Wizualizacja Sieci Neuronowych

Tomasz Rajchel
Analiza obrazów
2020.01

Spis Treści

Opis

Celem projektu jest stworzenie narzędzia pozwalającego na wizualizację wag neuronów sztucznej sieci neuronowej użytej do klasyfikacji obrazów.

Instalacja i uruchamianie

Niestandardowe biblioteki python'a znajdują się w pliku **requirements.txt**
Można je zainstalować poprzez: **pip3 install -r requirements.txt**
Należy również zainstalować bibliotekę tkinter: **sudo apt install python3-tk**

Aby uruchomić program należy z terminala uruchomić skrypt **src/NNviewer.py**

Sieć neuronowa

Implementacja sieci neuronowej pochodzi z książki Neural Networks and Deep Learning której autorem jest Michael A. Nielsen. Książka jest dostępna na licencji Creative Commons.

Technologie

Algorytm uczenia sieci neuronowej, kod opisujący strukturę sieci oraz kod wczytujący dane w odpowiednim formacie był oryginalnie napisany w języku Python wersja 2.7 i został przeze mnie dostosowany do wersji 3.6. Biblioteki użyte w projekcie to:

- numpy – do obliczeń numerycznych
- pickle – serializacja i zapis danych
- matplotlib – wizualizacja danych
- tkinter - GUI

Dane

Danymi wejściowymi są ręcznie pisane cyfry (0-9) ze zbioru MNIST Database (Modified National Institute of Standards and Technology database). Zostały one przeskalowane do formatu 28x28 pikseli, wycentrowane oraz znormalizowane.

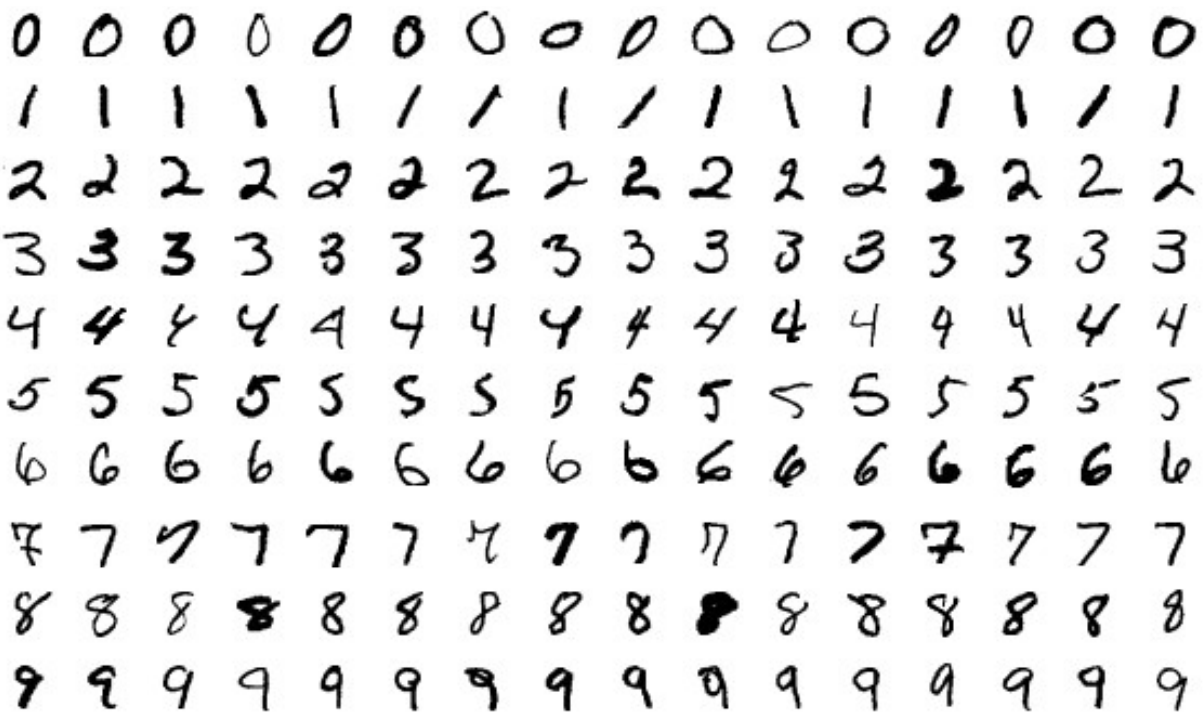

Illustration 1: Zbiór cyfr MNIST DATASET

70 000 zdjęć zostało podzielone na:

- 50 000 – zbiór uczący
- 10 000 – zbiór walidujący
- 10 000 – zbiór testujący

Pełen zbiór jest dostępny na tej stronie: http://yann.lecun.com/exdb/mnist/
Na stronie jest również dostępny ranking różnych metod automatycznej klasyfikacji tych danych.

Klasyfikacja

W zakładce ‘examples’ możemy uruchamiać sieć neuronową dla poszczególnych obrazów ze zbioru testowego. W kolumnie po prawej stronie zobaczymy z jaką pewnością sieć klasyfikuje obrazy. Poprawne zaklasyfikowanie oznaczone jest kolorem zielonym.

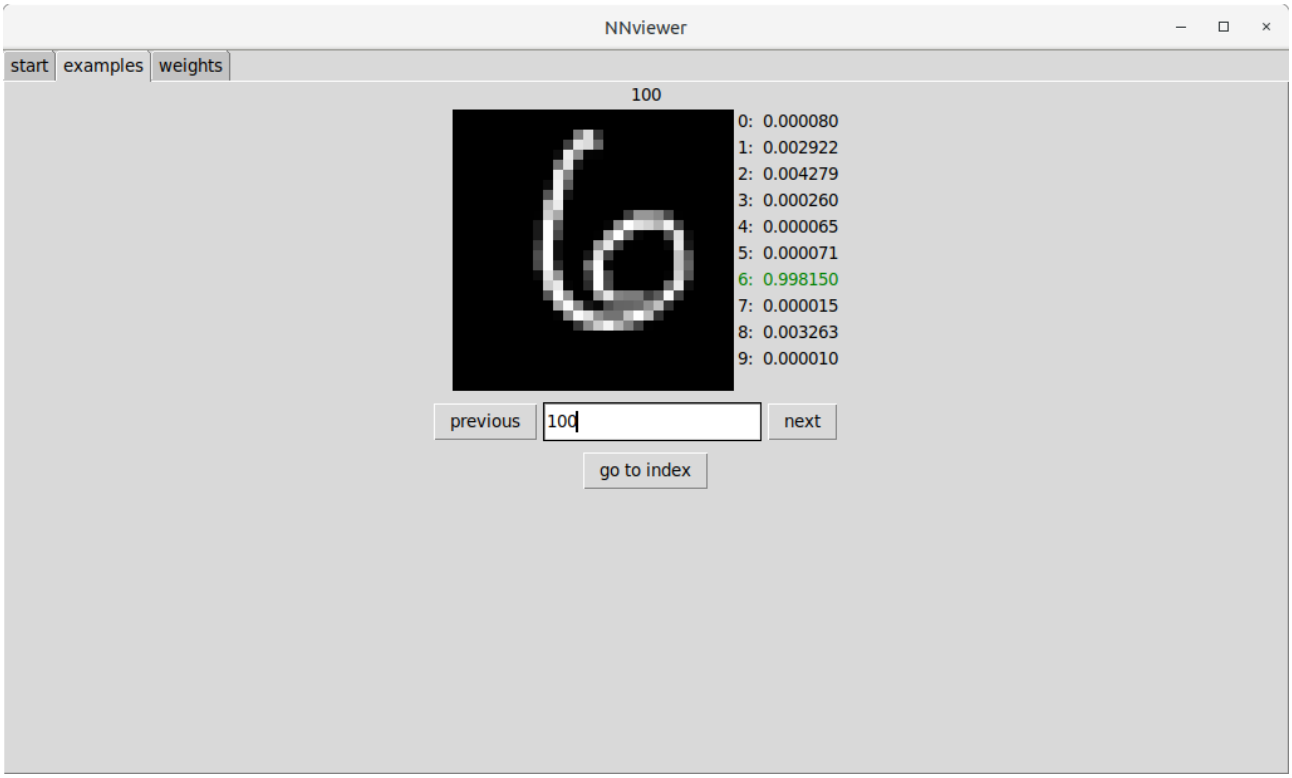

Illustration 2: Poprawnie zaklasyfikowana cyfra

Nieprawidłowe zaklasyfikowanie oznaczone jest kolorem czerwonym.

