

# Biblioteka MLP

Mikołaj Nowak Jakub Jasiński  
280082 280109

# Zaimplementowana sieć

# Struktura biblioteki

- [network.py](#) - zawiera główną klasę MLP i jej funkcje
- [layers.py](#) - zawiera klasę warstwy, definicje jej parametrów i algorytm propagacji wstecznej
- [activations.py](#) - zawiera funkcje aktywacji
- [losses.py](#) - zawiera funkcje straty
- [utils.py](#) - zawiera funkcje potrzebne do odczytu bazy MNIST

# Tworzenie sieci

- Zainicjalizowanie sieci wraz z współczynnikiem uczenia.
- Dodanie pierwszej warstwy o danym rozmiarze, funkcji aktywacji i jej pochodnej oraz rozmiar wejścia.
- Dodanie kolejnych warstw o danym rozmiarze, funkcji aktywacji i jej pochodnej.
- Dodanie ostatniej warstwy o rozmiarze wyjścia, funkcji aktywacji wyjścia i jej pochodnej.
- Ustawienie funkcji straty.

# Klasyfikacja MNIST

# Dane Wejściowe

Do klasyfikacji obrazów użyta została baza MNIST z serwisu Kaggle

<https://www.kaggle.com/datasets/hojjatk/mnist-dataset>, zawierająca 60 tys. zdjęć treningowych oraz 10 tys. zdjęć testowych. Dane są spłaszczone do wektora 784 elementów. Etykiety kodowane są metodą one-hot encoding.

Przykładowe dane wejściowe (MNIST)

Label: 7



Label: 8



Label: 1



Label: 5



Label: 7



Label: 1



Label: 4



Label: 2



Label: 2



Label: 3



Label: 9



Label: 7



Label: 4



Label: 2



Label: 8



Label: 7



# Dobór parametrów sieci

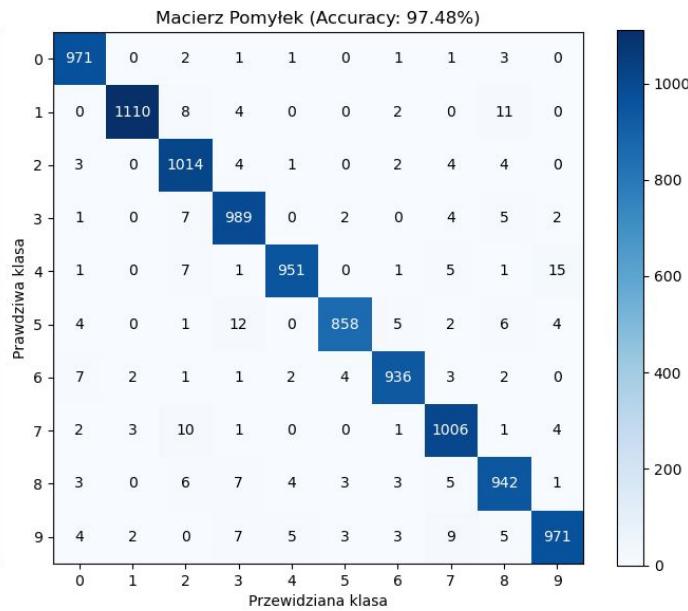
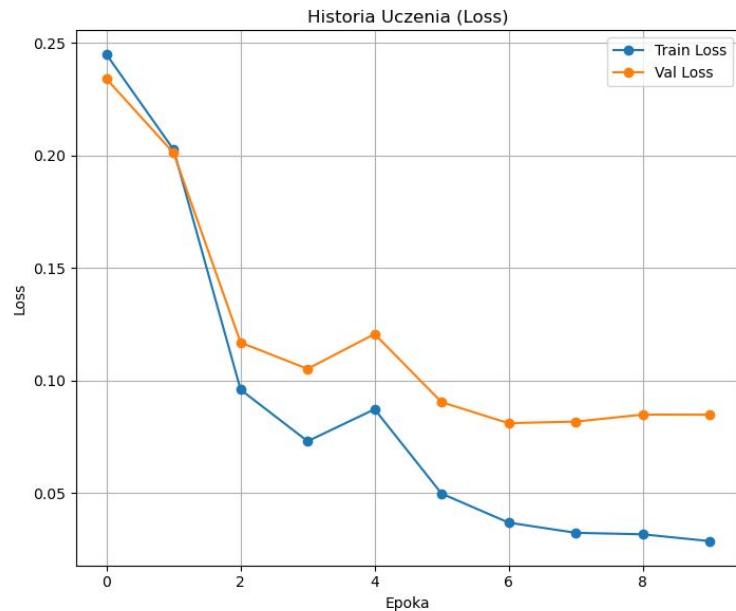
## Wartości Stałe

- Ilość wejść : 784
- Ilość wyjść : 10
- Funkcja aktywacji : ReLU
- Funkcja straty : cross entropy loss

