

WSI 2022Z Sieci neuronowe

Raport ćw. 5

Łukasz Wójcicki 318746, Piotr Kowalski 318679

Cel ćwiczenia:

Ćwiczenie polega na implementacji perceptronu wielowarstwowego z algorytmem propagacji wstecznej oraz metodą gradientu prostego. Trenujemy perceptron w celu identyfikacji ręcznie napisanych liczb z bazy danych MNIST.

Postanowienia projektowe:

Zdjęcia pobrane z bazy danych reprezentowane są jako wektory jasności pikseli. Zdjęcie posiada rozmiary 8x8, co daje nam wektory długości 64.

Pobrany zbiór dzielimy na część treningową, walidacyjną oraz testową w stosunku 7:2:1. Każdą sieć budujemy na największej części: treningowej. Zbiór walidacyjny służy do wyszukania najlepszego współczynnika uczenia oraz do wyszukania najlepszej liczby iteracji. Te parametry poszukujemy osobno dla stałego drugiego parametru. Część testowa wykorzystywana jest do uzyskania ostatecznego wyniku.

W algorytmie wykorzystujemy metodę gradientu prostego. Sieć uczy się na poszczególnych zdjęciach, nie na grupach.

Testujemy wyniki na sieci, której neurony ustawione są w warstwach: 64, 25, 10, 10.

Dla każdej warstwy neuronów korzystamy z sigmoidalnej funkcji aktywacji:

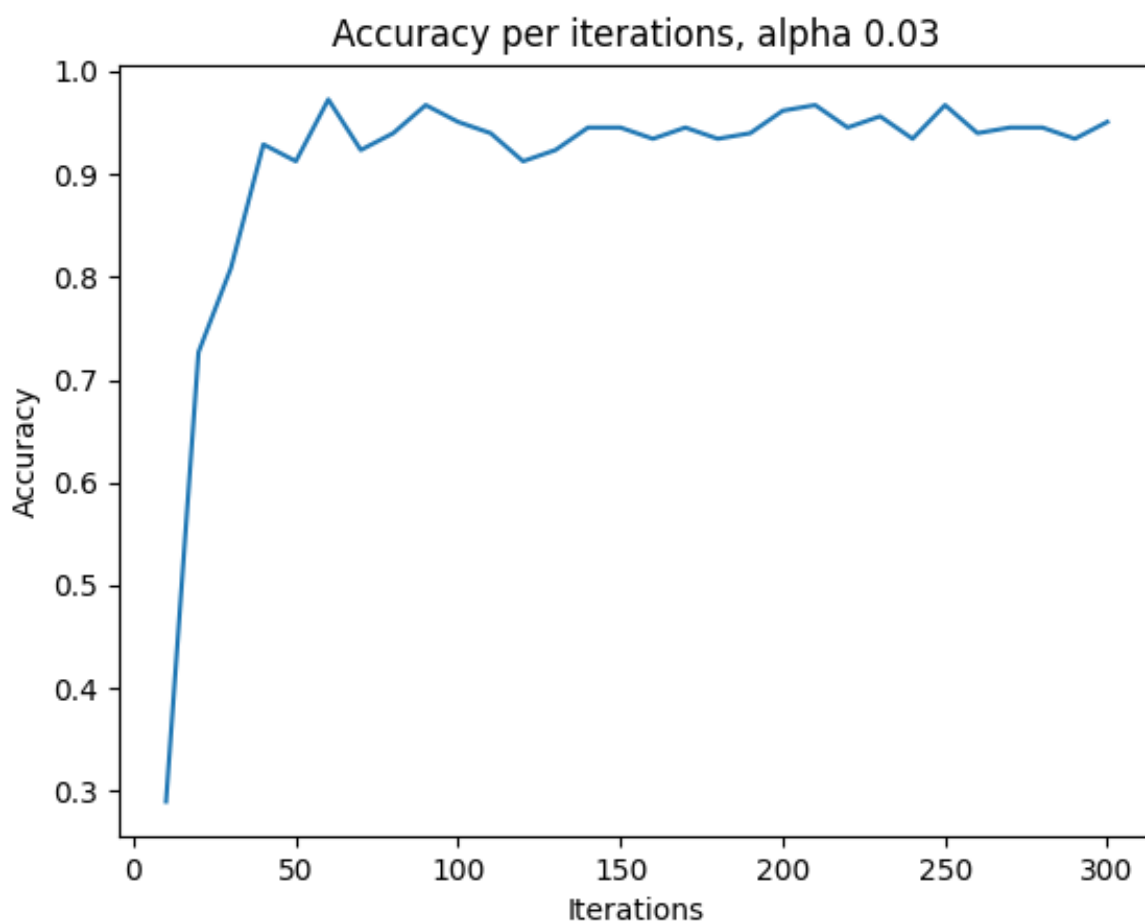
$$\psi(s) = \frac{1}{1 + e^{-s}}$$

Początkowe wartości wag i biasów inicjujemy jako losowe z rozkładu:

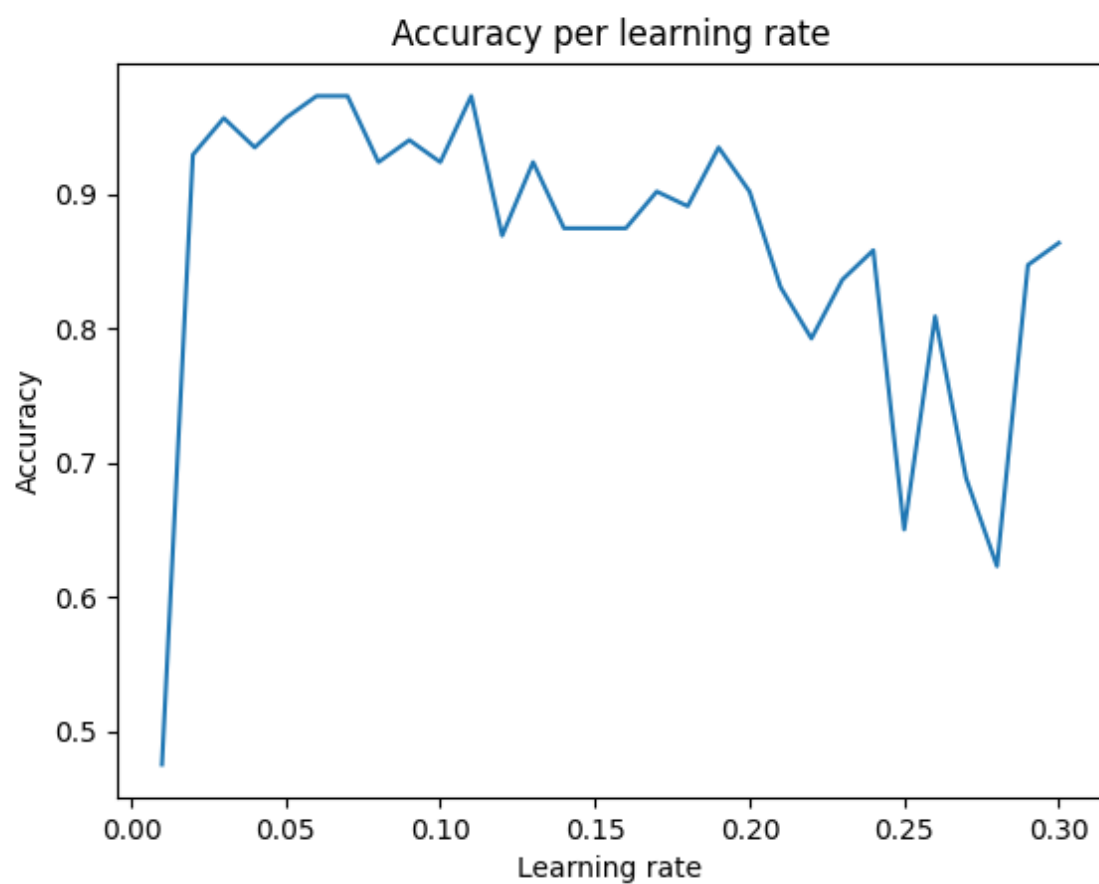
$$u\left(\frac{-1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$$

Badania:

Pokazane wykresy tworzone są przy stałym współczynniku uczenia albo przy stałej liczbie iteracji.

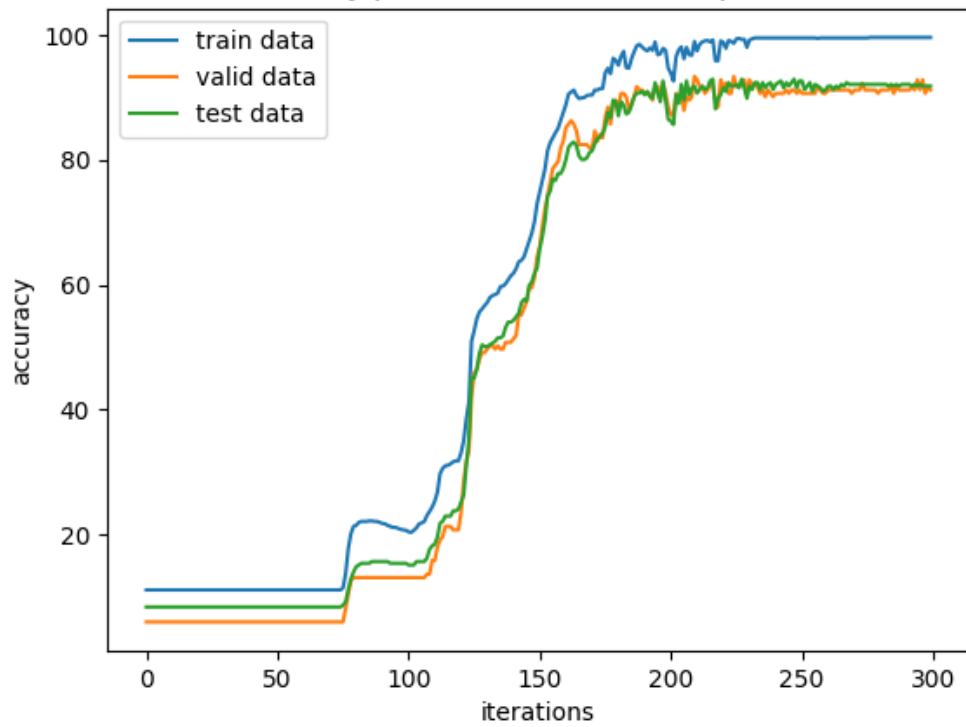


Wykres zależności dokładności przewidywania od liczby iteracji tworzony był przy współczynniku uczenia równym 0.03 oraz na zbiorze walidacyjnym.