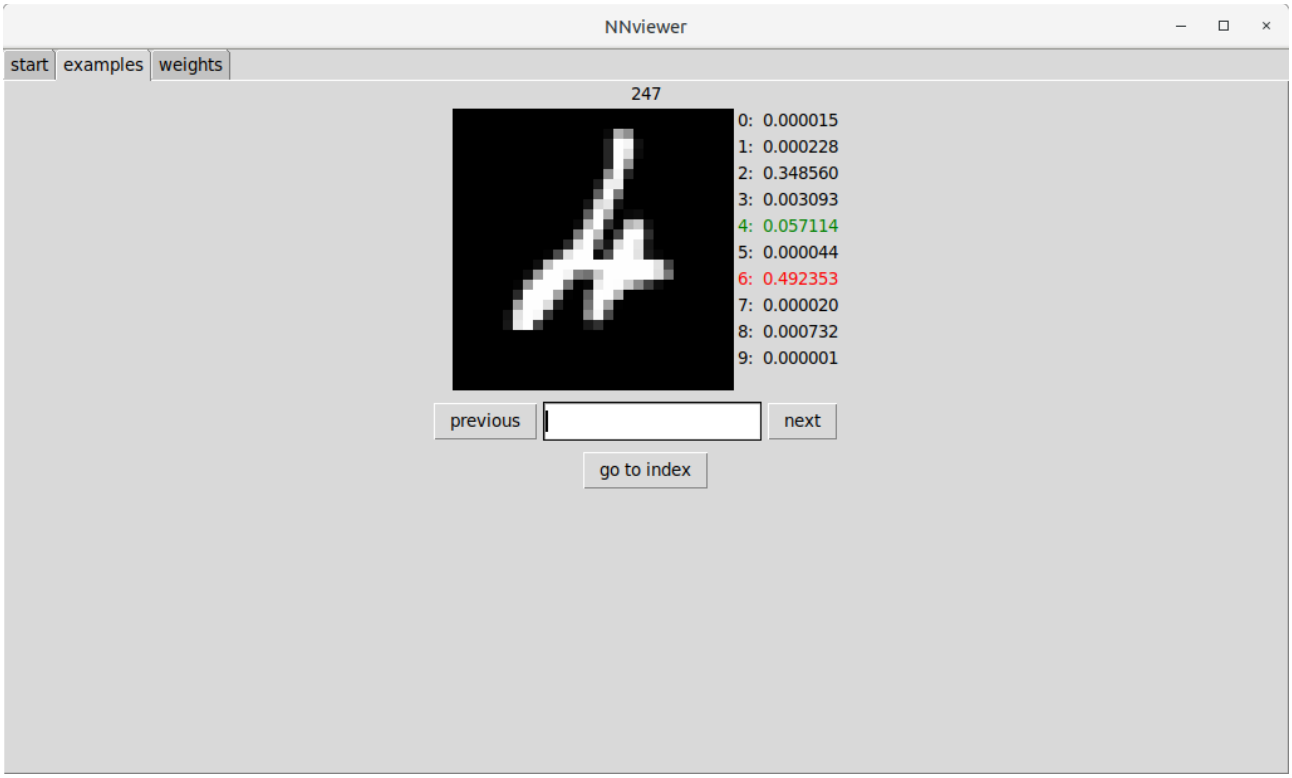

Illustration 3: Niepoprawnie zaklasyfikowana cyfra

Wizualizacja wag poszczególnych neuronów

W zakładce ‘weights’ zobaczymy wagi neuronów warstwy pierwszej. Dzięki wizualizacji można zauważyć, że dobra sieć nauczyła się wykrywać niektóre kształty: poziome i pionowe linie, okręgi itp. które w kolejnych warstwach mogą się złożyć na pełne cyfry.
Oprócz wag zaznaczone jest również bias (B) oraz wartość średnia, influence (I)

