
Regularyzacja Sieci Neuronowe 2020

Jakub Ciszek
238035

Spis treści

1	Opis badań	3
1.1	Plan eksperymentów	3
1.2	Charakterystyka zbiorów danych	3
2	Eksperymenty	4
2.1	Wpływ dropoutu na przebieg procesu uczenia	4
2.2	Wpływ regularyzacji L1 na przebieg procesu uczenia	7
2.3	Wpływ regularyzacji L2 na przebieg procesu uczenia	9
2.4	Wpływ regularyzacji łączonej L1 i L2 na przebieg procesu uczenia	11
2.5	Porównanie metod	13
3	Wnioski	15

Cały kod wykorzystany w zadaniu znajduje się pod adresem: <https://github.com/Greenpp/sieci-neuronowe-pwr-2020>

1 Opis badań

1.1 Plan eksperymentów

Wszystkie eksperymenty zostały przeprowadzone 10 razy. Losowość przy inicjalizacji wag oraz generacji danych nie została narzucona żadnym ziarnem. Podczas badań przyjęto górną granicę 40 epok, po przekroczeniu której, uczenie zostawało przerywane. Ze względu na charakter zadania (klasyfikacja) na ostatniej warstwie użyto funkcji Softmax, a za funkcję straty przyjęto Entropię krzyżową. Sieć posiadała jedną warstwę ukrytą, złożoną z 512 neuronów. Za funkcję aktywacyjną posłużyła ReLU, wagi zostały wylosowane z przedziału $-0.1 - 0.1$. W celu przyspieszenia procesu uczenia do aktualizacji parametrów użyto metody Adam. W procesie uczenia użyto paczek wielkości 1024 elementów.

Zostały przeprowadzone następujące badania:

- Wpływ dropoutu na przebieg procesu uczenia.
- Wpływ regularyzacji L1 na przebieg procesu uczenia.
- Wpływ regularyzacji L2 na przebieg procesu uczenia.
- Wpływ regularyzacji łączonej L1 i L2 na przebieg procesu uczenia.
- Porównanie metod.

Podczas wizualizacji funkcji straty pominięto pierwsze 10 pomiarów dla lepszej czytelności.

1.2 Charakterystyka zbiorów danych

Danymi użytymi w zadaniu jest zbiór ręcznie pisanych cyfr 0 – 9 - MNIST. Na zbiór składa się 70,000 obrazów wielkości 28x28 pikseli, co po przekształceniu odpowiadało 784 elementom wektorowi wejściowemu i 10 klasom na wyjściu. Użyta w zadaniu wersja została podzielona na 3 zbiory:

- Uczący - 50,000 przykładów.
- Walidujący - 10,000 przykładów.
- Testowy - 10,000 przykładów.

W trakcie eksperymentów wykorzystano jedynie pierwsze 10,000 przykładów zbioru uczącego i testowy. Zbiór uczący został ograniczony aby łatwiej pokazać wpływ regularyzacji na przeuczenie.

2 Eksperymenty

2.1 Wpływ dropoutu na przebieg procesu uczenia

Założenia

Tabela 1: Stałe dla eksperymentu 1

Parametr	Wartość
Regularyzacja	Dropout

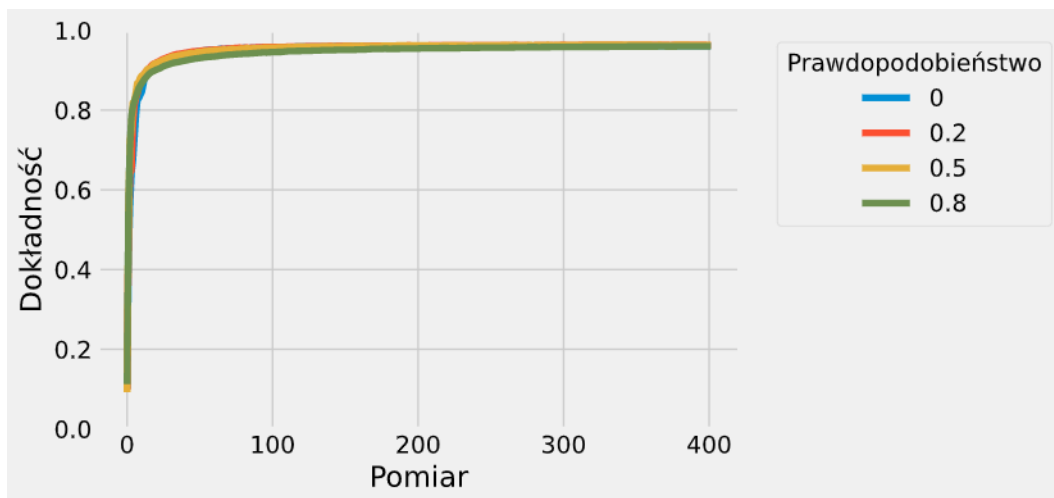
Zmienną w tym eksperymencie była szansa na wyłączenie połączenia. Prawdopodobieństwo przyjmowało wartości ze zbioru $\{0.0, 0.2, 0.5, 0.8\}$

