

Sprawozdanie z laboratoriów: Perceptrony Wielowarstwowe (MLP)

Bartłomiej Wójcik

Numer indeksu: 327327

Spis treści

Opis tematyki laboratoriów	2
NN1: Bazowa implementacja MLP	2
NN2: Propagacja wsteczna błędu	4
NN3: Moment i RMSProp	5
NN4: Klasyfikacja z funkcją softmax	8
NN5: Różne funkcje aktywacji	12
NN6: Przeuczenie i regularyzacja	17
Podsumowanie	22

Opis tematyki laboratoriów

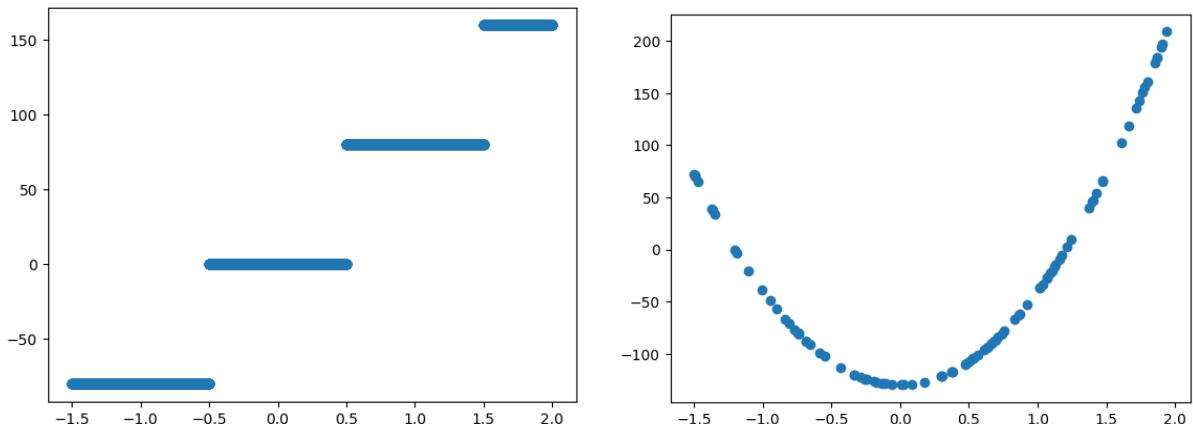
Celem laboratoriów było zapoznanie się z perceptronami wielowarstwowymi (MLP), ich ręczną implementacją oraz technikami uczenia nadzorowanego. Zajęcia obejmowały projektowanie architektury sieci, propagację wsteczną, optymalizację gradientową z momentem i RMSProp, zastosowania w regresji i klasyfikacji oraz techniki przeciwdziałania przeuczeniu. Kolejne tematy laboratoriów zapoznawały nas z nowymi zagadnieniami, które należało zaimplementować jako funkcjonalności sieci. W tym sprawozdaniu opisany jest rozwój tego projektu, wraz z testami na różnorodnych zbiorach danych.

NN1: Bazowa implementacja MLP

Opis zadania

Bazowa implementacja architektury sieci neuronowej typu MLP z możliwością konfiguracji liczby warstw, neuronów oraz wag. Zaimplementowanie funkcji sigmoidalnej jako aktywacji (opcjonalnie funkcji liniowej na wyjściu). Następnie przeprowadzone zostały eksperymenty dla zbiorów **square-simple** oraz **steps-large** z manualnym dobraniem wag i biasów.

Eksperymenty i wyniki



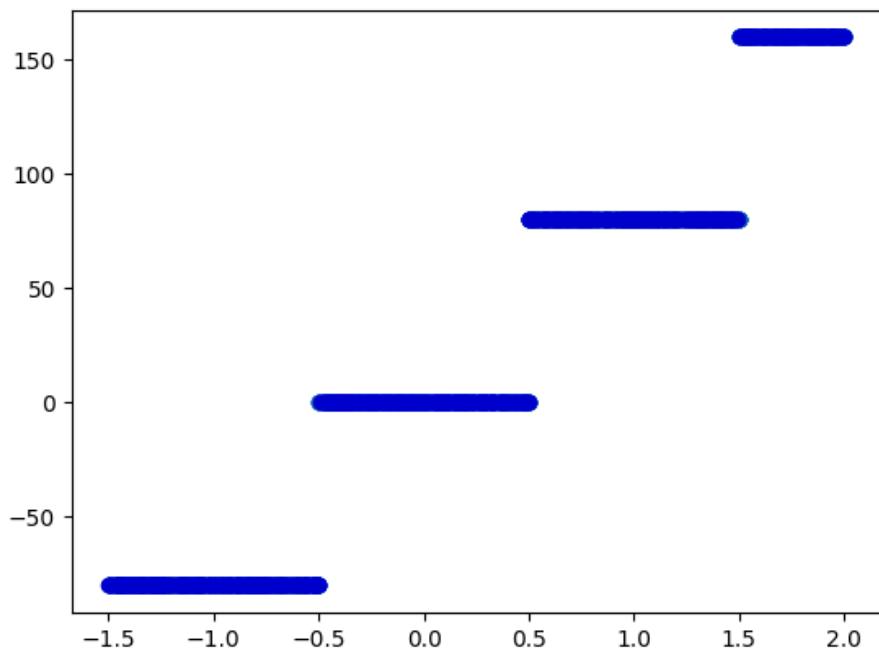
Rys. 1. Wizualizacja zbiorów danych, do których należało dobrać wagi.

Dla zbioru steps dobranie wag było dosyć proste i wystarczyła jedynie jedna warstwa ukryta o 5 neuronach (wystrzyczyło dobrać wagi tak aby na kolejnych przedziałach wzrosła aktywowana liczba neuronów). Udało się osiągnąć idealne dopasowanie ($MSE=0$) zarówno na zbiorze treningowym jak i testowym.

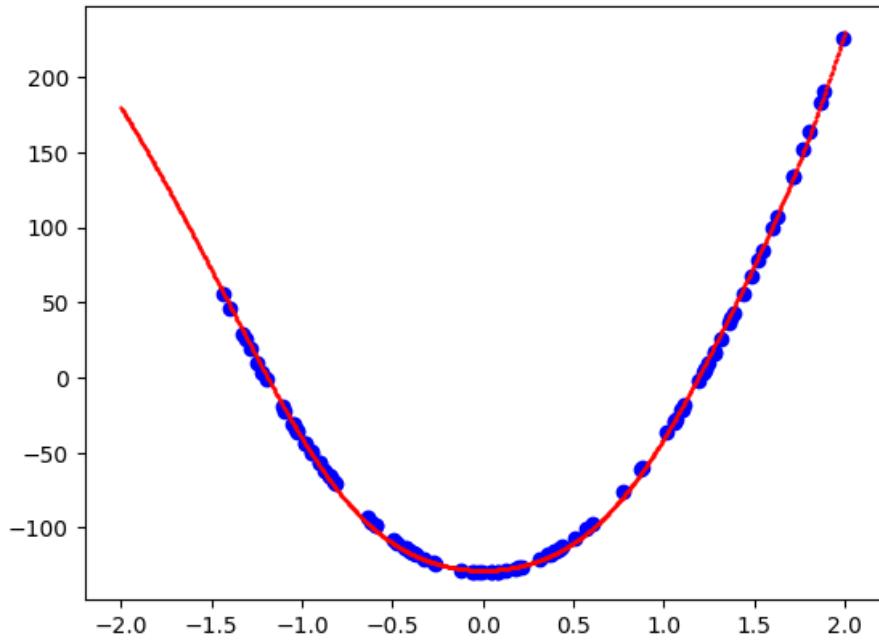
Dla zbioru square metodologia pozostała analogiczna natomiast do otrzymania satysfakcjonującego wyniku potrzebne było 10 neuronów.

