Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Sieci neuronowe w zastosowaniach biomedycznych

Rozpoznawanie cyfr pisanych ręcznie ze zbioru MNIST

Sprawozdanie

Kamil Gabryjelski, Antoni Różański

Prowadzący: mgr inż. Piotr Płoński

Warszawa, 2017

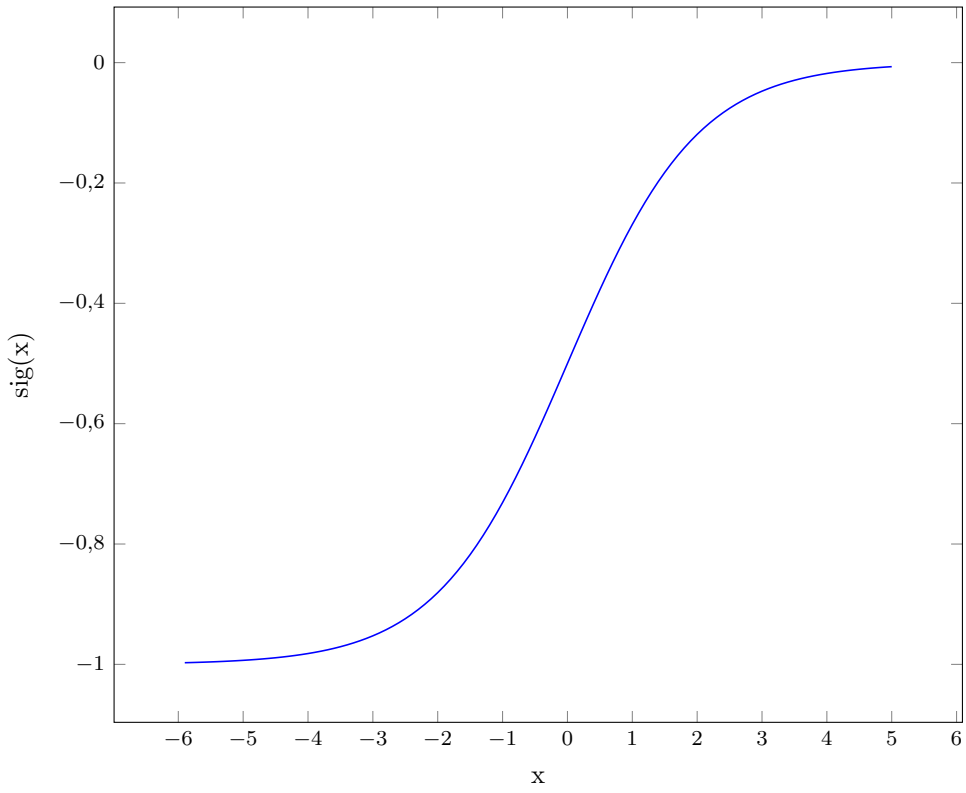
1. Dane

Mieliśmy do dyspozycji 70000 próbek cyfr pisanych ręcznie, każda składająca się z 784 pikseli. Dane podzielone zostały na:
— zbiór uczący - 55000 próbek
— zbiór testowy - 10000 próbek
— zbiór walidacyjny - 5000 próbek

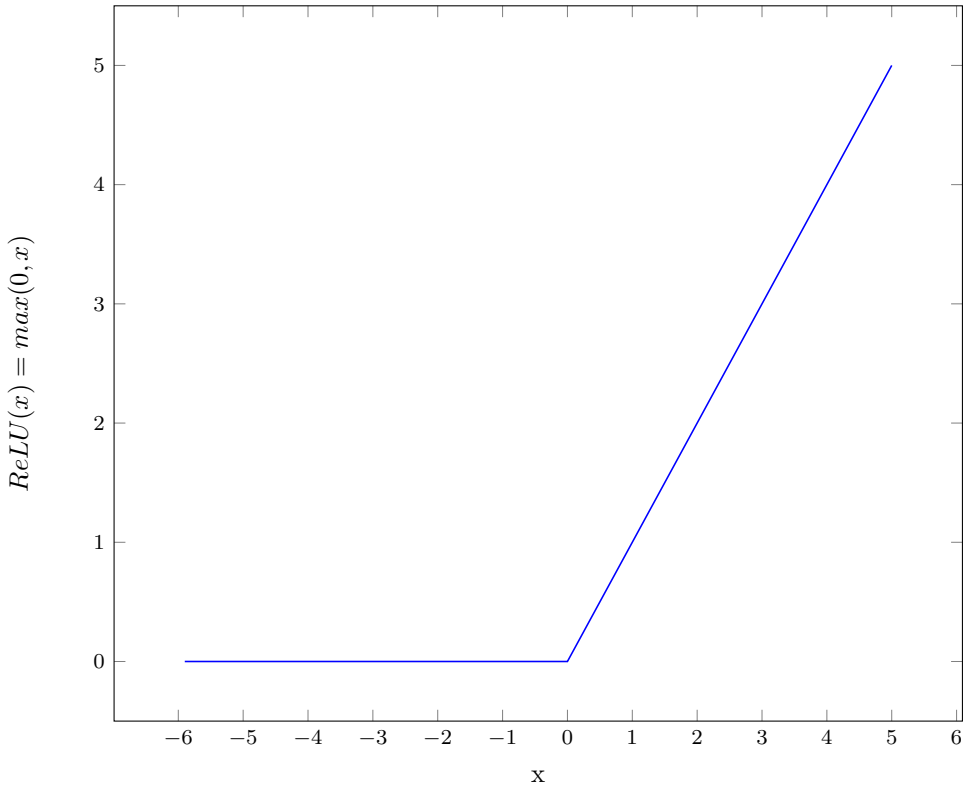
2. Implementacja sieci

2.1. Funkcje aktywacji

Przetestowanymi przez nas funkcjami aktywacji neuronów są funkcja sigmoidalna(wykres 2.1) oraz ReLU, która przedstawiona jest na wykresie 2.2.



Rys. 2.1: Sigmoidalna funkcja aktywacji



Rys. 2.2: Funkcja aktywacji ReLU

2.2. Struktura sieci

Warstwa wejściowa składa się z 784 neuronów, po jednym na każdy piksel obrazka z cyfrą. W trakcie eksperymentów zbadane zostały struktury sieci z dwiema i trzema warstwami ukrytymi. Warstwę wyjściową tworzy 10 neuronów, gdyż oczekujemy, że sieć zwróci jedną z dziesięciu cyfr. Na warstwie wyjściowej używana jest funkcja $softmax$, dana wzorem 2.1, dzięki której wyniki możemy interpretować jako prawdopodobieństwa.

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{n} e^{x_i}} \tag{2.1}$$

2.3. Błąd sieci

W celu obliczenia błędu warstwy wyjściowej stosowana jest metoda cross entropy, dana wzorem 2.2.

$$L(w) = -\frac{1}{N}\sum_{n=1}^{N}[y_n log\hat{y}_n + (1-y_n)log(1-\hat{y}_n)] \tag{2.2}$$

2.4. Proces uczenia

Testowanymi algorytmami uczącymi są stochastyczny spadek gradientu oraz jego modyfikacja wykorzystująca pęd. Ponadto, zastosowaliśmy technikę *dropout* polegającą na usuwaniu (zerowaniu) losowych połączeń między neuronami sąsiadujących warstw. Ma to na celu zapobiegnięcie zjawisku dopasowywania się sieci do danych uczących. Stosowane przez nas prawdopodobieństwo zachowania połączenia wynosi 0,95.

Zastosowaliśmy technikę wykładniczego spadku wartości kroku, dzięki czemu możemy użyć dużej wartości początkowej kroku. Sprawia to, że sieć uczy się szybko na początku eksperymentu, a zwalnia gdy jest w pobliżu optymalnego rozwiązania.

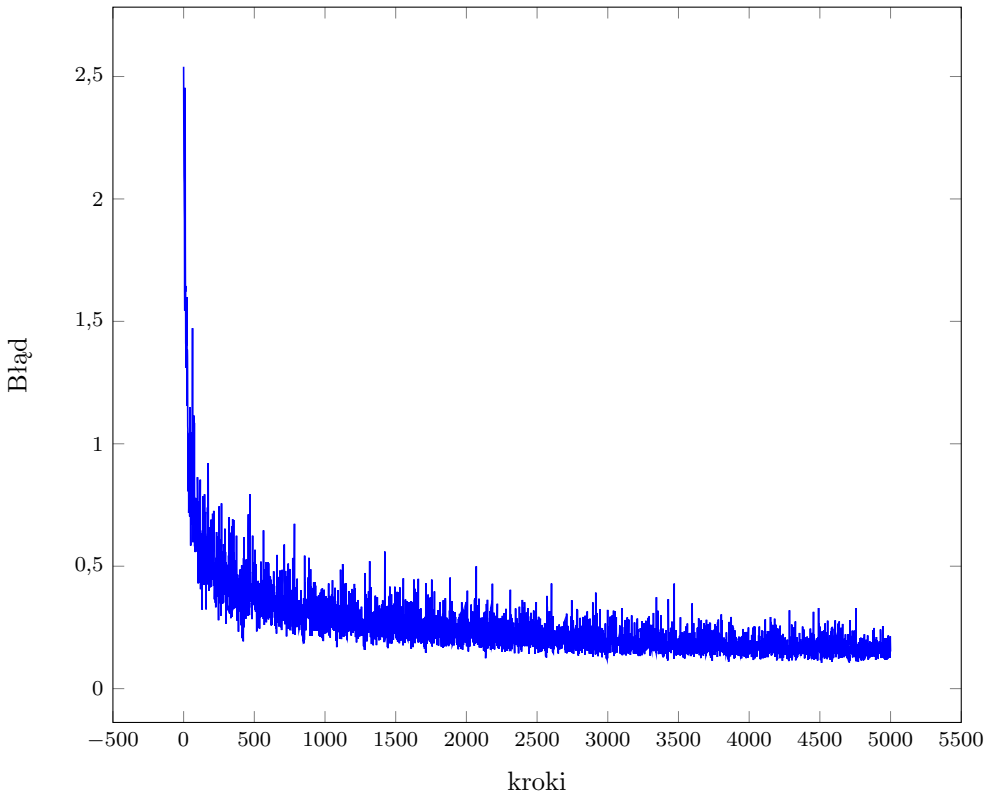
3. Eksperymenty

Badane będą różne struktury sieci, funkcje aktywacji, algorytmy uczenia, szybkości uczenia oraz wielkości serii (dalej nazywane batch size).

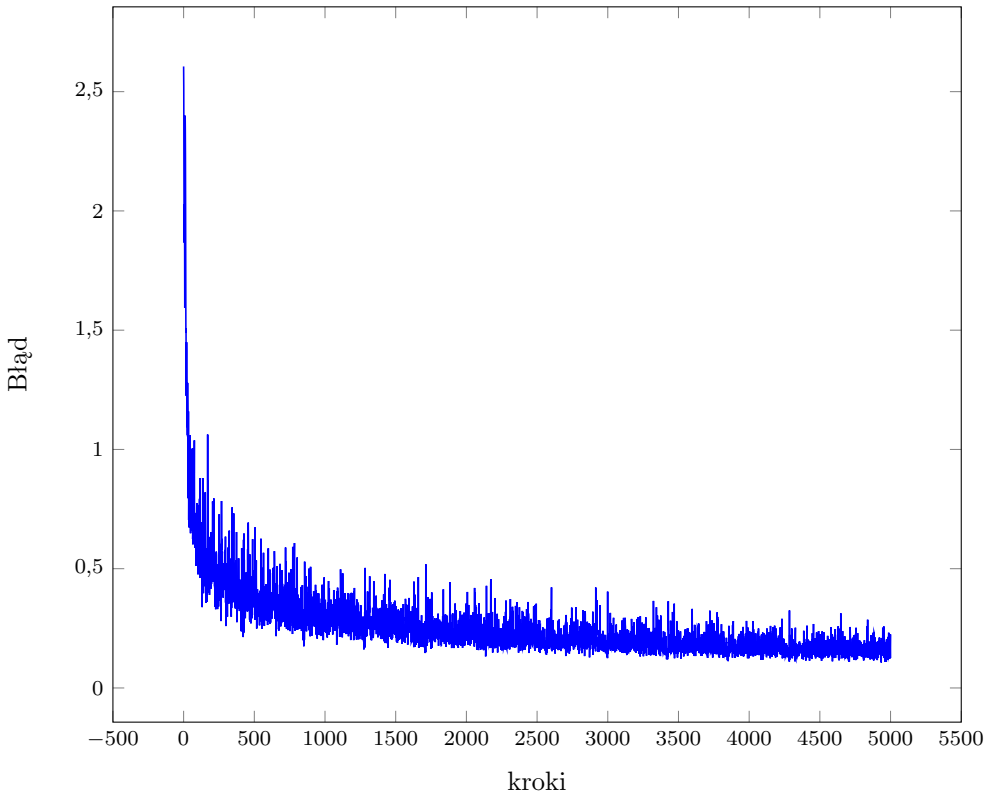
Ze względu na niedeterministyczny charakter sieci neuronowych, każdy zestaw parametrów był testowany kilkukrotnie i wybierany był najlepszy z uzyskanych wyników.