Przebieg

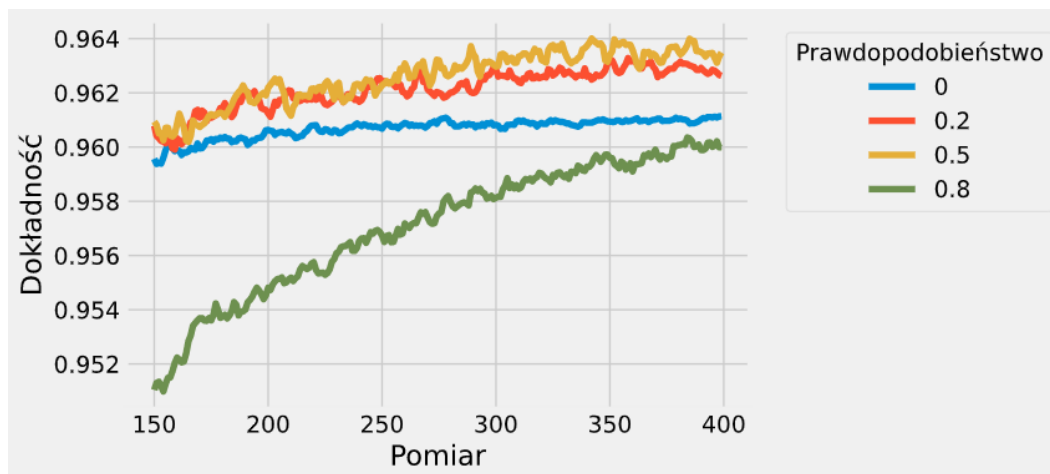
Podczas eksperymentu model został zainicjalizowany 10 razy dla każdej z badanych wartości oraz wyuczony, uzyskane wyniki zostały zapisane w postaci pliku .plk do dalszej analizy.

Wyniki

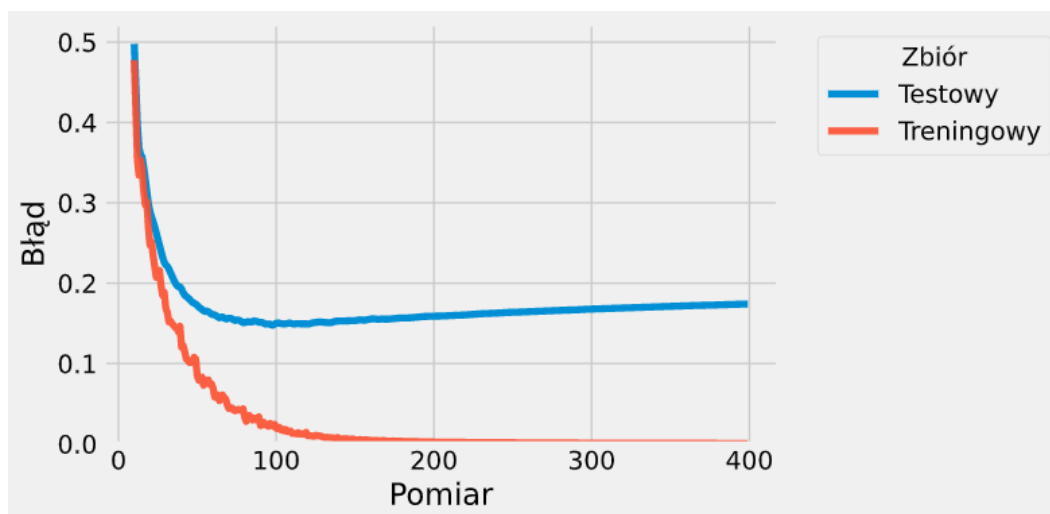
Wykres 1: Dokładność modelu w zależności od prawdopodobieństwa wyłączenia połączenia