Dataset	MSE train	MSE test
Steps large	0.0	0.0
Square simple	5.49	4.98

Tabela 1. Wyniki NN1



Rys. 2. Rezultat po dobraniu odpowiednich wag (zbiór steps large)



Rys. 3. Rezultat po dobraniu odpowiednich wag (zbiór square simple)

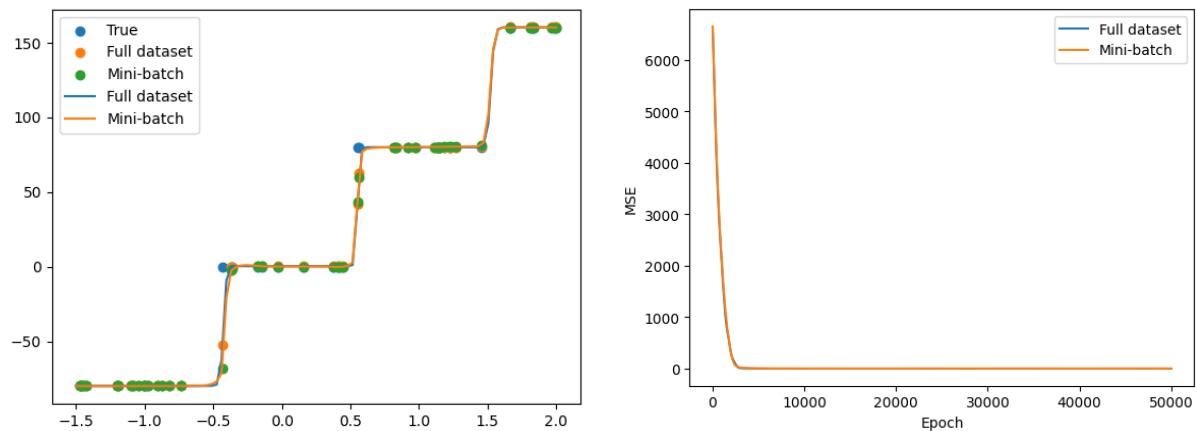
NN2: Propagacja wsteczna błędu

Opis zadania

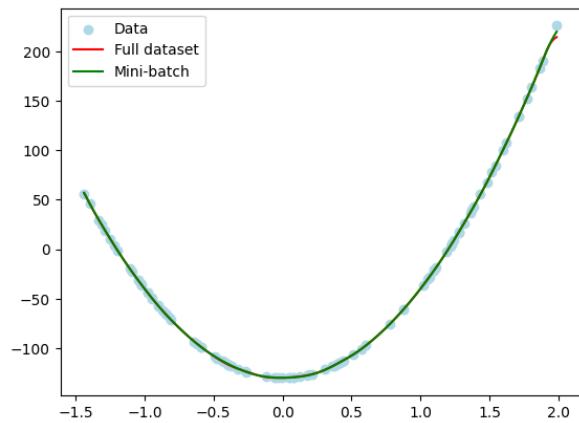
Implementacja propagacji wstecznej z aktualizacją wag i biasów oraz metody trenowania mini-batch. Umożliwia to modelowi naukę i korektę błędów.

Eksperymenty i wyniki

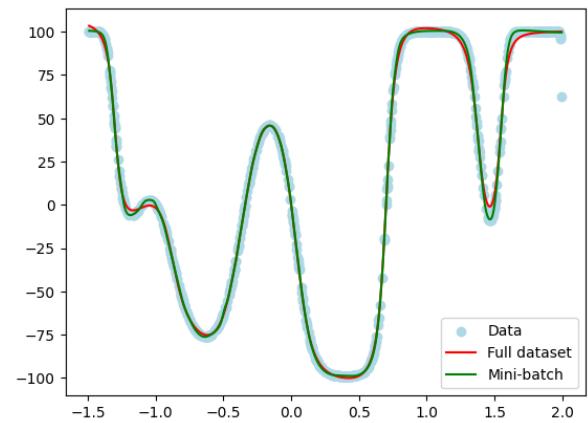
Po testowaniu i odpowiednim dobraniu architektury, rozmiaru batcha czy współczynnika uczenia, otrzymałem następujące wyniki.



Rys. 4. Wyniki dla zbioru Steps.



Rys. 5. square simple



Rys. 6. multimodal large

Rys. 7. Wyniki dla zbiorów Square i Multimodal.

Dataset	MSE full	MSE mini-batch
Steps large	89.77	127.39
Square simple	1.35	0.40
Multimodal large	6.18	1.54

Tabela 2. Wyniki NN2

Dla zbioru steps, niemożliwe było osiągnięcie satysfakcyjnych wyników ze względu na niefortunne dobranie zbiorów treningowego i testowego (testowy jest delikatnie przesunięty względem treningowego). Z obserwacji podczas tego laboratorium zauważałem, że przy podziale na batche model uczy się szybciej i lepiej radzi sobie ze zmiennością danych czy dziwnymi skokami. Na ten moment sieć ma problemy z nauką skrajnych punktów, takich jak np. na granicy "schodka" w zbiorze steps czy na prawym skraju zbioru multimedial. Temu zjawisku powinna zaradzić implementacja uczenia z momentem, która zostanie zaimplementowana w kolejnym etapie.

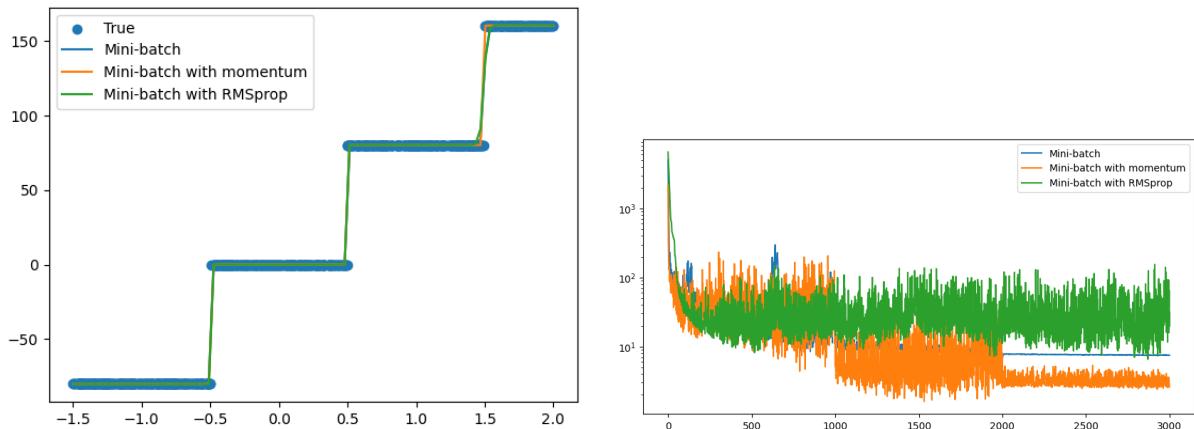
NN3: Moment i RMSProp

Opis zadania

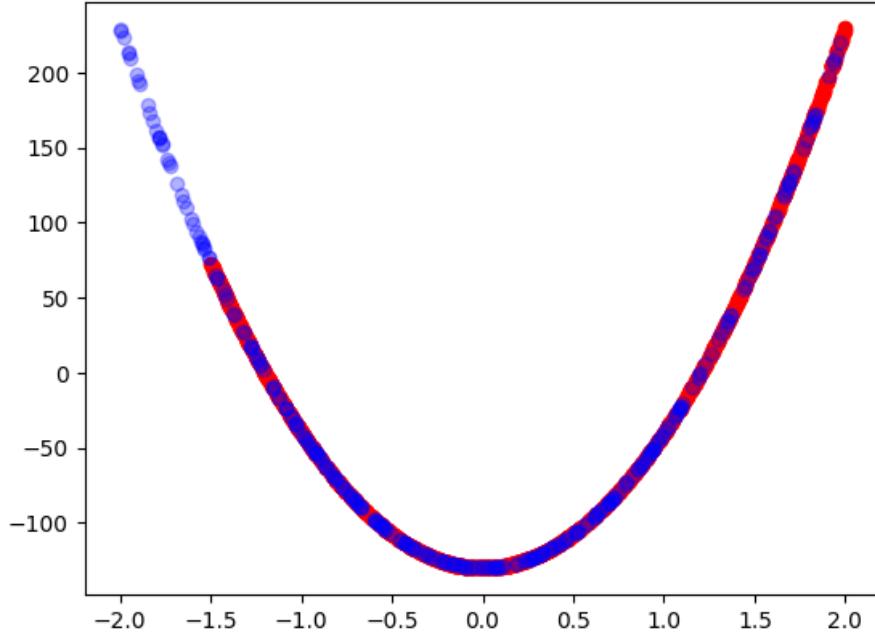
W tym laboratorium zostały dodane metody uczenia z momentem i RMSProp, które jako dodatkowa funkcjonalność backpropagacji, mają przyczynić się do większej dokładności modelu i szybszej zbieżności.

Eksperymenty i wyniki

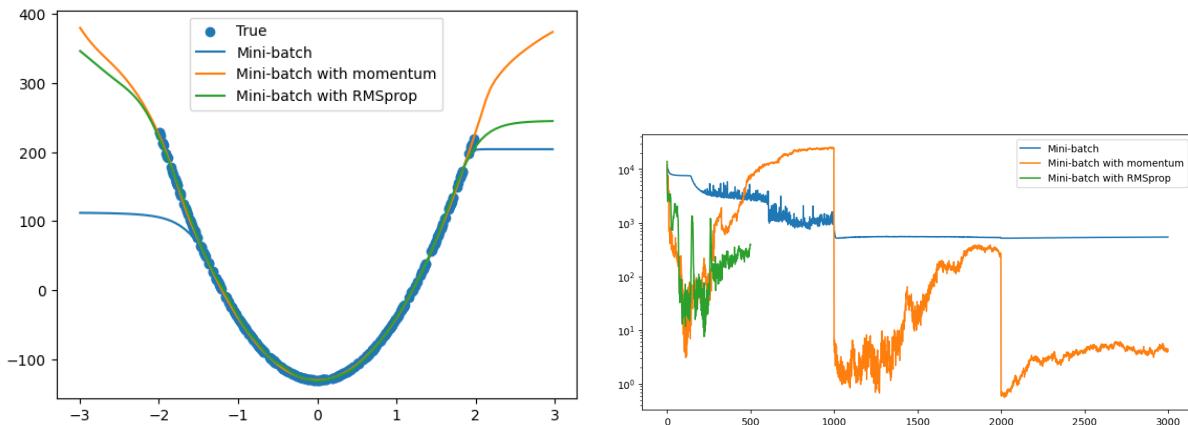
(Wykresy MSE w czasie dla różnych metod, testy na zbiorach.)



