

Zadanie 5 - Raport

Piotr Niedziałek, Jan Stusio

Maj 2024

1 Wstęp

Celem laboratorium była implementacja perceptronów wielowarstwowych (MLP) oraz przeprowadzenie badań hiperparametrów tego typu sieci neuronowych. Jest to jeden z najpopularniejszych rodzajów sieci, składa się z warstwy wejściowej, warstw ukrytych i warstwy wyjściowej. Trenowanie sieci polega na zastosowaniu propagacji w przód, czyli otrzymania wartości na wyjściu sieci, a następnie na obliczeniu funkcji oceny i wykorzystaniu propagacji w tył - obliczeniu gradientu i zmienieniu parametrów w sieci przeciwnie do wartości gradientu.

2 Implementacja

Zadanie zaimplementowano w notatniku jupyter. Zastosowano biblioteki numpy, matplotlib oraz sklearn. Struktura kodu:

- przygotowanie danych (Jan Stusio)
- funkcje aktywacji (JS)
- inicjalizacja parametrów (Piotr Niedziałek)
- propagacja w przód (PN)
- propagacja w tył (PN)
- aktualizacja parametrów (PN)
- trenowanie sieci (JS)
- ewaluacja (JS)
- eksperymenty (PN+JS)

3 Badania

Eksperymenty zostały przeprowadzone na bazie danych "digits" z "sklearn.datasets.load_digits". Z danych 30% zostało wybrane na dane testowe. Przeprowadzono badania wpływu ilości warstw ukrytych, rozmiaru batch'a i wyboru funkcji aktywacji na wyniki. Badania przeprowadzono dla następujących parametrów:

- liczba warstw ukrytych: od 1 do 3
- rozmiary batch'a: [8, 16, 32, 64]
- funkcje aktywacji: ReLU i sigmoid

4 Wyniki eksperymentów

4.1 Liczba warstw ukrytych

Liczba Warstw	Accuracy	F1	Accuracy STD	F1 STD
1	0.971111	0.971111	0.002222	0.002222
2	0.143704	0.143704	0.036974	0.036974
3	0.092593	0.092593	0.000000	0.000000

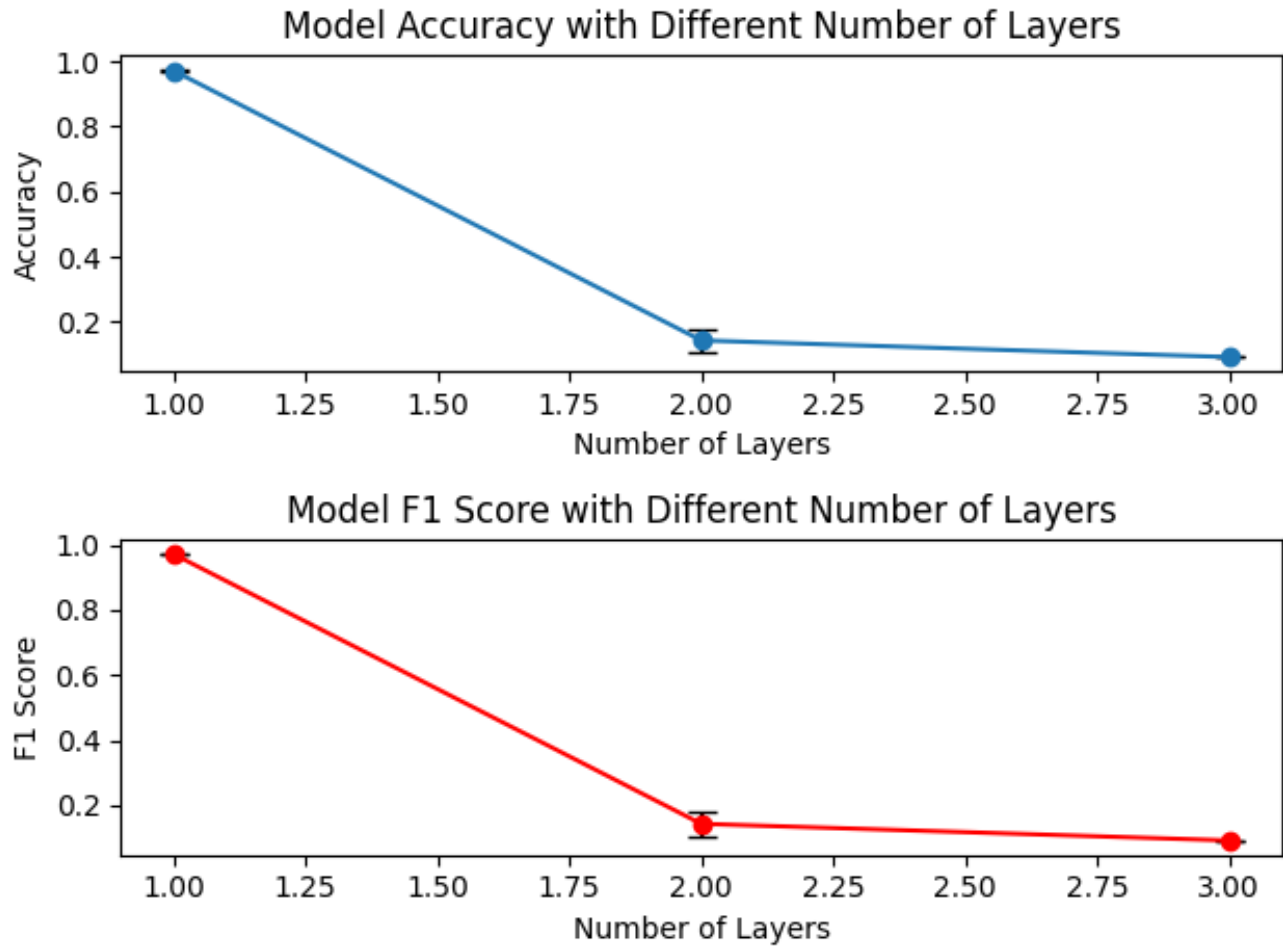


Figure 1: Liczba warstw ukrytych 2000 epok

4.2 Batch size

Batch Size	Accuracy	F1	Accuracy STD	F1 STD
8	0.966296	0.966296	0.002722	0.002722
16	0.862963	0.862963	0.026215	0.026215
32	0.148519	0.148519	0.068337	0.068337
64	0.092593	0.092593	0.000000	0.000000