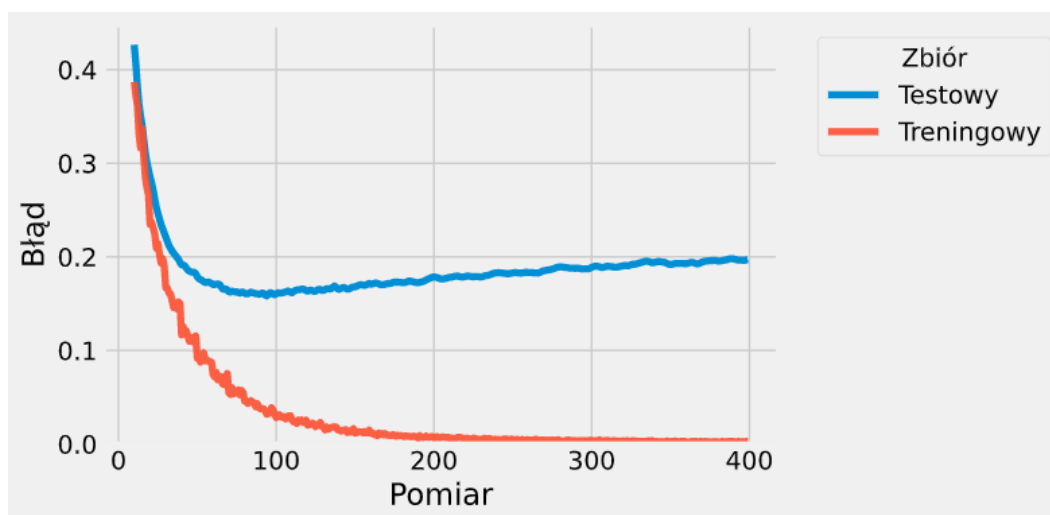
Wykres 2: Dokładność modelu w końcowym etapie uczenia w zależności od prawdopodobieństwa wyłączenia połączenia



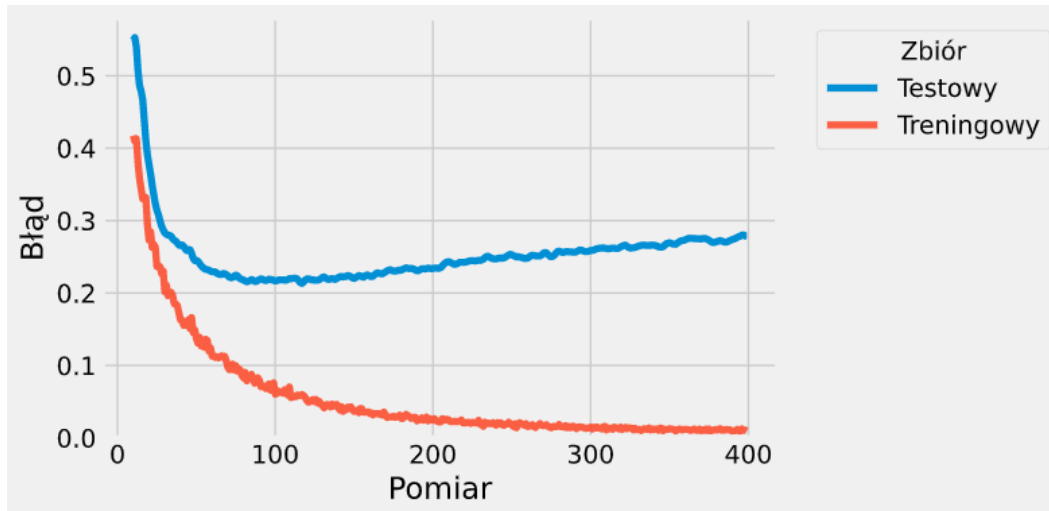
Wykres 3: Zachowanie funkcji błędu dla prawdopodobieństwa 0.0