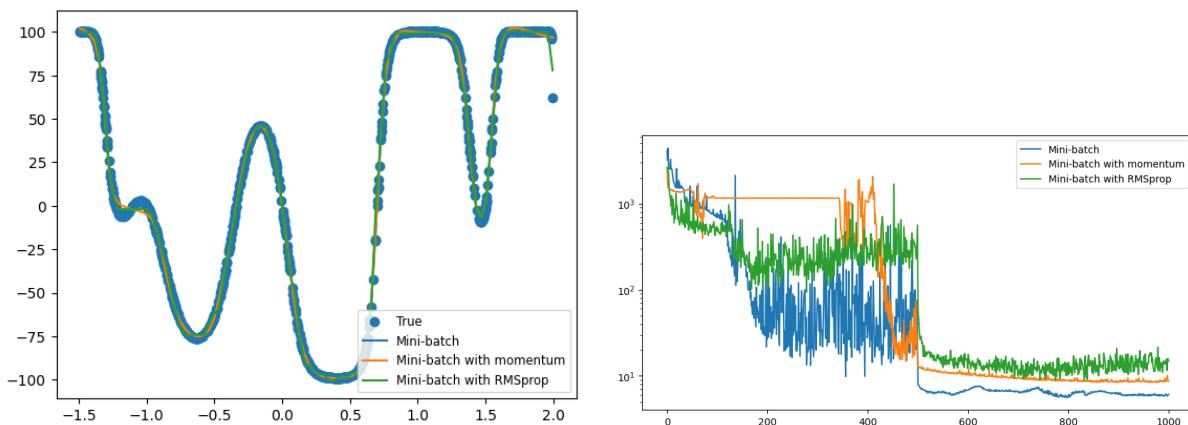
Rys. 8. Wyniki dla zbioru Steps.



Rys. 9. Tym razem dla zbioru square mamy różnice pomiędzy zbiorem testowym a treningowym (zbiór testowy [kolor niebieski] został rozszerzony aby sprawdzić jak sieć radzi sobie z rozpoznawaniem patternów).



Rys. 10. Wyniki dla zbioru Square.



Rys. 11. Wyniki dla zbioru Multimodal.

Na podstawie otrzymanych wyników mogę śmiało stwierdzić że zarówno uczenie z momentem, jak i RMSProp poprawiły szybkość uczenia. Uczenie z momentem radzi sobie również o wiele lepiej z wykrywaniem patternów i generalizacja rozwiązania oraz lepiej przewiduje obserwacje z poza zakresu uczenia (dobrym na to przykładem jest wizualizacja wyników dla zbioru Square)

Dataset	MSE mini-batch	MSE mini-batch momentum	MSE mini-batch rmsprop
Steps large	6.51	1.60	6.57
Square large	516.27	0.56	7.60
Multimodal large	1.56	3.56	5.97

Tabela 3. Wyniki NN3

NN4: Klasyfikacja z funkcją softmax

Opis zadania

W kolejnym etapie projektu, należało dodać do architektury sieci pozwalające na rozwiązywanie zadań klasyfikacji. Do tego idealnie sprawdza się funkcja Softmax, jako funkcja aktywacji warstwy wyjściowej, do lepszej ewaluacji modeli przy klasyfikacji jako funkcja straty posłużył cross-entropy loss. W ramach tego laboratorium należało porównać modele z zaimplementowaną funkcją softmax na wyjściowej warstwie, z inną funkcją aktywacji, ja wykorzystałem do tego Sigmoid.

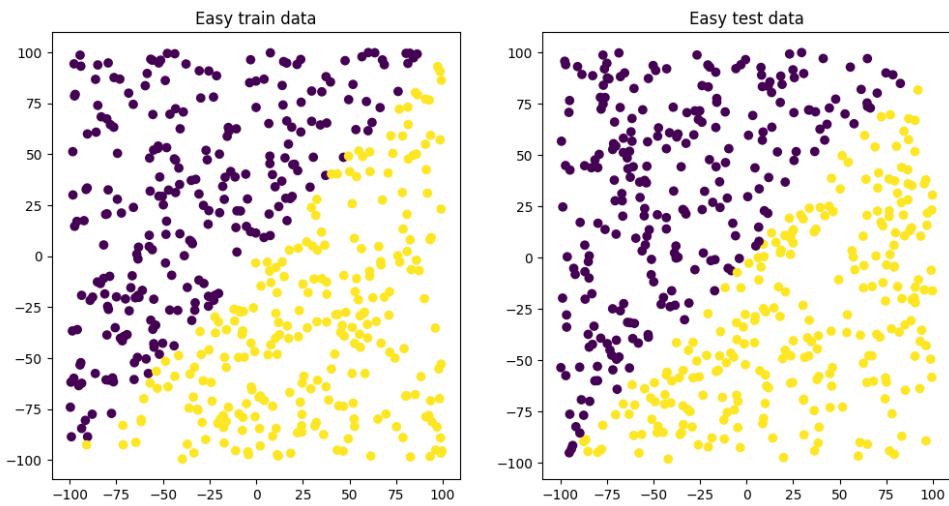
Eksperymenty i wyniki

Ze względu na prostote zbiorów, nie było zbyt wielkich różnic w wynikach, zarówno modele z softmaxem i sigmoidem zdołały bardzo dobrze nauczyć się zbiorów. Jednakże w przypadku bardziej skomplikowanych zbiorów wykorzystanie Softmaxa ukaże swoje zalety i interpretowalność jako prawdopodobieństwo przyporządkowania do danej klasy.

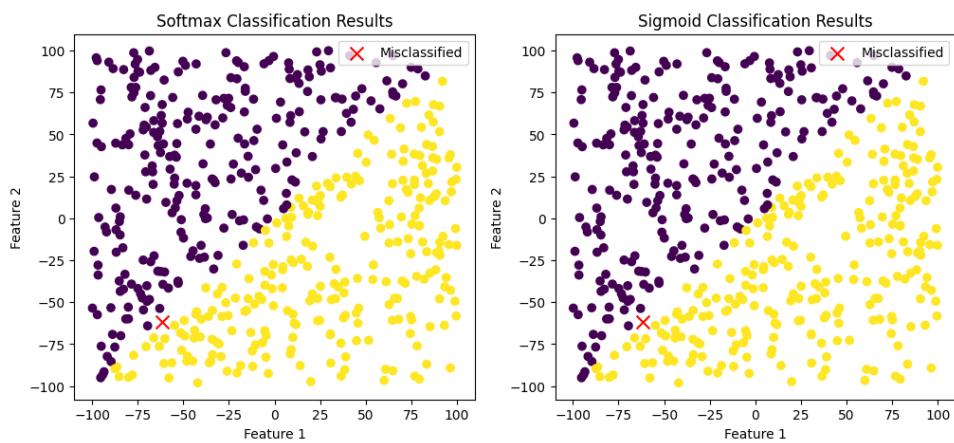
Dataset	F score	accuracy
Easy Softmax	0.998	0.998
Easy Sigmoid	0.998	0.998
Rings3 regular Softmax	0.974	0.971
Rings3 regular Sigmoid	0.974	0.971
XOR3 Softmax	0.974	0.978
XOR3 Sigmoid	0.951	0.958