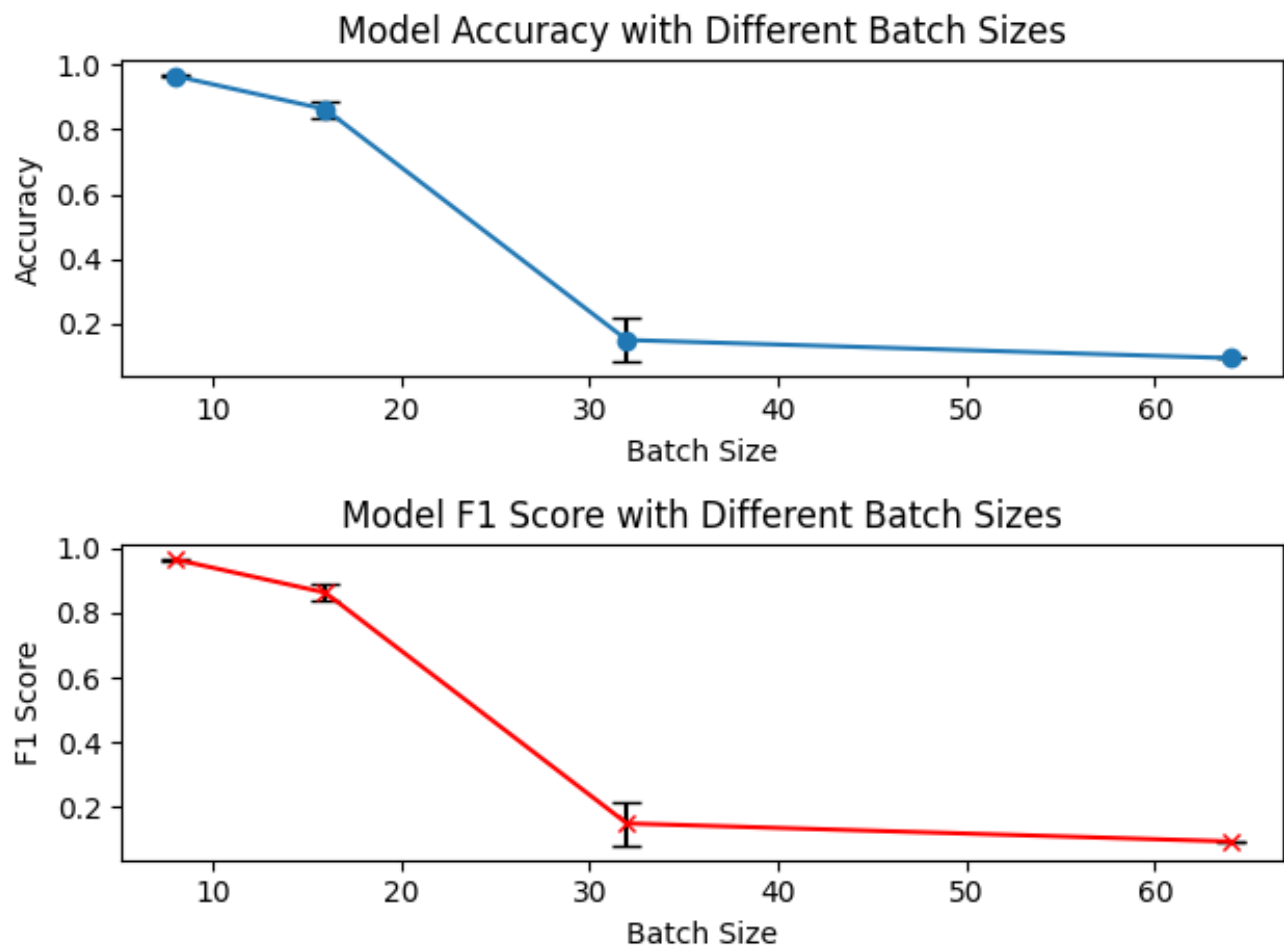


Figure 2: Rozmiar Batch'y 2000 epok

4.3 Funkcje aktywacji

Funkcja Aktywacji	Accuracy	F1	Accuracy STD	F1 STD
Sigmoid	0.920741	0.920741	0.007626	0.007626
ReLU	0.977778	0.977778	0.003313	0.003313

5 Wnioski

Zwiększenie ilości warstw w tym przypadku wpływa negatywnie na dokładność. Wynika to prawdopodobnie z dużej ilości parametrów, które powstają w takiej sieci. Liczba danych wejściowych jest zbyt mała, żeby nauczyć dobrze taką sieć.

Większa ilość danych w batch'u również powoduje pogorszenie wyników. Potwierdza to obserwacje badaczy, że większe batch'e powodują gorszą generalizację. Model może się łatwiej przeuczyć.

Lepsze wyniki otrzymano dla funkcji aktywacji ReLU.