Wykres 4: Zachowanie funkcji błędu dla prawdopodobieństwa 0.2



Wykres 5: Zachowanie funkcji błędu dla prawdopodobieństwa 0.5



Wykres 6: Zachowanie funkcji błędu dla prawdopodobieństwa 0.8

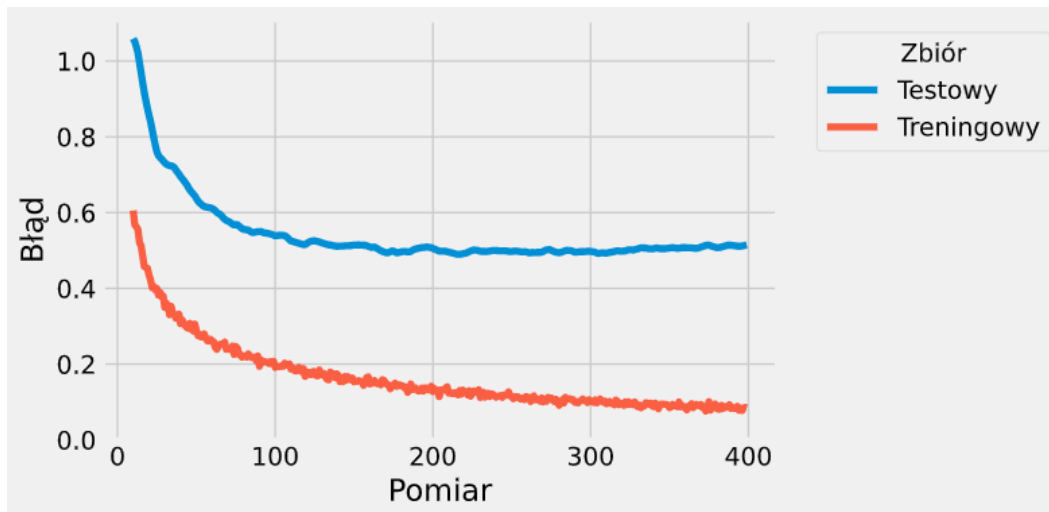


Tabela 2: TODO

Prawdopodobieństwo	Dokładność [%]
0.0	96.17
0.2	96.45
0.5	96.58
0.8	96.14

Wnioski

Na przedstawionych wykresach 1, 2 oraz tabeli 2 można zauważyć, że dokładność modelu nie zmieniała się znacznie, niezależnie od prawdopodobieństwa z jakim połączenia były wyłączone. Zauważalnym efektem jest natomiast zanikający efekt przeuczenia, widoczny na wykresach 3, 4 i 5, widoczny na wykresie 6, przy wyłączeniu 80% połączeń, gdzie błąd treningowy nie spada tak blisko 0 jak w pozostałych przypadkach, a błąd testowy nie rośnie tak gwałtownie.

2.2 Wpływ regularyzacji L1 na przebieg procesu uczenia

Założenia

Tabela 3: Stałe dla eksperymentu 2

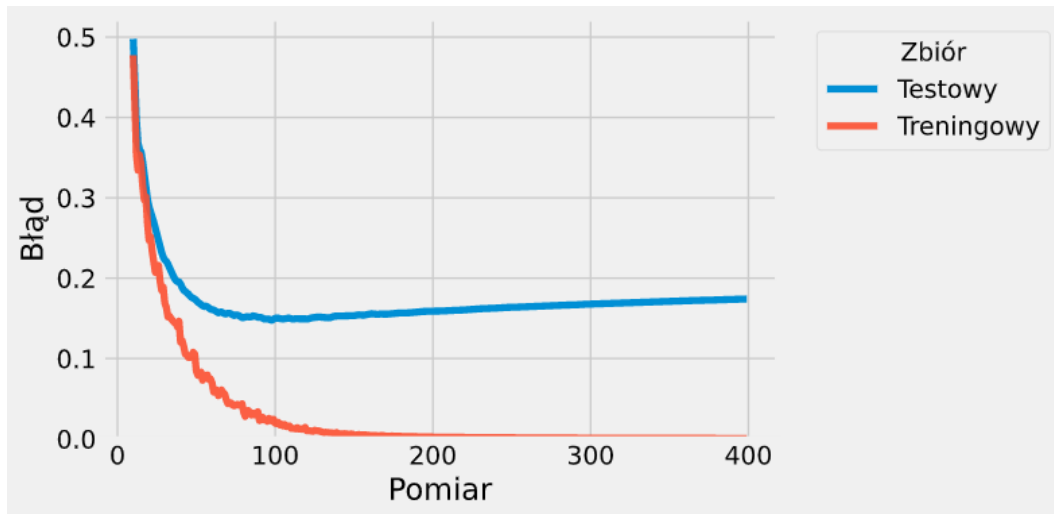
Parametr	Wartość
Regularyzacja	L1
Lambda	0.0001

