

WPROWADZANIE

Pogoda ma ogromny wpływ na nasze codzienne życie. Istnieje wiele aplikacji, które wymagają dokładnej prognozy pogody. Niektóre z zastosowań to rolnictwo i produkcja, operacje wojskowe, przemysł lotniczy, transport, komunikacja i tak dalej. Prognozowanie pogody jest kanonicznym wyzwaniem predykcyjnym, które wymaga wiedzy z wielu dyscyplin. Dynamiczny i chaotyczny charakter atmosfery czyni przewidywanie bardziej złożonym zadaniem. Wymaga większej mocy obliczeniowej, aby rozwiązać złożone równania matematyczne związane z atmosferą. To otwiera nową erę badań w prognozowaniu pogody, które będą działać lepiej pod względem kosztów obliczeniowych i dokładności prognoz. Ze względu na tak duże zastosowanie, badacze skłaniają się ku temu polu przez przewidywanie parametrów pogodowych, takich jak wilgotność, ciśnienie, prędkość wiatru, kierunek wiatru, parowanie, zanieczyszczenie itp. Wśród tych wszystkich parametrów bardzo istotnym parametrem jest temperatura, ponieważ od niej zależy wiele innych parametrów. Dodatkowo ma duży wpływ nie tylko na życie ludzkie, ale także na regionalną działalność rolniczą i firmy użyteczności publicznej do szacowania zapotrzebowania w nadchodzących dniach itp. Obecnie prawie wszystkie stacje pogodowe są podłączone do Internetu, co zapewnia dane meteorologiczne w czasie rzeczywistym. Dane w czasie rzeczywistym to coś, co jest dostarczane natychmiast po ich zebraniu. Systemy czasu rzeczywistego są używane na całym świecie w takich zastosowaniach, jak monitorowanie ruchu, dostarczanie treści informacyjnych, wykrywanie oszustw w domenie finansowej, prognozowanie pogody itp. Zatem przewidywanie temperatury w czasie rzeczywistym jest kluczowym wyzwaniem. Metody prognozowania pogody można ogólnie podzielić na modele empiryczne i modele predykcyjne. Modele empiryczne wymagają dużej ilości obliczeń, podczas gdy modele predykcyjne wymagają dużej ilości danych. Ze względu na rosnące wykorzystanie komputerów dane internetowe są generowane ogromnie, a ze względu na łatwą dostępność mocy przetwarzania chipów i przystępne techniki przechowywania danych, modele prognostyczne są dość popularne wśród badaczy w porównaniu z modelami empirycznymi, ponieważ są całkowicie uzależnione od danych i opracować realistyczny model. Różne techniki, takie jak regresja liniowa, autoregresja, sieci wielowarstwowych perceptronów (MLP) i radialnych funkcji bazowych (RBF) są stosowane do przewidywania parametrów atmosferycznych, takich jak temperatura, prędkość wiatru, opady, zanieczyszczenie meteorologiczne itp. Oprócz tego narzędzia do eksploracji danych, takie jak wsparcie maszyna wektorowa (SVM), drzewa decyzyjne (CART), sztuczna sieć neuronowa (ANN) i klastry są wykorzystywane do badania ukrytych wzorców i trendów w tak ogromnym i statycznym zestawie danych meteorologicznych. Jednak metody eksploracji danych mają pewne ograniczenia, gdy są używane do szybkiej analizy strumieni danych, które są niestacjonarne, chaotyczne i niespójne lub mają dryf koncepcyjny. Do tej kategorii należy pogoda, ponieważ jej natura jest niestacjonarna. Wyżej wymienione techniki są całkiem odpowiednie dla danych statycznych, ale w czasie rzeczywistym dane przewidywane taką techniką to prawdziwa motywacja i duże wyzwanie. W kolejnych sekcjach zostaną szczegółowo omówione powiązane prace w zakresie prognozowania pogody z wykorzystaniem modeli uczenia maszynowego wraz z ograniczeniami i modelem głębokiego uczenia wykorzystywanym do prognozowania temperatury w czasie rzeczywistym.

PRACA POWIĄZANA

W celu przewidzenia trendu pogodowego ważnym czynnikiem jest czas, a zatem szeregi czasowe są w tym kontekście podstawowym pojęciem. Niezbędna jest znajomość trzech głównych cech szeregów czasowych, a mianowicie autokorelacji, sezonowości i stacjonarności. Stacjonarne szeregi czasowe są idealne do modelowania, ale uzyskanie takiego modelu jest naprawdę bardzo trudne, więc różne przekształcenia, takie jak średnia ruchoma (MA), wygładzanie wykładnicze (ES) i zintegrowana autoregresyjna średnia ruchoma (ARIMA), są wykorzystywane do budowania lepszego modelu

predykcyjnego. W tej sekcji opisano różne modele uczenia maszynowego, które przewidują dane pogodowe.

