|  |  |  |  |
| --- | --- | --- | --- |
|  | Politechnika Bydgoska im. J. J. Śniadeckich  Wydział Telekomunikacji,  Informatyki i Elektrotechniki  **Zakład Systemów Teleinformatycznych** | |  |
| **Przedmiot** | Sztuczne sieci neuronowe | | |
| **Prowadzący** | prof. dr hab. inż. prof. PBŚ Piotr Cofta | | |
| **Temat** | *Project Raport 1* | | |
| **Student** | Cezary Tytko | | |
| **Ocena** |  | **Data oddania spr.** |  |

Etap 1. – kontrola danych.

Weryfikacji możemy dokonać przez wczytanie danych jako pliku csv i sprawdzić czy wymiary otrzymanej tablicy zgadzają się z przewidywanymi, to znaczy czy mamy odpowiednia liczbę rekordów i czy każdy rekord zawiera odpowiednią liczbę danych, to jest liczba oczek na kostce i 28 X 28 pikseli ułożonych w jednym wymiarze. Należy również sprawdzić czy wartości etykiet jak o danych są zgodne z założeniami, np. czy nie wychodzą poza przewidziany zakres, albo czy nie ma wartości brakujących, można to sprawdzić wyświetlając wartości unikatowe (Select distinct w konwencji sql), jeżeli dane będą zawierały błędy na poziome typów wartości tzn. string nie konwertowany na int, to dostaniemy błąd na etapie odczytu pliku csv (przynajmniej dla implementacji z pandas).

Przykładowy kod weryfikacji:

1. dice\_y, dice\_x = ReadDiceCSV(dice\_dir\_csv)

2. dice\_x = dice\_x.reshape((60000, 28, -1))

3. print("dice:")

4. print(f"Label Shape: {dice\_y.shape} has NAN: {np.isnan(dice\_y).any()}")

5. print(f"Values : {np.unique(dice\_y)}")

6. print(f"Data Shape: {dice\_x.shape} has NAN: {np.isnan(dice\_x).any()}")

7. print(f"Values : {np.unique(dice\_x)}")

8.

9. mnist\_y, mnist\_x = ReadDiceCSV(mnist\_train\_dir\_csv)

10. mnist\_x = mnist\_x.reshape((60000, 28, -1))

11. print("minst:")

12. print(f"Label Shape: {mnist\_y.shape} has NAN: {np.isnan(mnist\_y).any()}")

13. print(f"Values : {np.unique(mnist\_y)}")

14. print(f"Data Shape: {mnist\_x.shape} has NAN: {np.isnan(mnist\_x).any()}")

15. print(f"Values : {np.unique(mnist\_x)}")

16.

17. cifar\_y, cifar\_x = ReadDiceCSV(cifar\_all\_dir\_csv)

18. cifar\_x = cifar\_x.reshape((50000, 28, -1))

19. print("cifar:")

20. print(f"Label Shape: {cifar\_y.shape} has NAN: {np.isnan(cifar\_y).any()}")

21. print(f"Values : {np.unique(cifar\_y)}")

22. print(f"Data Shape: {cifar\_x.shape} has NAN: {np.isnan(cifar\_x).any()}")

23. print(f"Values : {np.unique(cifar\_x)}")

24.

Powyższa walidacja wskazuje na poprawność zbiorów danych wraz z etykietami. Można by również dokonać weryfikacji ręcznej sprawdzając treść obrazków (czy na zdjęciach kostek, faktycznie znajdują się kostki), ale na tym etapie zakładem weryfikacje na poziomie wartości i typów, wyświetlając tylko po jednym przykładowym elemencie zgodnie z poleceniem zadania.

Cały kod umieściłem (w pliku main.ipynb) na:

https://github.com/Czarkowski/SSN\_Project

Obraz zawierający tekst, zrzut ekranu, czarne i białe, projekt graficzny

Opis wygenerowany automatycznie