Programowanie sieciowe

Perceptron i Adaline - Lab1

Data:	11.04.2022	Dzień:	Wtorek TN + 1/2
Grupa:	Y02-15b	Godzina:	17:05
Numer indeksu:	252889	Prowadzący:	dr inż. Marek Bazan
Nazwisko i imię: Nowek Jakub			

Spis treści

1. Opis problemu

Należało dokonać klasyfikacji gatunków irysów za pomocą modeli neuronów - Perceptron oraz Adaline. Źrodłem danych była baza danych Iris Data. Klasyfikator miał być w stanie rozpoznać do jakiego gatunku (klasy) należy kwiat po podaniu jego parametrów - w tym przypadku długości kielich oraz długości płatków.

1.1. Zad.1

W pierwszym zadaniu należało zaimplementować klasyfikację osobników jednej, dowolnej klasy za pomocą modelu perceptronu.

1.2. Zad.2

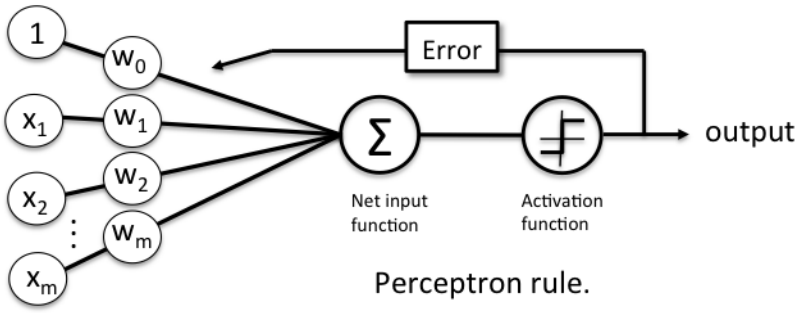
W drugim zadaniu należało zaimplementować klasyfikację osobników jednej, dowolnej klasy za pomocą modelu Adaline.

1.3. Zad.3

Trzecie zadanie polegało na zbudowaniu klasyfikatora dla trzech klas.

2. Opis użytych algorytmów

2.1. Perceptron



Rys. 1: Schemat perceptronu.

Cały algorytm uczenia perceptronu opiera się na powyższym modelu. Na początku inicjowane są wagi, jako losowe liczby o niewielkich wartościach lub o wartości 0. Pierwszą wagą o indeksie 0 jest tzw. 'bias'. Następnie liczony jest iloczyn skalarny wektora wejść (cech) i wektora wag:

$$o_1 = w \cdot x \tag{1}$$

Iloczyn ten jest kierowany na wejście funkcji aktywacji - dla perceptronu jest nią funkcja Heaveside'a. Tak otrzymane wyjście porównuje się ze znanym wynikiem ze zbioru uczącego a ich różnicę wykorzystuje się do zaktualizowania wektora wag zgodnie z zależnościami:

$$o(x) = f(w \cdot x) = f\left(\sum_{i=0}^{n} w_i \cdot x_i\right) \tag{2}$$

$$\begin{gathered} error(i) = y^{(i)} - o(x^{(i)}) \\ \Delta w = \eta \cdot error(i) \cdot x^{(i)} \\ w = w + \Delta w \end{gathered} \tag{3}$$

gdzie:

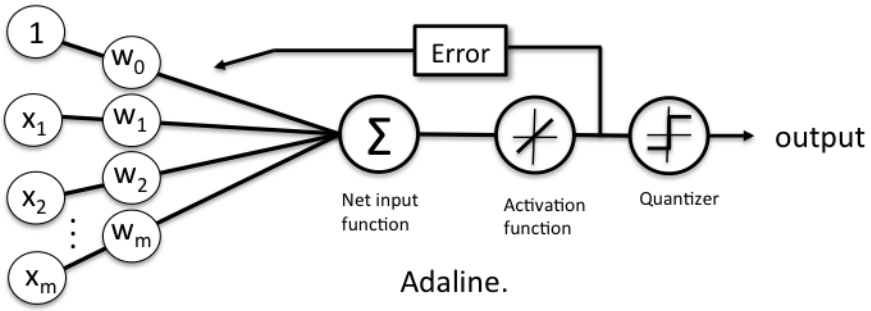
- η - szybkość uczenia zbioru
- i - numer elementu ze zbioru
- Epoka (epoch) - określa ilość przejść po zbiorze treningowym.