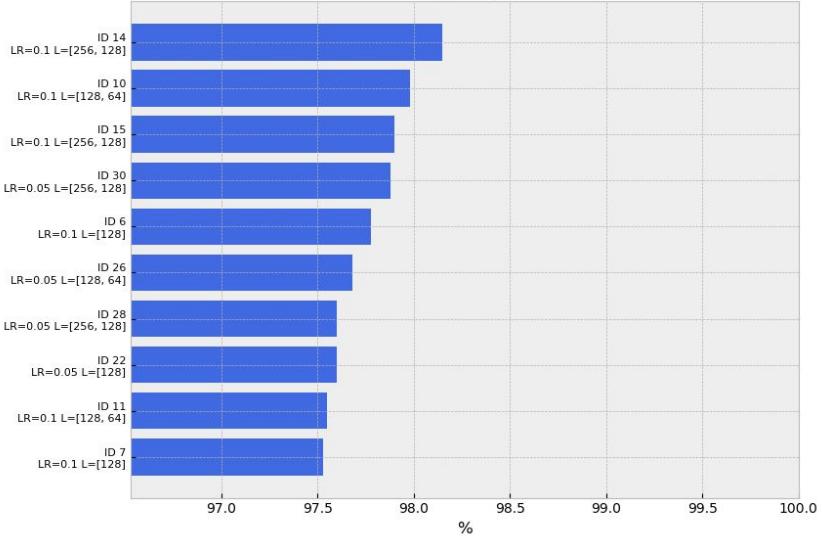
## Siatka Parametrów do testów:

- Współczynnik uczenia (LR): 0.1, 0.05, 0.01
- Warstwy ukryte: [64], [128], [64, 64], [128, 64], [256, 128]
- Epoki: 5, 10
- Batch size: 32, 64

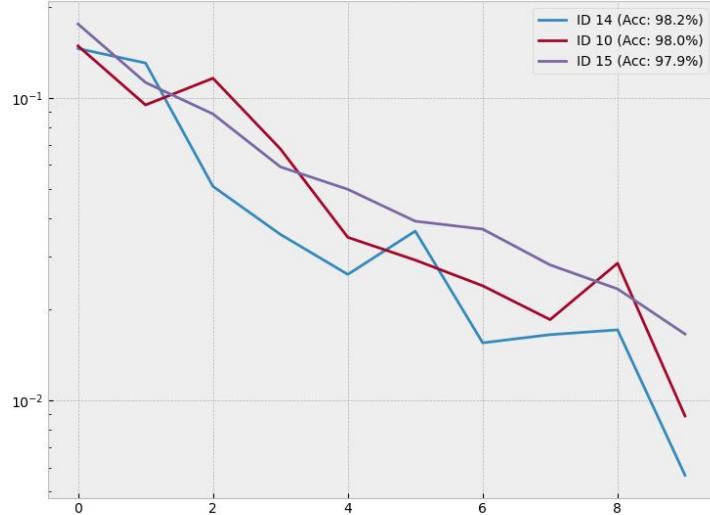
# Wyniki eksperymentów



### Top 10 Konfiguracji (Accuracy)



### Loss History (Top 3)



# Aproksymacja funkcji Ackleya

# Dane Wejściowe

Do aproksymacji użyto danych generowanych losowo (20 tys. próbek), aby zapewnić lepszą generalizację w dziedzinie  $x, y \in [-2, 2]$ .

Funkcja Ackleya:

$$f(x, y) = -20 \exp \left[ -0.2 \sqrt{0.5(x^2 + y^2)} \right] - \exp [0.5(\cos(2\pi x) + \cos(2\pi y))] + e + 20$$

# Dobór parametrów sieci

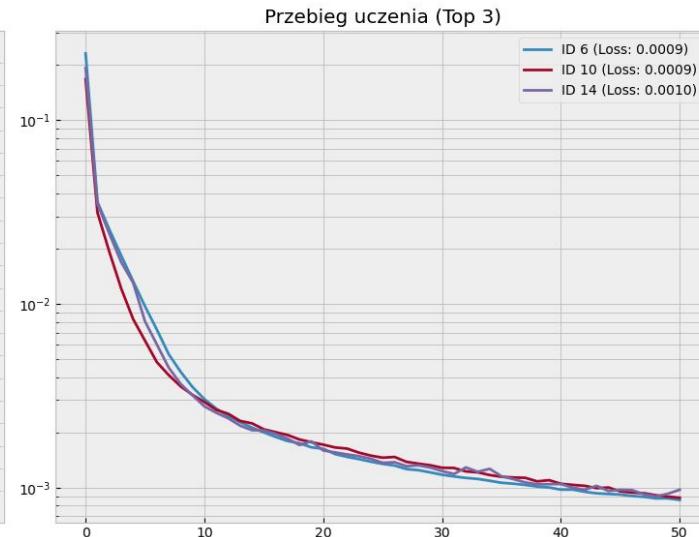
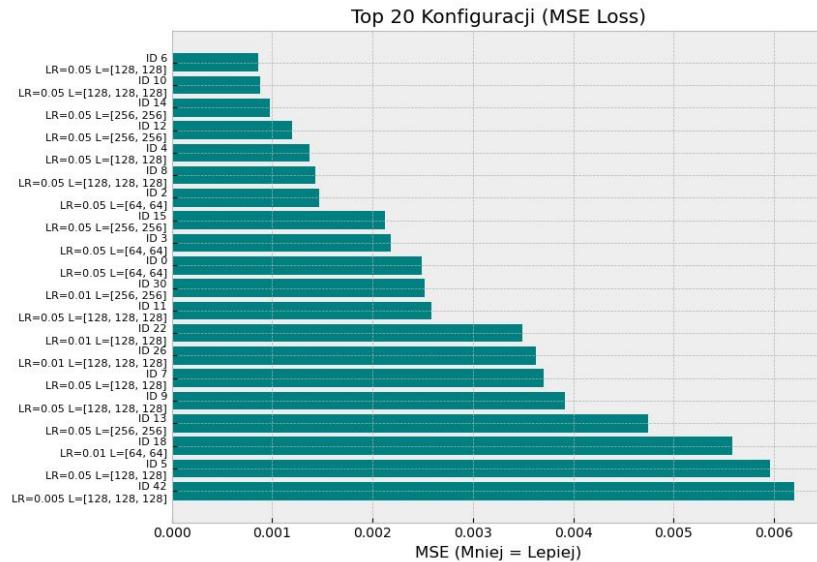
## Wartości Stałe