3.1. Struktura sieci

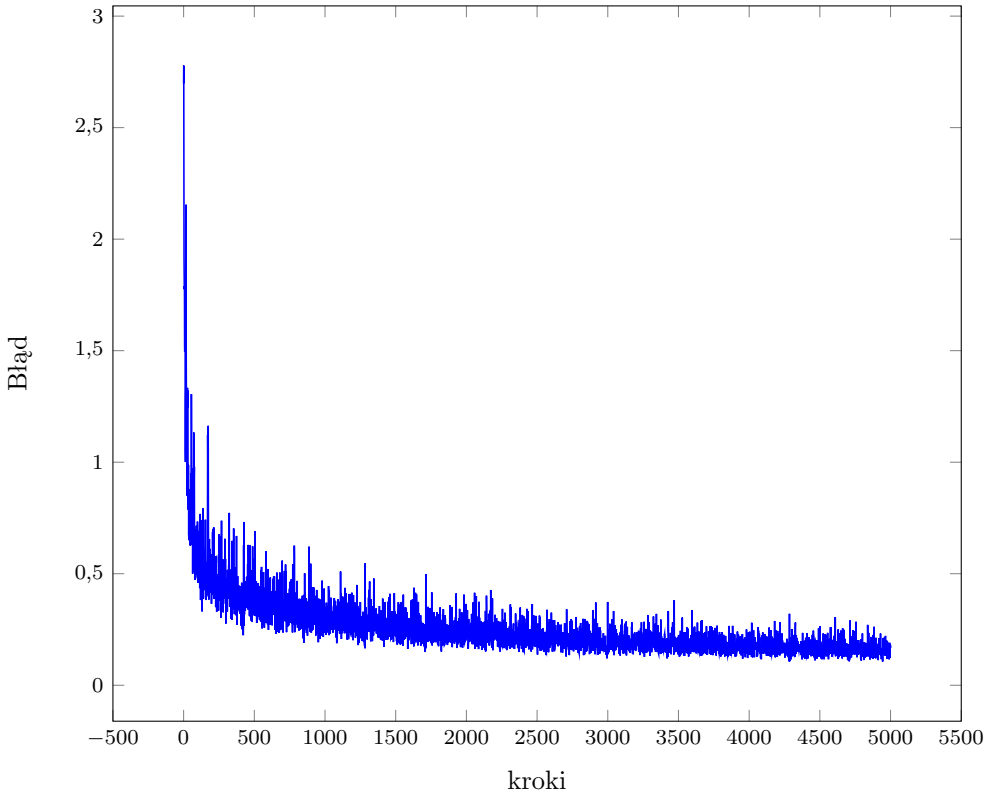
Wyniki przedstawione w tabeli 3.1 pozwalają stwierdzić, że różnice w jakości rozpoznawania cyfry przez sieć w zależności od jej struktury są niewielkie. Sieć prawidłowo rozpoznaje cyfrę w około 98% przypadków. Analizując wykresy błędów można jednak zauważyć, że im więcej neuronów w sieci, tym mniejsza wariancja funkcji błędu. Tę zależność wyraźnie widać, porównując przykładowo wykresy 3.2 i 3.8. Biorąc tę obserwację pod uwagę, w kolejnych eksperymentach używana będzie struktura sieci z 2 warstwami ukrytymi po 400 neuronów w każdej.



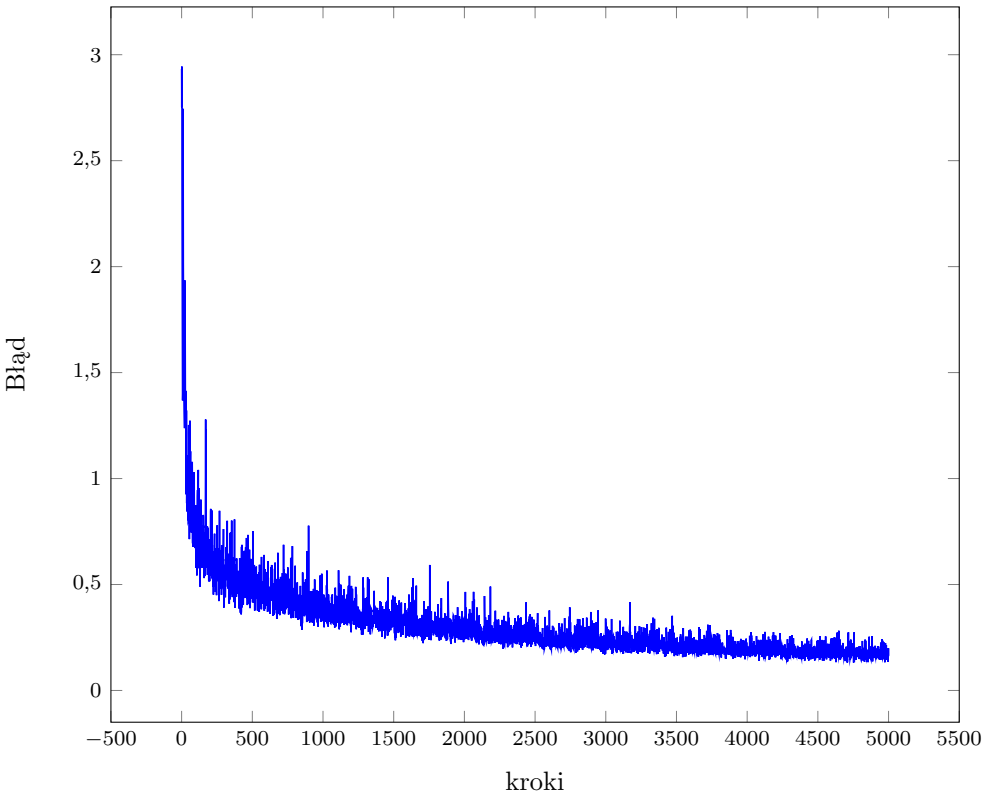
Rys. 3.1: 2 warstwy ukryte, (100, 50) neuronów



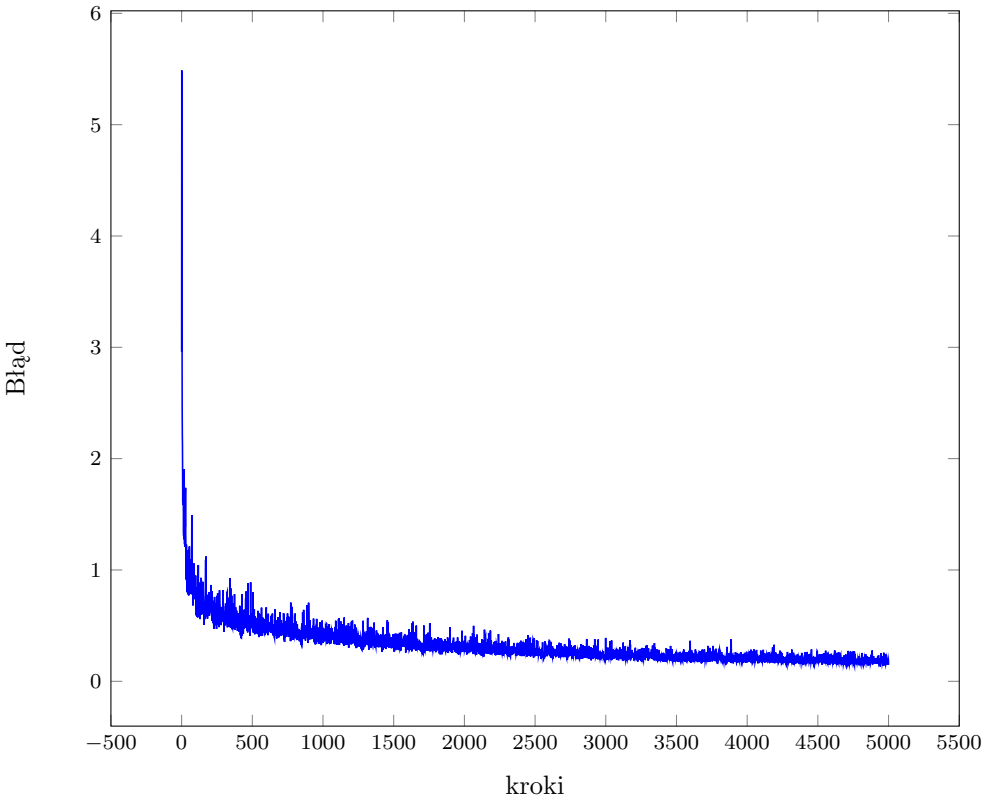
Rys. 3.2: 2 warstwy ukryte, (100, 100) neuronów



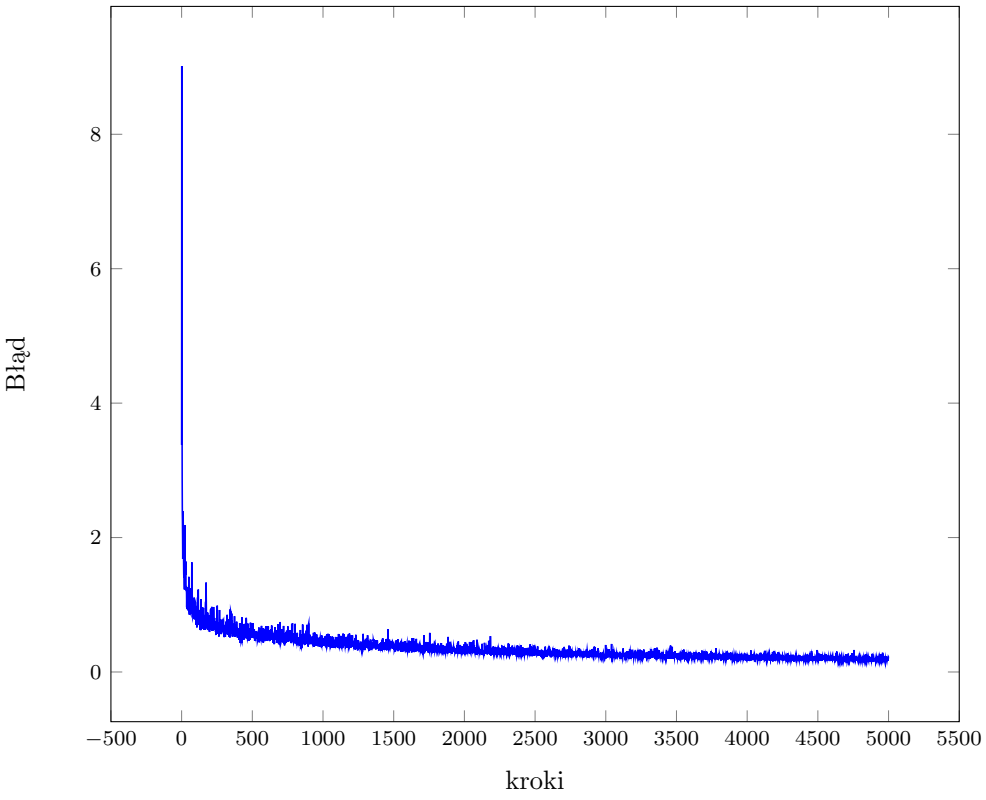
Rys. 3.3: 2 warstwy ukryte, (100, 150) neuronów



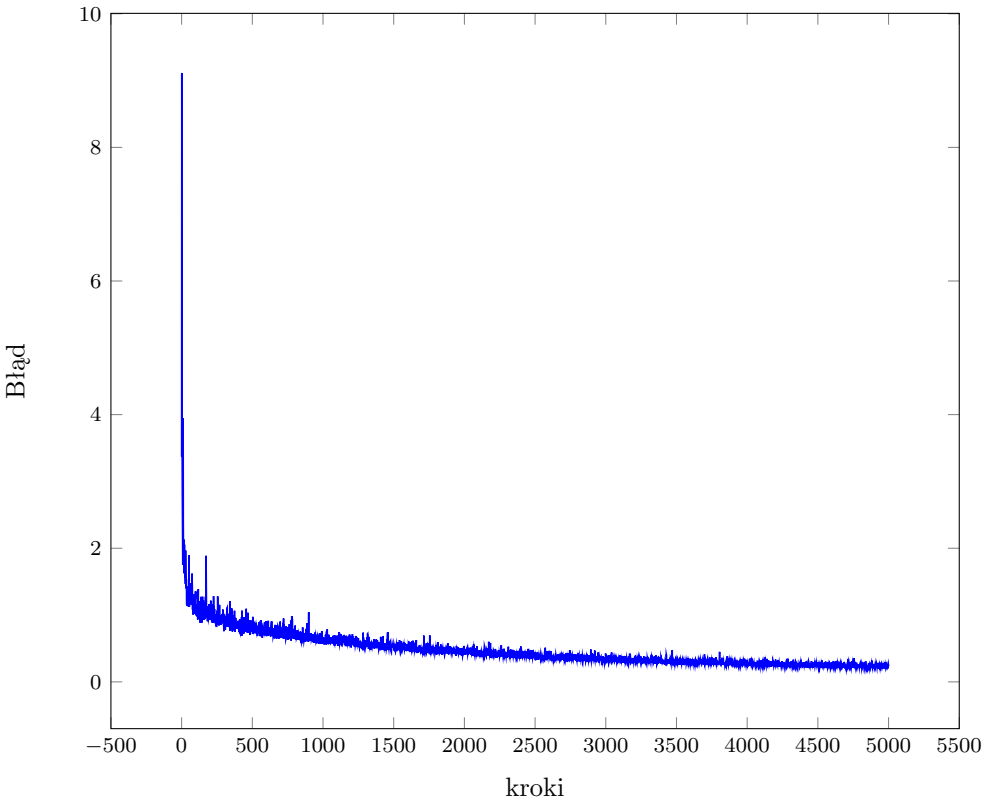
Rys. 3.4: 2 warstwy ukryte, (200, 100) neuronów