Następnie cała procedura jest powtarzana dla wszystkich elementów zbioru uczącego. Wynikiem działania algorytmu trenowania neuronu jest końcowy wektor wag. Gdy neuron został nauczony zbiorem uczącym (treningowym), można przystąpić do testowania neuronu na zbiorze testowym. Ważne jest, aby w zbiorze testowym nie znajdowały się elementy ze zbioru uczącego - daje to pewność że klasyfikacja została dokonana poprawnie, na podstawie obliczeń, a nie poprzez zwykłe przepisanie znanego już wyniku.

Klasyfikacji dokonuje się przez skalarne wymnożenie wektora danych testowych z wektorem wag uzyskanym w wyniku treningu oraz dodaniu do otrzymanego wyniku wartości bias'u. W przypadku gdy w zbiorze treningowym wynik '-1' oznaczał aktywację neuronu, a '1' brak aktywacji neuronu, wartość mniejsza od *zera*, oznacza, że dany element został wykryty jako element wskazany w zbiorze uczącym, natomiast wartość większa, bądź równa zeru oznacza, że element nie należy do pożądanej klasy.

Perceptron nie sprawdza się jednak w przypadku gdy klasy przez niego rozpoznawane są liniowo zależne. Wyliczony błąd nie zbiega w takim przypadku do *zera*, ponieważ perceptron przestaje aktualizować wektor wag dopiero, gdy wszystkie próbki zostaną sklasyfikowane poprawnie, a gdy jakaś próbka zostanie błędnie sklasyfikowana, perceptron nigdy nie przestanie aktualizować wektora wag. Aby uniknąć zbyt dużej ilości błędnych klasyfikacji można ustawić odpowiednią maksymalną liczbę epok - przejść po zbiorze treningowym.

2.2. Adaline - Adaptive Linear Neuron



Rys. 2: Schemat Adaline.

Powyższy rysunek przedstawia model Adaline. Jest on bardzo podobny do modelu perceptronu, z tą różnicą, że w Adaline błąd obliczany jest przed przejściem iloczynu skalarnego przez funkcję aktywacji Heavyside'a, tzn:

$$error(i) = y^{(i)} - o_1(x^{(i)}) \tag{4}$$

gdzie:

- y - znana wartość ze zbioru treningowego
- o_1 - tzw. soft output

Dodatkowo model Adaline jest typowo przedstawany w dwóch wersjach, różniących się pod względem aktualizacji wag.

2.2.1. Stochastic Gradient Descent - online learning

Rozważany tu model neuronu Adaline został zaimplementowany jako AdalineSGD w ramach 'testów grupy uderzeniowej'. W modelu SGD wagi są aktualizowane po każdej próbce - stąd 'online learning'. Innym podejściem byłoby aktualizowanie wag po przejściu przez wszystkie próbki treningowe.

2.3. Klasyfikator dla 3 klas

Klasyfikator 3 klas został stworzony zarówno na podstawie modeli Perceptronu, jak i Adaline. W każdym z tych dwóch przypadków działanie algorytmu było następujące:

1. Każdy z trzech klasyfikatorów klasyfikuje jako pozytywną inną klasę oraz jako negatywną pozostałe dwie klasy.
2. Każdy z klasyfikatorów został wytrenowany na zbiorze przygotowanym zgodnie z punktem powyżej.
3. Klasyfikacja polegała na podaniu tego samego zbioru testowego do wszystkich trzech klasyfikatorów i wybraniu dla każdej próbki tej klasy gdzie wskazanie pozytywne było największe.