Tabela 4. Wyniki NN4

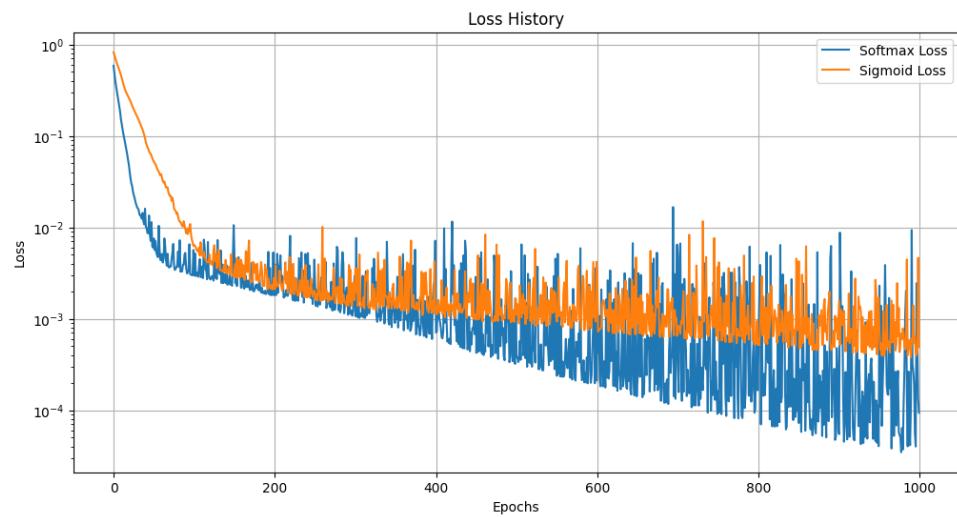
Easy dataset



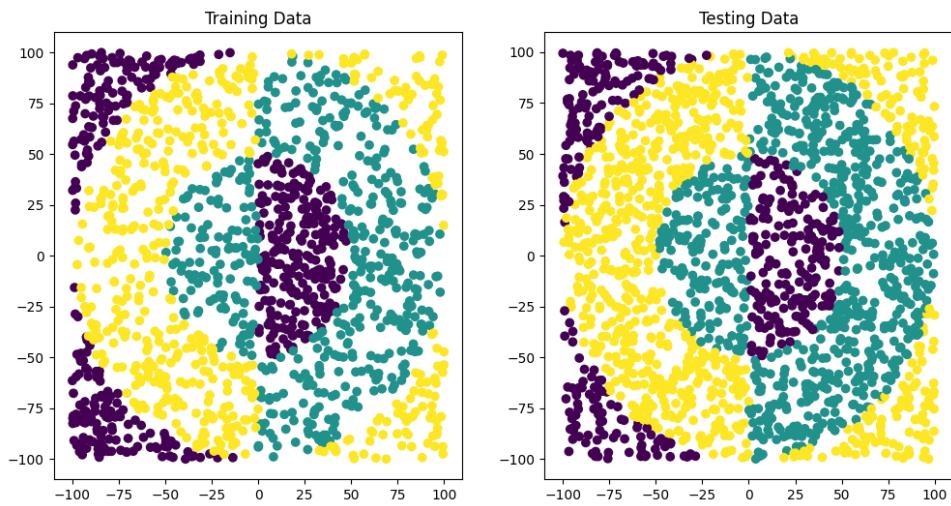
Rys. 12. Podział na klasy w zbiorze Easy



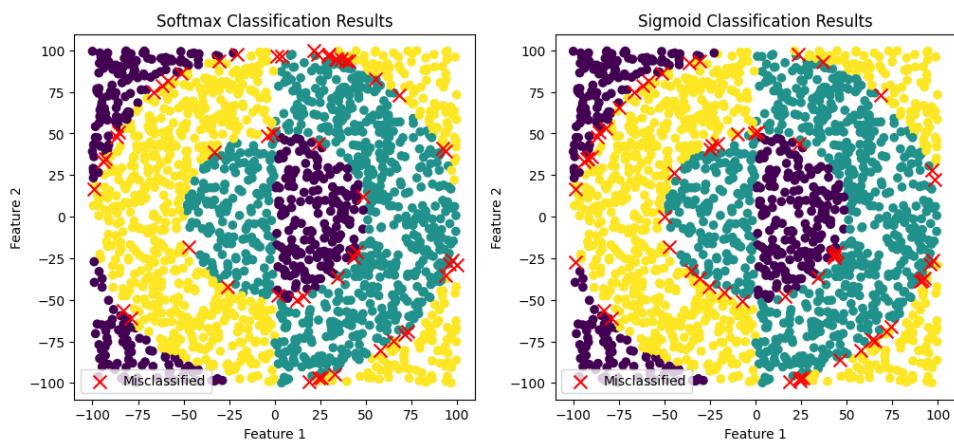
Rys. 13. Wizualizacja klasyfikacji w zbiorze Easy



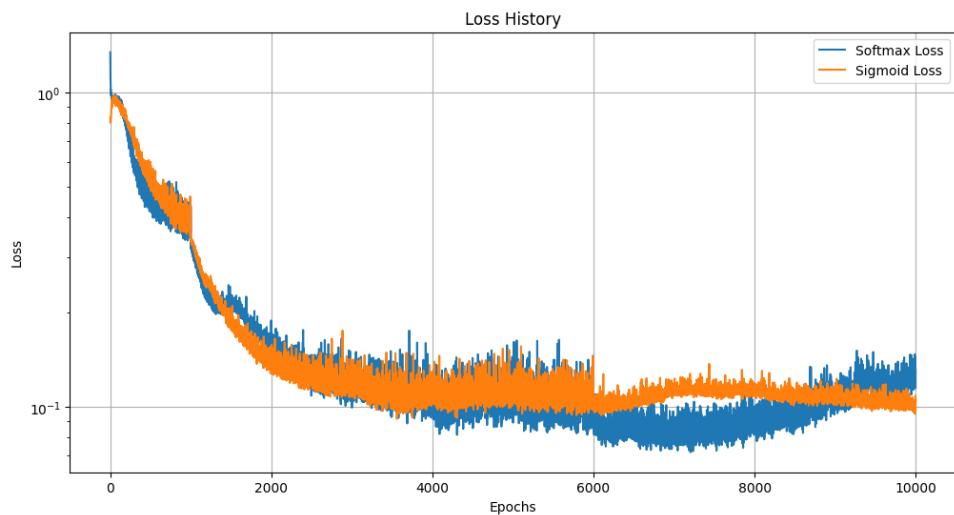
Rings3 regular



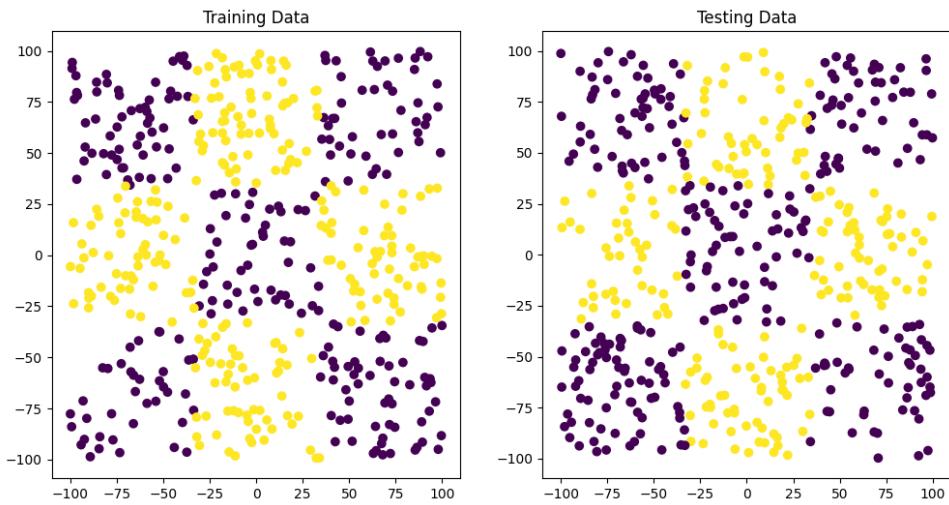
Rys. 14. Podział na klasy w zbiorze Rings3



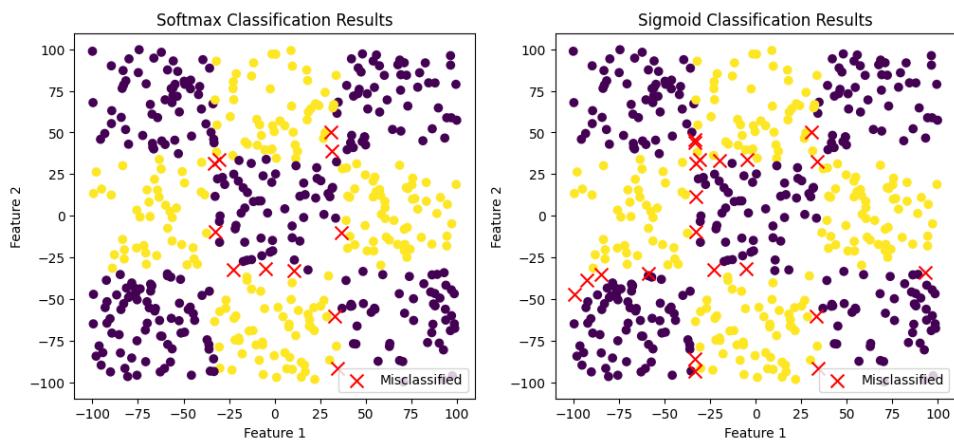
Rys. 15. Wizualizacja klasyfikacji w zbiorze Rings3



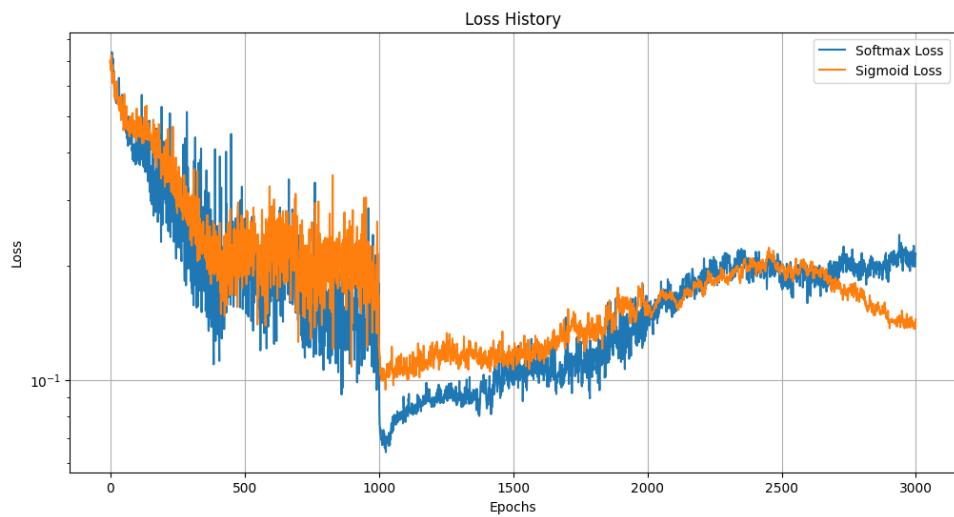
Xor3 regular



Rys. 16. Podział na klasy w zbiorze Xor3



Rys. 17. Wizualizacja klasyfikacji w zbiorze Xor3



NN5: Różne funkcje aktywacji

Opis zadania

Podczas tego laboratorium należało porównać użycie w sieci różnych funkcji aktywacji. Do tej pory w każdej z sieci używaną funkcją aktywacji był Sigmoid. Teraz dodatkowo zaimplementowana została możliwość wyboru funkcji ReLU czy tanh.

