

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Sieci neuronowe w zastosowaniach biomedycznych

Rozpoznawanie cyfr pisanych ręcznie ze zbioru  
MNIST

Sprawozdanie

Kamil Gabryjelski, Antoni Róžański

Prowadzący: mgr inż. Piotr Płoński

Warszawa, 2017

# 1. Dane

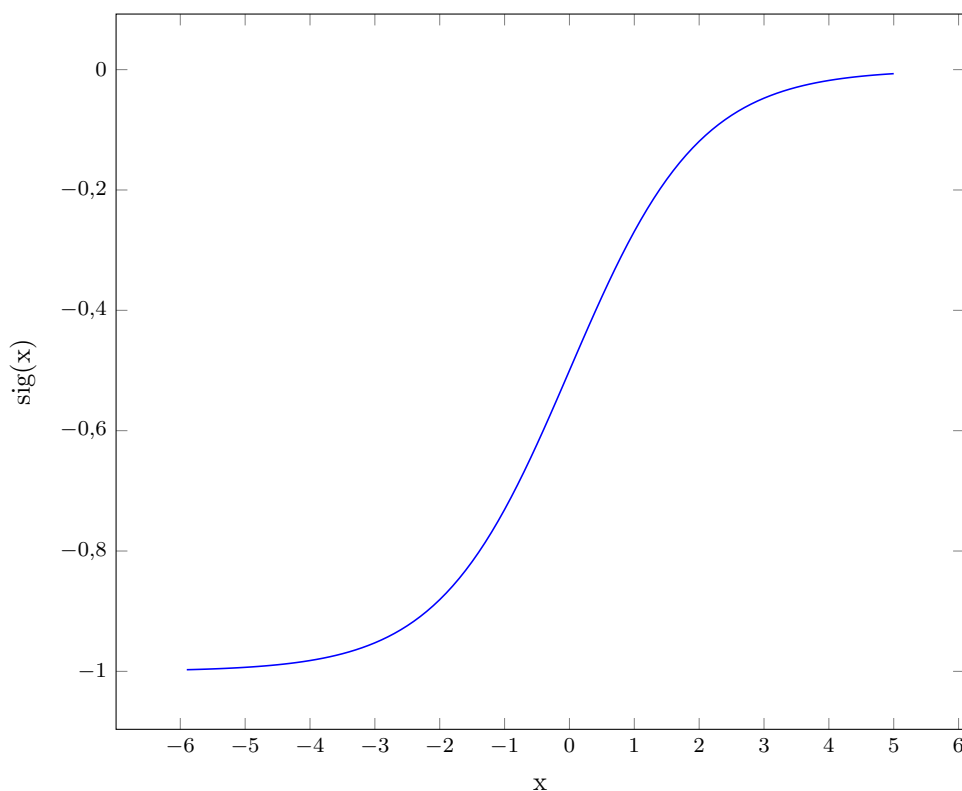
Mieliśmy do dyspozycji 70000 próbek cyfr pisanych ręcznie, każda składająca się z 784 pikseli.. Dane podzielone zostały na:

- zbiór uczący - 55000 próbek
- zbiór testowy - 10000 próbek
- zbiór walidacyjny - 5000 próbek

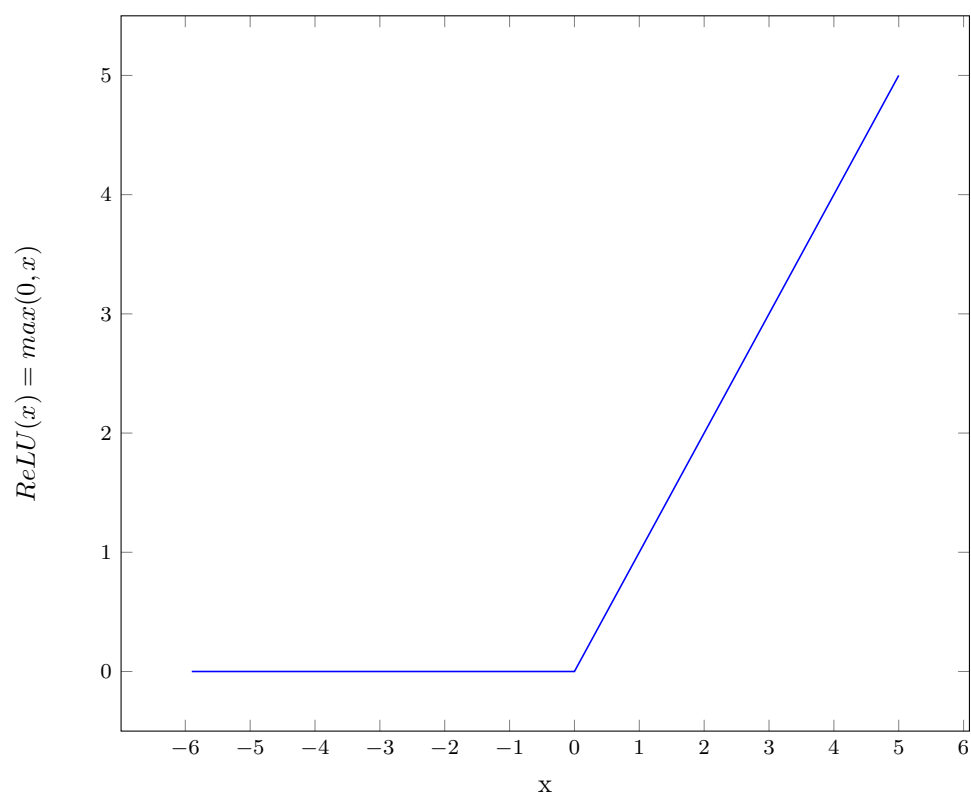
## 2. Implementacja sieci

### 2.1. Funkcje aktywacji

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



**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 ??, dzięki której wyniki możemy interpretować jako prawdopodobieństwa.

