل شد که اندازه پنجره ۳ نتیجه خوبی را دارد.

---

[1] Rectified Linear Unit (Relu)

[2] Leaky ReLU

[3] Softmax

[4] Root Mean Square Propogation (RMSprop)

[5] Adaptive delta (Adadelta)

[6] Adaptive Gradient (Adagrad)

فصل چهارم

نتایج عددی

پیاده‌سازی، نتایج عددی و مقایسه روش‌ها

در این فصل، سامانه پیشنهادی فصل قبل را روی دو مجموعه دادگان واقعی تراکنش‌های بارندگی پیاده می‌کنیم. سپس، نتایج به‌دست‌آمده را به کمک تعدادی از معیارهای ارزیابی مناسب می‌سنجیم.

۴-۱- مجموعه دادگان

برای ارزیابی سامانه پیشنهادی، از مجموعه دادگان شامل بارش‌های واقعی استرالیا بهره می‌گیریم. در این بخش، مشخصات و ویژگی‌های آن‌ها را شرح می‌دهیم.

دیتاست Rain in Australia در سایت Kaggle موجود است و حاوی حدود ۱۰ سال مشاهدات روزانه آب‌وهوا از مکان‌های مختلف در سراسر استرالیا است

در ادامه به برخی از ویژگی‌ها و جزئیات این دیتاست اشاره می‌کنیم:

تعداد نمونه‌ها: این دیتاست شامل ۱۴۲,۱۹۳ نمونه است.

تعداد ویژگی‌ها: در این دیتاست ۲۳ ویژگی وجود دارد.

ویژگی‌ها:

در این دیتاست، ویژگی‌های مربوط به باد، رطوبت، فشار و دما به شرح زیر است:

WindGustDir (جهت باد وزشی): جهتی که باد وزشی (افزایش ناگهانی سرعت باد) از آن می‌آید.

WindGustSpeed (سرعت باد وزشی): سرعت باد وزشی به کیلومتر بر ساعت.

WindDir9am (جهت باد ساعت ۹ صبح): جهت باد در ساعت ۹ صبح.

WindDir3pm (جهت باد ساعت ۳ بعدازظهر): جهت باد در ساعت ۳ بعدازظهر.

WindSpeed9am (سرعت باد ساعت ۹ صبح): سرعت باد در ساعت ۹ صبح به کیلومتر بر ساعت.

WindSpeed3pm (سرعت باد ساعت ۳ بعدازظهر): سرعت باد در ساعت ۳ بعدازظهر به کیلومتر بر ساعت.

Humidity9am (رطوبت ساعت ۹ صبح): رطوبت نسبی هوا در ساعت ۹ صبح به درصد.

Humidity3pm (رطوبت ساعت ۳ بعدازظهر): رطوبت نسبی هوا در ساعت ۳ بعدازظهر به درصد.

Pressure9am (فشار هوا ساعت ۹ صبح): فشار هوا در ساعت ۹ صبح به هکتوپاسکال.

Pressure3pm (فشار هوا ساعت ۳ بعدازظهر): فشار هوا در ساعت ۳ بعدازظهر به هکتوپاسکال.

Cloud9am (ابر ساعت ۹ صبح): پوشش ابر در ساعت ۹ صبح به درصد.

Cloud3pm (ابر ساعت ۳ بعدازظهر): پوشش ابر در ساعت ۳ بعدازظهر به درصد.

Temp9am (دمای ساعت ۹ صبح): دمای هوا در ساعت ۹ صبح به درجه سانتی‌گراد.

Temp3pm (دمای ساعت ۳ بعدازظهر): دمای هوا در ساعت ۳ بعدازظهر به درجه سانتی‌گراد.

RainToday (باران امروز): نشان‌دهنده‌ی اینکه آیا امروز باران باریده است یا خیر.

این ویژگی‌ها اطلاعات مفیدی در مورد شـــرایط آب‌وهوایی در اســـترالیا ارائه می‌دهند و می‌توانند برای پیش‌بینی بارش‌های باران کاربرد داشته باشند.

مقادیر گمشـــده: این دیتاســت دارای مقادیر گمشـــده اســت و برخی از ویژگی‌ها تعداد قابل‌توجهی مقادیر گمشده دارند.

انواع داده‌ها: ویژگی‌ها دارای انواع مختلف داده‌ها هستند، ازجمله عددی، رده‌ای و تاریخ/زمان.

در کل، دیتاست Rain in Australia منبع ارزشمندی برای مطالعه الگوهای آب‌وهوایی و توسعه مدل‌های پیش‌بینی بارش است.