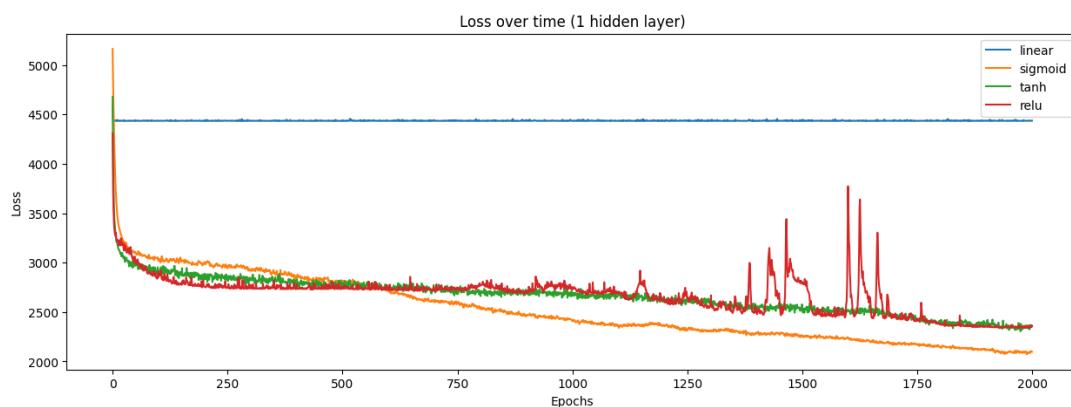
Testowanie różnych funkcji aktywacji

Porównane zostały 4 funkcje aktywacji: liniowa, sigmoid, relu oraz tanh. Dla każdej z nich zostały stworzone 3 następujące sieci, które następnie zostały użyte do nauki zbioru multimedialnego:

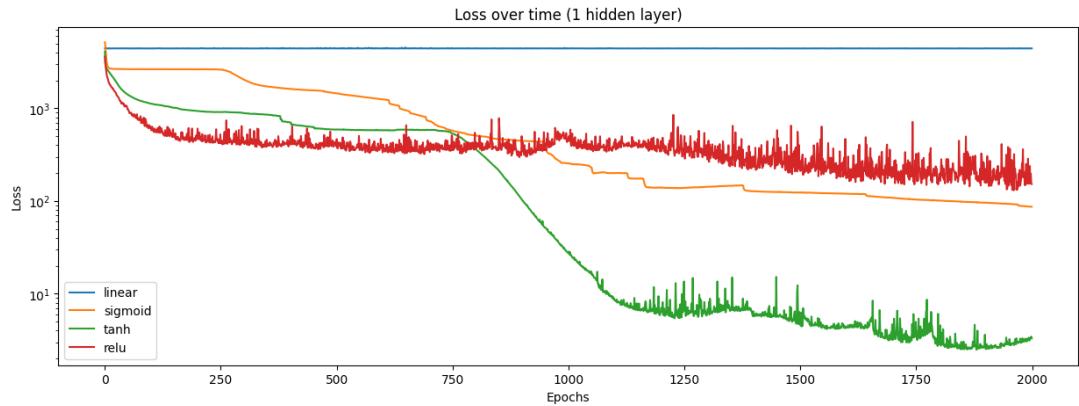
- 1 warstwa ukryta z 96 neuronami
- 2 warstwy ukryte po 48 neuronów
- 3 warstwy ukryte po 32 neurony

Liczba warstw	MSE linear	MSE sigmoid	MSE tanh	MSE ReLU
1	4433.57	2072.88	2305.07	2341.50
2	4433.57	86.71	2.48	130.61
3	4433.57	2.28	2.20	142.34

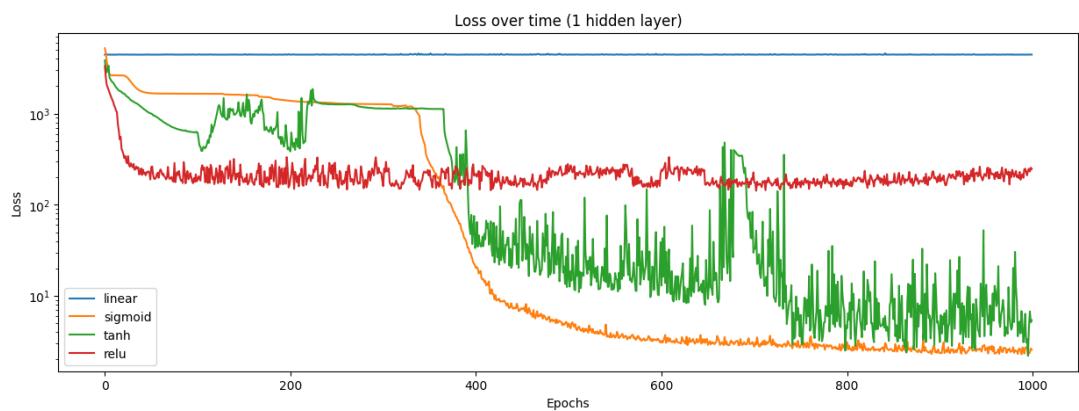
Tabela 5. Wyniki otrzymane podczas porównania funkcji aktywacji.



Rys. 18. Proces nauki przy 1 warstwie ukrytej.



Rys. 19. Proces nauki przy 2 warstwach ukrytych.



Rys. 20. Proces nauki przy 3 warstwach ukrytych.

Szybkie wnioski:

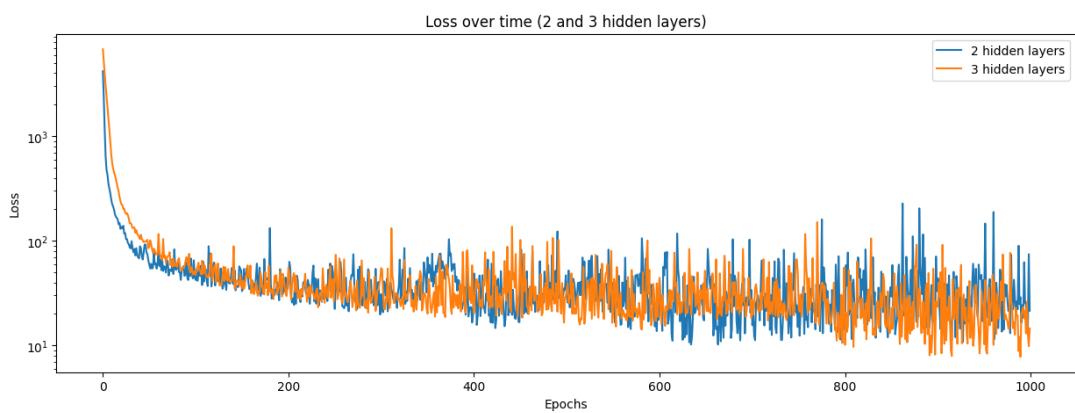
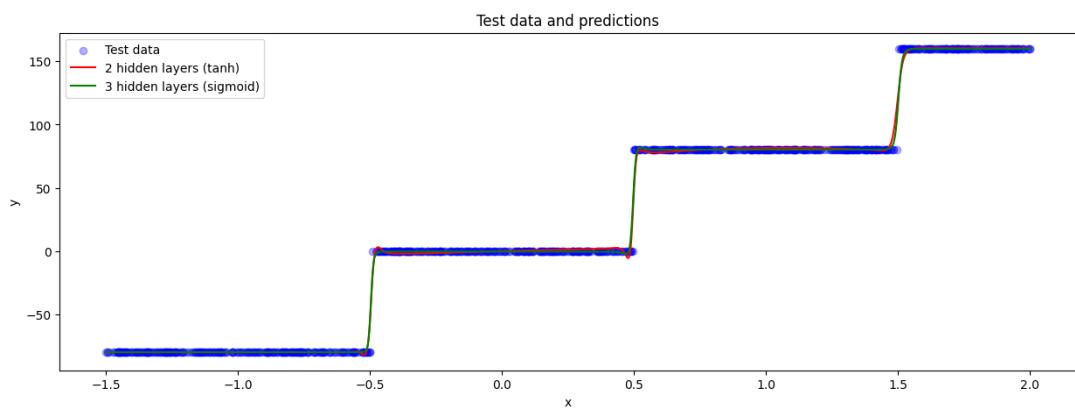
- jak można się było przewidzieć liniowa aktywacja nie była w stanie nauczyć się zbioru multimodal bedącego wielomianem
- ReLU napoczątku nauki osiągało lepsze wyniki ale finalnie otrzymane wyniki były zdecydowanie gorsze niż dla tanh czy sigmoida
- ogólnie najlepsze wyniki osiągnięte zostały przy użyciu tanh
- dla sigmoida i ReLU sieć z 2 warstwami okazała się zbyt "pływka", model nie zdołał się dostatecznie dobrze nauczyć
- jako dwa najlepsze modele, które zostały użyte w dalszej części tego laboratorium wybrałem **tanh z 2 warstwami ukrytymi** i **sigmoid z 3 warstwami ukrytymi**