WIELOKROTNY MODEL REGRESJI LINIOWEJ (MLR)

MLR opisuje liniową zależność między zbiorem zmiennych niezależnych (X_1, X_2, \dots, X_k) a zmienną niezależną (Y) i jest ogólnie przedstawiana jako

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

Tutaj Y jest predyktorem, a X_1, X_2, \dots, X_k są predyktorami; $\beta_0, \beta_1, \dots, \beta_k$ są nazywane współczynnikami regresji, a ε jest składnikiem błędu standardowego. W celu opracowania modelu obliczane są współczynniki regresji, a następnie obliczana jest wartość Y . Rzeczywiste i przewidywane wyniki są porównywane w celu zidentyfikowania terminu błędu. Ten prosty model jest rygorystycznie stosowany w wielu procesach prognostycznych. Paras i Mathur opracowali model statystyczny oparty na MLR i przewidzieli różne parametry pogodowe. Do prognozowania zbierano lokalnie warunki pogodowe dla danej stacji. Dane zostały przetworzone w celu uzyskania pewnych wskaźników statystycznych w celu wydobycia ukrytych informacji obecnych w szeregach czasowych. Te wskaźniki statystyczne, tj. MA, wykładnicza średnia krocząca (EMA), tempo zmian (ROC), OSCillator (OSC), momenty (μ_2, μ_3 i μ_4) oraz współczynniki skośności i kurtozy, zostały obliczone dla pewnych okresów. Na podstawie korelacji wybrano cechy jako dane wejściowe do modeli i otrzymano równania regresji do przewidywania parametrów pogodowych. Sreehari opracował model prognozowania opadów oparty na MLR. Amral opracował model oparty na MLR, aby przewidzieć zapotrzebowanie systemu elektroenergetycznego na obciążenie krótkoterminowe (do 24 godzin) w południowym regionie Sulawesi. Patrząc na łatwość tego modelu, ten model jest łatwy do wdrożenia; jednak ten model jest całkiem odpowiedni, jeśli istnieje liniowa zależność między predyktorami a predyktorem; w środowisku czasu rzeczywistego wszystkie zmienne mają relację nieliniową. Dodatkowo brakujące dane lub nieprawidłowe wartości mogą utrudnić dokładność modelu, dlatego modele oparte na SSN zyskują coraz większą uwagę.

Sztuczna sieć neuronowa (ANN)

Podstawową jednostką obliczeniową w sieci neuronowej jest neuron lub perceptron symulujący zachowanie ludzkiego mózgu. Wejścia neuronowe odpowiadają wartościom zmiennych predykcyjnych pogody. Neurony warstwy ukrytej i warstwy wyjściowej akceptują dowolną liczbę wejść w zależności od typu synapsy wybranego do połączenia neuronów. Najpopularniejszą strategią jest wykorzystanie synaps ważonych, które łączą każdy neuron w warstwie źródłowej z każdym neuronem w warstwie docelowej w celu utworzenia sieci ze sprzężeniem do przodu. Komponent funkcji aktywacji skaluje wyjście z ukrytej warstwy do użytecznej wartości. Obliczona wartość jest porównywana z wartością rzeczywistą, a składnik błędu jest obliczany i propagowany wstecznie w całej sieci, a ten typ algorytmu nazywa się algorytmem wstecznej propagacji błędów. Kumar dokonał przeglądu prac wykonanych przez różnych badaczy korzystających z SSN, opracował model SNN, który zmniejsza koszty przetwarzania danych poprzez zwiększenie liczby warstw ukrytych, i zaobserwował zachowanie większej liczby warstw ukrytych w zakresie wydajności i uogólnienia modelu. Chattopadhyay i Chattopadhyay zaproponowali model ANN do prognozowania opadów w porze monsunowej dla indyjskich danych meteorologicznych. Do przeprowadzenia tego eksperymentu wykorzystano wyłącznie dane szeregów czasowych opadów. Model oparty na ANN okazał się lepszy od modelu MLR, ale nadal nie nadawał się do prognozowania pogody w czasie rzeczywistym. W związku z tym modele oparte na głębokim uczeniu się zyskują obecnie wiele uwagi. Shabana i Bewoor szczegółowo

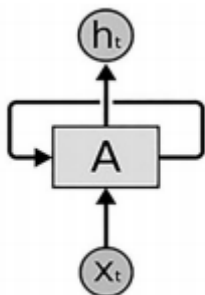
przeanalizowali różne modele głębokiego uczenia wykorzystywane do analizy danych w czasie rzeczywistym. Kolejna sekcja ujawnia te modele głębokiego uczenia się.