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

## 2.3. Błąd sieci

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

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

## 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,

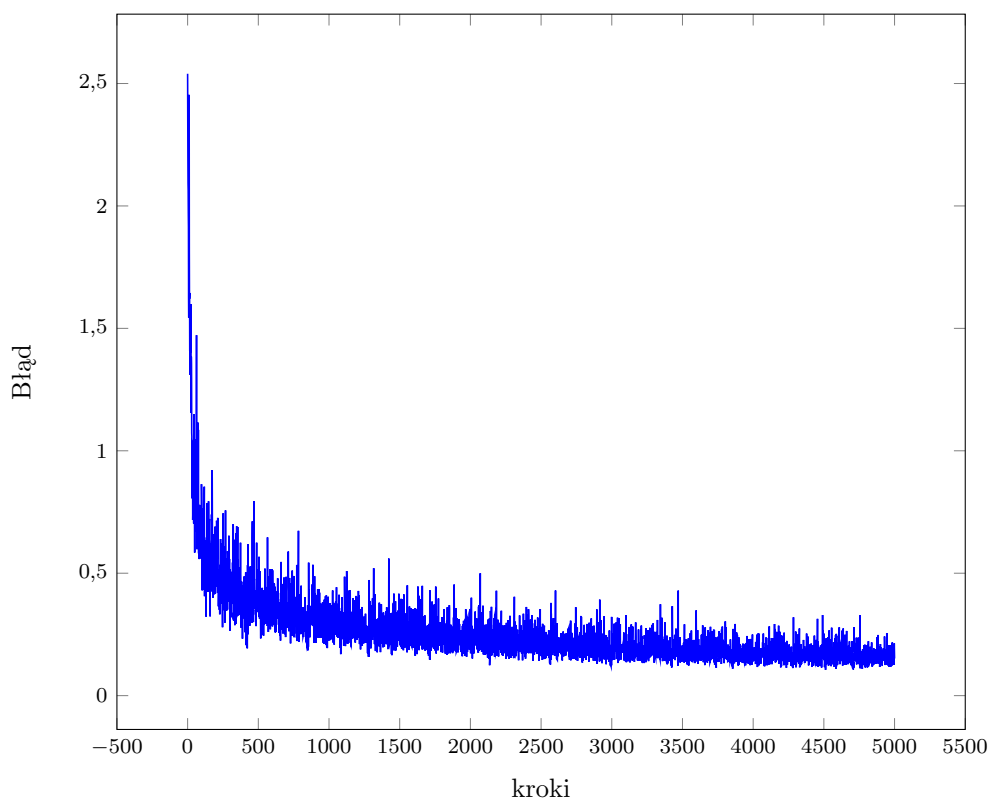
### 3. Eksperymenty

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

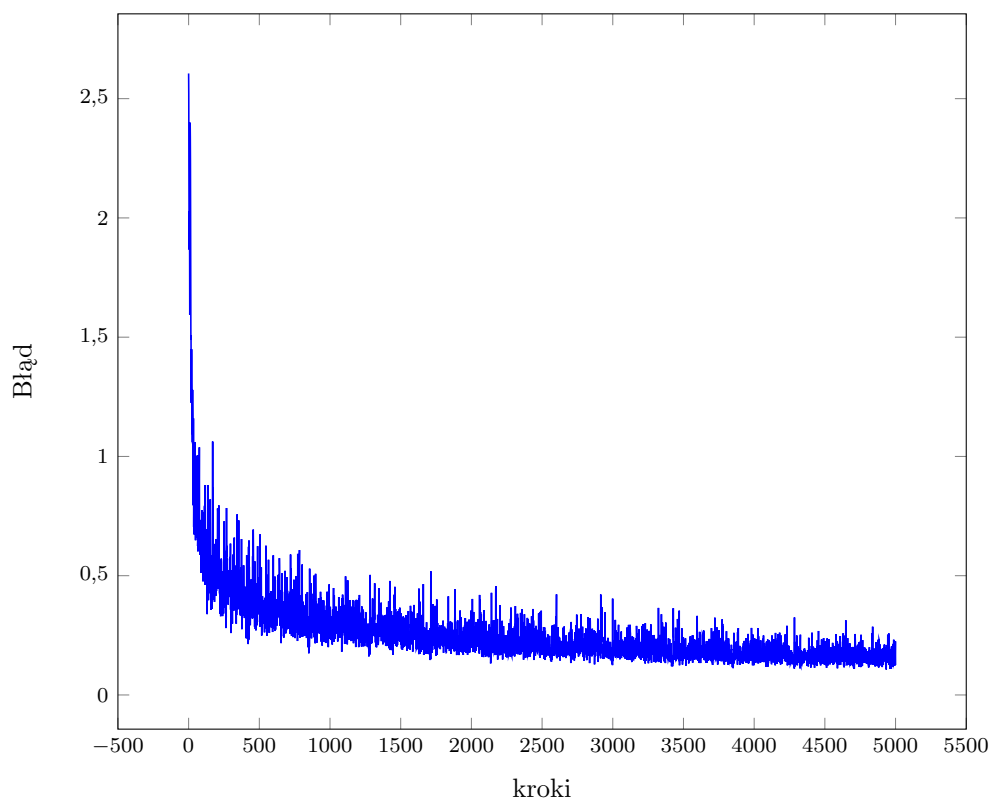
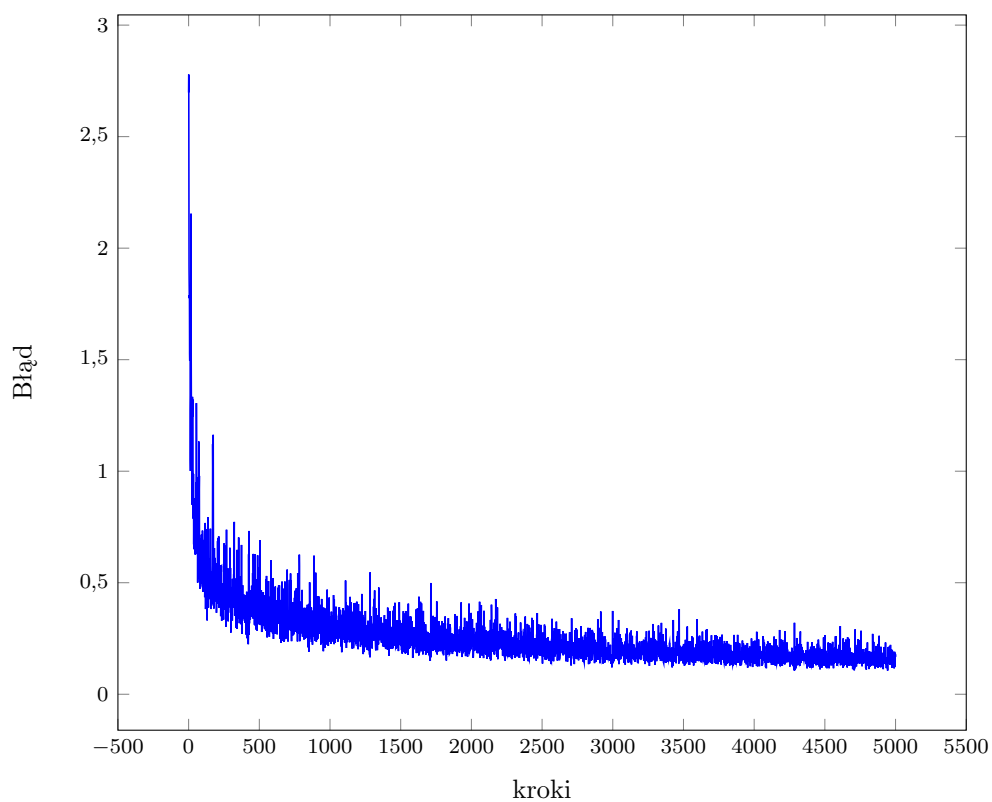
Ze względu na niedeterministyczny charakter sieci neuronowych, każdy zestaw parametrów był testowany kilkakrotnie i wybierany był najlepszy z uzyskanych wyników.