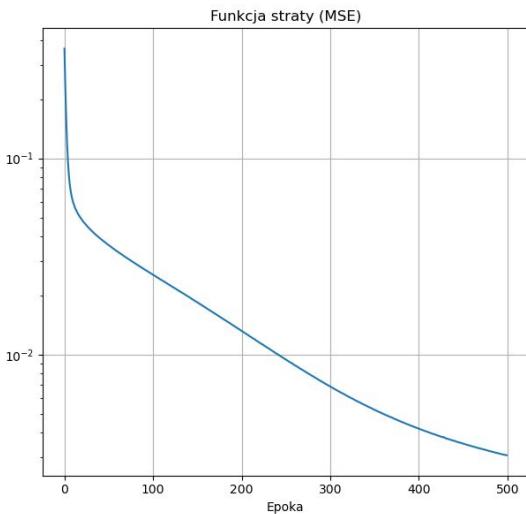
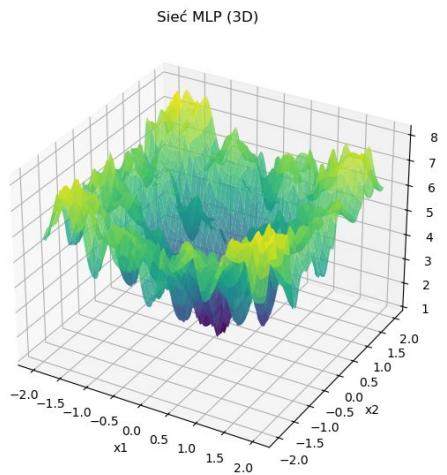
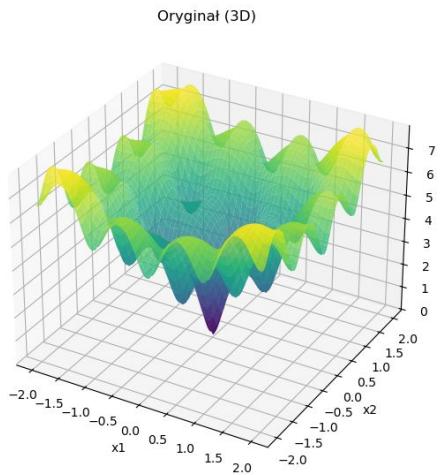
- Ilość wejść : 4
- Ilość wyjść : 1
- Funkcja aktywacji : Tanh
- Funkcja straty : MSE

## Siatka parametrów do testów:

- Współczynnik uczenia (LR): 0.05, 0.01, 0.005
- Warstwy ukryte: [64, 64], [128, 128], [128, 128, 128], [256, 256]
- Epoki: 500, 1000
- Batch size: 64, 128

# Wyniki Eksperymentów





# Wnioski

- Klasyfikacja MNIST:
  - Najlepszy model: 98.15%
  - Dla 10 epok najlepsze wyniki zostały osiągnięte przy wyższym współczynniku uczenia oraz mniejszym rozmiarze danych
  - Większe sieci zajmują wyższe miejsca w rankingu, co wskazuje na wyższą ekstrakcję cech przy bardziej pojemnych modelach.
- Aproksymacja funkcji Ackleya:
  - Najlepszy model: MSE < 0.001
  - Najlepsze MSE uzyskane przy 1000 epok, 500 nie wystarczające do uzyskania pełnej konwergencji.
  - Głębsza sieć [128,128,128] okazała się lepsza od płytowych dwu warstwowych sieciach.
  - Mniejsza seria danych 64 dała nam lepsze wyniki od serii 128.
  - Zarówno współczynnik uczenia 0.05 jak i 0.005 pozwolił na osiągnięcie dobrych wyników.

Dziękujemy