MODELE DO GŁĘBOKIEJ NAUKI

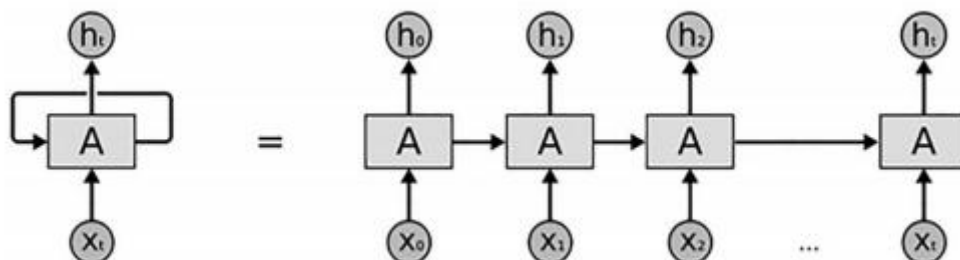
Sieci neuronowe ze sprzężeniem do przodu dają bardzo dobre wyniki, gdy są stosowane do rozwiązywania problemów z klasyfikacją, ale mogą zawieść, jeśli chodzi o dane szeregów czasowych. Powodem tego jest to, że zawierają statyczne mapowanie wektora wejściowego na wektor wyjściowy. Jednak problemy czasowe polegają na dynamicznie zmieniających się stanach, których należy się nauczyć podczas reprezentacji modelu. Jednym z przykładów tej czasowej natury jest prognoza pogody, w której podczas przewidywania przyszłych warunków pogodowych konieczne jest uwzględnienie parametrów pogody z przeszłości. Problemy te są również nazywane problemami przewidywania sekwencji, w których sygnał wejściowy do sieci neuronowej jest sekwencją wektorów cech, a oczekiwanym wynikiem jest prawidłowe przewidywanie nadchodzącego parametru sekwencji. W problemach przewidywania sekwencji prezentacja wektora wejściowego sieci neuronowej określonej sekwencji odbywa się w różnych krokach czasowych. Jest to główna przyczyna problemów z przewidywaniem sekwencji, które pociągają za sobą zastąpienie statycznej sieci neuronowej ze sprzężeniem do przodu modelem sieci neuronowej, który ujawnia nieliniowe zachowanie dynamiczne. Rekurencyjna sieć neuronowa (RNN) posiada tę właściwość.

Rekurencyjne sieci neuronowe

RNN wykazuje wysoce nieliniowe zachowanie dynamiczne, ponieważ jest najpotężniejszym i najbardziej niezawodnym typem architektury sieci neuronowej. RNN to naturalne uogólnienie sprzężenia zwrotnego sieci neuronowej na sekwencje. Architektura RNN zawiera pętlę w sieci, która pomaga w trwałości informacji. RNN to jedyne sieci, które mają pamięć wewnętrzną. Powodem preferowania RNN do sekwencyjnego przewidywania danych są RNN, które są w stanie zapamiętać ważne punkty dotyczące dostarczanych im danych wejściowych. Ta właściwość sprawia, że RNN może być bardzo specyficzny w przewidywaniu następnego wartości. RNN jest z powodzeniem wdrażany dla danych sekwencyjnych, tj. szeregów czasowych, głosu, tekstu, wideo, dźwięku, mowy, informacji finansowych, treści wiadomości, pogody i tak dalej. RNN może zbudować bardzo intensywne zrozumienie sekwencji i jej kontekstu. Dlatego wszędzie tam, gdzie znacznik czasu każdej ramki danych w sekwencji danych jest ważniejszy niż strukturalna zawartość pojedynczej ramki, należy zastosować RNN. Dane sekwencyjne to po prostu dane ciągłe, w których wzajemnie powiązane lub tego samego rodzaju rzeczy następują po sobie. Niektóre przykłady takich danych to sekwencja DNA lub informacje finansowe. Istnieje bardzo popularny rodzaj danych sekwencyjnych, tj. dane szeregów czasowych, które są po prostu szeregiem punktów danych uporządkowanych w czasie. Tradycyjne sieci neuronowe, takie jak sieć neuronowa ze sprzężeniem do przodu, nie były zdolne do utrwalania informacji. RNN zarządza tym problemem za pomocą pętli w swojej sieci. Na rysunku 1 sieć neuronowa pobiera trochę danych wejściowych x_t i daje wartość wyjściową h_t .