Przebieg

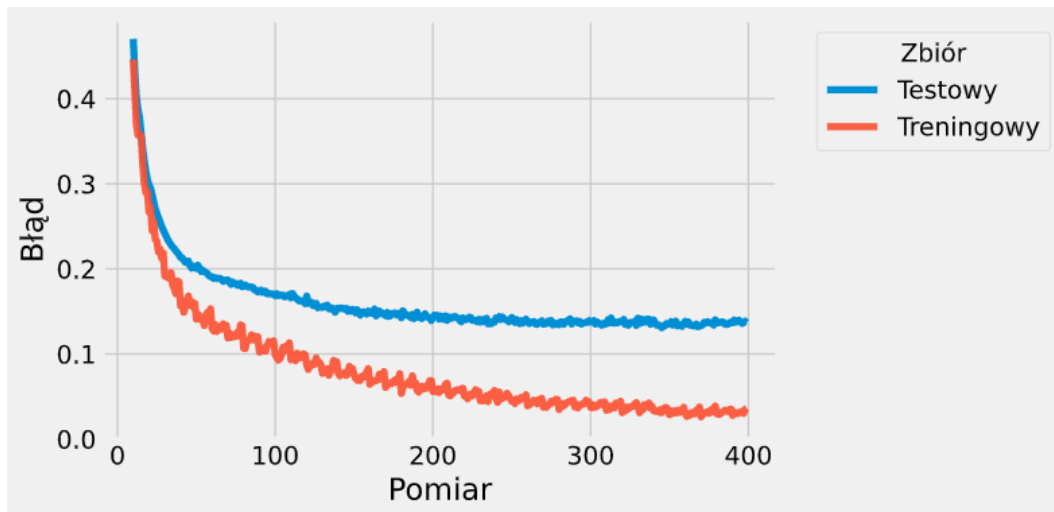
Podczas eksperymentu model został zainicjalizowany 10 razy dla każdej z badanych wartości oraz wyuczony, uzyskane wyniki zostały zapisane w postaci pliku .plk do dalszej analizy.

Wyniki

Wykres 7: Zachowanie funkcji błędu bez regularyzacji



Wykres 8: Zachowanie funkcji błędu dla regularyzacji L1



Wnioski

Na przedstawionym wykresie 8 można zaobserwować eliminację problemu przeuczenia, zaobserwowanego przy uczeniu bez regularyzacji (wykres 7). Błąd testowy w przeprowadzonym eksperymencie nie przestaje spadać, a błąd treningowy nie osiąga 0, co sugeruje lepszą generalizację.

2.3 Wpływ regularyzacji L2 na przebieg procesu uczenia

Założenia

Tabela 4: Stałe dla eksperymentu 3

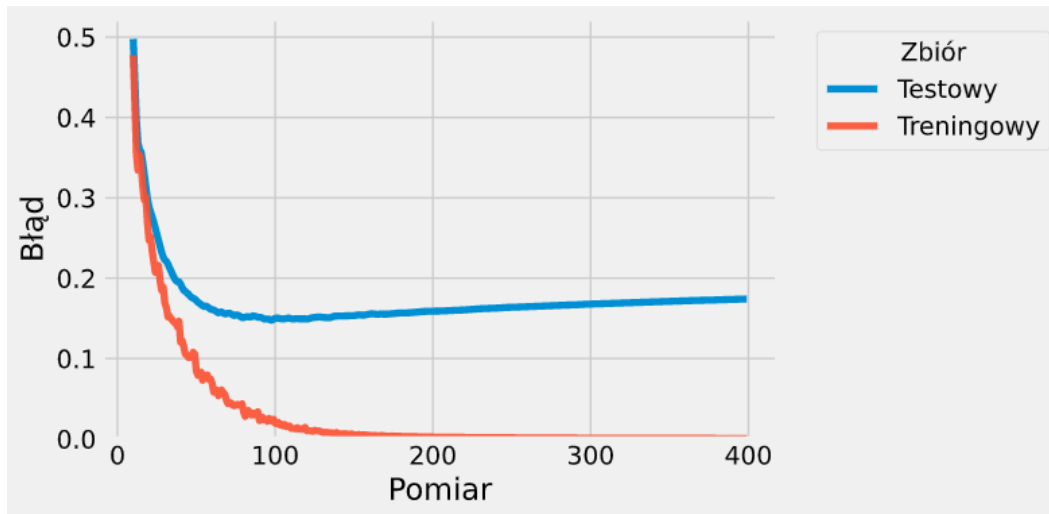
Parametr	Wartość
Regularyzacja	L2
Lambda	0.0001