3. Testy numeryczne

3.1. Definicja testów

Na początku neuron lub zestaw neuronów są trenowane na zbiorze treningowym, zawierającym po 80% danych z każdej kasy zbioru Iris (łącznie 120 próbek, po 40 z każdej klasy). Pozostałe 20% z każdej klasy umieszczamy w zbiorze testowym (łącznie 30 próbek, po 10 z każdej klasy). Dane w obu zbiorach są następnie standaryzowane.Potem dla różnych wartości prędkości uczenia η oraz dla różnych ilości epok wykonujemy uczenie neuronu, a później test predykcji klasy (gatunku kwiatu).

Dla klasyfikatorów 2 klas - Perceptron i Adaline wygenerowano obszar decyzyjny klasyfikatora oraz wykres zależności wartości błędu *error* w zależności od ilości iteracji.

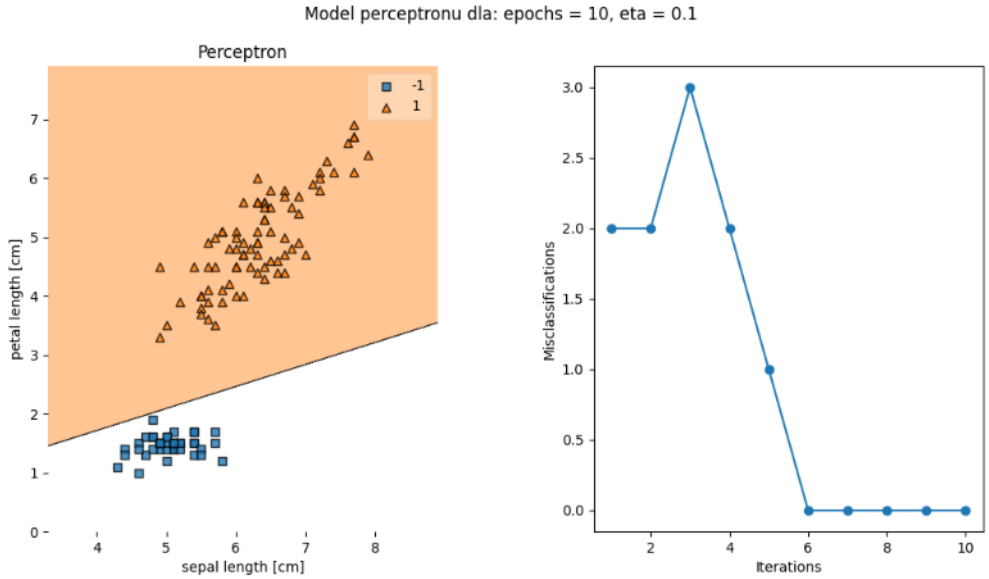
Dodatkowo dla modelu Adaline wygenerowany został wykres wartości soft output ($o_1(x)$) dla każdej próbki zbioru walidacyjnego.

Dla klasyfikatora 3 klas nie generowano wykresów, ponieważ klasyfikator 3 klas składa się z wyżej wspomnianych neuronów. Próbki o najbardziej pozytywnym wskazaniu są wyświetlane wraz z ilościami powtórzeń oraz prędkościami uczenia każdego z trzech neuronów.

