**Weather Prediction**

Projekt na przedmiot Sztuczna Inteligencja

Jakub Stachowicz

Krzysztof Pecyna

Marcin Stenka

Spis treści:

1. [Opis problemu](#OpisProblemu)
2. [Teoretyczny opis użytych metod](#TeoretycznyOpisUżytychMetod)
3. [Opis realizacji zadania](#OpisRealizacjiZadania)
4. [Prezentacja osiągniętych wyników](#PrezentacjaOsiągniętychWyników)
5. [Dyskusja](#Dysusja)
6. Opis Problemu

Weather Prediction (przewidywanie pogody) to problem polegający na wyznaczeniu temperatury kolejnych dni na podstawie danych historycznych pobranych ze strony [danepubliczne.imgw.pl](https://danepubliczne.imgw.pl/). Trudność problemu polega na przewidzeniu zachowania przyrody, która jest niemonotonna i nie zawiera zależności. Chcemy wygenerować dane jak najbardziej zbliżone do poprawnych wyników natomiast jesteśmy świadomi, że nie da się tego zrobić dokładnie.

1. Teoretyczny opis użytych metod
   1. **Linear Regression**

Model polegający na wyznaczenie funkcji liniowej (prostej), której funkcja straty MSE (mean square error) czyli tak zwana suma błędów kwadratu jest jak najmniejsza.

Na samym początku należy podzieli dane na zbiór treningowy do nauki modelu i zbiór testowy służący do faktycznego sprawdzania dokładności modelu. Najczęściej zbiory te dzielą się w proporcji 80 do 20 lub 67 do 33.

Aby wyznaczyć funkcję liniową najlepiej użyć metody gradientu prostego (gradient descent), która jest iteracyjną metodą poszukiwania parametrów optymalizujących funkcję straty. Gradientem określamy tempo wzrostu funkcji.

W metodzie tej potrzebny jest wektor θ, który przechowuje współczynniki naszej funkcji. Na początku trzeba go wypełnić wartościami losowymi - najlepiej z przedziału (0,1). Następnie w każdej iteracji należy obliczyć gradient dla wybranej funkcji kosztu przemnożoną przez współczynnik uczenia η (learning rate) i odjąć tę wartość od aktualnej wartości wektora θ.

A picture containing font, typography, white, calligraphy

Description automatically generated

Im MSE mniejszy tym model jest bardziej precyzyjny. Jeżeli MSE równałoby się 0 to oznaczałoby, że dla danych testowych model zwracałby wartości przewidywane równe wartościom rzeczywistym.

A black text on a white background

Description automatically generated with low confidence

gdzie:  
n - ilość danych  
 - wartość rzeczywista  
 - wartość przewidziana  
t - indeks danej

Czasami pojawia się problem pracy na bardzo dużych i bardzo małych wartościach jednocześnie. Sprawia to wiele problemów optymalizacyjnych i może doprowadzać błędnych wyników. Aby temu zapobiec należy przeprowadzić Standaryzację, której celem jest zbliżenie wartości do siebie według poniższego wzoru:

A picture containing font, number, white, line

Description automatically generated

gdzie:  
z - zestandaryzowane dane wyjściowe  
x - dane wejściowe  
µ - średnia wartości danych  
σ - odchylenie standardowe danych

Należy pamiętać, że w celu obliczenia średniej i odchylenia standardowego nie można wykorzystywać zbioru treningowego ze względu na ryzyko wycieku danych testowych (testing dataset leakage).

* 1. **Linear Autoregressive**

Parametryczny model szeregu czasowego, w którym nowsze dane mają większy wpływ na wyznaczanie współczynników funkcji liniowej niż dane starsze.

Model AR można zdefiniować za pomocą poniższego wzoru:

A picture containing font, text, white, typography

Description automatically generated

gdzie:   
c - stała  
 - i-ty parametr modelu  
- biały szum

* 1. **Linear Autoregressive with lag**

Model liniowy podobny do Linear Autoregressive z tą różnicą, że każdy dzień ma taką samą wartość. W modelu tym ważne jest ustalenie ilości danych wejściowych, które mają znaczący wpływ na dane wyjściowe.