Przebieg

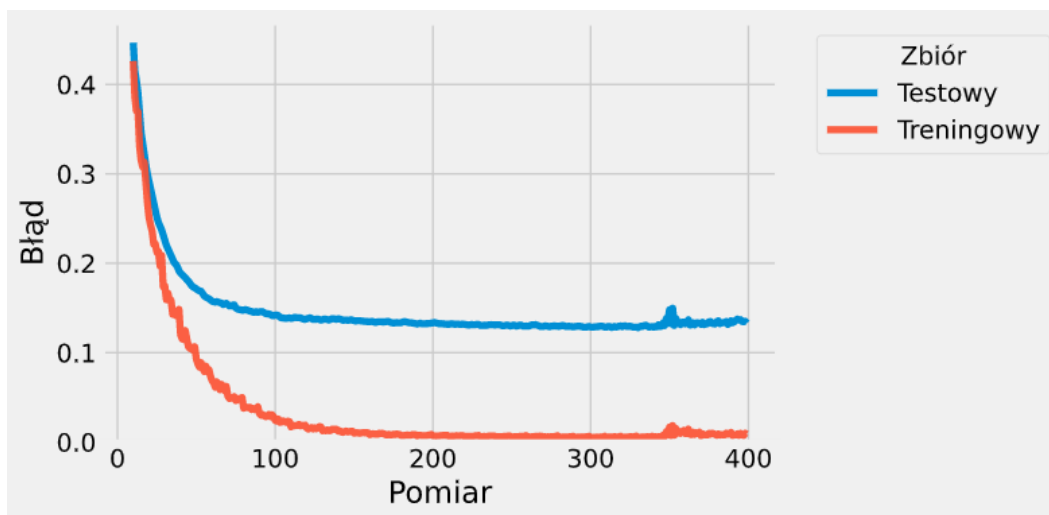
Podczas eksperymentu model został zainicjalizowany 10 razy dla każdej z badanych wartości oraz wyuczony, uzyskane wyniki zostały zapisane w postaci pliku .plk do dalszej analizy.

Wyniki

Wykres 9: Zachowanie funkcji błędu bez regularyzacji



Wykres 10: Zachowanie funkcji błędu dla regularyzacji L2



Wnioski

Na przedstawionym wykresie 10 można zaobserwować eliminację problemu przeuczenia, zaobserwowanego przy uczeniu bez regularyzacji (wykres 9). Błąd testowy w przeprowadzonym eksperymencie stabilizuje się, a błąd treningowy nie osiąga 0, co sugeruje lepszą generalizację.

2.4 Wpływ regularyzacji łączonej L1 i L2 na przebieg procesu uczenia

Założenia

Tabela 5: Stałe dla eksperymentu 4

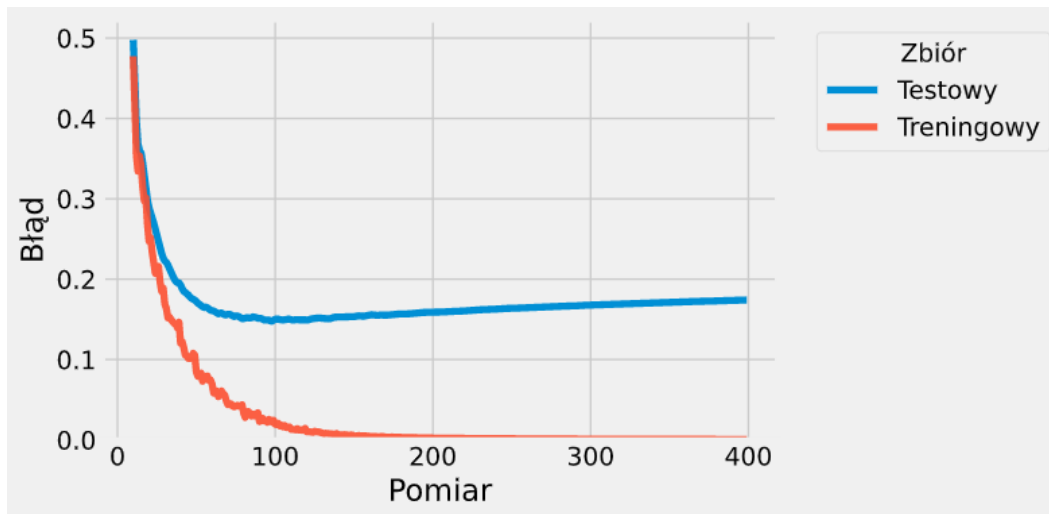
Parametr	Wartość
Regularyzacja	L1 L2
Lambda1	0.0001
Lambda2	0.0001