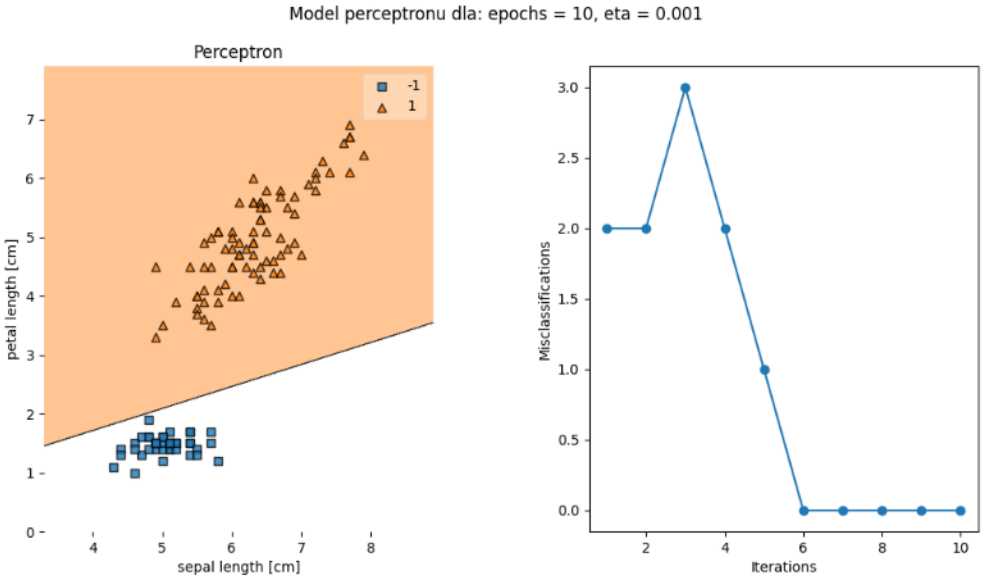
3.2. Wyniki

Wyniki zaprezentowano na poniższych rysunkach.

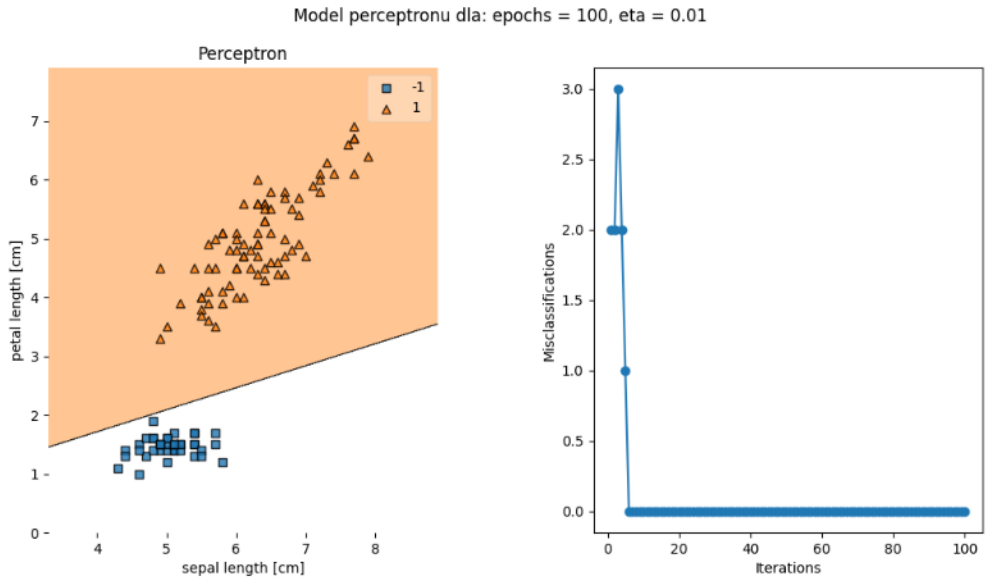
3.3. Perceptron - klasy liniowo separowalne



Rys. 3: Perceptron - klasyfikacja Iris Setosa.