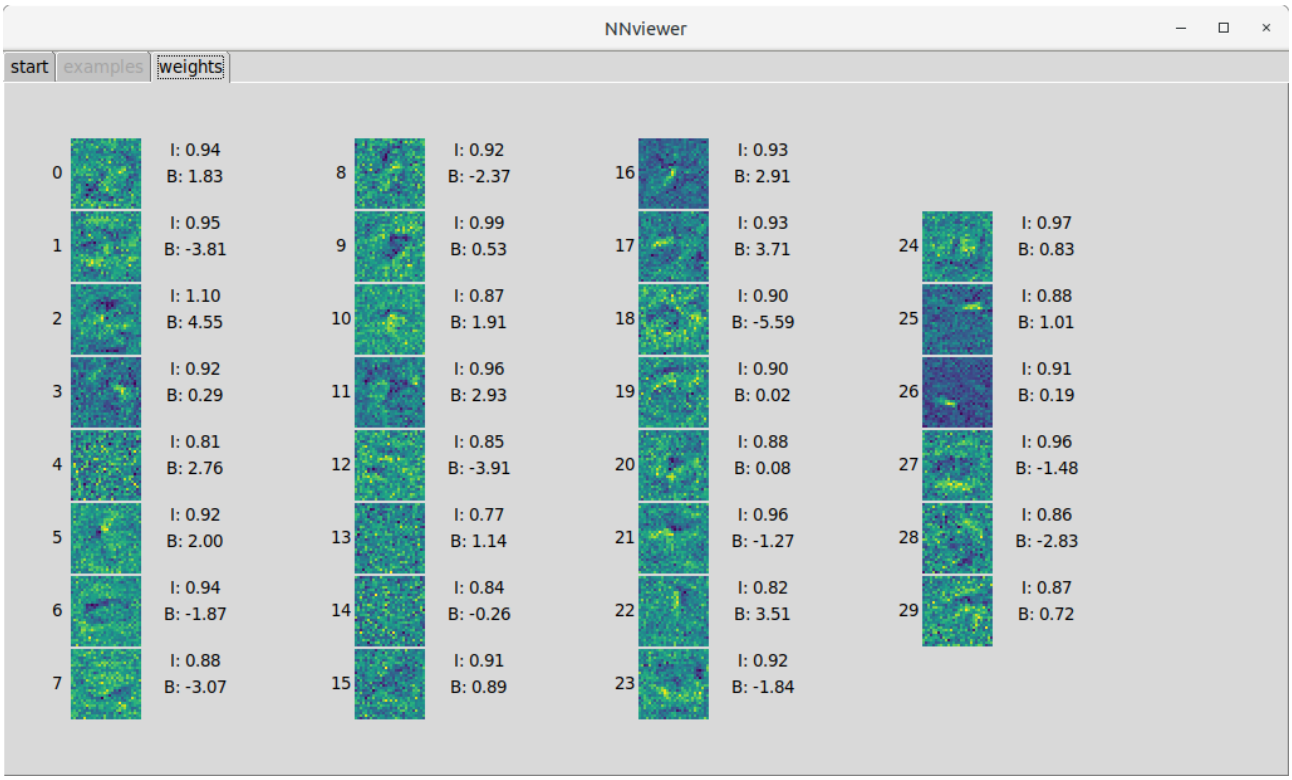

Illustration 4: Wagi neuronów nauczonej sieci

Dla porównania zobaczmy jak wygląda nienauczona sieć.