Eksperymenty i wyniki dla dwóch najlepszych modeli

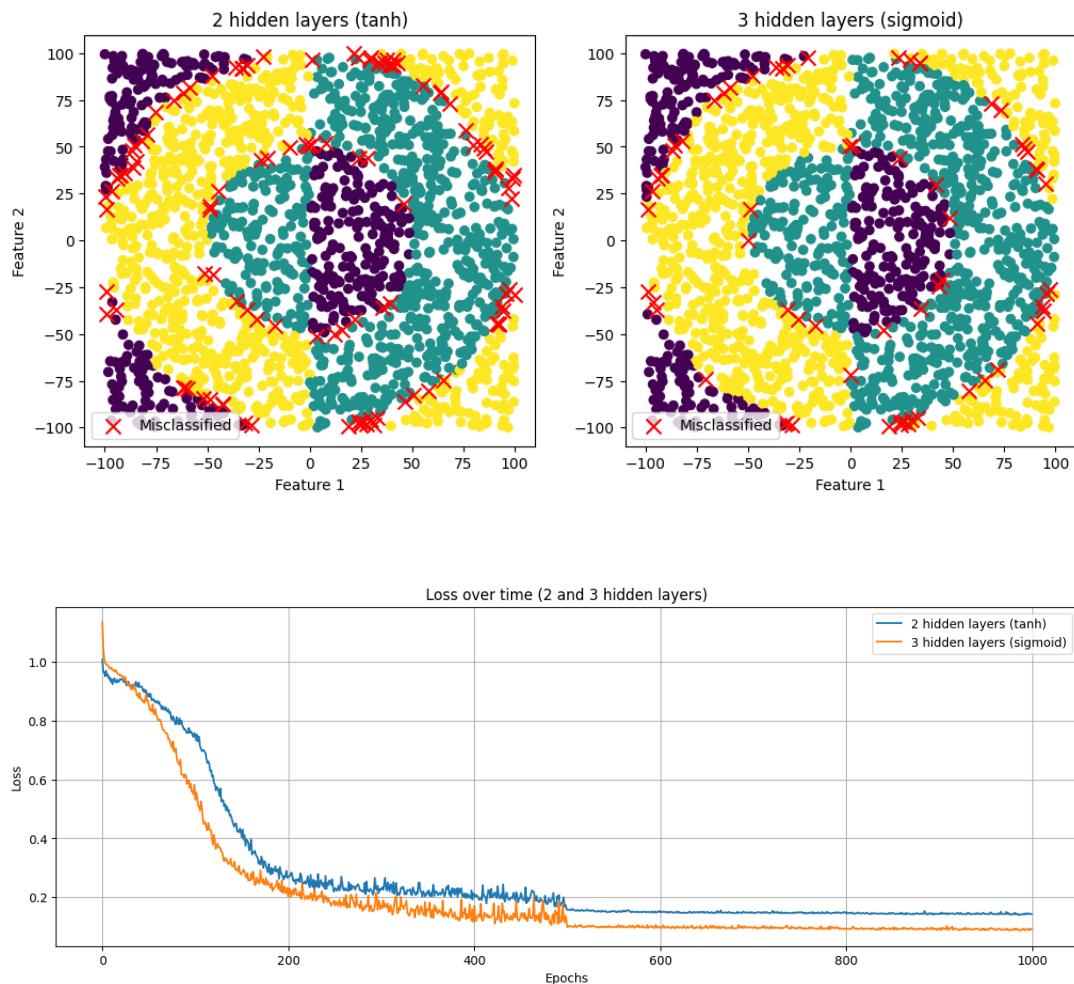
Dataset	MSE tanh 2 hidden layers	MSE sigmoid 3 hidden layers
Steps large	10.18	7.77
Dataset	F score tanh 2 hidden layers	F score sigmoid 3 hidden layers
Rings3 regular	0.9454	0.9675
Rings5 regular	0.9446	0.9396

Tabela 6. Wyniki NN5

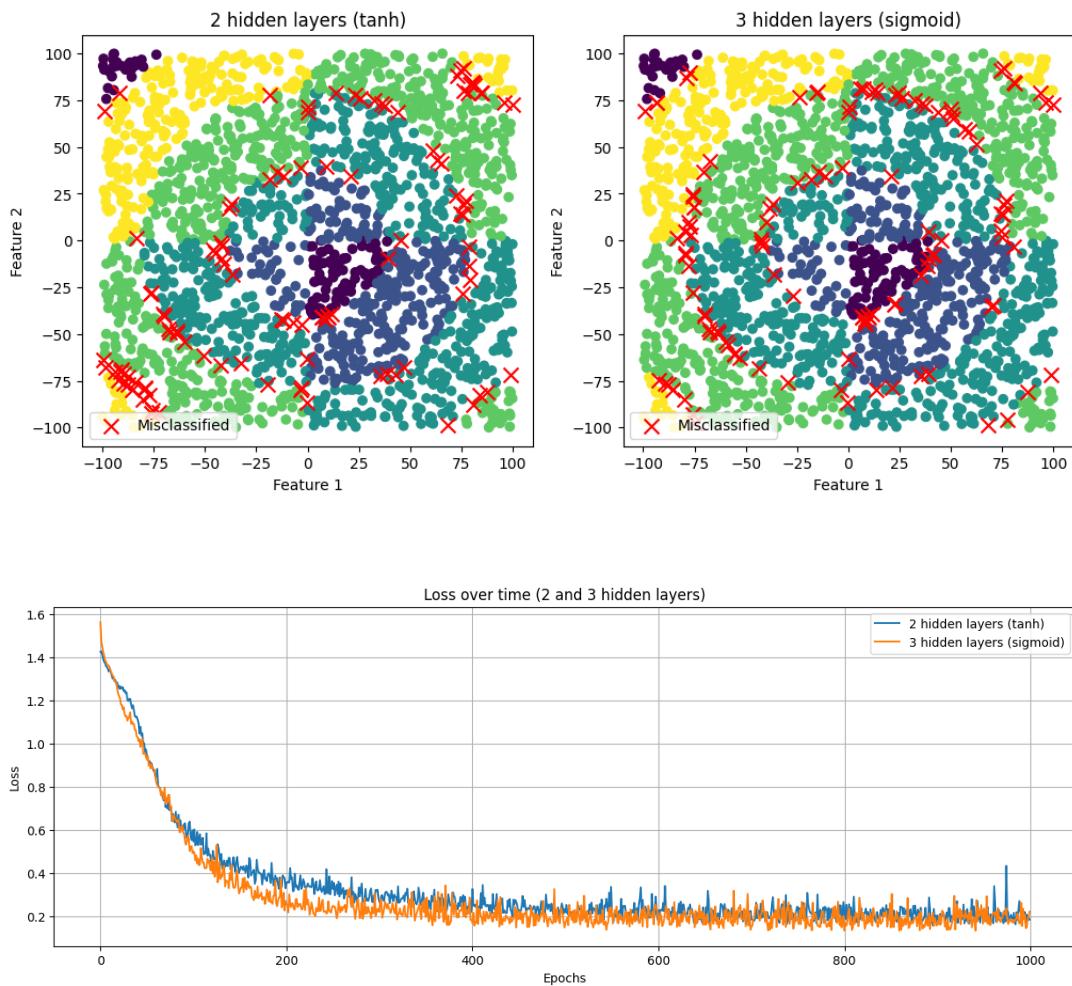
Steps large



Rings3 regular



Rings5 regular



Wnioski

Na podstawie powyższych wyników ciężko jednoznacznie wybrać lepszą architekturę, obydwie poradziły sobie ze zbiorami bardzo dobrze. Z ogólnych wniosków zaobserwować można szybszą zbieżność i lepsze wyniki dla tanh w porównaniu do sigmoida w przypadku wykorzystania takich samych architektur sieci.