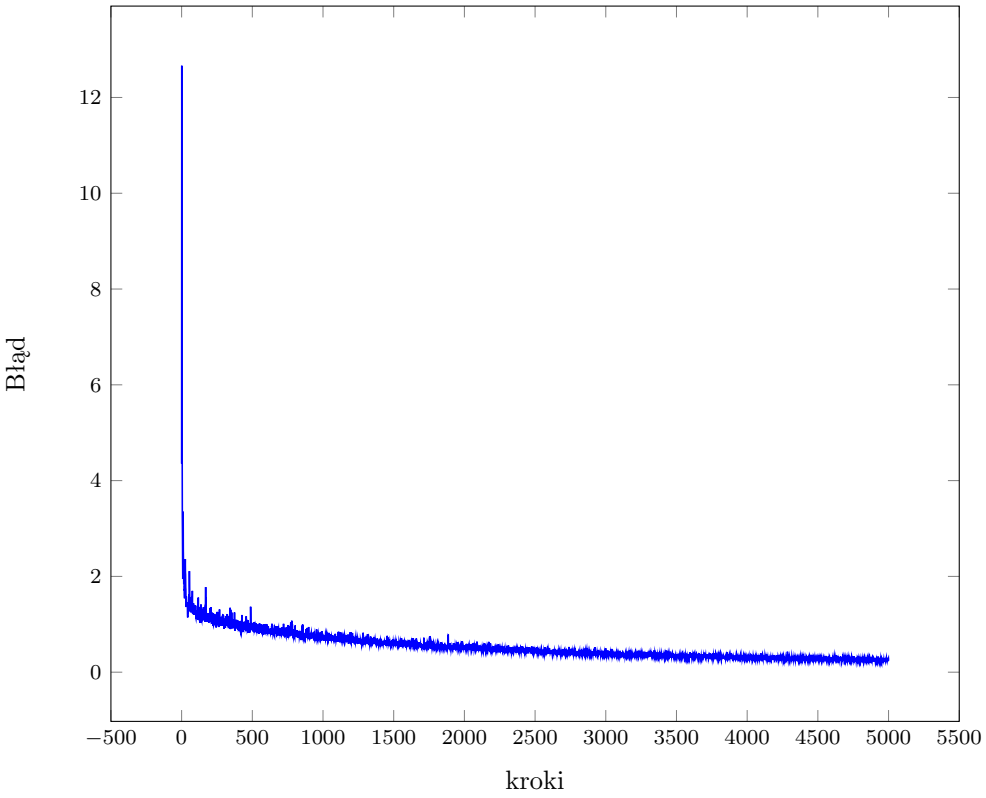
Rys. 3.5: 2 warstwy ukryte, (200, 200) neuronów



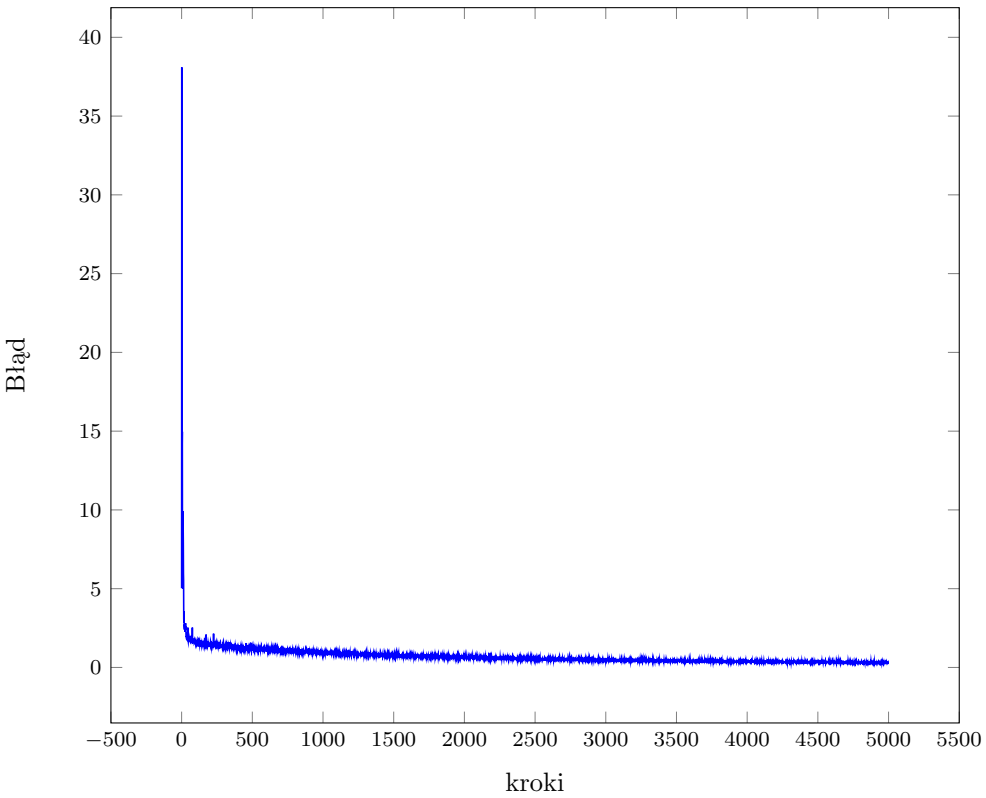
Rys. 3.6: 2 warstwy ukryte, (200, 300) neuronów



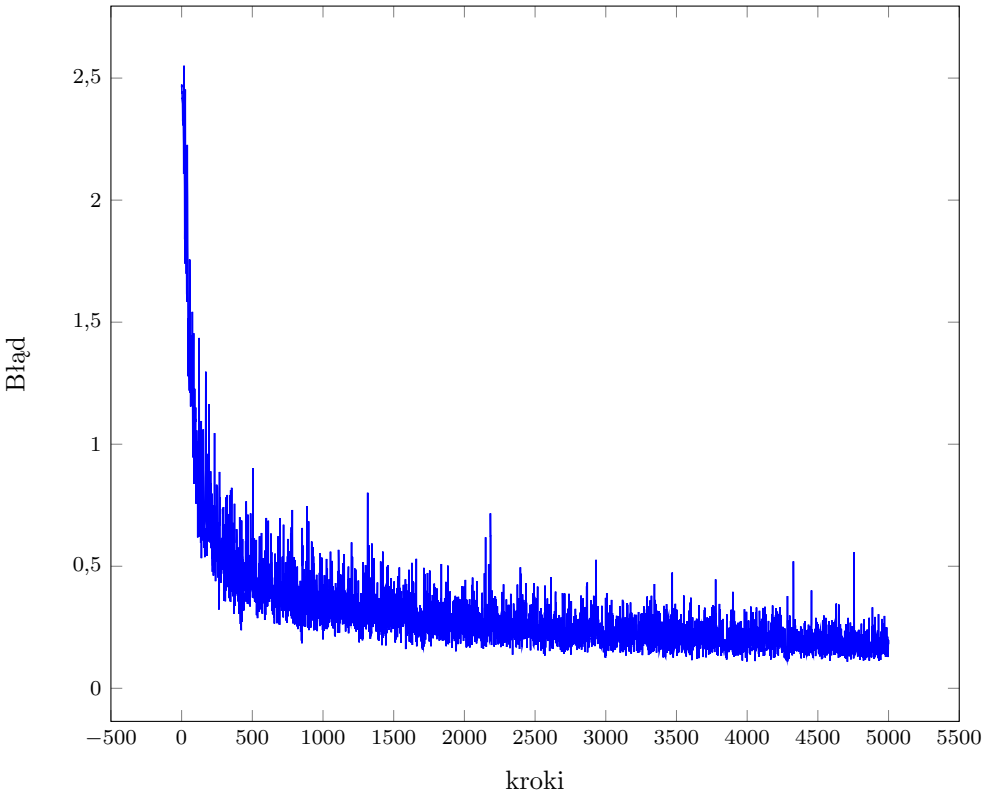
Rys. 3.7: 2 warstwy ukryte, (400, 200) neuronów



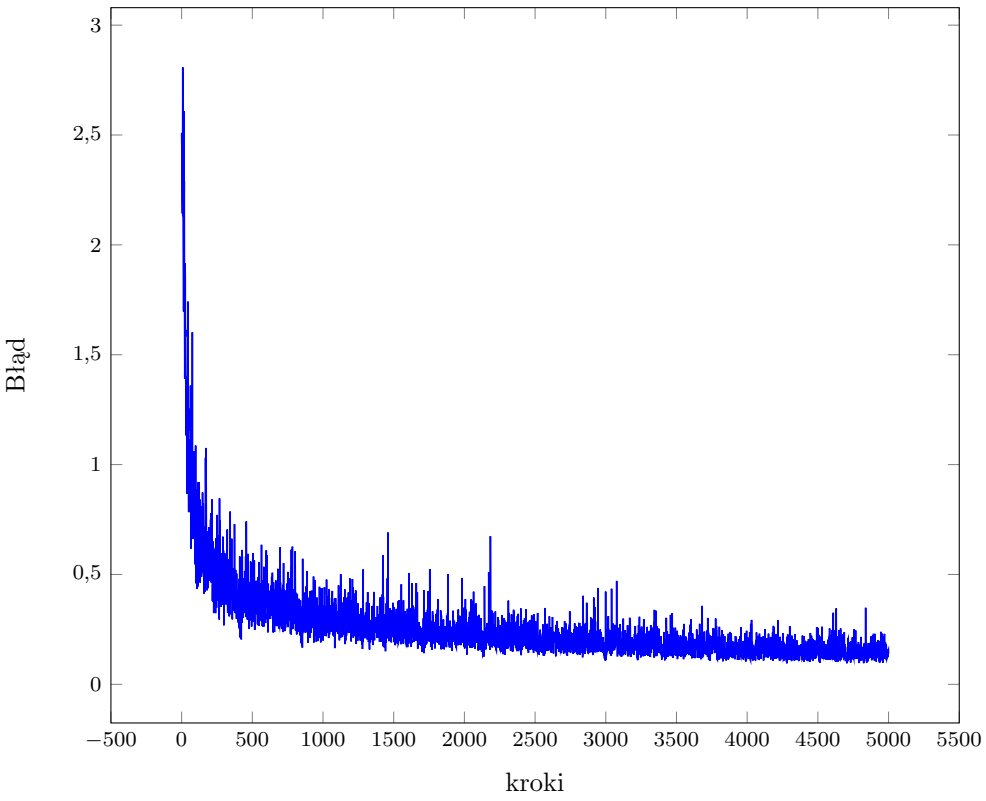
Rys. 3.8: 2 warstwy ukryte, (400, 400) neuronów



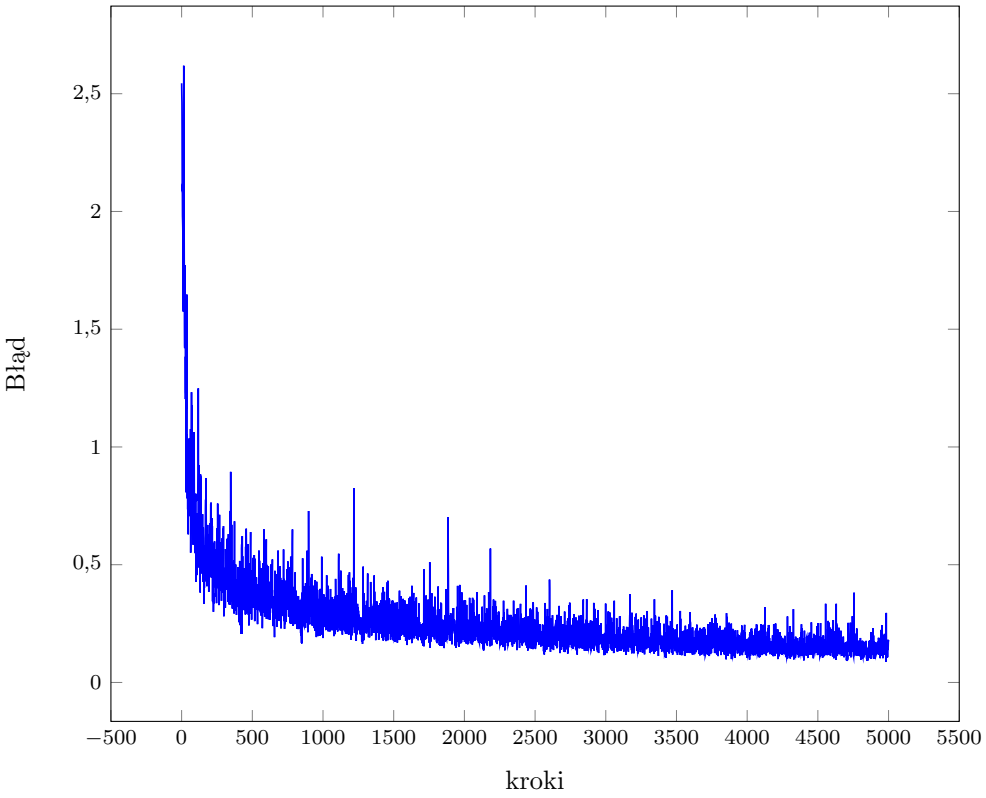
Rys. 3.9: 2 warstwy ukryte, (400, 600) neuronów