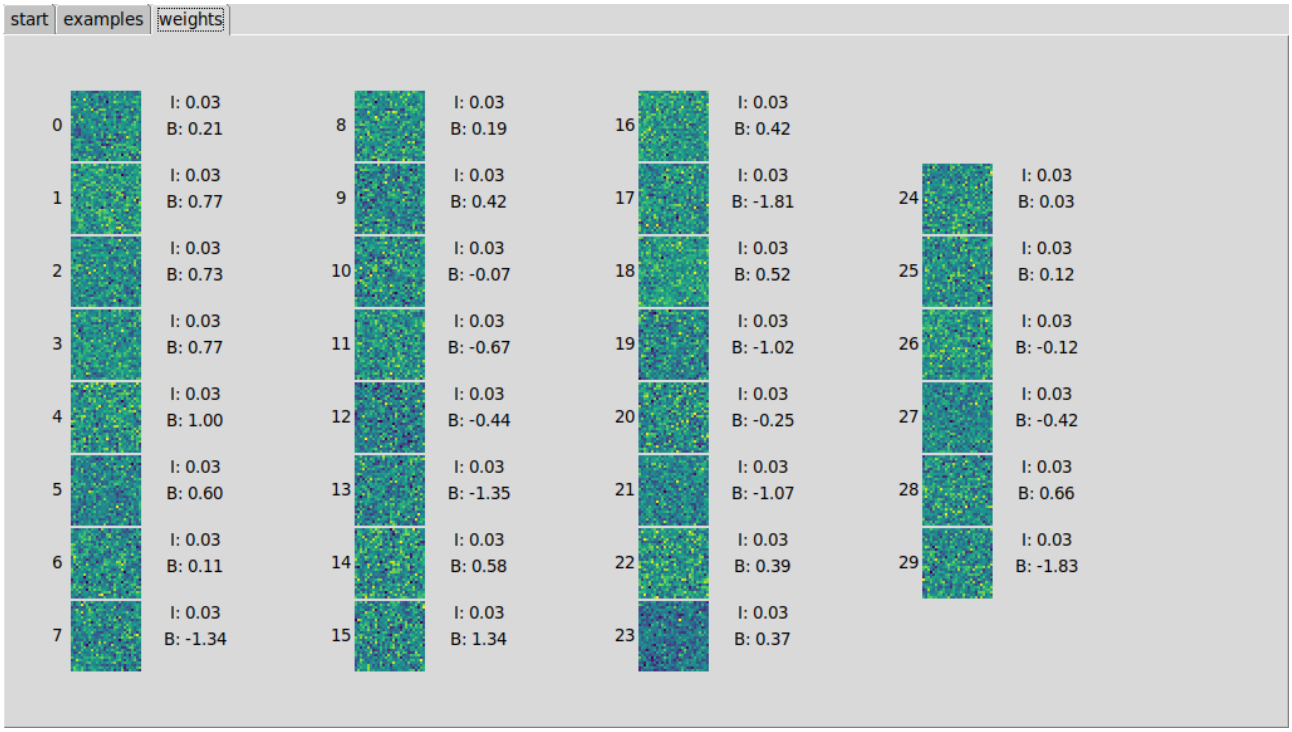

Illustration 5: Wagi neuronów nienauczonej sieci

Losowy szum.

Uogólniony obraz klasyfikowanych cyfr

Możemy dowiedzieć się jakie piksele na obrazie najbardziej wpływają na klasyfikacje obrazu do odpowiednich cyfr poprzez mnożenie wszystkich wag sieci „wstecz”.

Załóżmy, że topologia naszej sieci jest następująca: [784, 30, 10]
784 – piksele wejściowe (28x28)
30 – neurony warstwy pierwszej
10 – neurony warstwy drugiej, wyjściowej

Obraz wyjściowy ‘img’ możemy obliczyć następująco:

$$img = \sum_{i=0}^{9} \sum_{j=0}^{29} w_j^0 \cdot w_{j,i}^1$$

gdzie: $w_{j,i}^l$ - i-ta waga, j-tego neuronu w warstwie l-tej

Wygenerujmy teraz te obrazy.