W sieci istnieje pętla w punkcie A, która jest powodem przekazywania informacji z jednego kroku do drugiego. Z rysunku 2 jasno wynika, że RNN to nic innego jak wielokrotne powtórzenia tej samej sieci neuronowej, w której każda sieć przekazuje wiadomość do potomka.



Aktualny stan sieci obliczany jest według wzoru:

$$h_t = f(h_{t-1}, x_t)$$

gdzie

h_t = stan bieżący

h_{t-1} = poprzedni krok

x_t = stan wejścia

Wzór na funkcję aktywacji:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

gdzie

W_{hh} = waga w neuronie powtarzalnym

W_{xh} = waga na neuronie wejściowym

Wydajność sieci obliczana jest według następującego wzoru:

$$y_t = W_{hy}h_t$$

gdzie

y_t = wyjście

W_{hy} = waga w warstwie wyjściowej

Podczas uczenia modelu przez RNN brane są pod uwagę następujące kroki:

1. x_t jest dostarczany do sieci, co jest pojedynczym krokiem czasowym wejścia.
2. Aktualny stan, tj. h_t , jest obliczany za pomocą bieżącego wejścia i poprzedniego stanu.
3. Dla następnego kroku czasowego bieżące h_t traktowane jest jako h_{t-1} .

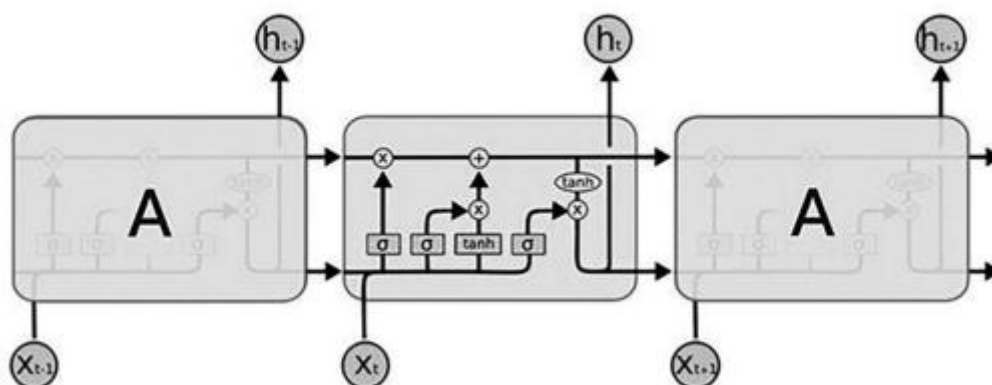
4. Ta procedura jest wykonywana w tylu krokach, ile wymaga problem. Następnie połącz dane ze wszystkich poprzednich kroków.
 5. Następnie y_t , który jest bieżącym krokiem końcowym, jest obliczane po zakończeniu przetwarzania wszystkich kroków czasowych.
 6. Na koniec wytworzona wartość wyjściowa jest porównywana z rzeczywistą wyjściową w celu obliczenia błędu.
 7. Wartość błędu jest następnie propagowana wstecznie do sieci i odpowiednio aktualizowane są wagi.
- RNN jest szkolony w ten sposób.

Problem z powtarzającą się siecią neuronową

Szkolenie w RNN składa się z trzech głównych części. Początkowo sieć wykonuje przejście do przodu i przewiduje pewną wartość. Po drugie, oblicza funkcję straty. Dane wyjściowe funkcji strat wskazują, jak słabo działa sieć. Na koniec propagowano wstecznie wartość błędu, aby zmierzyć gradienty dla każdego węzła w sieci. Wagi wewnętrzne w sieci są dostosowywane w zależności od wartości gradientu. Gradient to punkt decyzyjny, w jaki sposób sieć będzie się uczyć. Jeśli nachylenie jest duże, oznacza to, że wymagana jest większa regulacja i odwrotnie. Stwarza to problem, że jeśli gradient do poprzednich warstw jest mały, to dopasowanie wagi do bieżącej warstwy staje się jeszcze mniejsze. W ten sposób gradient zmniejsza się, gdy przechodzi przez sieć, i ostatecznie sieć przestaje się uczyć. Ten problem nazywa się problemem znikającego gradientu. Ten problem powoduje, że RNN nie uczy się długoterminowych zależności w krokach czasowych; tj. RNN mają pamięć krótkotrwałą.