Przebieg

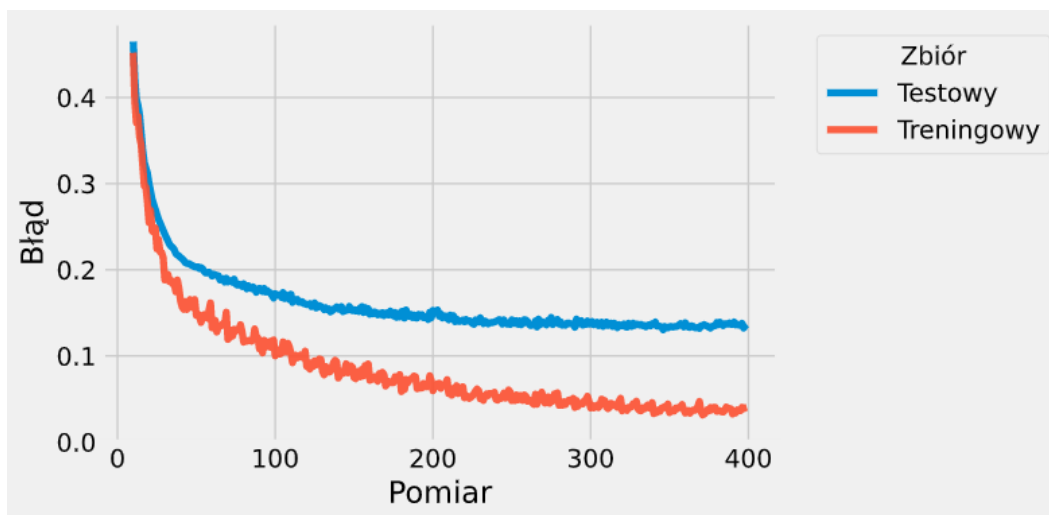
Podczas eksperymentu model został zainicjalizowany 10 razy dla każdej z badanych wartości oraz wyuczony, uzyskane wyniki zostały zapisane w postaci pliku .plk do dalszej analizy.

Wyniki

Wykres 11: Zachowanie funkcji błędu bez regularyzacji



Wykres 12: Zachowanie funkcji błędu dla regularyzacji L1 L2



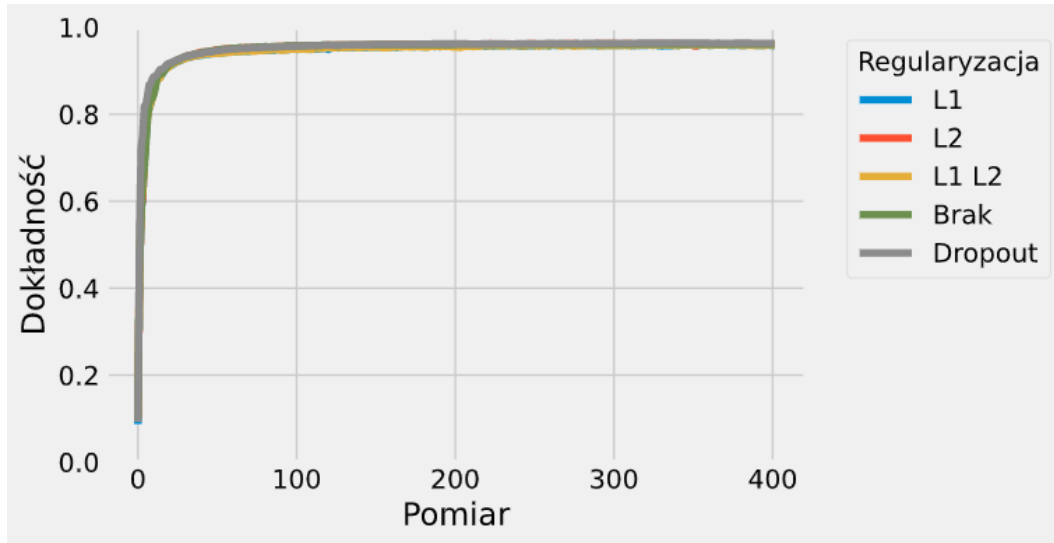
Wnioski

Na przedstawionym wykresie 12 można zaobserwować eliminację problemu przeuczenia, zaobserwowanego przy uczeniu bez regularyzacji (wykres 11). Błąd testowy w przeprowadzonym eksperymencie nie przestaje spadać, a błąd treningowy nie osiąga 0, co sugeruje lepszą generalizację.

