

Systemy Sztucznej Inteligencji

Dokumentacja projektu

Porównanie algorytmu KNN oraz Naiwnego Klasyfikatora Bayesa przy klasyfikacji odręcznie
pisanym cyfr.

Piotr Skowroński gr. 3, Krzysztof Czuba gr. 4, Jakub Poreda gr. 3

19 czerwca 2023

Część I

Opis programu

Celem zadania jest klasyfikowanie odręcznie pisanych cyfr przez użytkownika. Program jest prostą aplikacją okienkową, która pozwala użytkownikowi narysowanie cyfry na ekranie, a następnie po wciśnięciu przycisku 'Recognize' program klasyfikuje cyfrę za pomocą jednego z klasyfikatorów. Program wyświetla wyniki klasyfikacji w bloku po prawej stronie.

Instrukcja obsługi

Program można uruchomić z poziomu konsoli lub IDE. Po uruchomieniu programu wyświetla się główne okno programu. Dane na podstawie których program klasyfikuje cyfry są zapisane w katalogu 'imgs'.

Dodatkowe informacje

Wymagania: Python 3.11

Program korzysta z następujących zewnętrznych bibliotek:

- Pillow
 - Do transformacji zapisanych obrazów na znormalizowaną macierz od 0 do 1.
 - Do transformacji narysowanej cyfry na znormalizowaną macierz od 0 do 1.
- numpy
- scikit-learn

Część II

Opis działania

Wartości obrazu są konwertowane na skalę szarości oraz normalizowane co ułatwia modelowi dopasowanie cyfr.

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

Następnie przechodzimy do fazy uczenia, gdzie model otrzymuje nasze dane treningowe. Model wykorzystując klasyfikator KNN, umieszcza w przestrzeni metrycznej cech wartości pikseli cyfr z zestawu treningowego. Wykorzystuje metrykę odległości Manhattan mierzy on sumę bezwzględnych różnic pomiędzy cechami

$$d_n = \sum_{i=0}^n |\text{wektor1}_i - \text{wektor2}_i|$$

Przewidywanie odbywa się poprzez wybranie trzech cyfr z zbioru treningowego, których cechy różnią się najmniej od cech sprawdzanego rysunku. Wybór odbywa się poprzez znalezienie dominującej spośród wybranych cyfr

Algorytm

Tutaj opisujemy rozwiązanie zadania. Dla przedmiotu programowanie będzie to wykorzystanie matematyki z poprzedniego zadania itd. Dla SSI będzie to ogólne działanie przetwarzania danych w oparciu o modele matematyczne z poprzedniego zadania.

Pseudokod tworzymy w L^AT_EX. Przykład:

Data: Dane wejściowe liczba k

Result: Brak

$i := 0$;

while $i < k$ **do**

 Drukuj na ekran liczbę i ;

if $i \% 2 == 0$ **then**

 | Wydrukuj informację, że liczba i jest liczbą parzystą;

else

 | Wydrukuj informację, że liczba i nie jest liczbą parzystą;

end

end

Algorithm 1: Algorytm drukowania informacji o liczbie parzystej/nieparzystej.

Bazy danych

Sekcja występuje tylko w przypadku projektów bazodanowych.

Należy pokazać przykładowe dane, które były wykorzystywane podczas uczenia klasyfikatorów.

Strukturę bazy i relacje.

Implementacja

1. Program składa się z statycznej klasy przechowującej ustawienia,
2. Funkcji zamieniającej obrazy na wektory
3. Funkcji przewidującej cyfry
4. Widgeta Tkintera

Widget wyświetla UI, a na nim canvas po którym użytkownik może pisać. Użytkownik bazgrze po nim i klika przycisk z napisem Recognize. Rysunek jest pobierany i serializowany do postaci odpowiadającej danym testowym. Uruchamiana jest procedura klasyfikująca, a jej rezultat - Dopasowana cyfra - Wyświetlany jest w GUI.

Tumogobywejzdjcieaplikacji

Testy

Tutaj powinna pojawić się analiza uzyskanych wyników oraz wykresy/pomiary.

Eksperymenty

Sekcję używamy gdy porównywaliśmy dwa lub więcej algorytmów, albo wykonywaliśmy jakies pomiary.

Warto dodać jakies wykresy jako obraz, albo tabele z wynikami.

Wszystkie wyniki powinny być opisane/poddane komentarzowi i poddane analizie statystycznej.

Pełen kod aplikacji

`1 Tutaj wklejamy pełen kod.`