* 1. **Feedforward artificial neural**

Jednokierunkowa sieć neuronowa

A diagram of a network

Description automatically generated with low confidence

Sieć ta składa się z:

* Input layer - dane wejściowe. Jest to macierz o wymiarach gdzie: N to ilość danych, a M to ilość cech
* Hidden layer - powłoka ukryta. Jest to wektor, który posiada sumę wartości z input layer przemnożoną przez wartości wagi każdego połączenia. Dodatkowo do obliczonej macierzy dodaje się jeszcze bias (obciążenie). Ukrytych powłok może być wiele o różnych rozmiarach.
* Output layer - dane wyjściowe. Jest to wektor posiadający dane obliczone przez nasz model. Wartość oblicza się w ten sam sposób co w powłoce ukrytej.

Wartość wagi mówi jak bardzo ma wpływ dany neuron, a obciążenie mówi jak bardzo wpływ ma dana powłoka.

A picture containing clock, diagram, circle, sketch

Description automatically generated

gdzie:  
 - dane wejściowe i-tego wiersza  
 - waga i-tego wiersza  
b - bias (obciążenie)  
z - waga z danego neuronu.

Sieć składa się z dwóch trybów pracy:

* Propagating forward - przewidywanie wprzód. Jest to wyznaczenie wartości przewidywanej na podstawie danych wejściowych, wartości wag i obciążeń
* Backpropagation - nauczanie sieci. Na podstawie obliczenia funkcji kosztu i gradientu prostego wyznacza się wartości wszystkich wag i obciążeń sieci.

W sieciach używa się funkcji aktywacyjnych (Activation functions), które określają zasady tworzenia funkcji. Najczęściej używane funkcje to:

* sigmoid - zwraca wartość z przedziału <0;1>

A graph with a blue line

Description automatically generated with low confidence

* tanh - funkcja hiperboliczna zwracająca wartości z przedziału <-1,1>

A picture containing line, diagram, plot, parallel

Description automatically generated

* ReLU - funkcja zwracająca wartości nie ujemne: max(0, <wartość>)

A graph with a blue line

Description automatically generated with low confidence

1. Opis realizacji zadania

Wzięliśmy za cel stworzenie kilku modeli liniowych i sieci neuronowych, aby porównać ich wyniki i sprawdzić ich zastosowania w obrębie problemu pogodowego.

Modele liniowe:

1. Linear Regression
2. Linear Autoregressive
3. Linear Autoregressive with lag

Model sieci neuronowej:

1. Feedforward artificial neural networks
2. Dane brane tylko z Gdańska do przewidywania temperatury dzień do przodu
3. Dane brane tylko z Gdańska do przewidywania temperatury dwa dni do przodu
4. Dane brane ze wszystkich ww. miejscowości do przewidywania temperatury dzień do przodu

Zostały użyte dane historyczne z lat 1966-2023.

Miasta z których były brane dane:

* Gdańsk
* Warszawa
* Białystok
* Bielsko-Biała
* Szczecin
* Wrocław

Cechy pogody, które były brane pod uwagę w algorytmach:

* Średnia temperatura
* Maksymalna temperatura
* Minimalna temperatura
* Suma opadów
* Typ opadów
* Zachmurzenie
* Prędkość wiatru
* Wilgotność
* Ciśnienie

Na podstawie tych wszystkich danych naszym celem było utworzenie jak najbardziej precyzyjnego algorytmu sztucznej inteligencji do przewidzenia kolejnych dni.

1. Prezentacja osiągniętych wyników
   1. **Linear Regression**

A picture containing screenshot, text, line, diagram

Description automatically generated

Tak przedstawia się utworzona funkcja. Jej wzór to 0 = [0.95691274]x + [0.41918928]y.

Funkcje kosztu modelu:



* 1. **Linear Autoregressive**

Porównanie wyników przewidzianych z rzeczywistymi:

A graph with red and blue lines

Description automatically generated with low confidence

Funkcje kosztu modelu:



* 1. **Linear Autoregressive with lag**

W tym modelu wykorzystali bibliotekę ”statsmodels” w celu znalezieniu odpowiedniej wartości lag, czyli ilości dni, które mają wpływ na temperaturę następnego dnia. Utworzyliśmy dwa wykresy, które powinny pokazać nam te informacje.

1. Funkcja autokorelacji ACF, która ocenia korelację między obserwacjami w szeregu czasowym w danym zakresie opóźnień:

A picture containing text, line, screenshot, plot

Description automatically generated

1. Funkcja częściowej autokorelacji PACF, która dla każdego opóźnienia jest unikalną korelacją między dwiema obserwacjami po usunięciu korelacji pośrednich:

A picture containing text, screenshot, line, plot

Description automatically generated

Dla obu wartości słupki, których wartość y jest w niebieskim prostokącie oznacza, że nie mają wystarczającego wpływu, aby brać je pod uwagę.