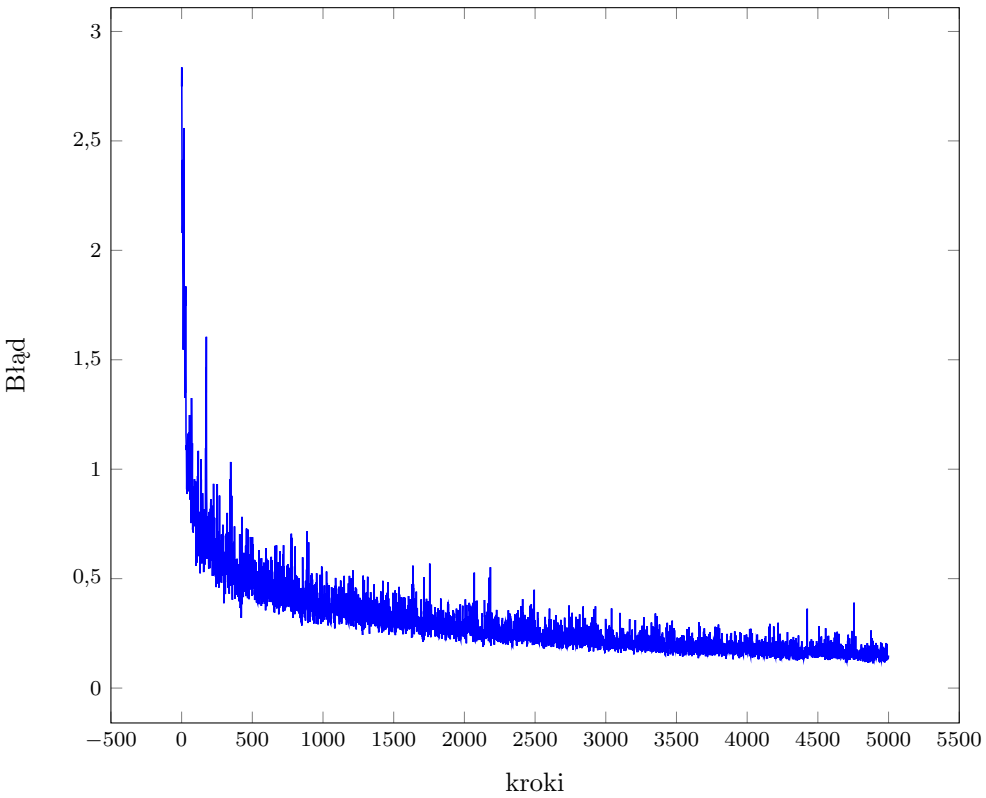
Rys. 3.10: 3 warstwy ukryte, (100, 50, 10) neuronów



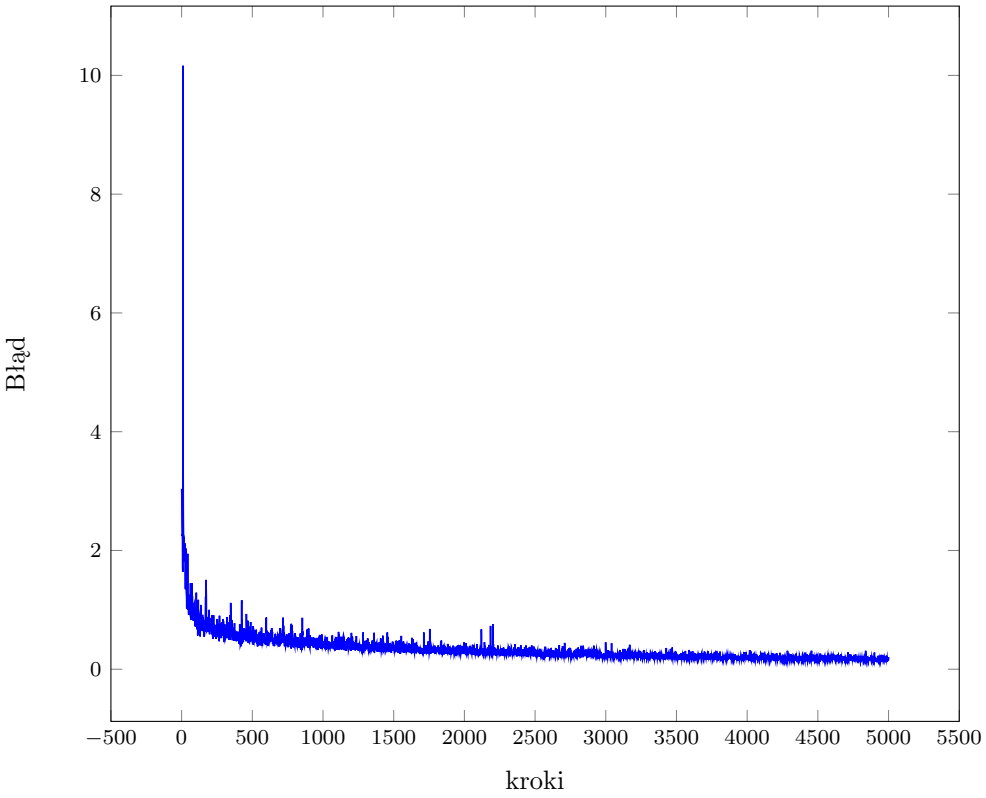
Rys. 3.11: 3 warstwy ukryte, (100, 100, 50) neuronów



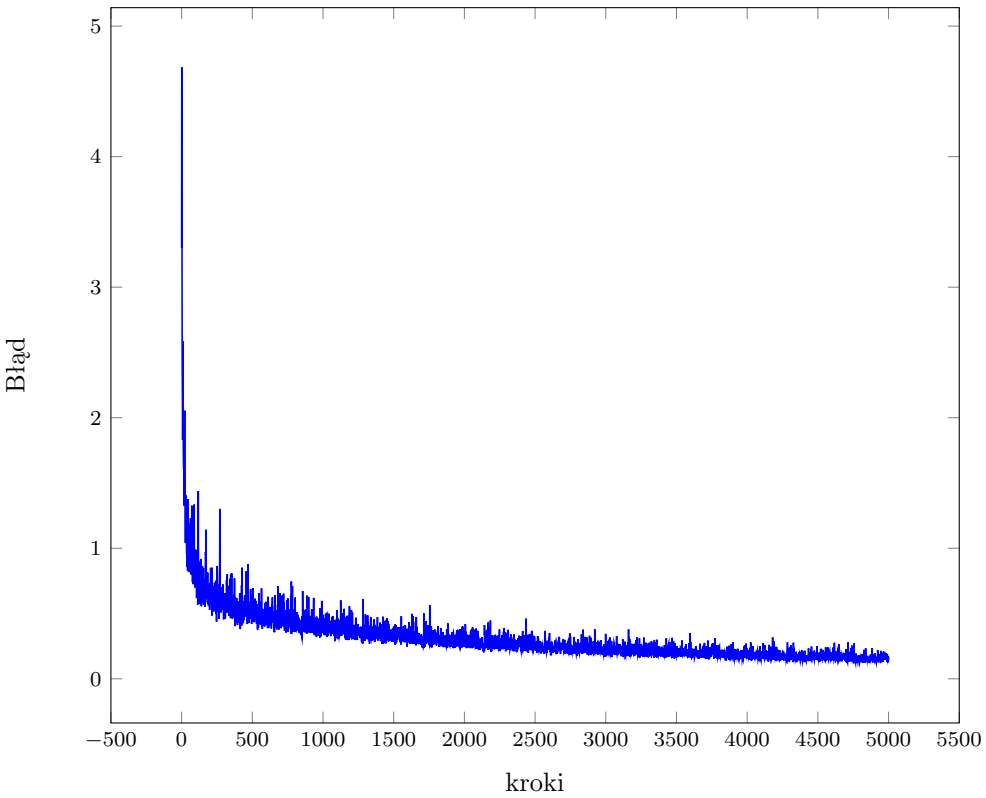
Rys. 3.12: 3 warstwy ukryte, (100, 100, 100) neuronów



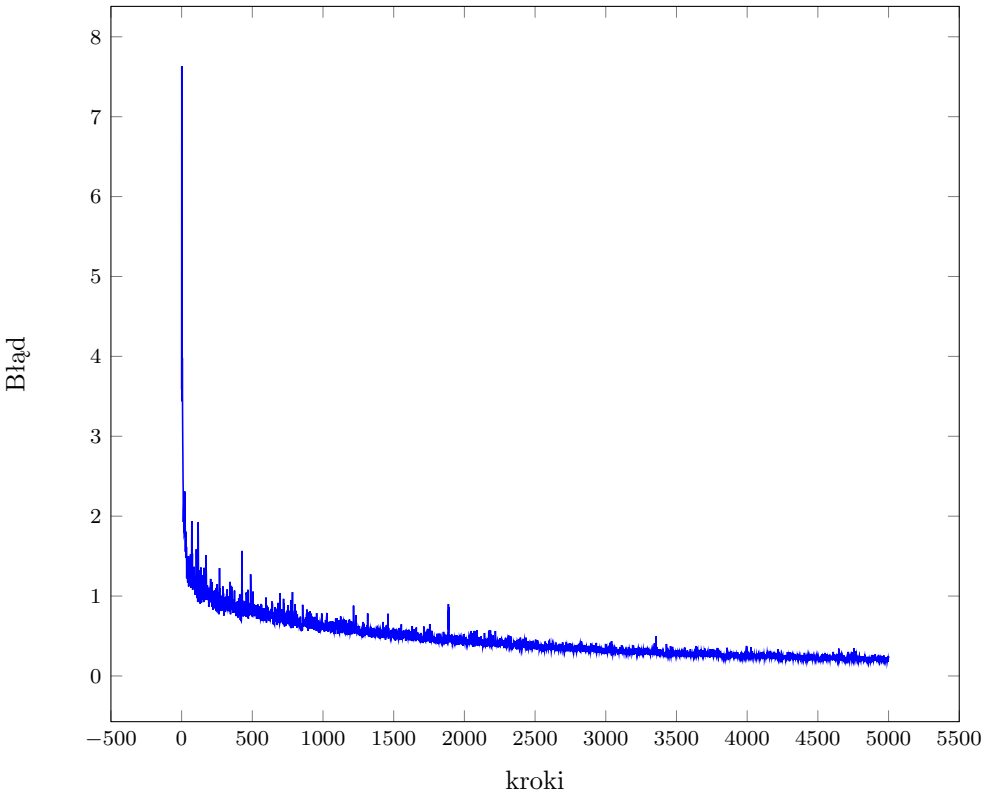
Rys. 3.13: 3 warstwy ukryte, (200, 100, 100) neuronów



Rys. 3.14: 3 warstwy ukryte, (200, 200, 100) neuronów



Rys. 3.15: 3 warstwy ukryte, (200, 200, 200) neuronów



Rys. 3.16: 3 warstwy ukryte, (400, 200, 200) neuronów

Ilość warstw ukrytych	Ilość neuronów	Wyniki
2	100, 50	Zbiór uczący: 97,7% Zbiór testowy: 97,5%
2	100, 100	Zbiór uczący: 99,2% Zbiór testowy: 97,8%
2	100, 150	Zbiór uczący: 97,7% Zbiór testowy: 97,9%
2	200, 100	Zbiór uczący: 98,4% Zbiór testowy: 97,9%
2	200, 200	Zbiór uczący: 98,4% Zbiór testowy: 98%
2	200, 300	Zbiór uczący: 97,7% Zbiór testowy: 97,9%
2	400, 200	Zbiór uczący: 98,4% Zbiór testowy: 98%
2	400, 400	Zbiór uczący: 98,4% Zbiór testowy: 98%
2	400, 600	Zbiór uczący: 97,7% Zbiór testowy: 97,9%
3	100, 50, 10	Zbiór uczący: 96,9% Zbiór testowy: 97,7%
3	100, 100, 50	Zbiór uczący: 97,7% Zbiór testowy: 98%
3	100, 100, 100	Zbiór uczący: 97,7% Zbiór testowy: 97,7%
3	200, 100, 100	Zbiór uczący: 97,7% Zbiór testowy: 98%
3	200, 200, 100	Zbiór uczący: 98,4% Zbiór testowy: 97,9%
3	200, 200, 200	Zbiór uczący: 99,2% Zbiór testowy: 98,1%
3	400, 200, 200	Zbiór uczący: 97,7% Zbiór testowy: 98,1%