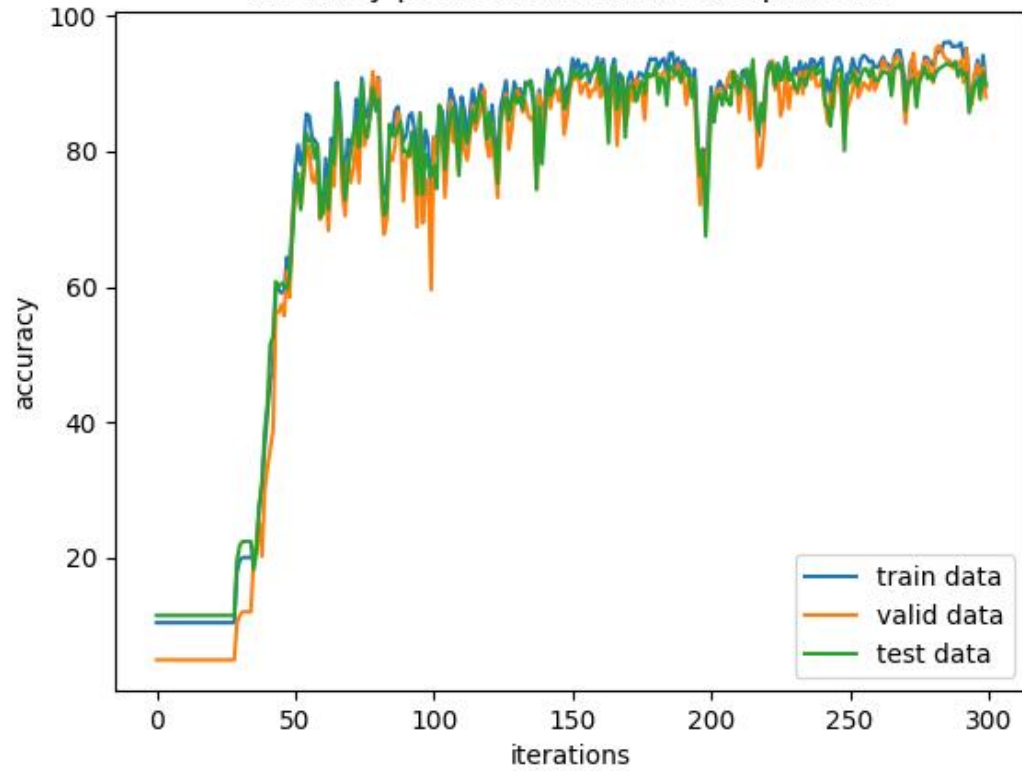


Wykres zależności dokładności przewidywania od współczynnika uczenia tworzony był przy stałej ilości iteracji algorytmu równej 100. Użyty został zbiór walidacyjny.

Accuracy per iteration number, alpha 0.03



Accuracy per iteration number, alpha 0.10



Wnioski:

Dla zbioru testowego dokładność przewidywania wynosi ponad 93.5%, co jest wysokim wynikiem. Wynik jest jednak zależny od jakości danych. W zbiorze danych pojawiają się zdjęcia, których nie mógłby ocenić człowiek.

Na obu pierwszych wykresach widzimy słabe wyniki dla niskiej ilości iteracji albo dla małego współczynnika uczenia. Sieć nie ma możliwości do dobrego wyuczenia się. Mały współczynnik nauki potrzebuje wielu iteracji, 100 jest ilością niewystarczającą.

Dwa ostatnie wykresy ukazują nam skutki dużego współczynnika uczenia algorytmu. Przy wartości 0.1 sieć staje się niestabilna. Skuteczność przewidywań spada do nawet 65%. Niski współczynnik stabilizuje program, wymaga jednak większej ilości iteracji do przewidzenia ponad 10% przykładów, które powinny być uzyskiwane przy nieskutecznym algorytmie. Na obu wykresach dominują wyniki z danych treningowych. To na nich algorytm się tworzył.

Najlepsze wyniki uzyskiwane były dla parametrów z zakresu:

Iterations: +150

Alpha: 0.03-0.07

Dodatkowe informacje

Zbiór danych jest dostępny w bibliotece sklearn . W algorytmie wykorzystałem biblioteki:

- numpy – wykorzystywana w obliczeniach do przechowywania i przetwarzania danych -
- scikit-learn – wykorzystywana do pobrania i podziału danych na zbiory
- matplotlib – tworzenie wykresów