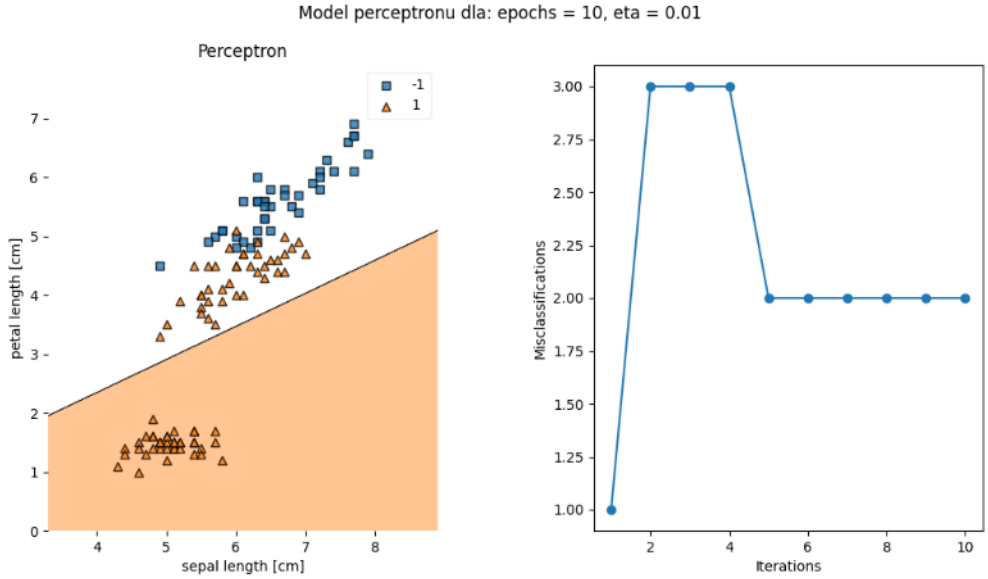


Rys. 4: Perceptron - klasyfikacja Iris Setosa - dla innej η.

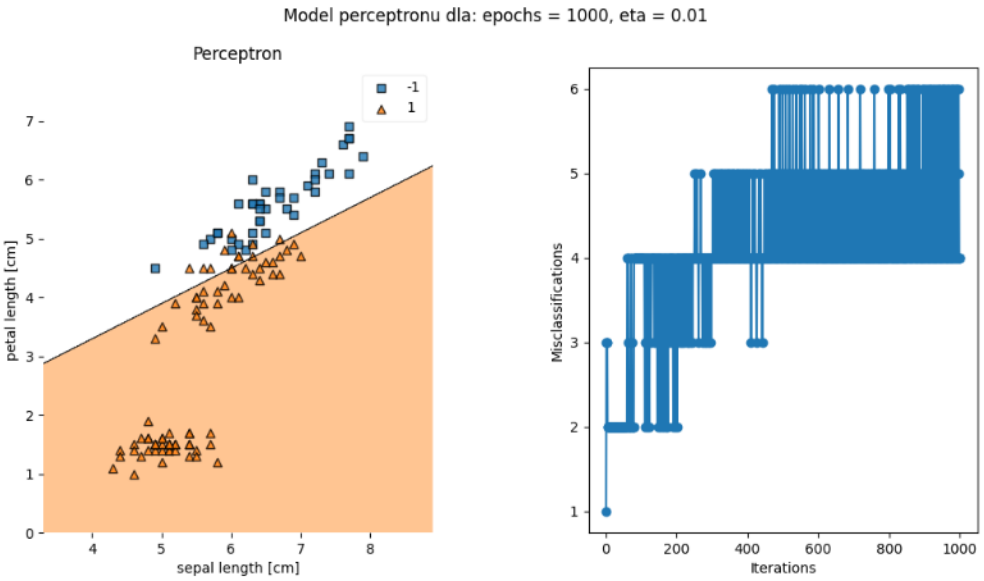


Rys. 5: Perceptron - klasyfikacja Iris Setosa - dla innej liczby epok.