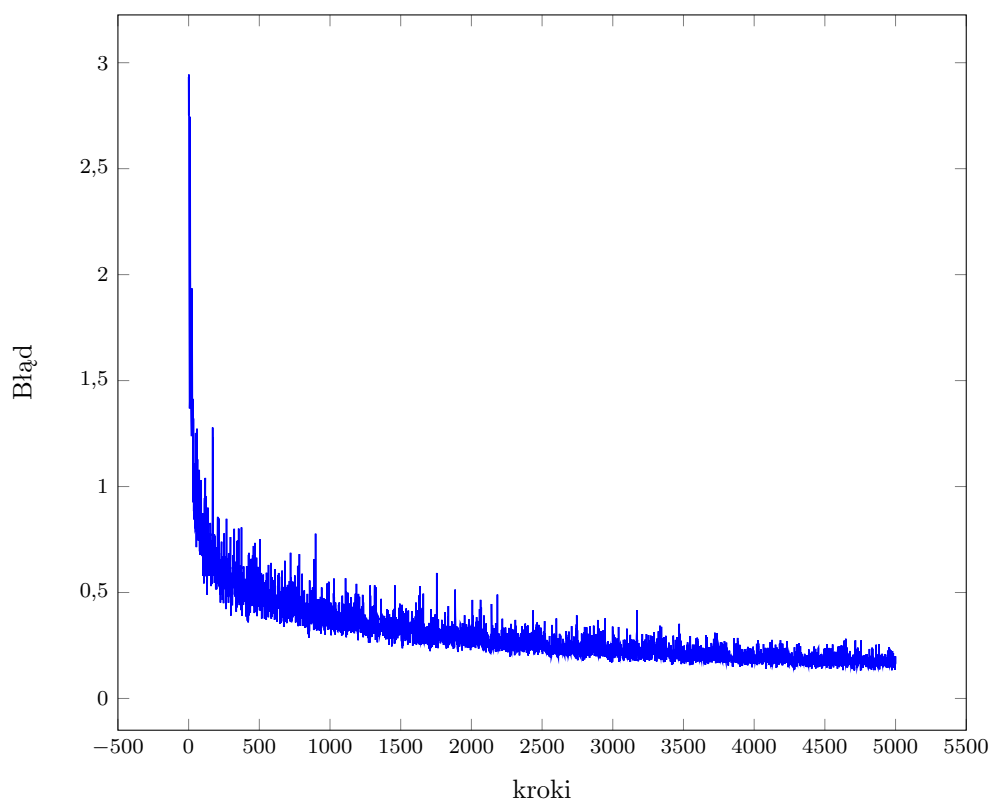
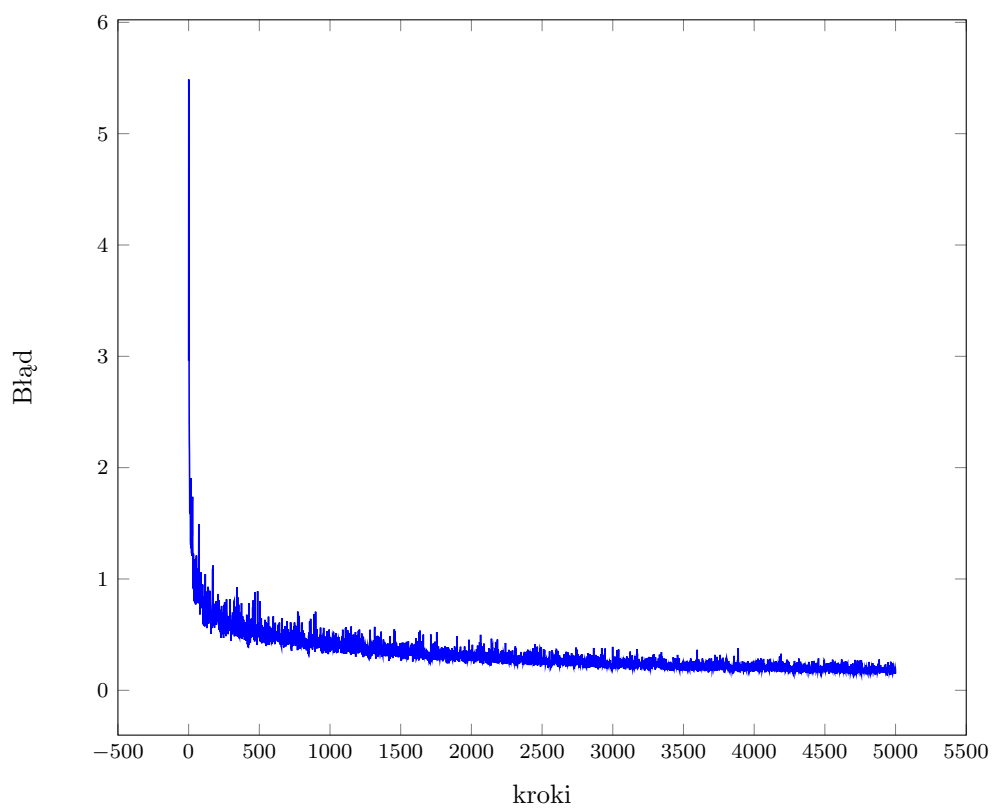
#### 3.1. Struktura sieci