۴-۲- معیارهای ارزیابی

در این قسمت، ۴ معیار ارزیابی مناسب برای سنجیدن کیفیت دسته‌بندی مطرح می‌کنیم. در این پایان‌نامه این چهار معیار مختلف را تحلیل و بررسی خواهیم کرد و برای ارزیابی الگوریتم‌ها و سامانه پیشنهادی این پایان‌نامه برای حل مسئله، از آن‌ها استفاده و نتایج به‌دست‌آمده را باهم مقایسه می‌کنیم.

با در نظر داشتن نمادهای TP[۱](مثبت واقعی)، TN[۲](منفی واقعی)، FP[3] (مثبت کاذب) و FN[۴](منفی کاذب) در این بخش چهار معیار معرفی‌شده به شرح ذیل هستند:

۱. معیار دقت: تعداد پیش‌بینی‌های صحیح ساخته‌شده توسط مدل نسبت به انواع پیش‌بینی‌های انجام‌شده است. دقت، یک معیار عالی برای سنجش کارکرد مدل است اما وقتی مجموعه دادگان متوازن و متعادل باشد. بااین‌حال، وقتی داده‌ها مانند مجموعه دادگان موردمطالعه ما متعادل نباشند، معیار دقت، اثربخشی دسته‌بند را جلب نمی‌کند، زیرا ممکن است مدل تحت تأثیر داده قرار بگیرد و در تصمیم‌گیری تنها با پیش‌بینی عدم بارش به دقت بالایی برسد و دچار تناقض دقت شود.

$$\text{دقت} = \frac{TP + TN}{TP + FP + TN + FN}$$

۲. معیار فراخوانی: معیاری است که به ما می‌گوید چه نسبت ادعاهایی که در واقع بارش بوده‌اند، توسط الگوریتم به‌عنوان بارش پیش‌بینی‌شده است.

$$\text{فراخوانی} = \frac{TP}{TP + FN}$$

در بسیاری از حوزه‌های تشخیص داده‌های نامتوازن، شناسایی این داده‌ها تا حد امکان بسیار مهم‌تر از سوگیری نمونه‌های عادی به‌عنوان داده‌های پرت است.

۳. معیار صحت: معیاری است که به ما می‌گوید چه نسبت ادعاهایی که در واقع بارش نبوده‌اند، توسط مدل پیش‌بینی‌شده است که بارشی نیست.

---

[1] True Positive

[2] True Negative

[3] False Positive

[4] False Negative

$$\text{صحت} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

۴. معیار F1: میانگین هارمونیک دو معیار فراخوانی و صحت است و از این معیار به‌عنوان معیار اصلی در ارزیابی و مقایسه‌های خود استفاده خواهیم کرد.

$$\boldsymbol{F1} = ۲ \cdot \frac{\textbf{فراخوانی . صحت}}{\textbf{فراخوانی} + \textbf{صحت}}$$

۴-۳- نتایج عددی و مقایسه

در این بخش، عملکرد رویکردهای پیشنهادی را در مجموعه دادگان به کمک چهار معیار تعریف‌شده ارزیابی و نتایج عددی حاصل را گزارش می‌کنیم. جدول ۱ معیارهای ارزیابی مختلف را روی داده آزمون نمایش می‌دهد.

جدول ۱- نتایج عددی

F1	فراخوانی	صحت	معیار دقت
۰.۹۵	۰.۹۴	۰.۹۵	۰.۹۸

فصل پنجم

جمع‌بندی پایان‌نامه

جمع‌بندی

در این پایان‌نامه، یک سامانه پیشنهادی برای تشخیص بارش مطرح شد. این سامانه با رویکرد تشخیص بارش با توجه به توالی معرفی‌شد. معماری سامانه به‌گونه‌ای است که هر مجموعه ویژگی‌های مختلف را به‌عنوان ورودی می‌تواند بپذیرد. به‌عبارت‌دیگر، سامانه به‌صورت جامع طراحی‌شده تا قابلیت پردازش ویژگی‌های مختلف را داشته باشد.

در قسمت پیش‌پردازش، سه‌گام اصلی روی‌داده‌های خام ورودی انجام گرفت. ابتدا داده‌ها پاک‌سازی و نرمال‌سازی شد. سپس با توجه به رویکرد مدل، داده آموزش و آزمایش از همدیگر جداشدند.

سپس در بخش مدل‌سازی به دلیل اهمیت از توالی شبکه‌های عصبی بازگشتی استفاده شد که بارندگی را با دقت خوبی تشخیص داد.

برای ارزیابی کیفیت سامانه‌های پیشنهادی، از چهار معیار دقت، صحت، فراخوانی و F1 استفاده شد. درنهایت سامانه پیشنهادی ما توانست به دقت ۹۸ درصد روی دیتاست بارندگی استرالیا برسد.