Pamięć krótkoterminowa sieci LSTM (LSTM)

Sieć neuronowa LSTM jest rozszerzeniem RNN, które rozszerza jego pamięć. LSTM może uczyć się zależności dalekiego zasięgu. LSTM zawiera podobną strukturę łańcuchową jak RNN; jedyną różnicą jest liczba warstw sieci neuronowej. LSTM składa się z czterech warstw zamiast jednej warstwy. Jak pokazano na rysunku 3, powtarzający się moduł w LSTM zawiera cztery wzajemnie oddziałujące warstwy.

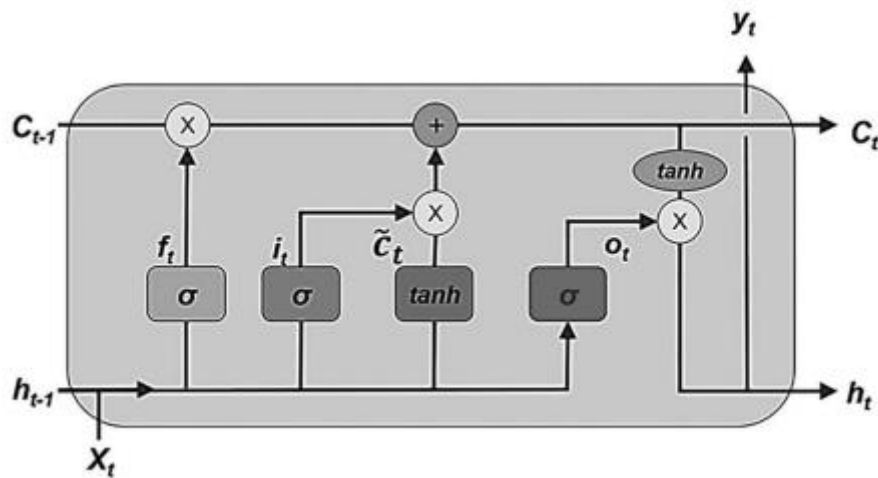


LSTM ma bramki do ochrony i kontroli stanu komórki. LSTM zawiera następujące składniki:

1. Brama zapomnij, 2. bramka wejściowa i 3. bramka wyjściowa

1. Bramka Zapomnij: Bramka ta służy do podejmowania decyzji, co należy usunąć z poprzedniego stanu, tj. $h_{(t-1)}$. Decyzję podejmuje się po otrzymaniu wyjścia stanu poprzedniego i wejścia bieżącego

x_t . Wizualna reprezentacja komórki LSTM jest pokazana na rysunku 4. Jest to warstwa sigmoidalna i daje wynik od 0 do 1.



2. Bramka wejściowa: Bramka wejściowa pobiera poprzednie wyjście i następne wejście i przepuszcza je przez kolejną warstwę sigmoidalną. Zwraca wartość z zakresu od 0 do 1. Wynik tej bramki jest mnożony przez wynik warstwy kandydującej.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$c_t = \tanh(W_c * [h_{t-1}, x_t] + b_c)$$

Ta warstwa stosuje tangens hiperboliczny, aby zmieszać dane wejściowe i poprzednie dane wyjściowe, zwracając wektor kandydujący do dodania do stanu wewnętrznego. Stan wewnętrzny jest aktualizowany według następującej formuły:

$$c_t = f_t * c_{t-1} + i_t * c_t$$

3. Bramka wyjściowa: Bramka wyjściowa kontroluje, jaka część stanu wewnętrznego jest przekazywana do wyjścia.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh c_t$$

RNN i LSTM to popularne modele używane do prognozowania pogody. Jednak modele hybrydowe można zastosować, aby uzyskać dokładniejsze wyniki. Ponadto modele te są bardziej kosztowne obliczeniowo, ale można je wykonywać w środowisku równoległym.

WNIOSKI

Uzyskanie bardziej aktualnych wyników w zależności od danych w czasie rzeczywistym wymaga godziny. Obecnie techniki głębokiego uczenia, takie jak RNN i LSTM, dają lepsze wyniki w przewidywaniu danych w czasie rzeczywistym. W przyszłości hybrydowe modele głębokiego uczenia mogą być z powodzeniem wykorzystywane do przewidywania parametrów pogodowych, takich jak

prędkość wiatru, wilgotność itp. Można je również zintegrować z istniejącymi już aplikacjami, takimi jak aplikacja internetowa, rolnictwo, produkcja, decyzje biznesowe itp. można wywnioskować, że technika głębokiego uczenia się oparta na sztucznej inteligencji może zastąpić ludzkie i tradycyjne metody w prognozowaniu pogody. Wydajność modelu można również zwiększyć, gdy zostanie zaimplementowany na górze GPU.