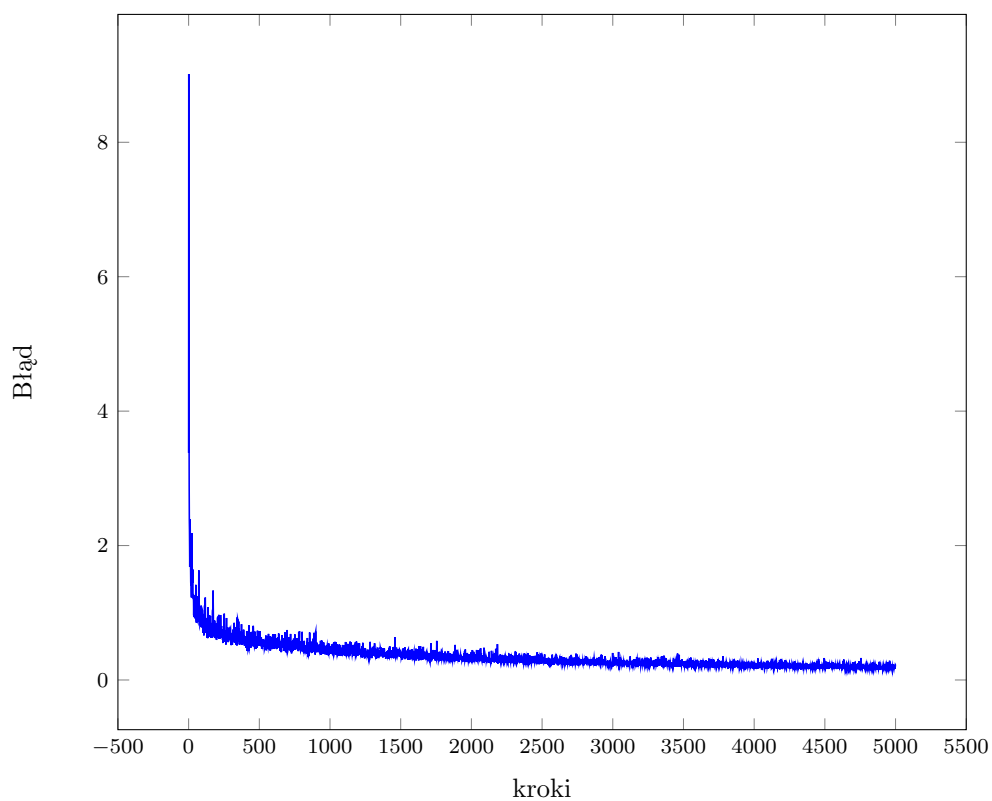
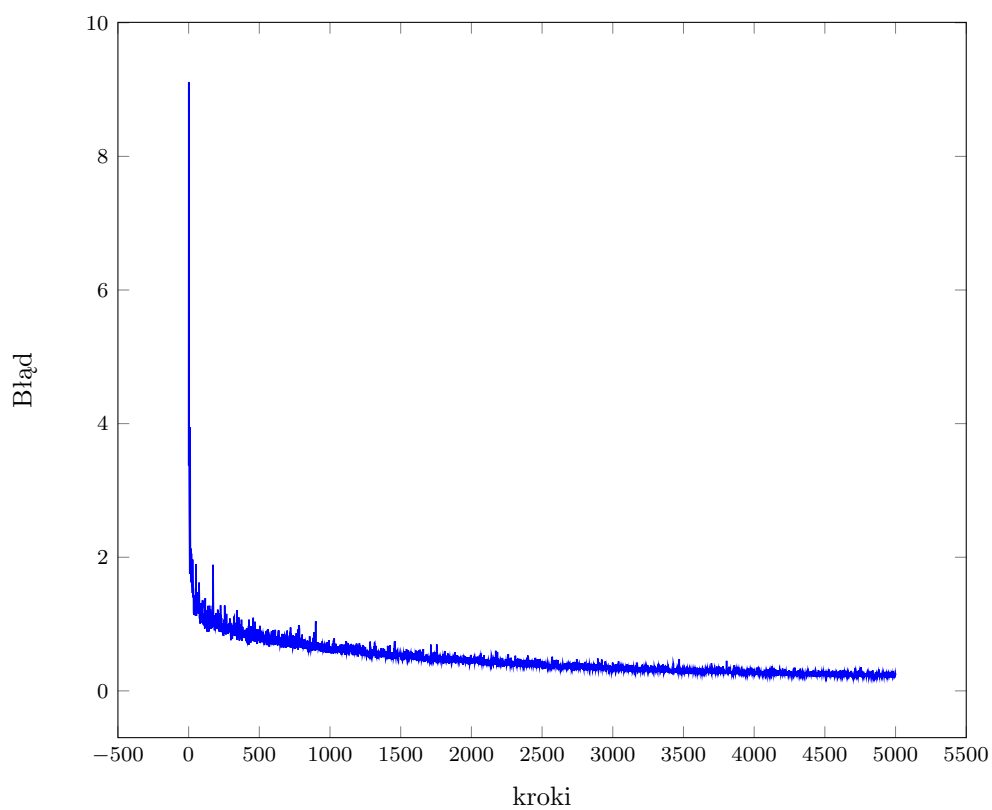
Wyniki przedstawione w tabeli ?? pozwalają stwierdzić, że różnice w jakości rozpoznawania cyfry przez sieć w zależności od jej struktury są niewielkie. Sieć prawidłowo rozpoznaje cyfrę w około 98% przypadków. Analizując wykresy błędów można jednak zauważyć, że im więcej neuronów w sieci, tym mniejsza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



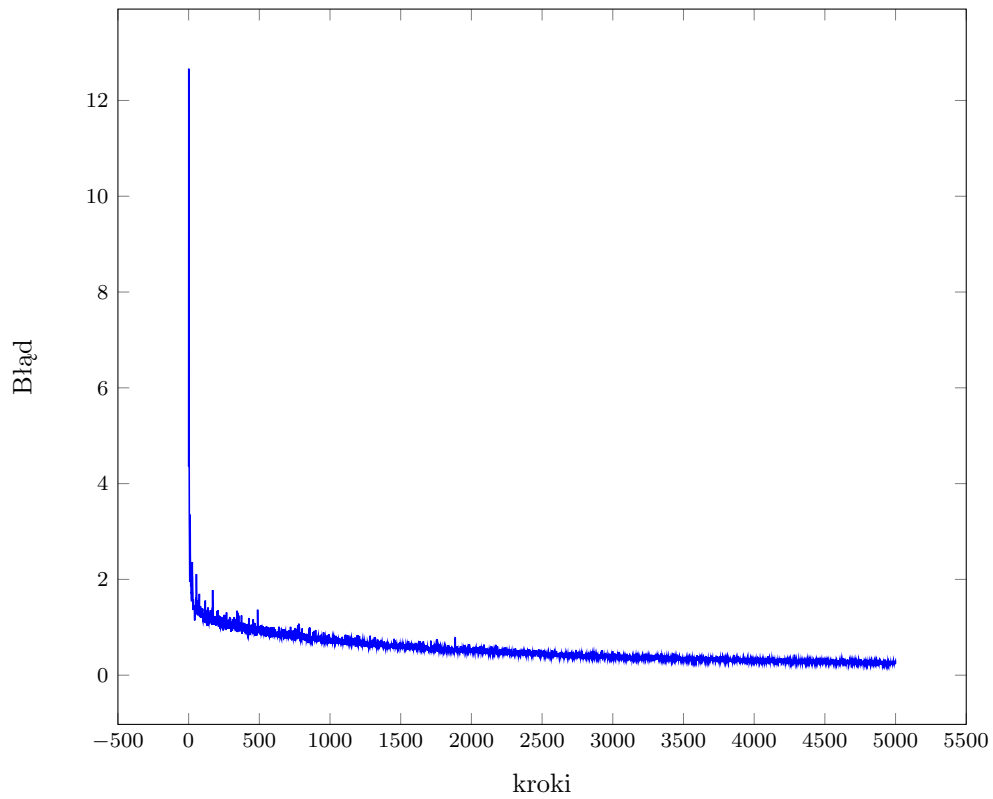
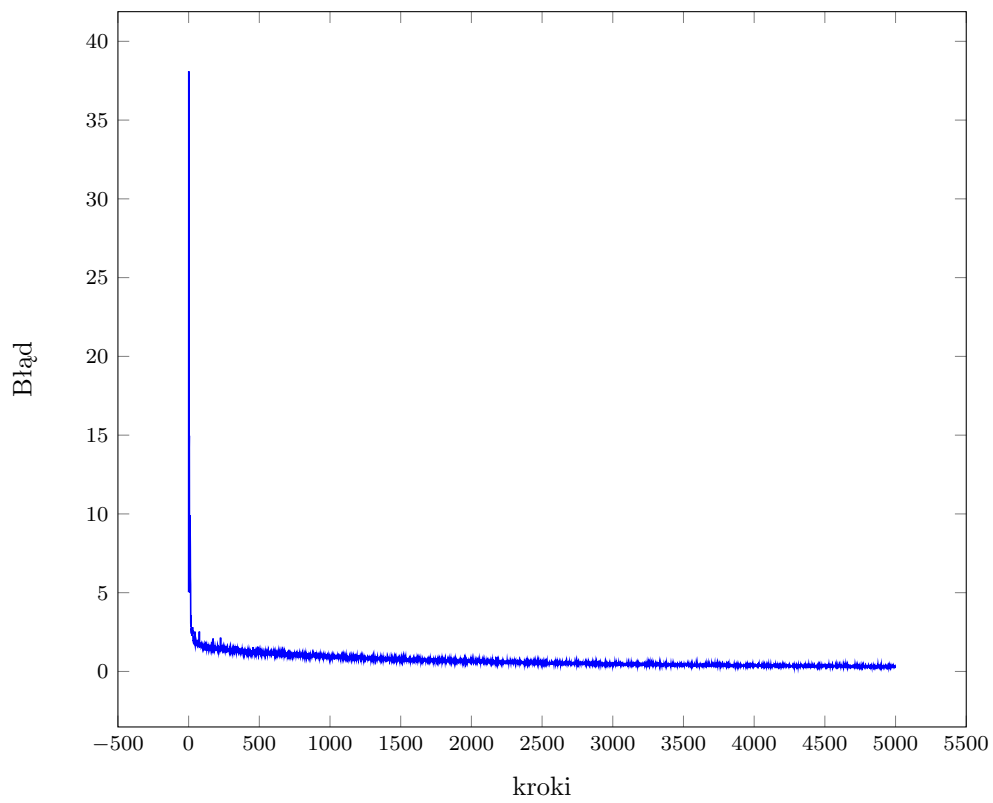
**Rys. 3.1:** 2 warstwy ukryte, (100, 50) neuronów