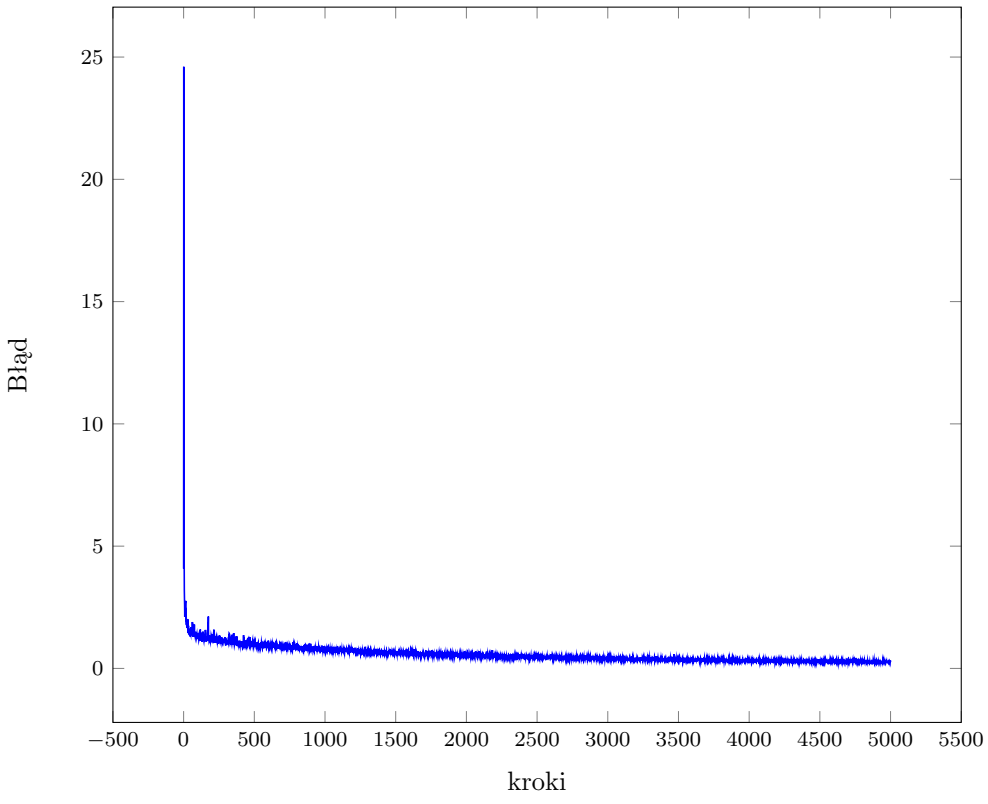
Tab. 3.1: Dane po 5000 kroków

3.2. Funkcja aktywacji

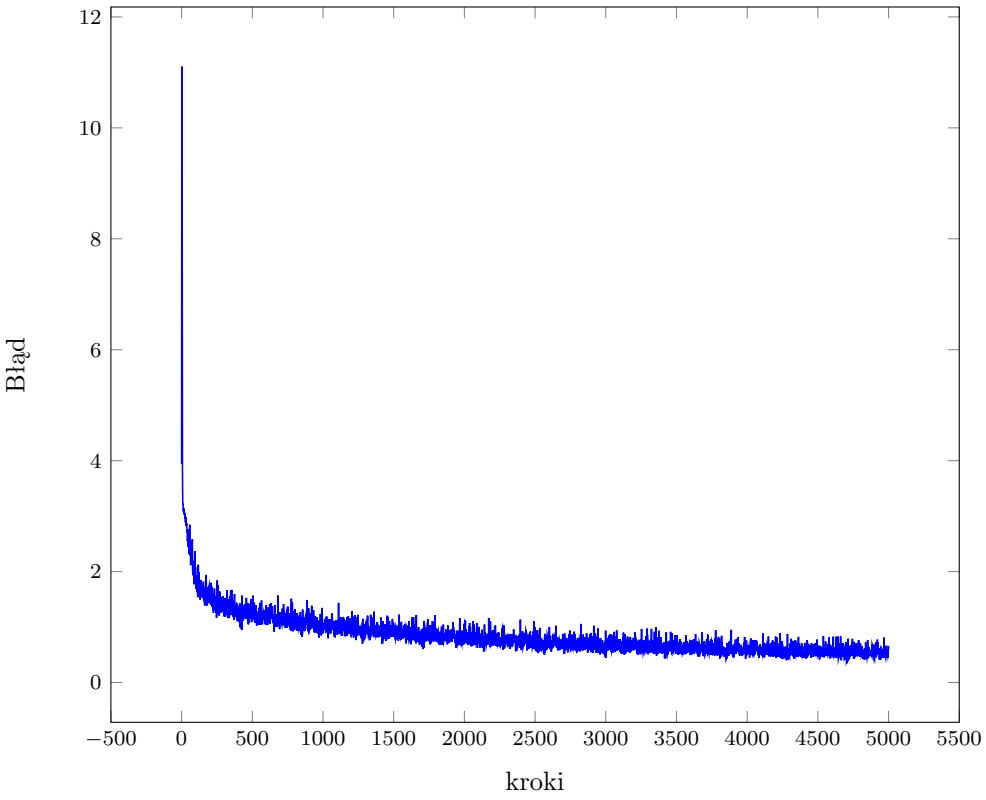
Wyniki przedstawione w tabeli 3.2 pozwalają jednoznacznie stwierdzić, że funkcja aktywacji ReLU daje znacznie lepsze wyniki niż sigmoidalna. Ponadto porównując wykresy 3.17 i 3.18 można zauważyć, że funkcja błędu ma znacznie mniejszą wariancję dla ReLU niż w przypadku funkcji sigmoidalnej. W kolejnych eksperymentach używana będzie funkcja aktywacji ReLU.

Funkcja aktywacji	Wyniki
ReLU	Zbiór uczący: 98,4% Zbiór testowy: 98%
Sigmoidalna	Zbiór uczący: 90.6% Zbiór testowy: 94.2%

Tab. 3.2: Porównanie działania funkcji aktywacji ReLU i sigmoidalnej



Rys. 3.17: Funkcja aktywacji ReLU



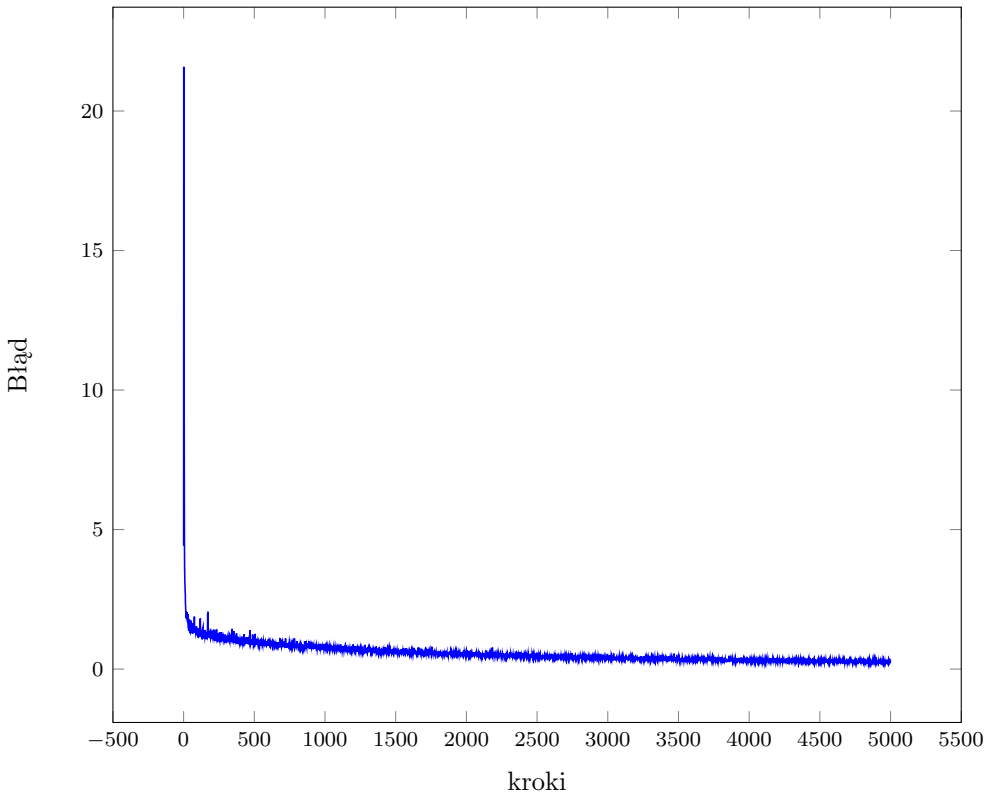
Rys. 3.18: Funkcja aktywacji sigmoidalna

3.3. Algorytm uczenia

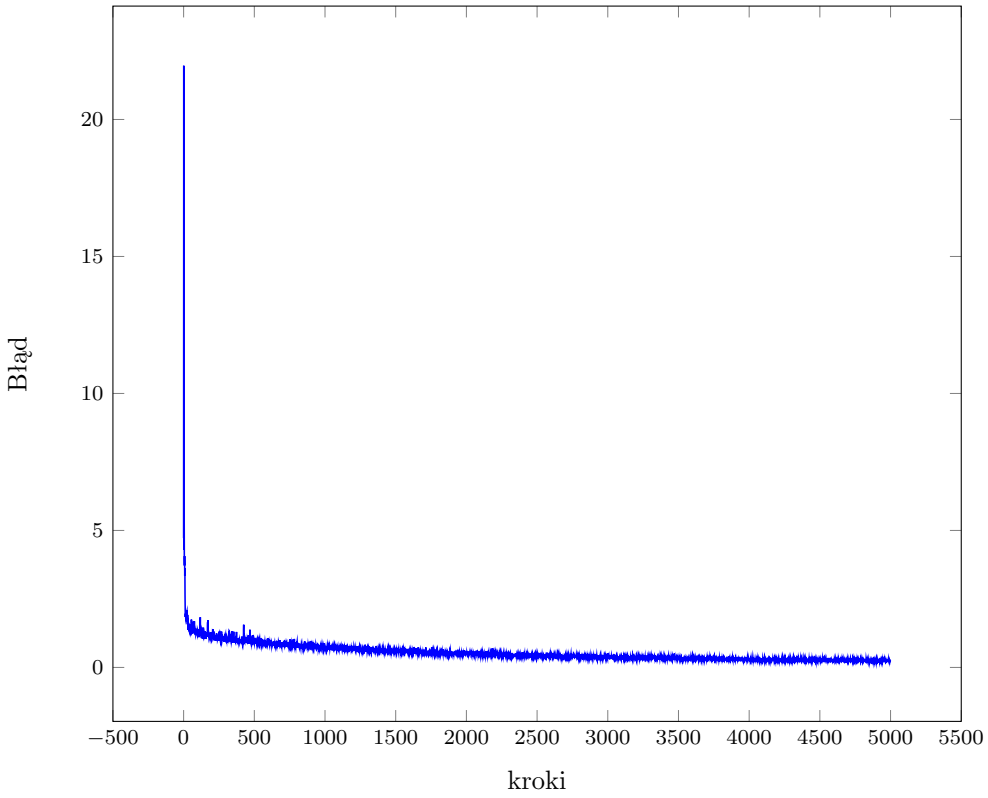
Testy w poprzednich sekcjach były przeprowadzane z użyciem algorytmu SGD (Stochastic Gradient Descent). Tabela 3.3 przedstawia wyniki dla algorytmu wykorzystującego pęd. Jak widać, dla niskich wartości parametru wyniki są praktycznie identyczne jak dla algorytmu SGD. Z kolei dla wyższych wartości parametru, wyniki okazują się być nieznacznie gorsze. Z uwagi na brak poprawy rezultatów mimo wprowadzenia dodatkowego parametru, kolejne testy będą przeprowadzane z wykorzystaniem algorytmu SGD.

Parametr pędu	Wyniki
0,001	Zbiór uczący: 98,4% Zbiór testowy: 98%
0,005	Zbiór uczący: 98,4% Zbiór testowy: 98,1%
0,01	Zbiór uczący: 99,2% Zbiór testowy: 98%
0,05	Zbiór uczący: 97,7% Zbiór testowy: 98%
0,1	Zbiór uczący: 96,9% Zbiór testowy: 98%
0,2	Zbiór uczący: 96,7% Zbiór testowy: 98%
0,3	Zbiór uczący: 97,7% Zbiór testowy: 97,7%
0,3	Zbiór uczący: 97,7% Zbiór testowy: 97,5%

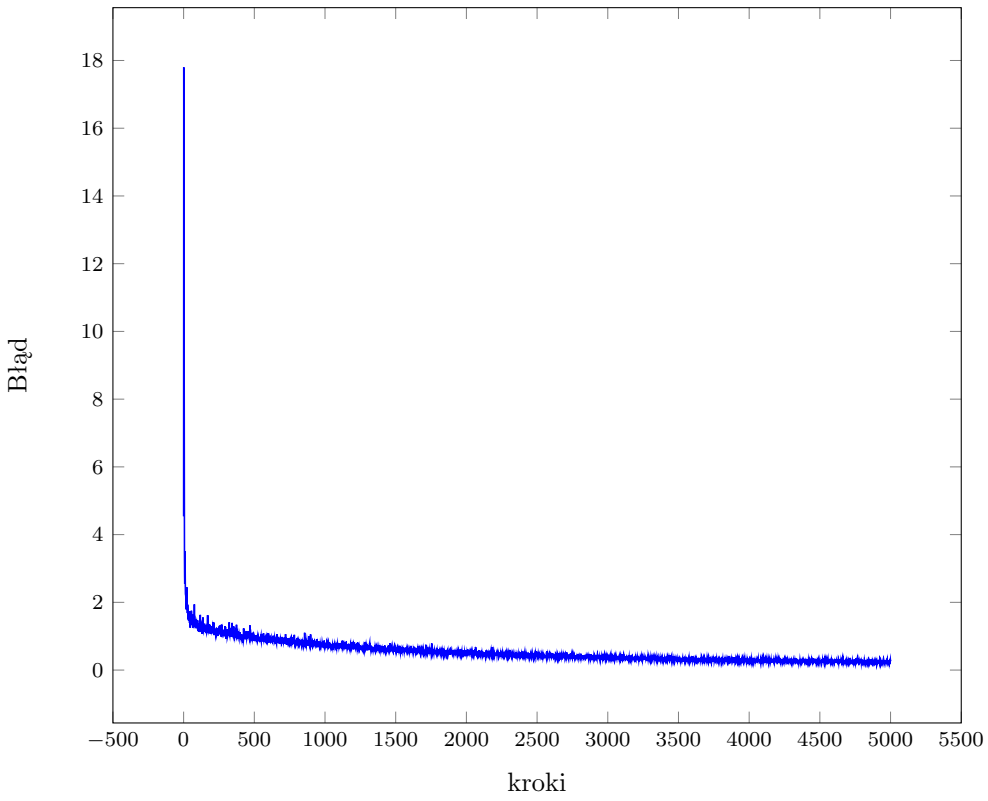
Tab. 3.3: Wyniki dla algorytmu wykorzystującego pęd