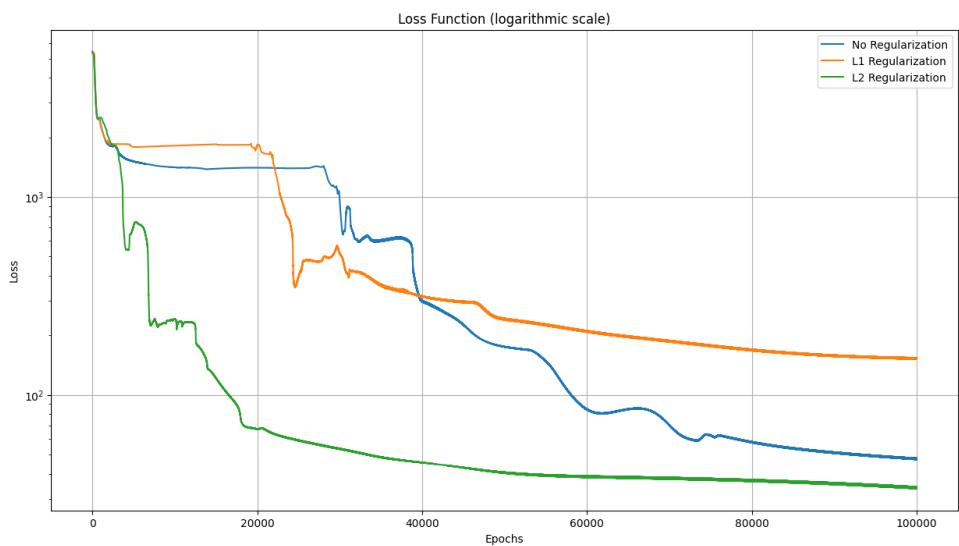
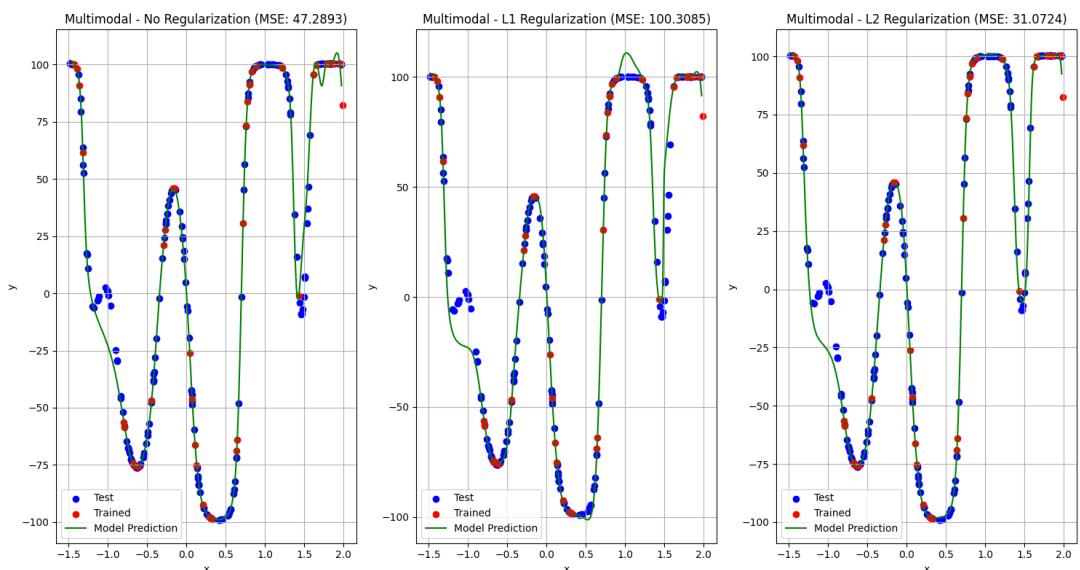
NN6: Przeuczenie i regularyzacja

Opis zadania

W ostatnim etapie należało zaimplementować mechanizm regularyzacji wag w sieci oraz mechanizm zatrzymywania uczenia przy wzroście błędu na zbiorze walidacyjnym. Do nauki cały czas wykorzystywany jest zbiór treningowy, natomiast do ewaluacji najlepszego modelu (który jest zapamiętywany przez sieć) wykorzystywany jest zbiór walidacyjny niezależny od treningowego. Zbiory danych podczas tego laboratorium były rzadkie i niezbalansowane, jak można zauważyć na poniższych wizualizacjach. Dla każdego ze zbiorów sprawdzone zostały różne wartości parametrów regularyzacji i do predykcji zostały wybrane wartości dające najlepsze wyniki zarówno dla regularyzacji L1 i L2.