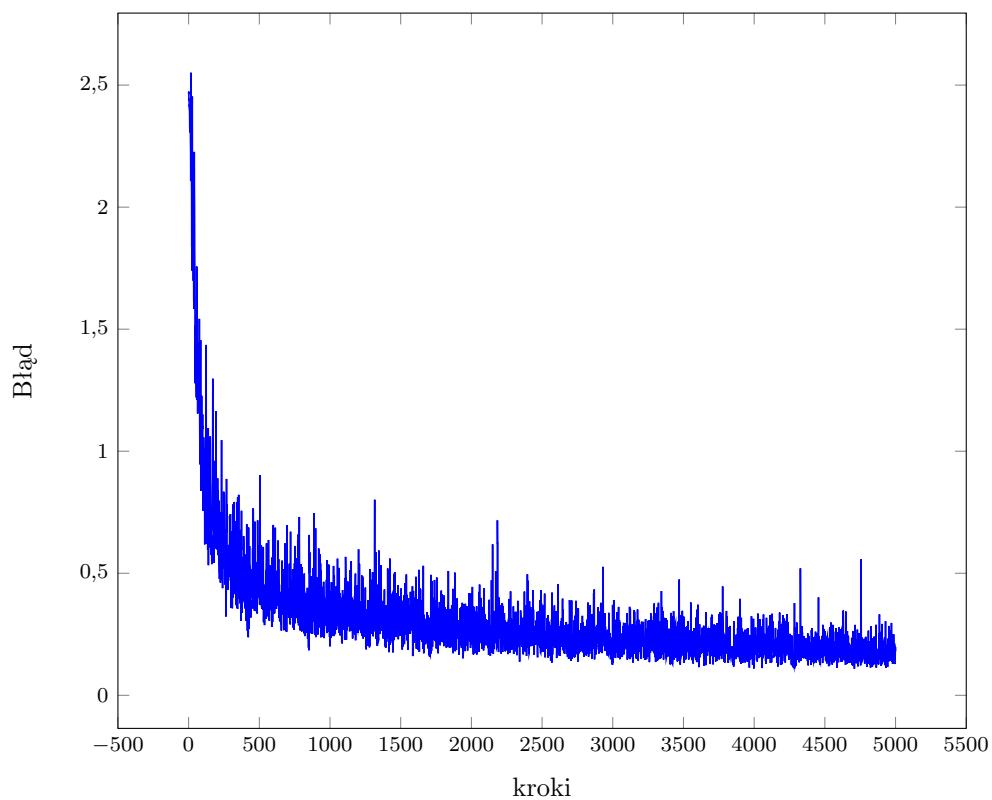
**Rys. 3.2:** 2 warstwy ukryte, (100, 100) neuronów**Rys. 3.3:** 2 warstwy ukryte, (100, 150) neuronów

**Rys. 3.4:** 2 warstwy ukryte, (200, 100) neuronów**Rys. 3.5:** 2 warstwy ukryte, (200, 200) neuronów

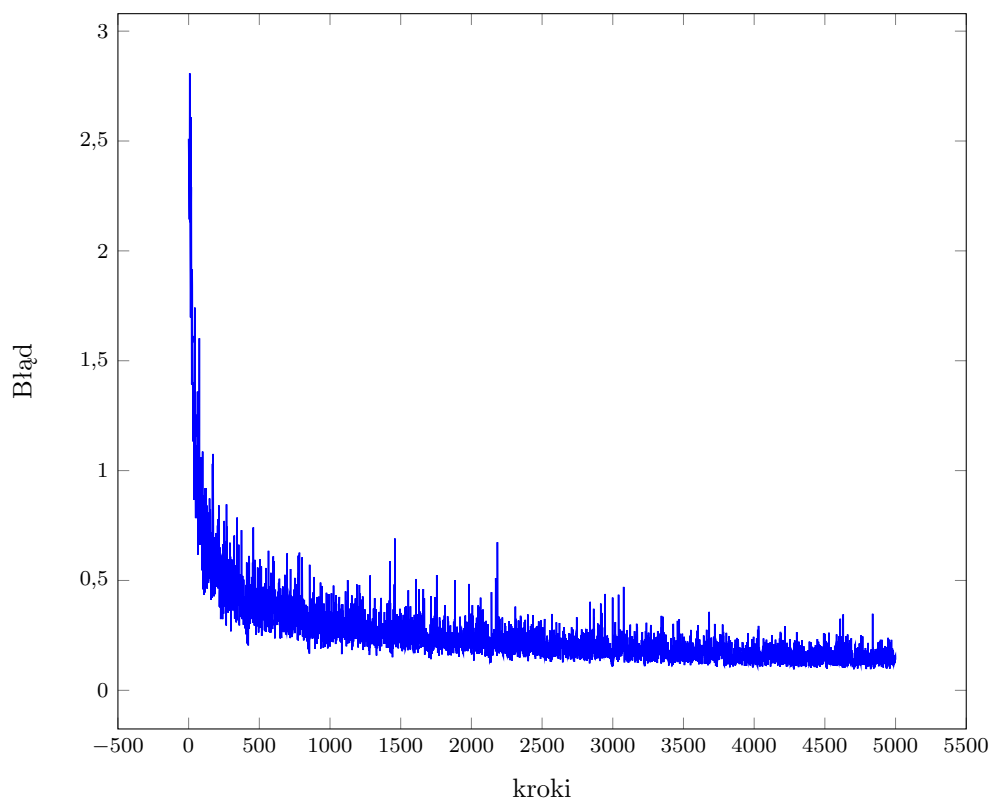
**Rys. 3.6:** 2 warstwy ukryte, (200, 300) neuronów**Rys. 3.7:** 2 warstwy ukryte, (400, 200) neuronów