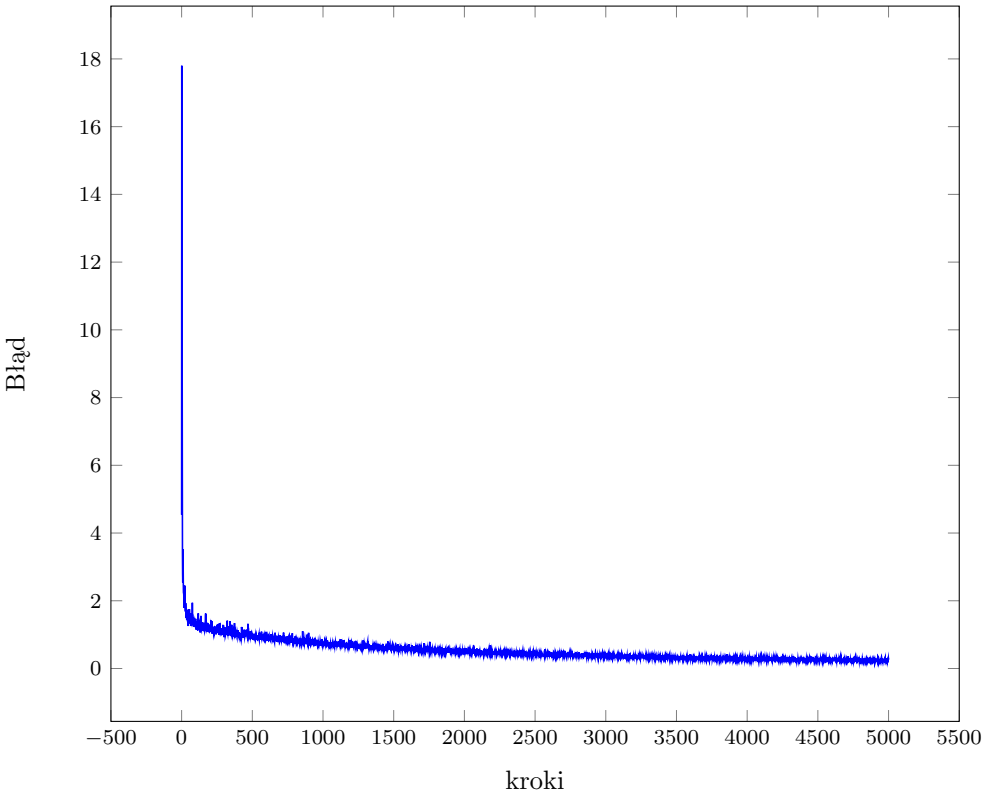
Rys. 3.19: Parametr pędu - 0,001



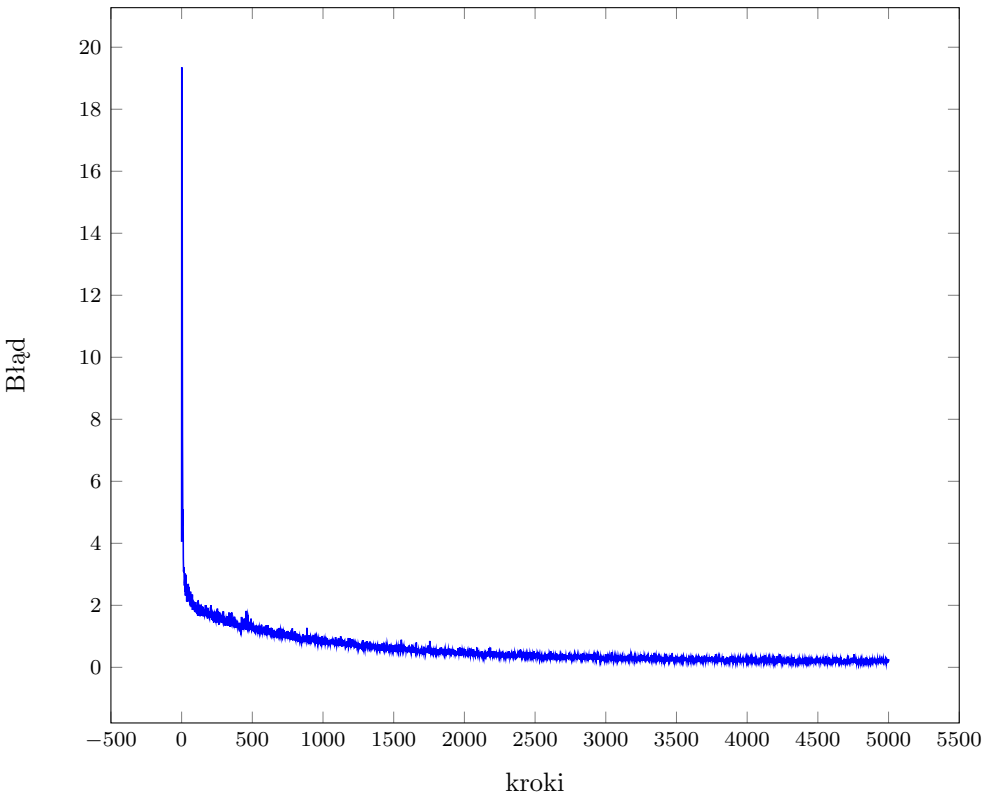
Rys. 3.20: Parametr pędu - 0,01



Rys. 3.21: Parametr pędu - 0,1



Rys. 3.22: Parametr pędu - 0,2



Rys. 3.23: Parametr pędu - 0,4

3.4. Szybkość uczenia

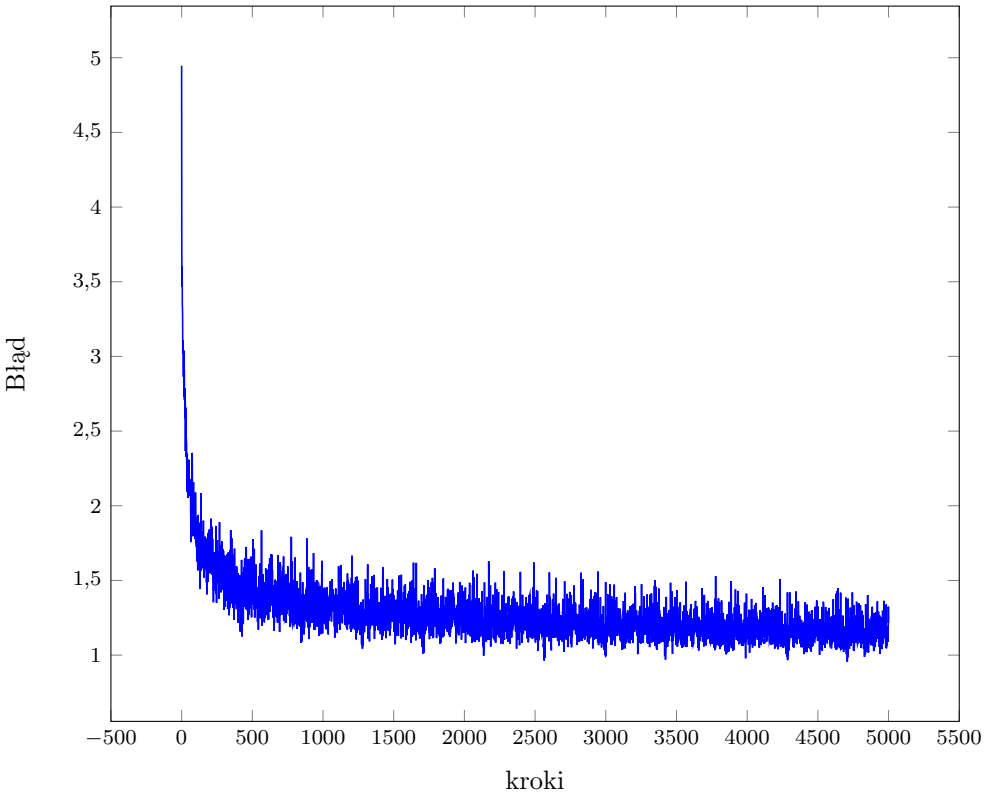
Tabela 3.4 przedstawia wyniki dla różnych początkowych kroków uczenia. Krok jest w każdej iteracji zmniejszany według wzoru 3.1, gdzie i to aktualna iteracja algorytmu.

$$\eta_i = \eta_{i-1} * 0,9^{\frac{i}{800}} \tag{3.1}$$

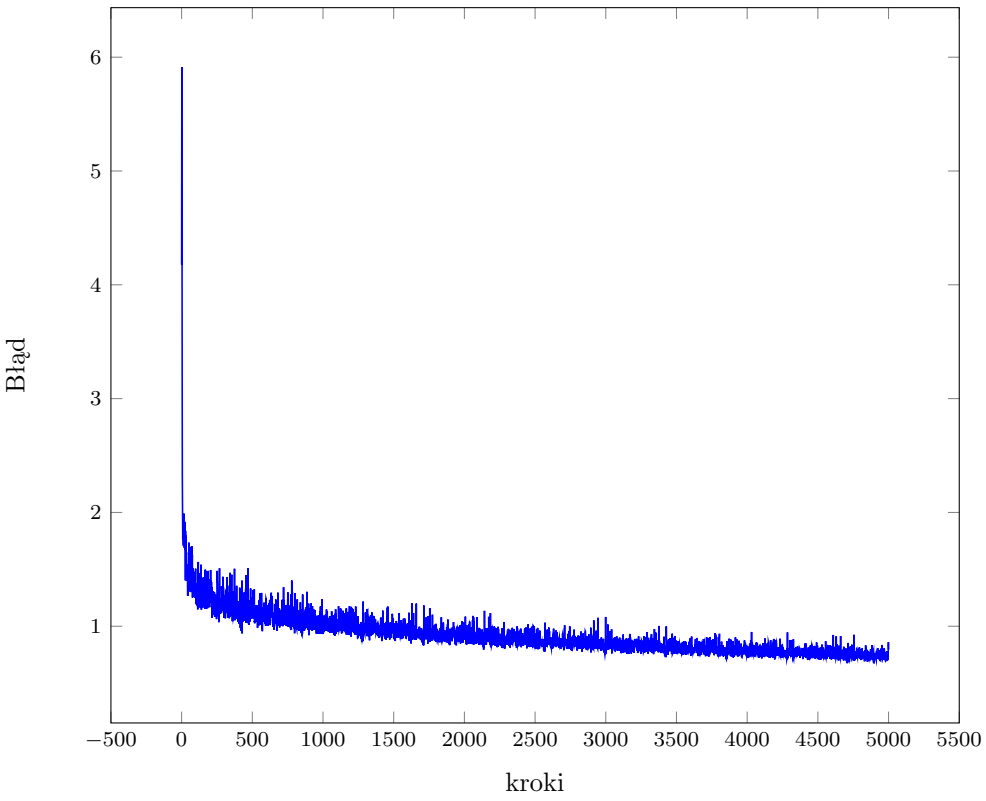
Testy przeprowadzanie w poprzednich testach miały początkową szybkość uczenia $\eta = 0,5$. Tabela 3.4 przedstawia wyniki dla innych wartości parametru. Jak można zauważyć, zmiana wartości kroku w przedziale od 0,2 do 0,6 nie zmieniła znacząco otrzymywanych wyników. Dla parametrów η spoza tego przedziału następuje pogorszenie jakości klasyfikacji.

