
	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	<i>Project Raport 1</i>		
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 odpowiednią 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 poziomie 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("mnist:")
12. print(f"Label Shape: {mnist_y.shape} has NAN: {np.isnan(mnist_y).any()}")
13. print(f"Values : {np.unique(mnist_y)}")

```

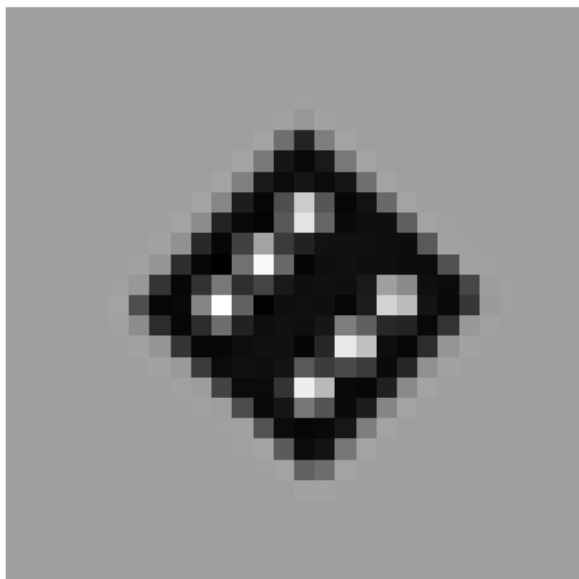
```
14. print(f"Data Shape: {mnist_x.shape} has NAN: {np.isnan(mnist_x).any()}")
15. print(f"Values : {np.unique(mnist_x)}")
16.
17. cifar_y, cifar_x = ReadDiceCSV(cifar_all_dir_csv)
18. cifar_x = cifar_x.reshape((50000, 28, -1))
19. print("cifar:")
20. print(f"Label Shape: {cifar_y.shape} has NAN: {np.isnan(cifar_y).any()}")
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 weryfikację na poziomie wartości i typów, wyświetlając tylko po jednym przykładowym elemencie zgodnie z poleceniem zadania.

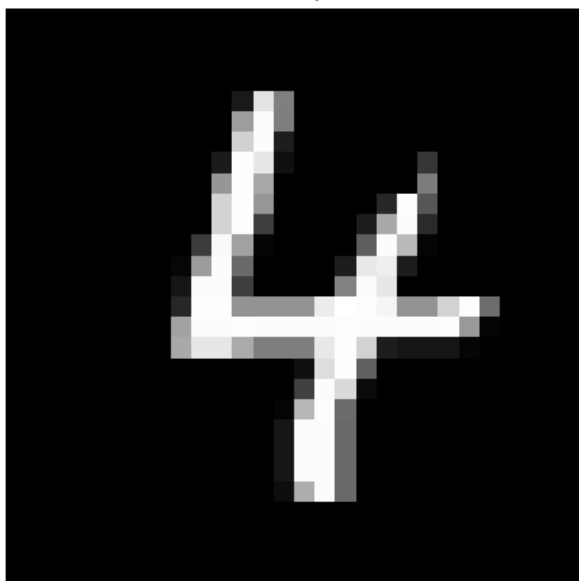
Cały kod umieściłem (w pliku main.ipynb) na:

https://github.com/Czarkowski/SSN_Project

Dice 58873, value: 6



Mnist 58873, value: 4



Cifar 18873, value: 12