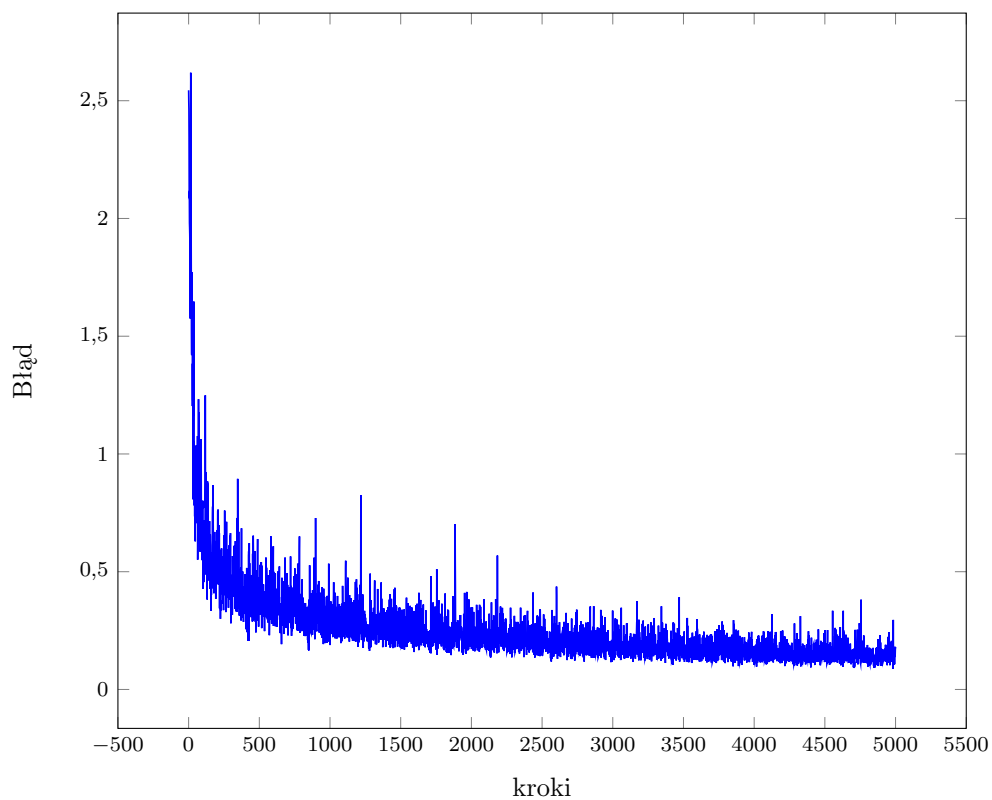
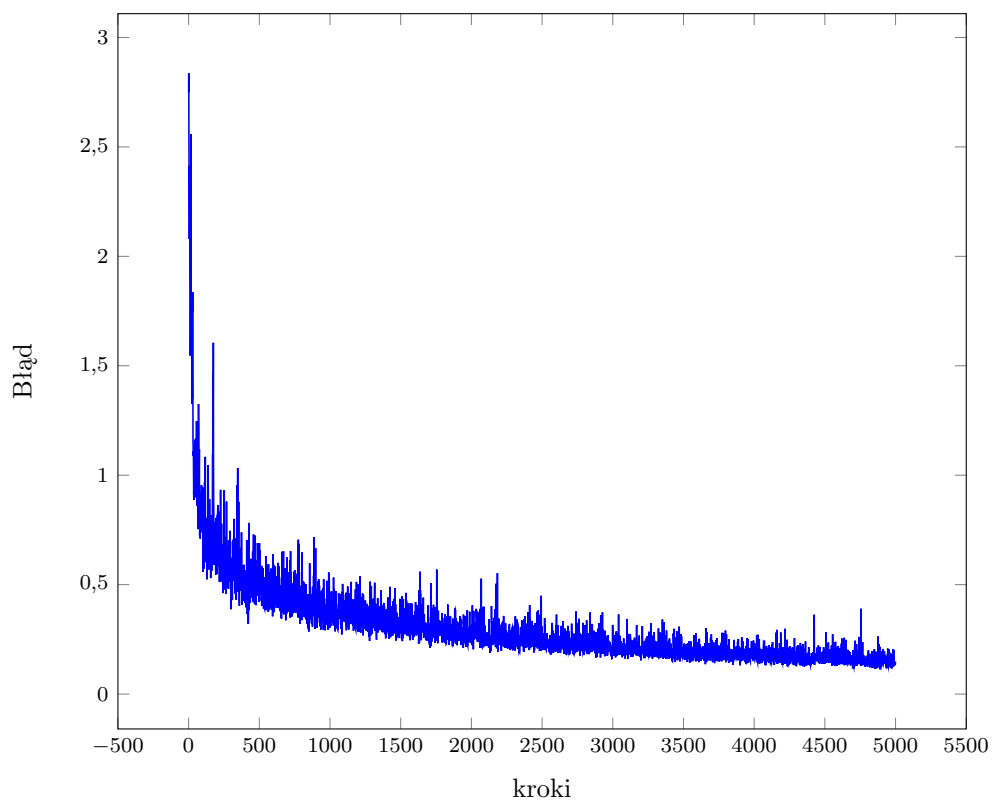
**Rys. 3.8:** 2 warstwy ukryte, (400, 400) neuronów**Rys. 3.9:** 2 warstwy ukryte, (400, 600) neuronów