Krok uczenia	Wyniki
0,01	Zbiór uczący: 90,6% Zbiór testowy: 94,2%
0,1	Zbiór uczący: 96,9% Zbiór testowy: 97,5%
0,2	Zbiór uczący: 97,7% Zbiór testowy: 97,9%
0,3	Zbiór uczący: 99,2% Zbiór testowy: 98%
0,4	Zbiór uczący: 98,4% Zbiór testowy: 98%
0,6	Zbiór uczący: 95,3% Zbiór testowy: 97,9%
0,7	Zbiór uczący: 96,9% Zbiór testowy: 97,4%

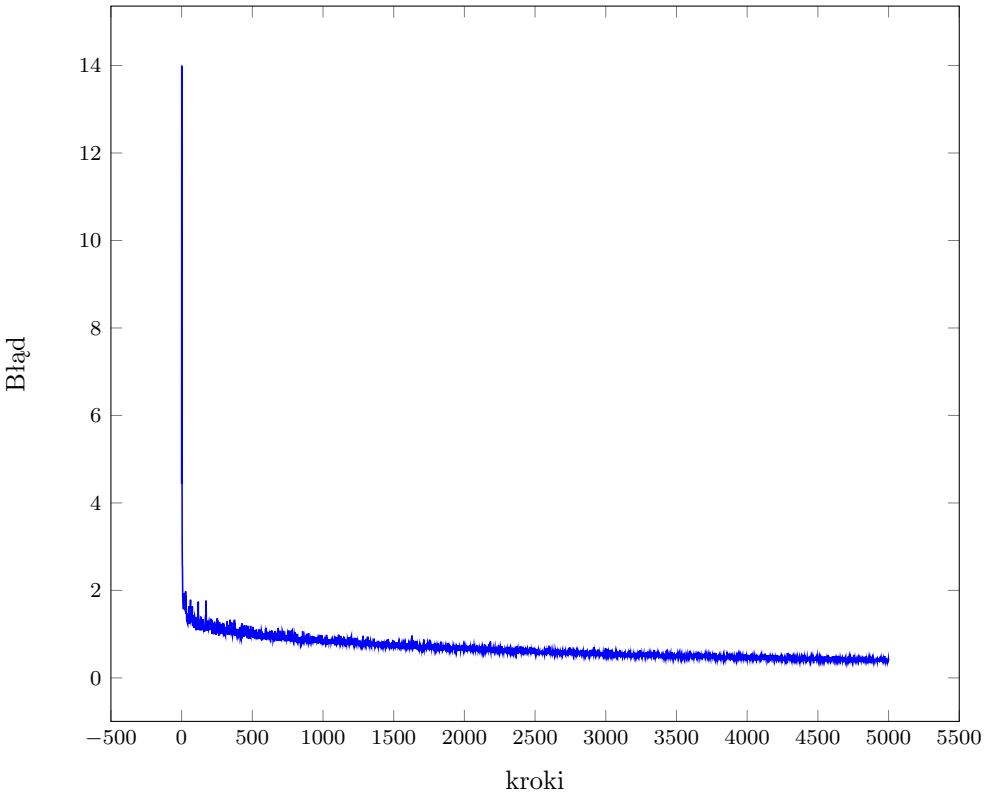
Tab. 3.4: Wyniki dla różnych wartości parametru η



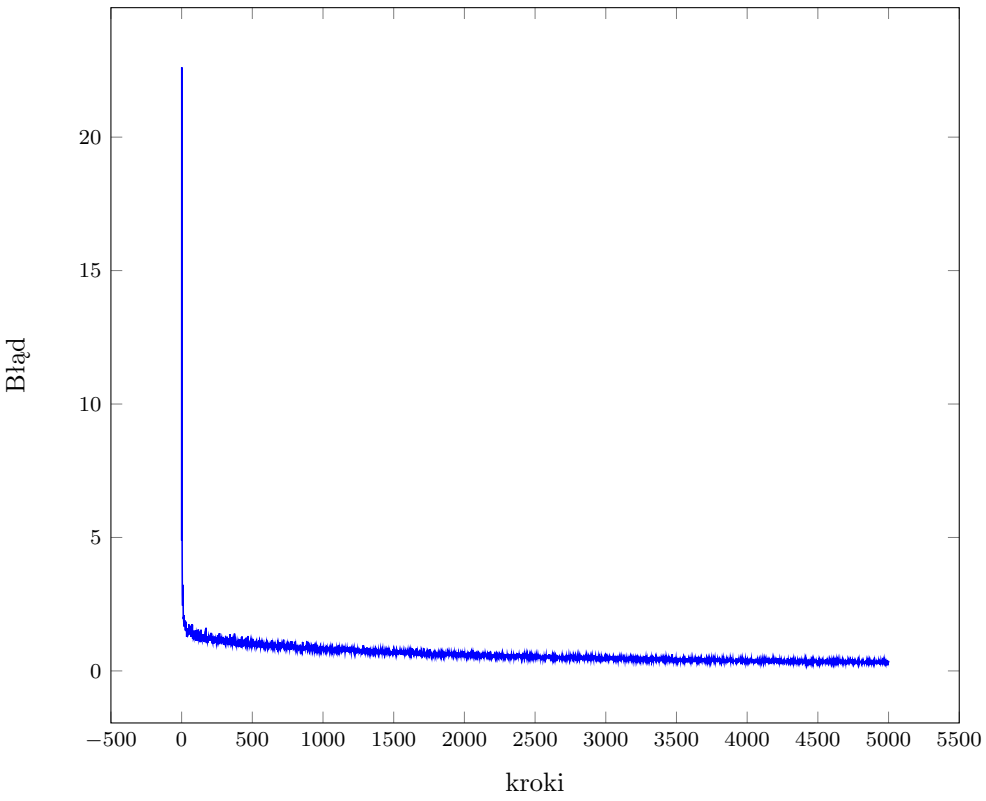
Rys. 3.24: $\eta = 0{,}01$



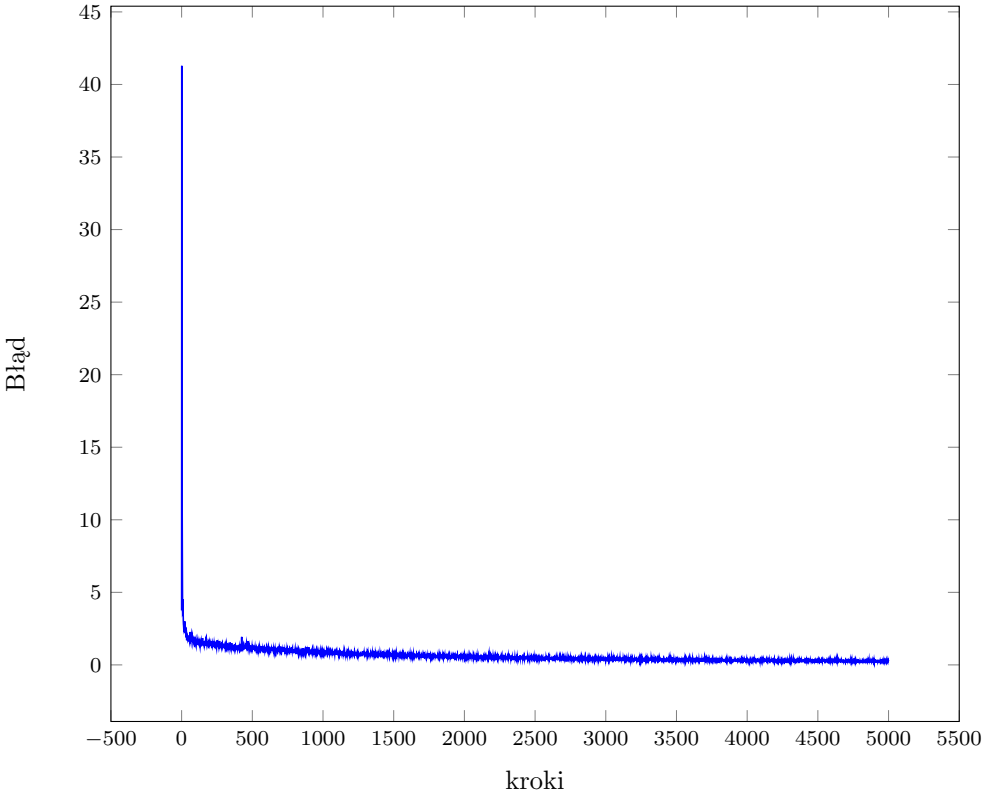
Rys. 3.25: $\eta = 0{,}1$



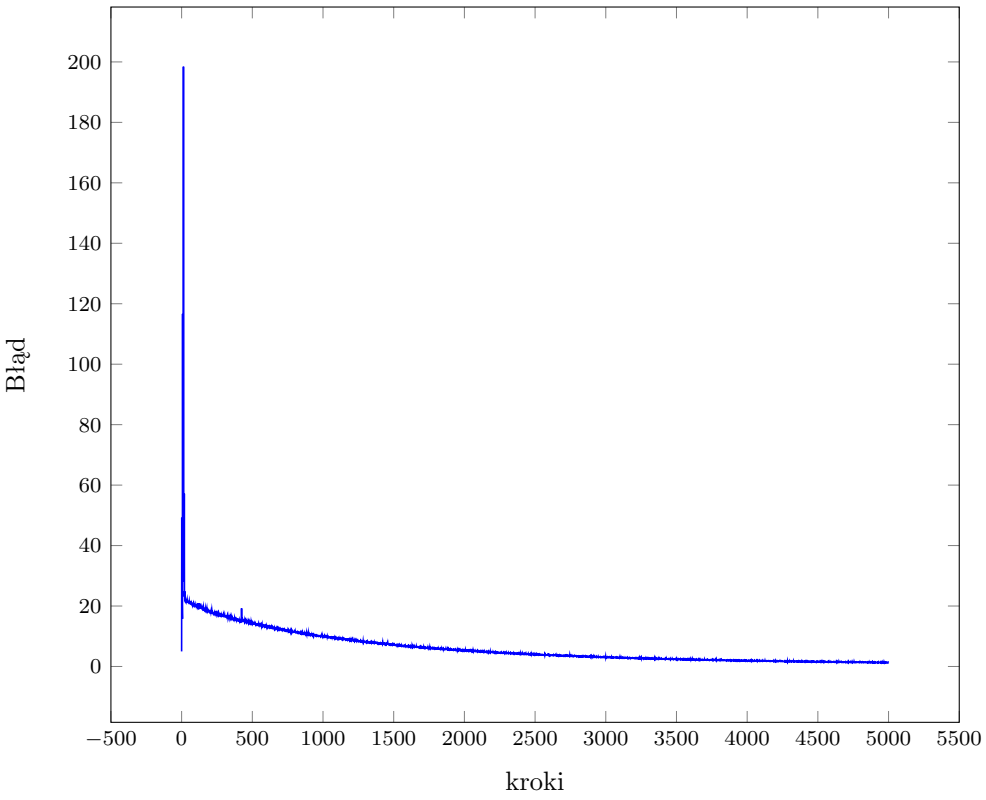
Rys. 3.26: $\eta = 0{,}3$



Rys. 3.27: $\eta = 0{,}4$



Rys. 3.28: $\eta = 0{,}6$



Rys. 3.29: $\eta = 0{,}7$

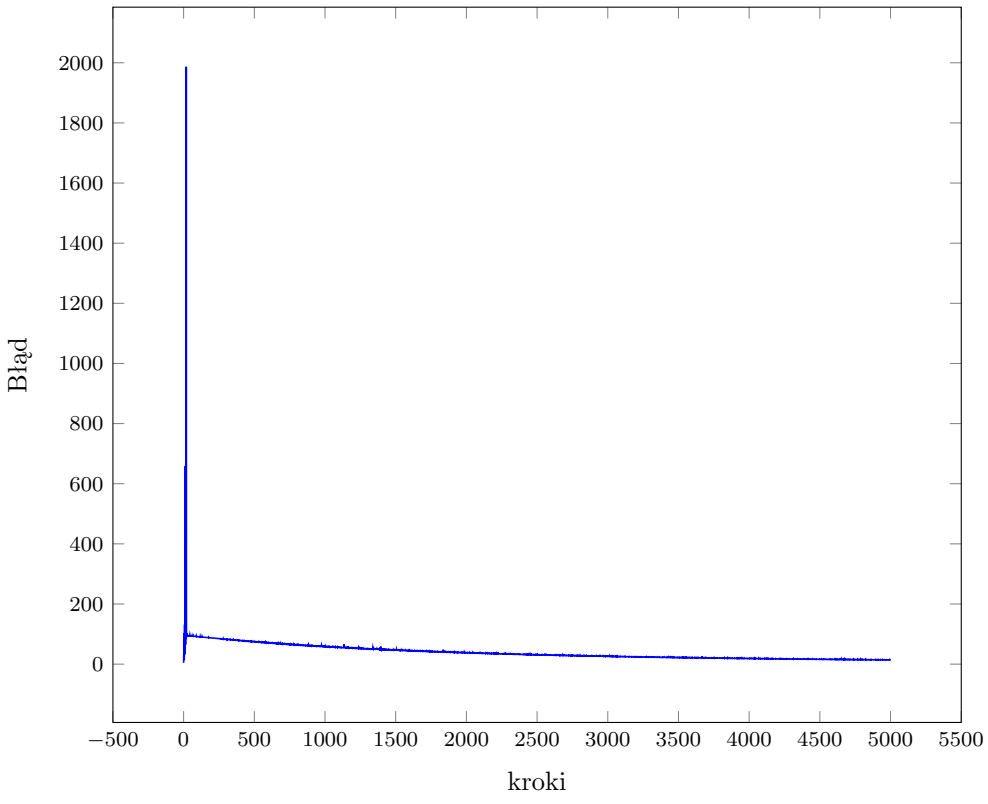
3.5. Batch size

Poprzednie eksperymenty były przeprowadzane dla batch size (wielkości serii) równej 128. Tabela 3.5 przedstawia wyniki dla innych wartości tego parametru. Jak widać, mała wielkość serii powoduje spadek jakości klasyfikacji, co prawdopodobnie spowodowane jest tym, że sieć dopasowuje się do danych uczących.