2.5 Porównanie metod

Wyniki

Wykres 13: Dokładność modelu w zależności od metody regularyzacji



Wykres 14: Dokładność modelu w końcowym etapie uczenia w zależności od metody regularyzacji

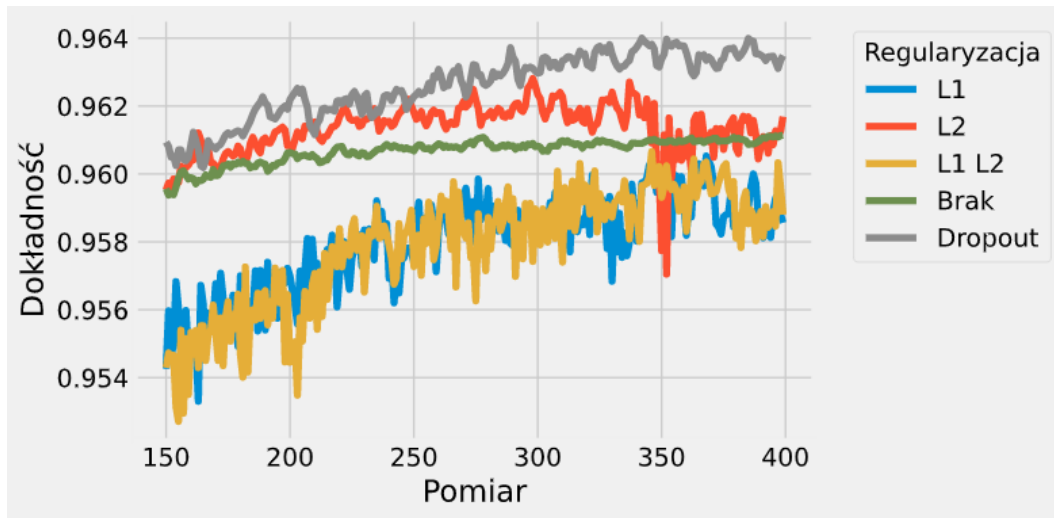


Tabela 6: TODO

Regularyzacja	Dokładność [%]
L1	96.34
L2	96.43
L1 L2	96.36
Brak	96.17
Dropout	96.58

Wnioski

Na przedstawionych wykresach 13 i 14 oraz tabeli 6 widać, że w przypadku rozpatrywanego zbioru danych zastosowanie regularyzacji nie dało znaczących zmian w dokładności modelu. Jednak każda z zastosowanych metod poradziła sobie w innym stopniu z problemem przeuczenia widocznym na wartościach funkcji błędu. Najgorsze wyniki dała metoda Dropout, która potrzebowała aż 80% wyłączonych połączeń dla stabilizacji funkcji błędu, jednak przez fakt, że połączenia te są wyłączane w sposób losowy, trening takiego modelu trwał dłużej, co widać na wykresie 1, gdzie sama dokładność zrównuje się z pozostałymi dopiero pod sam koniec procesu uczenia. Najlepsze efekty dały metody L1 oraz połączenie L1 i L2. Spowodowane jest to prawdopodobnie, popiera to też 80% Dropout, tym, że warstwa ukryta złożona z 512 neuronów była zdecydowanie za duża i wyłączenie większości z nich dawało najlepszy efekt regularyzacyjny. Sama metoda L2 była w stanie ustabilizować funkcję błędu i zapobiec przeuczeniu.

3 Wnioski

- Zbyt skomplikowane modele mogą zbyt przystosować się do danych uczących co negatywnie odbija się na ich generalizacji.
- Regularyzacja L1 pozwala na wyłączenie mniej ważnych połączeń w modelu co pozwala na jego zmniejszenie, a co za tym idzie przeciwdziałanie przeuczeniu.
- Regularyzacja L2 zmniejsza wartości wszystkich parametrów w modelu, jednak nie doprowadza ich do 0, co może być przydatne jeśli nie chcemy wykluczyć żadnych informacji.
- Regularyzacja pozwala na stosowanie większych modeli bez problemu przeuczenia.