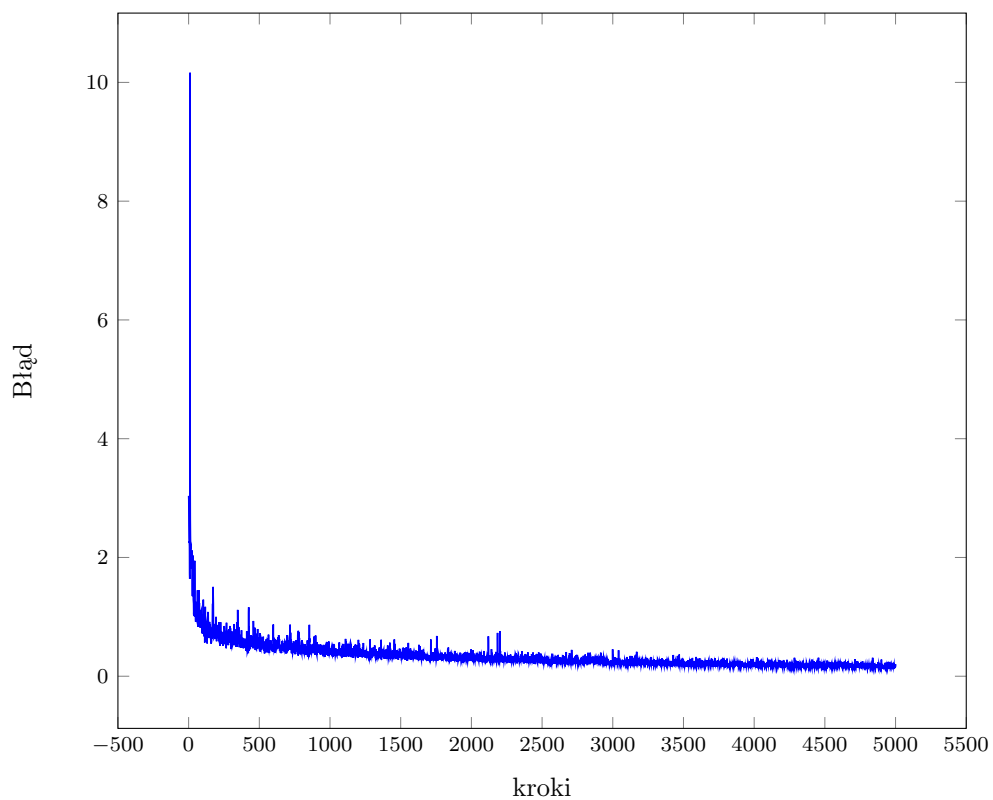
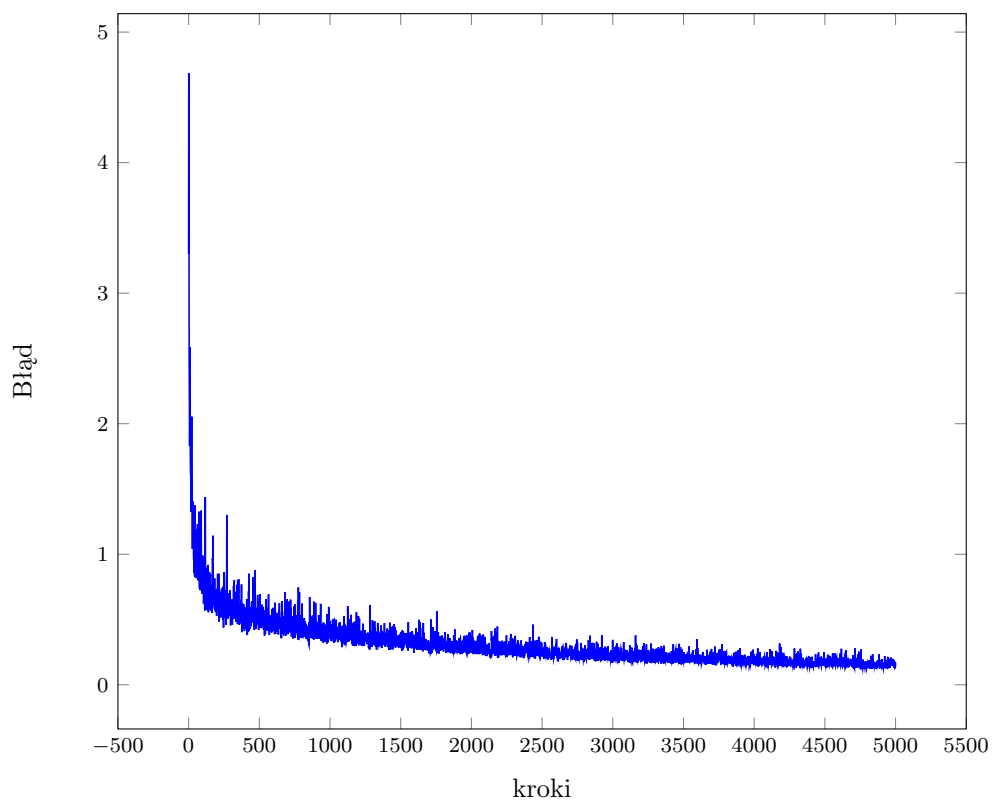


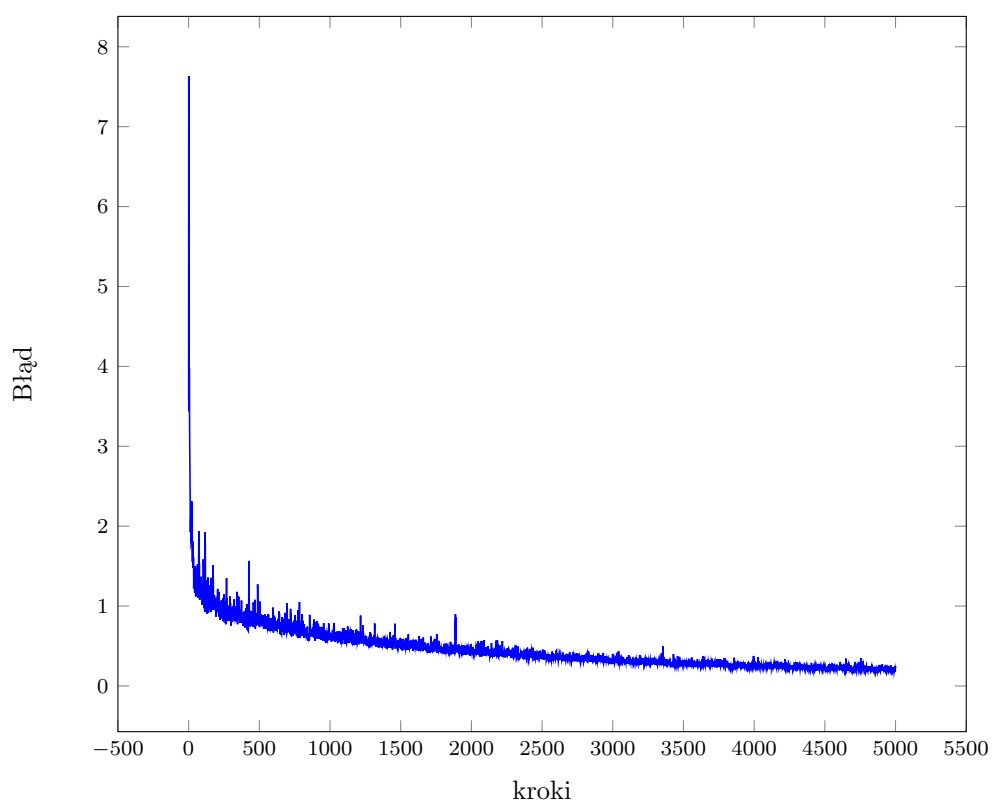
**Rys. 3.10:** 3 warstwy ukryte, (100, 50, 10) neuronów