پیوست

کد پیاده سازی

```
# Rain in Australia
# Get Reproducible Results within the notebook
from numpy.random import seed
seed(42)
import tensorflow
tensorflow.random.set_seed(42)
import warnings
warnings.filterwarnings('ignore')

import pandas as pd

# I think it is much easier to import the dataset from my github
repository than from kaggle itself
data = 'weatherAUS.csv'
df = pd.read_csv(data)

df.shape

df.head()

df.info()

df.describe()

# Are there any missing values (NaN)?
df['RainToday'].isnull().sum()
df.dropna(subset=['RainToday'], inplace=True)

df = df.drop(['RainTomorrow'],axis=1)

# Which are the unique values?
```

```
df['RainToday'].unique()

df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
df.drop('Date', axis=1, inplace=True)
df.head()

#uniqe year
df['Year'].unique()

#split df[2015 2016 2017] for test
X_test = df[(df['Year'] == 2015) | (df['Year'] == 2016) | (df['Year'] ==
2017)]
X_train = df[(df['Year'] != 2015) & (df['Year'] != 2016) & (df['Year']
!= 2017)]

categorical = [var for var in df.columns if df[var].dtype=='O']
print('There are {} categorical variables\n'.format(len(categorical)))
print('The categorical variables are :', categorical)

Numerical = [var for var in df.columns if df[var].dtype!='O']
print('There are {} numerical variables'.format(len(Numerical)))
print('The numerical variables are :', Numerical)

for col in ["WindGustDir", "WindDir9am", "WindDir3pm"]:
    col_mode=X_train[col].mode()
    X_train[col].fillna(col_mode, inplace=True)
    X_test[col].fillna(col_mode, inplace=True)

#label encoder
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
df['RainToday'] = le.fit_transform(df['RainToday'])

for col in Numerical:
    col_median=X_train[col].median()
    X_train[col].fillna(col_median, inplace=True)
    X_test[col].fillna(col_median, inplace=True)

categorical = [var for var in X_train.columns if
X_train[var].dtype=='O']
# label encoder
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for var in categorical:
    X_train[var] = le.fit_transform(X_train[var])
    X_test[var] = le.transform(X_test[var])

X_train.shape, X_test.shape

correlation = X_train.corr()

#matplotlib correlation matrix
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True,fmt='.1g')

y_train=X_train['RainToday']
y_test=X_test['RainToday']
X_train=X_train.drop(['RainToday'],axis=1)
X_test=X_test.drop(['RainToday'],axis=1)

X_train.shape, X_test.shape
```

```
y_train.shape, y_test.shape

X_train.dtypes

"""### Feature Scaling"""

X_train.describe()

from sklearn.preprocessing import StandardScaler
cols = X_train.columns
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
X_train = pd.DataFrame(X_train, columns=[cols])
X_test = pd.DataFrame(X_test, columns=[cols])

"""Now all the features are scaled between 0 and 1

### Model Training
"""

import numpy as np

# Assuming you have your X_train and X_test datasets available
# X_train.shape: (98987, 23)
# X_test.shape: (43212, 23)

# Define the number of time steps (sequence length) and the number of
features
time_steps = 3
num_features = 23
```

```
# Reshape the data to fit into LSTM model (input shape: (samples,
time_steps, num_features))

# Reshape X_train
num_samples_train = X_train.shape[0] - time_steps + 1
X_train_reshaped = np.zeros((num_samples_train, time_steps,
num_features))
for i in range(num_samples_train):
    X_train_reshaped[i] = X_train[i:i+time_steps]

# Reshape X_test
num_samples_test = X_test.shape[0] - time_steps + 1
X_test_reshaped = np.zeros((num_samples_test, time_steps, num_features))
for i in range(num_samples_test):
    X_test_reshaped[i] = X_test[i:i+time_steps]

# Now 'X_train_reshaped' and 'X_test_reshaped' are 3D numpy arrays
suitable for LSTM input
# They have shapes (num_samples_train, 3, 23) and (num_samples_test, 3,
23) respectively.

X_train.shape, y_train.shape, X_test.shape, y_test.shape

# Assuming you have X_train, y_train after windowing
# X_train.shape: (num_samples_train, time_steps, num_features)
# y_train.shape: (num_samples_train,)

# Assuming you have also windowed X_test and y_test if applicable
# X_test.shape: (num_samples_test, time_steps, num_features)
# y_test.shape: (num_samples_test,)

# Ensure the last few elements that do not form a complete sequence are
removed
```

```
num_samples_train = X_train_reshaped.shape[0]
num_samples_test = X_test_reshaped.shape[0]

num_complete_sequences_train = num_samples_train - time_steps + 1
num_complete_sequences_test = num_samples_test - time_steps + 1

X_train_reshaped = X_train_reshaped[:num_complete_sequences_train]
y_train = y_train[:num_complete_sequences_train]

X_test_reshaped = X_test_reshaped[:num_complete_sequences_test]
y_test = y_test[:num_complete_sequences_test]

X_train_reshaped.shape

import numpy as np
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense

# Prepare the target values (y_train and y_test) for training and
testing your LSTM model

# Create the LSTM model
model = Sequential()

# Add an LSTM layer with 8 units
model.add(SimpleRNN(1, input_shape=(time_steps, num_features)))

# Add a fully connected (dense) layer to output the prediction
model.add(Dense(1, activation='linear'))  # Assuming you are predicting
a continuous value

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
# Train the model
model.fit(X_train_reshaped, y_train, epochs=5, batch_size=32)

# Make predictions on the test set
predictions = model.predict(X_test_reshaped)

# You can now use the 'predictions' variable to evaluate the performance
of the model on the test set.

y_test.shape, predictions.shape

#classification report
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions>0.5)
```