A picture containing screenshot, plot, line, font

Description automatically generated

Funkcje kosztu modelu:



* 1. **Feedforward artificial neural networks**

Danymi testowymi był ostatni rok z dataset’u, natomiast reszta służyła do trenowania sieci.

1. **Dane brane tylko z Gdańska do przewidywania temperatury dzień do przodu**

A picture containing screenshot, plot, line, text

Description automatically generated

Funkcje kosztu modelu:

Wyniki modelu dostosowują się do temperatury z dni poprzednich bez żadnych gwałtownych zmian. Model dobrze reaguje na szybkie małe zmiany, natomiast źle reaguje na nagłe zmiany prognozy.

1. **Dane brane tylko z Gdańska do przewidywania temperatury dwa dni do przodu**

A picture containing screenshot, plot, line, text

Description automatically generated

Funkcje kosztu modelu:



Model na samym początku danych testowych generouje gorsze wyniki, natomiast dla ostatnich miesięcy podobne, a nawet lepsze niż w modelu pierwszym.

1. **Dane brane ze wszystkich ww. miejscowości do przewidywania temperatury dzień do przodu**

A picture containing screenshot, plot, line, font

Description automatically generated

Funkcje kosztu modelu:



Model bardzo szybko dostosowuje się do nadchodzących zmian, chociaż tak jak poprzednie dwa modele źle reaguje na nagłe zmiany. Charakteryzuje się on najmniejszą wartości funkcji kosztu, czyli najmniejszym błędem przewidywania.

1. Dyskusja

W ramach tego projektu przeprowadziliśmy badania mające na celu porównanie efektywności modeli liniowych i sieci neuronowych w prognozowaniu temperatury dnia następnego na podstawie różnych cech. Badania obejmowały wykorzystanie trzech różnych modeli liniowych oraz trzech modeli sieci neuronowych.

Wyniki naszych badań wykazały, że modele liniowe oraz sieci neuronowe mają potencjał w przewidywaniu temperatury. Spośród modeli liniowych, najlepsze rezultaty osiągnął model liniowy autoregresywny z opóźnieniem, który charakteryzował się średnim błędem absolutnym (MAE) wynoszącym 0.23861768. Natomiast w przypadku sieci neuronowych, najlepsze wyniki uzyskaliśmy dla ostatniego modelu, z wartością MAE wynoszącą 1.6984301.

Warto zauważyć, że modele liniowe są stosunkowo proste i łatwe do interpretacji, co może być istotne w niektórych zastosowaniach. Model liniowy autoregresywny z opóźnieniem uwzględniał wartości temperatury z dnia poprzedniego, co pozwoliło na uwzględnienie wpływu trendów i sezonowości w danych.

Z drugiej strony, sieci neuronowe mają większą zdolność do modelowania złożonych relacji między danymi, co może przynosić lepsze wyniki predykcyjne w niektórych przypadkach. Jednak istnieje ryzyko overfittingu, czyli dopasowania modelu do danych treningowych w sposób, który nie generalizuje się dobrze na nowe dane. W przypadku naszych badań, sieci neuronowe wykazywały większą pomyłkę w temperaturze, natomiast poprawiały swoje predykcje wraz ze wzrostem liczby przewidzianych temperatur.

Modele liniowe wykazały lepsze przybliżenia faktycznych temperatur niż sieci neuronowe. Reagują one lepiej na drastycznie zmieniające się dane. Może to być spowodowane wieloma czynnikami, na przykład zbyt małą ilością danych, nieodpowiednimi danymi lub złym doborem parametrów sieci neuronowych, czy też charakterystyką sieci neuronowych, która skupiona jest bardziej na wychwytywaniu trendów, niż dostosowywaniu się do obecnych danych.

Ważne jest również odnotowanie różnic w zestawie danych użytych do uczenia modeli. Dla modeli liniowych, wykorzystano temperatury średnie z trzech lat jednej miejscowości. Natomiast dla sieci neuronowych, dane obejmowały szeroki zakres cech z 6 miast. Tym bardziej zaskakująca jest dominacja modeli liniowych nad sieciami neuronowymi w naszym projekcie.