**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%

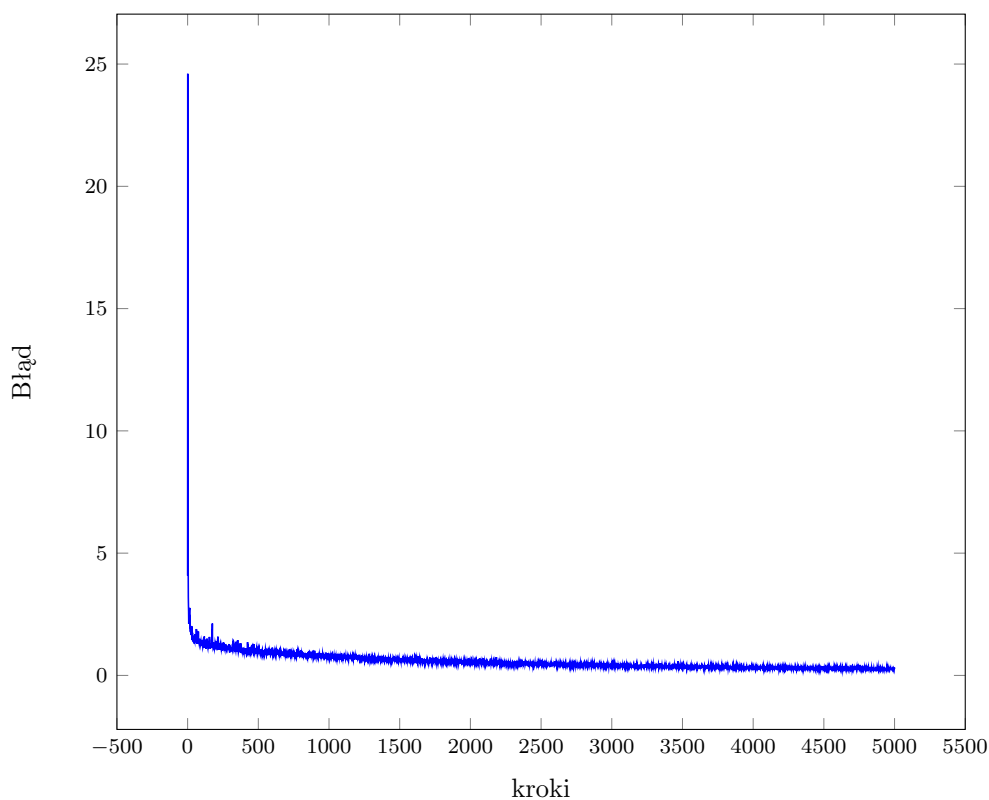
Tab. 3.1: Dane po 5000 kroków

### 3.2. Funkcja aktywacji

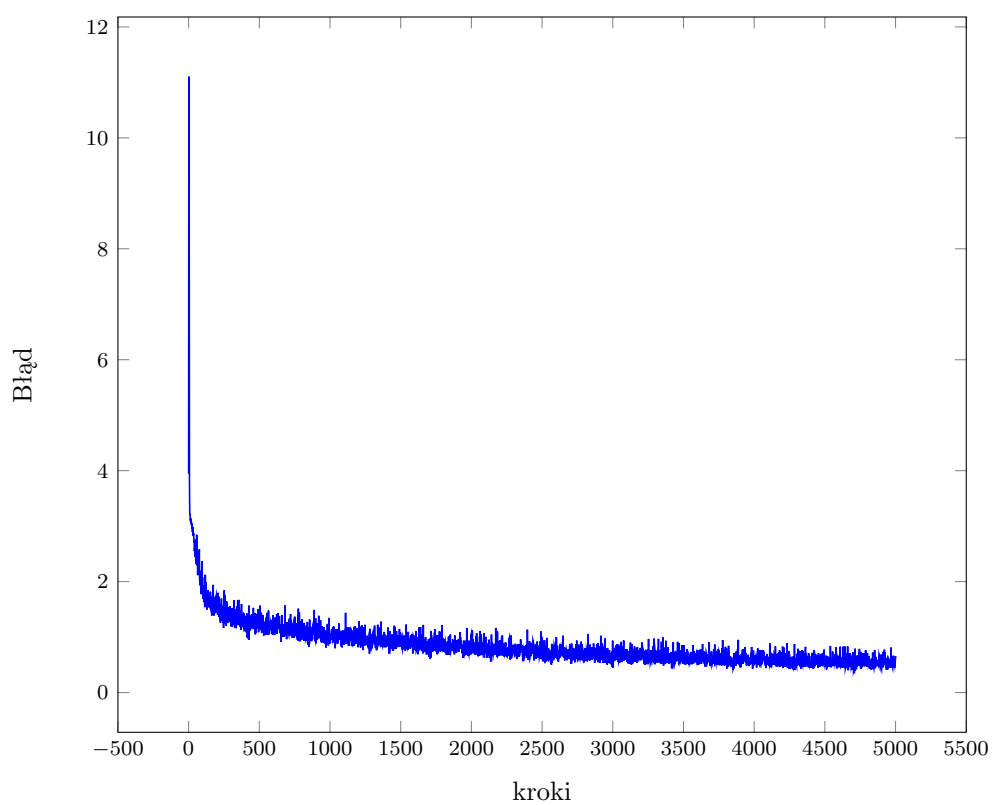
Wyniki przedstawione w tabeli ?? pozwalają jednoznacznie stwierdzić, że funkcja aktywacji ReLU daje znacznie lepsze wyniki niż sigmoidalna. Ponadto porównując wykresy ?? i ?? można zauważyć, że funkcja błędu ma znacznie mniejszą wariancję dla ReLU niż w przypadku funkcji sigmoidalnej. W kolejnych eksperymentach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

Funkcja aktywacji	Wyniki
hline 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

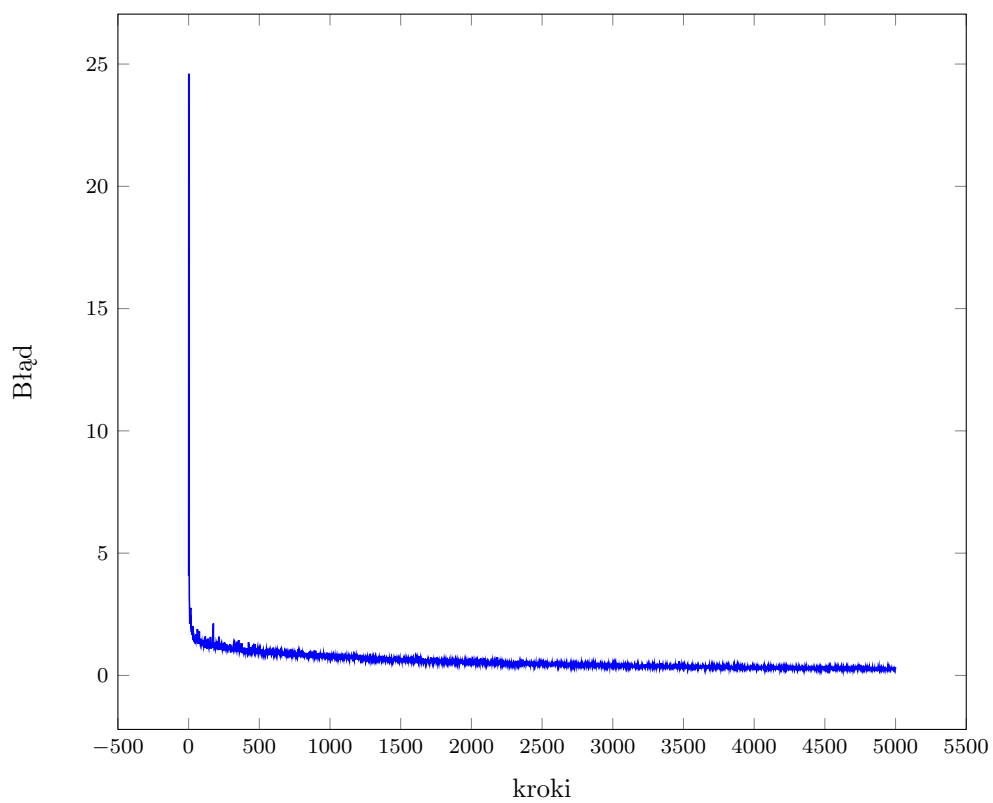


**Rys. 3.18:** Funkcja aktywacji sigmoidalna



## 4. Algorytm uczenia

Algorytm	Wyniki
SGD	Zbiór uczący: 98,4% Zbiór testowy: 98%
5—c—Momentum (pęd) sdvs	csvs



**Rys. 4.1:** Stochastyczny spadek gradientu (SGD)