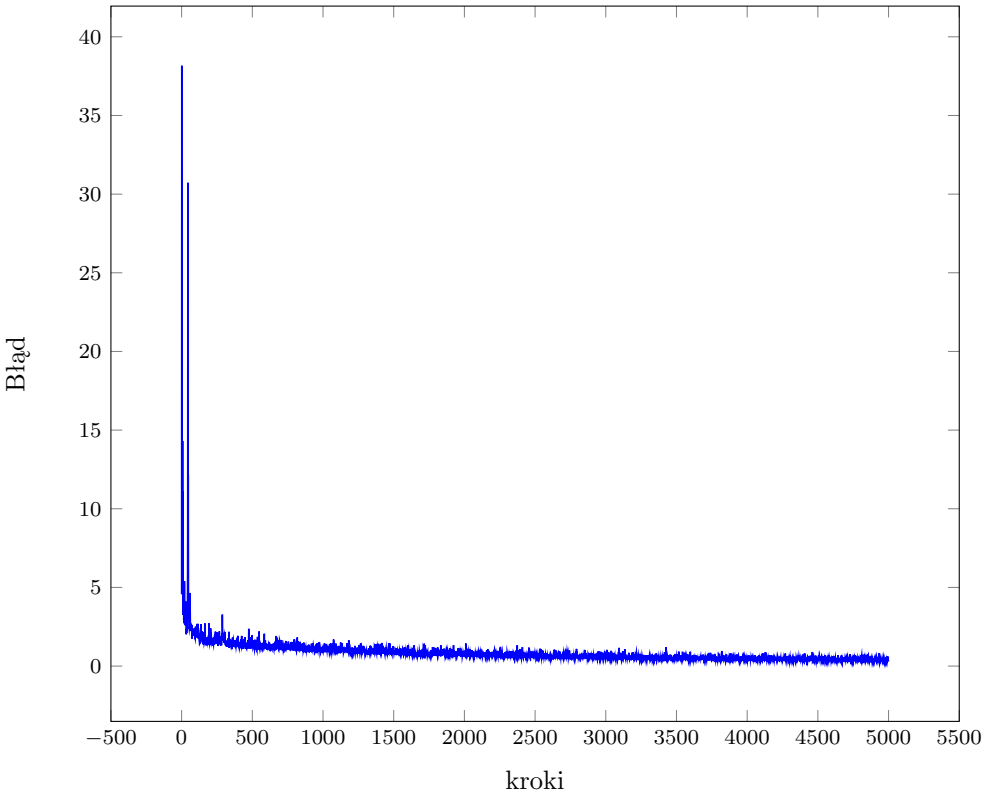
Duży batch size polepsza działanie sieci neuronowej, ale sprawia, że nauka trwa kilkukrotnie dłużej. Wielkości 256 lub 512 stanowią dobry kompromis między szybkością nauki a dobrymi wynikami.

Batch size	Wyniki
16	Zbiór uczący: 100% Zbiór testowy: 92,9%
32	Zbiór uczący: 100% Zbiór testowy: 96,8%
64	Zbiór uczący: 92,2% Zbiór testowy: 97,5%
256	Zbiór uczący: 99,6% Zbiór testowy: 98%
512	Zbiór uczący: 98,6% Zbiór testowy: 98,3%
1024	Zbiór uczący: 98,1% Zbiór testowy: 98,2%

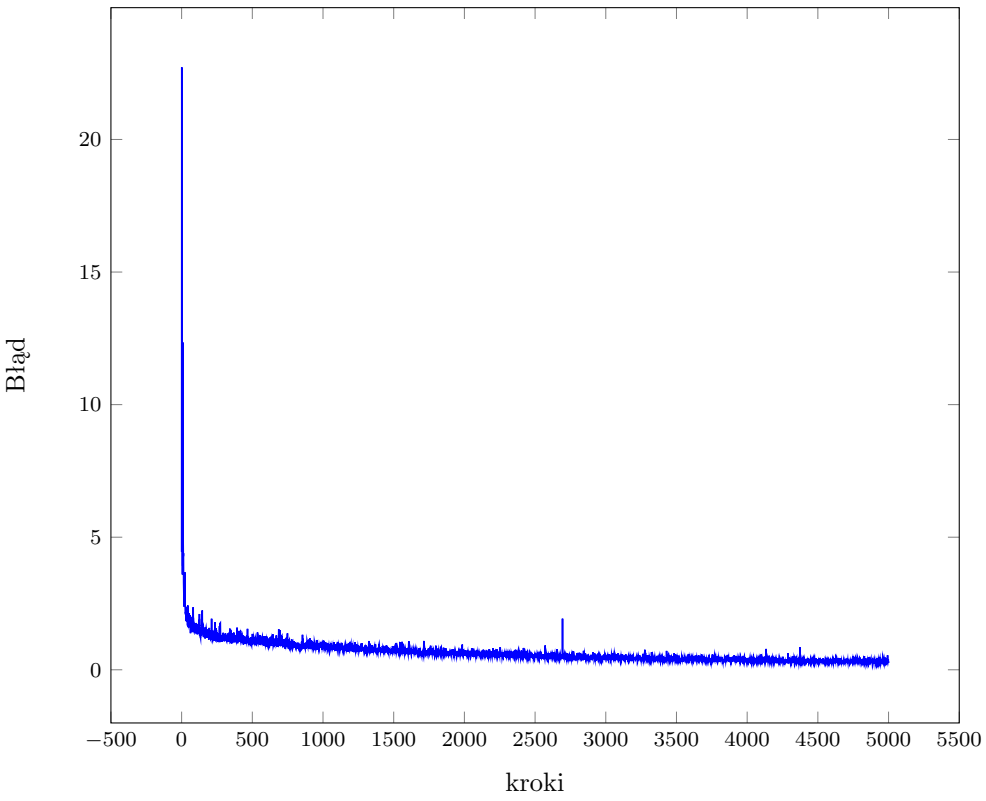
Tab. 3.5: Wyniki dla różnych wielkości serii



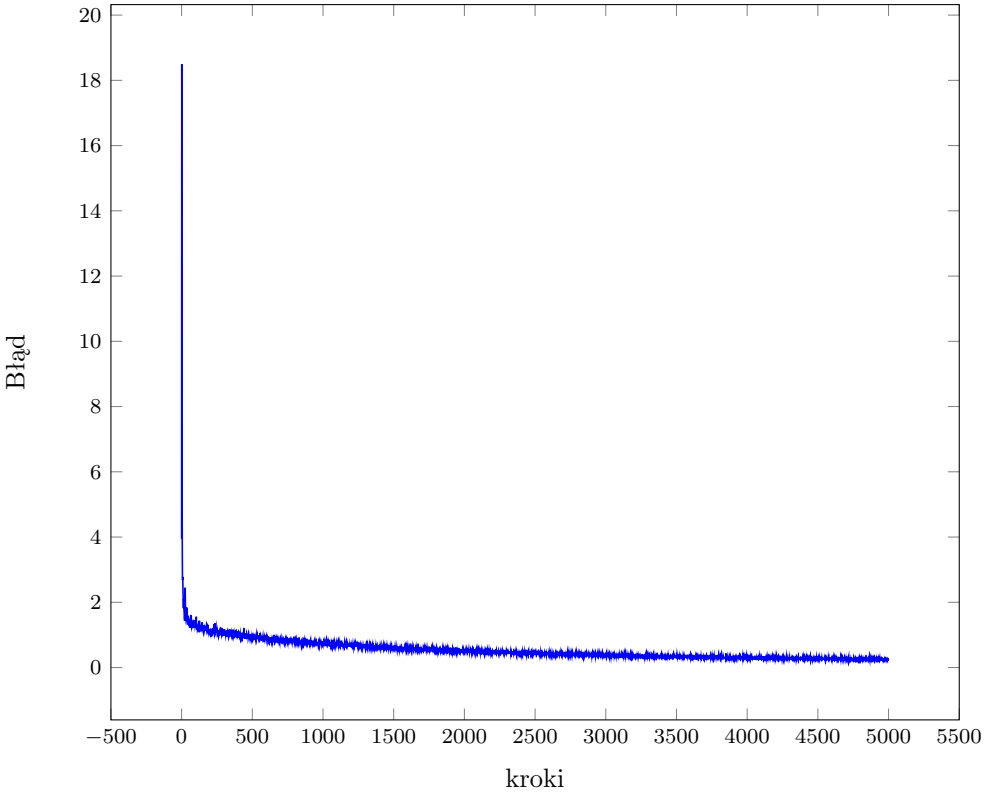
Rys. 3.30: Batch size = 16



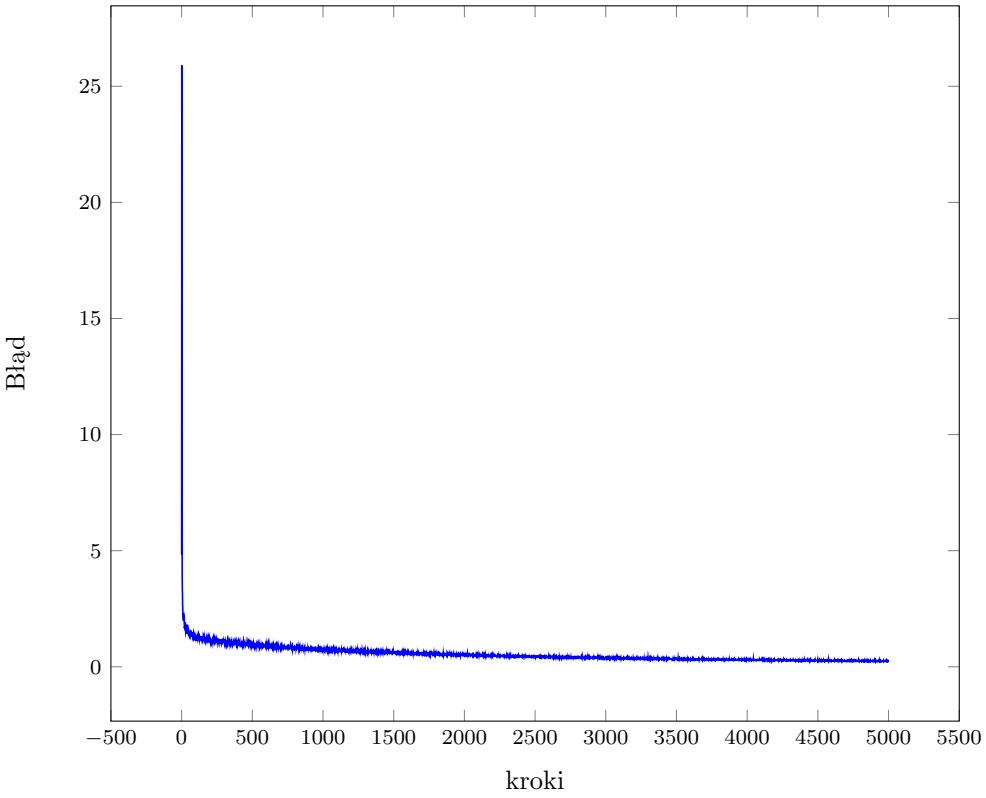
Rys. 3.31: Batch size = 32



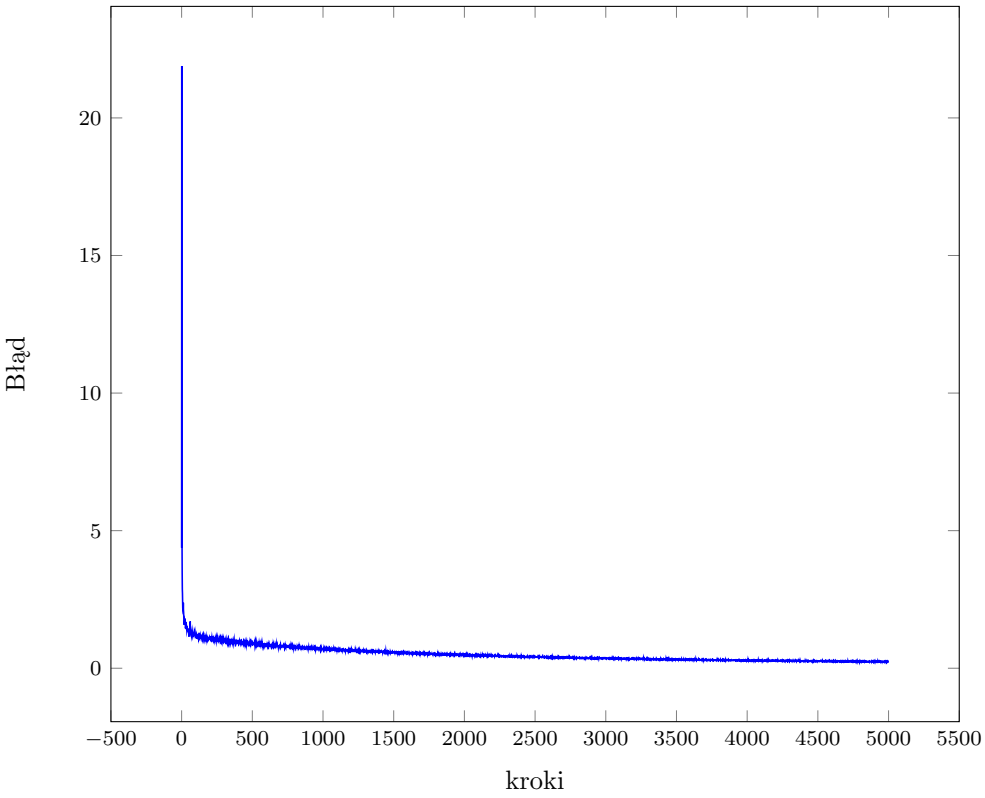
Rys. 3.32: Batch size = 64



Rys. 3.33: Batch size = 256



Rys. 3.34: Batch size = 512



Rys. 3.35: Batch size = 1024

4. Wnioski końcowe

Najlepszym wynikiem dla danych testowych, który udało nam się osiągnąć, było 98,3% poprawnych odpowiedzi. Parametry sieci, która osiągnęła ten wynik, są następujące:
— struktura sieci - 2 warstwy ukryte po 400 neuronów
— algorytm uczenia - Stochastic Gradient Descent
— szybkość uczenia - 0,5
— funkcja aktywacji - ReLU
— batch size - 512