Illustration 6: Wpływ pikseli wejściowych na klasyfikację poszczególnych cyfr - sieć nauczona

Warto zauważyć, że wagi poszczególnych neuronów tworzą obraz podobny do właściwych im cyfr. Szczególnie jest to widoczne dla obrazów 0, 1 i 3. Wyraźnie widać na nich zarys cyfry.
Dla porównanie zobaczmy to samo dla nienauczonej sieci.

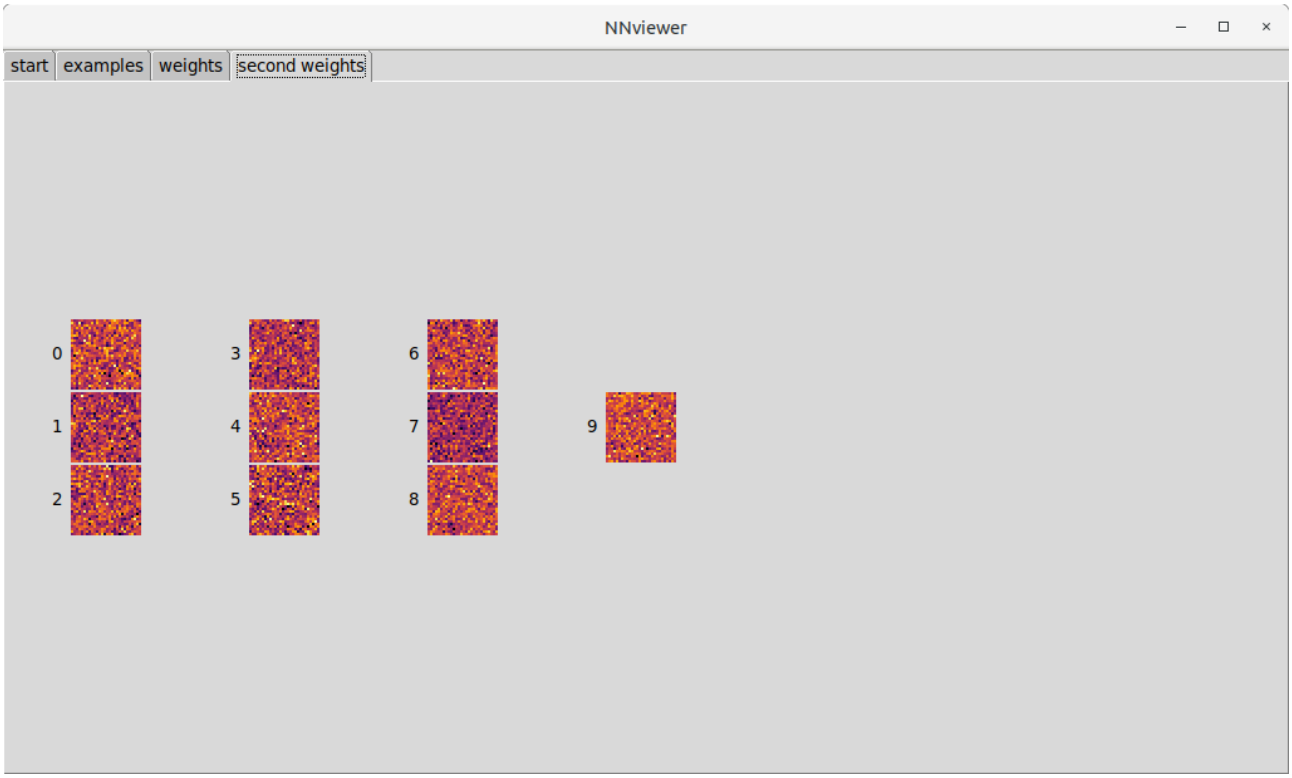

Illustration 7: Wpływ pikseli wejściowych na klasyfikację poszczególnych cyfr - sieć nienauczona

Bibliografia

[1] Michael A. Nielsen, „Neural Networks and Deep Learning”.
[2] Yann LeCun, Corinna Cortes, Christopher J.C. Burges, THE MNIST DATABASE, http://yann.lecun.com/exdb/mnist/