Eksperymenty i wyniki

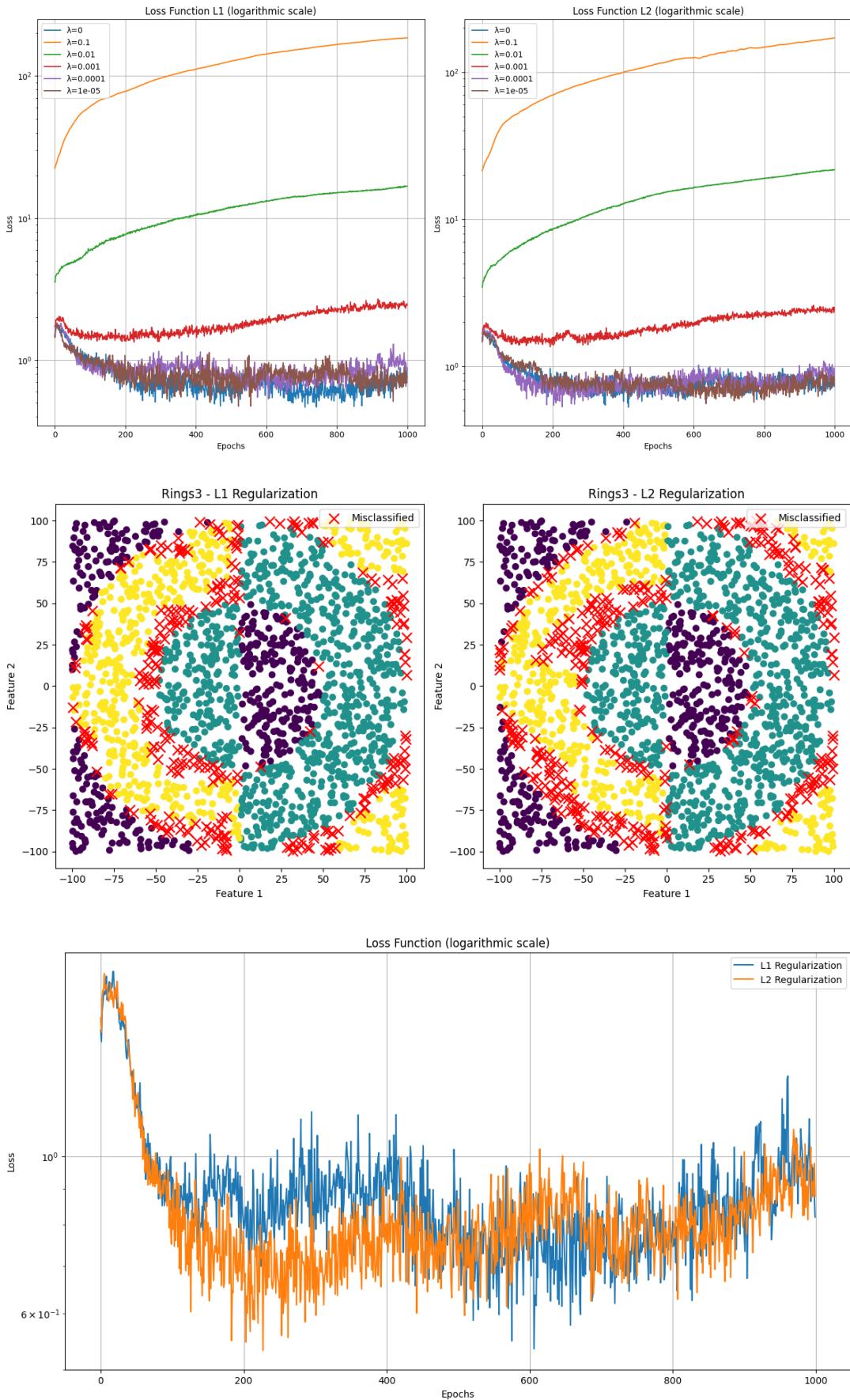
Multimodal sparse



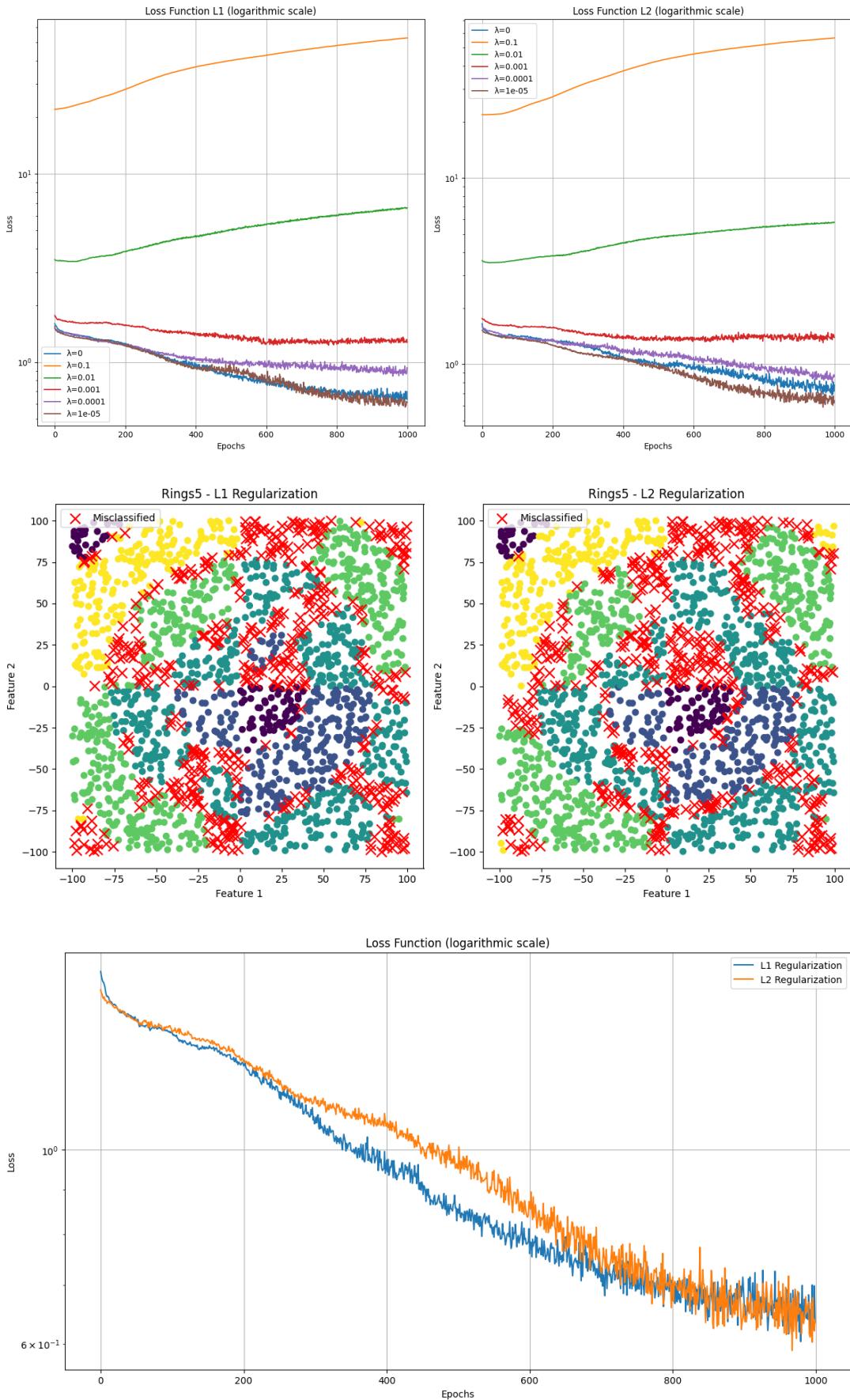
Wizualizacja zbiorów do klasyfikacji



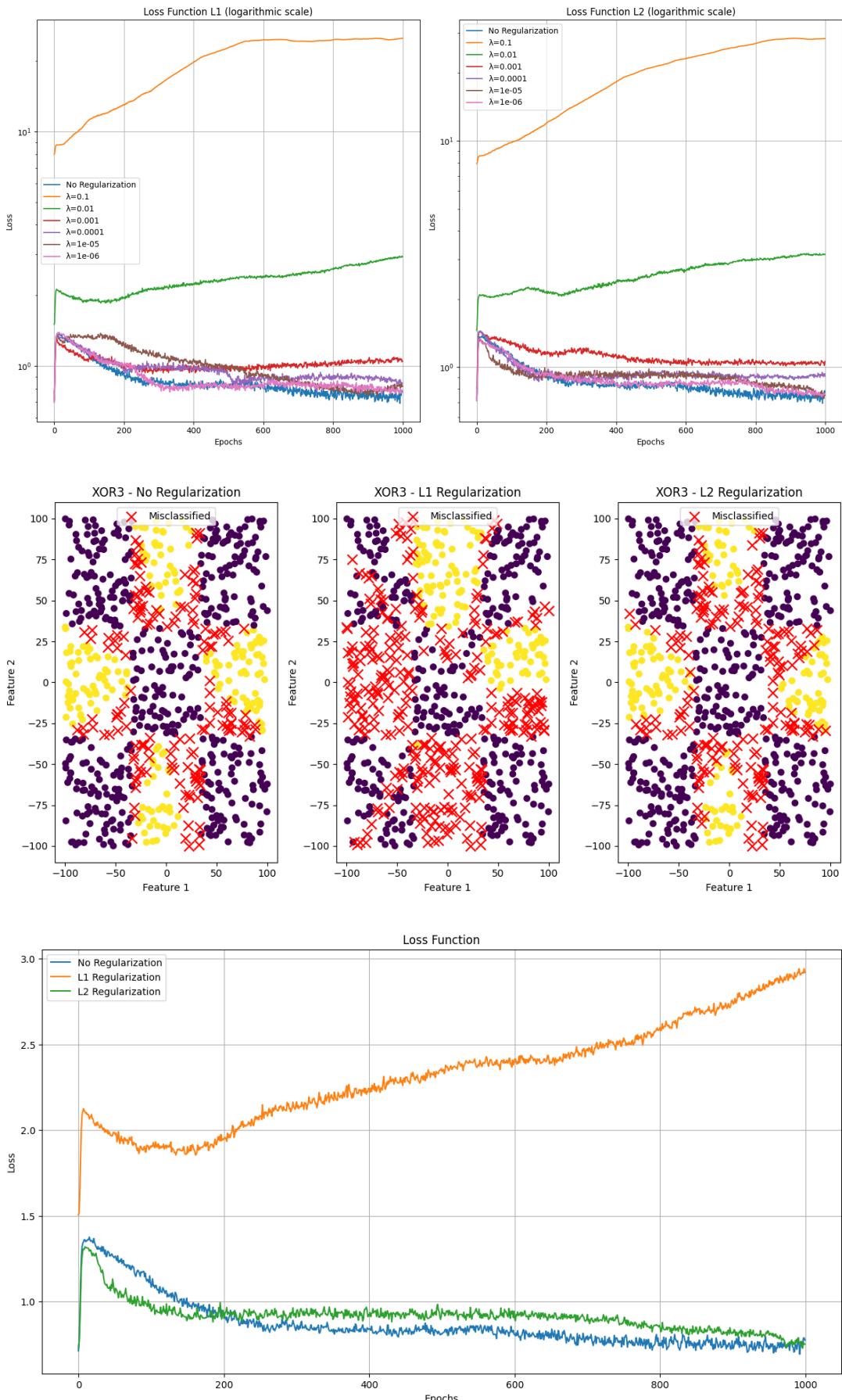
Rings3 sparse



Rings5 sparse



XOR3 sparse



Liczba warstw	MSE	MSE L1	MSE L2
Multimodal sparse	47.29	100.30	31.07
Liczba warstw	F score	F score L1	F score L2
Rings3 sparse	—	0.8598	0.8151
Rings5 sparse	—	0.7856	0.7809
Xor3 sparse	0.7860	5731	0.7418

Tabela 7. Wyniki NN6.

W przypadku wykorzystanych zbiorów, regularyzacja nie poprawiała zbytnio wyników uczenia, a jeśli to w bardzo nieznacznym stopniu. Przy dobraniu zbyt dużych parametrów regularyzacja uniemożliwiała naukę modelu. Dobrze prezentują to wykresy funkcji straty w czasie. Natomiast w przypadku bardzo małych wartości tych parametrów, model nie różnił się zbytnio od modelu bez regularyzacji.

Podsumowanie

- Przy wykorzystaniu jedynie bibliotek do podstawowych operacji matematycznych i wizualizacji danych, stworzony został model sieci neuronowej umożliwiający użytkownikowi dużą różnorodność w doborze architektury i treningu
- Dzięki propagacji wstępnej sieć jest w stanie ”uczyć się na swoich błędach”
- Dzięki możliwości wyboru różnych funkcji aktywacji i straty, sieć jest w stanie dobrze radzić sobie zarówno z zadaniami regresji, jak i klasyfikacji
- Dodanie uczenia z momentem czy RMSProp przyśpieszyło proces uczenia i pozwalało otrzymać jeszcze lepsze wyniki oraz lepszą generalizację modelu
- Zastosowanie metod regularyzacji czy walidacji modelu na niezależnym od trenin-gowego zbiorze pozwoliło na otrzymywanie satysfakcjonujących wyników również w przypadku ”rzadkich” zbiorów treningowych