منابع و مراجع

[1] Parmar, A., Mistree, K., & Sompura, M. (2017, March). Machine learning techniques for rainfall prediction: A review. In International conference on innovations in information embedded and communication systems (Vol. 3).

[2] Abhishek, K., Kumar, A., Ranjan, R., & Kumar, S. (2012, July). A rainfall prediction model using artificial neural network. In 2012 IEEE Control and System Graduate Research Colloquium (pp. 82-87). IEEE.

[3] Nayak, D. R., Mahapatra, A., & Mishra, P. (2013). A survey on rainfall prediction using artificial neural network. International journal of computer applications, 72(16).

[4] Hernández, E., Sanchez-Anguix, V., Julian, V., Palanca, J., & Duque, N. (2016). Rainfall prediction: A deep learning approach. In Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings 11 (pp. 151-162). Springer International Publishing.

[5] Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. Design and Applications, 5(64-67), 2.

[6] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

[7] Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. IEEE transactions on neural networks, 5(2), 240-254.

[8] Chu, C. S. J. (1995). Time series segmentation: A sliding window approach. Information Sciences, 85(1-3), 147-173.

[9] Hota, H. S., Handa, R., & Shrivas, A. K. (2017). Time series data prediction using sliding window based RBF neural network. International Journal of Computational Intelligence Research, 13(5), 1145-1156.

[10] Gupta, L., McAvoy, M., & Phegley, J. (2000). Classification of temporal sequences via prediction using the simple recurrent neural network. Pattern Recognition, 33(10), 1759-1770.

[11] Pham, B. T., Le, L. M., Le, T. T., Bui, K. T. T., Le, V. M., Ly, H. B., & Prakash, I. (2020). Development of advanced artificial intelligence models for daily rainfall prediction. Atmospheric Research, 237, 104845.

[12] Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. Machine Learning with Applications, 7, 100204.

[13] Basha, C. Z., Bhavana, N., Bhavya, P., & Sowmya, V. (2020, July). Rainfall prediction using machine learning & deep learning techniques. In 2020 international conference on electronics and sustainable communication systems (ICESC) (pp. 92-97). IEEE.

[14] Diez-Sierra, J., & Del Jesus, M. (2020). Long-term rainfall prediction using atmospheric synoptic patterns in semi-arid climates with statistical and machine learning methods. Journal of Hydrology, 586, 124789.

Abstract

Agriculture is the key point for survival. Rainfall is very important for agriculture. Rainfall forecasting has become a big problem these days. Rainfall forecasting informs people and they are informed in advance of rain so that they can take special precautions to protect their crops from rain. Many techniques have been developed to predict rainfall. Machine learning algorithms are more useful in rainfall forecasting. This thesis aims to detect rainfall on the Australian dataset using a simple RNN model. In this research, firstly, Australian meteorological data has been pre-processed. Then the simple RNN model is trained using these data. The trained model is evaluated with experimental data and 98% accuracy in rainfall detection is achieved. This thesis presents acceptable results in the area of rainfall detection on the Australian dataset using a simple RNN model. The results of this research can help researchers and those interested in meteorology and weather forecasting .

Key Words: Rainfall Prediction, Simple RNN, Australian Rainfall Dataset, Meteorology, Weather Forecasting.

Bachelor's Thesis

Rainfall prediction

By
Omid Aghili

Supervisor
Dr. Nazari

August 2023