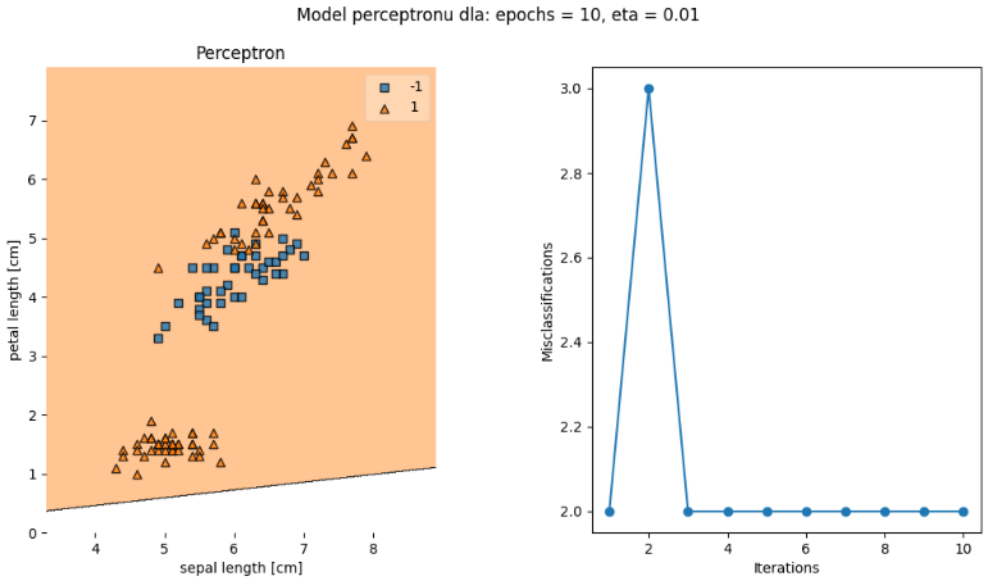
3.4. Perceptron - klasy nieliniowo separowalne



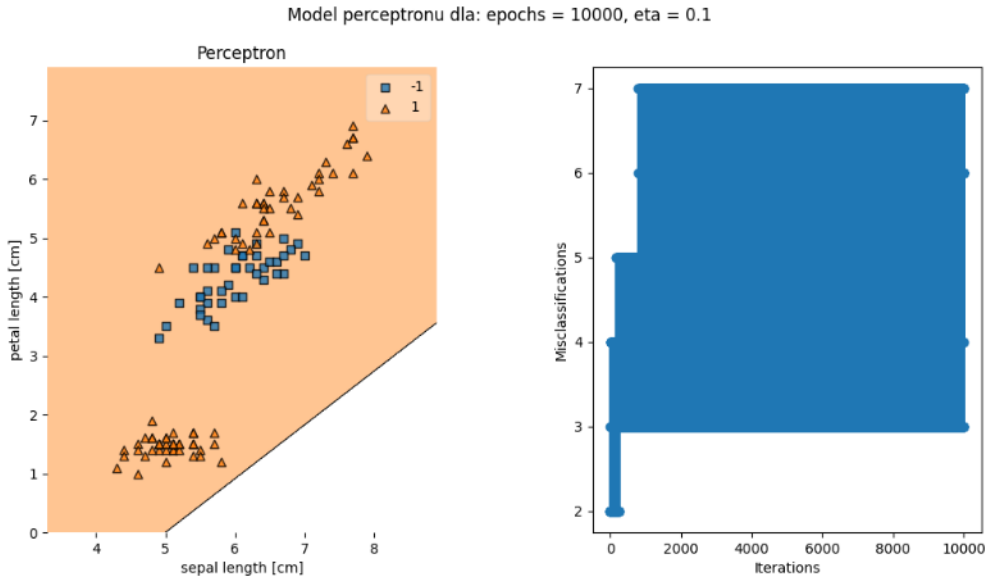
Rys. 6: Perceptron - klasyfkacja Iris Virginica.



Rys. 7: Perceptron - klasyfkacja Iris Virginica - większa liczba epok.

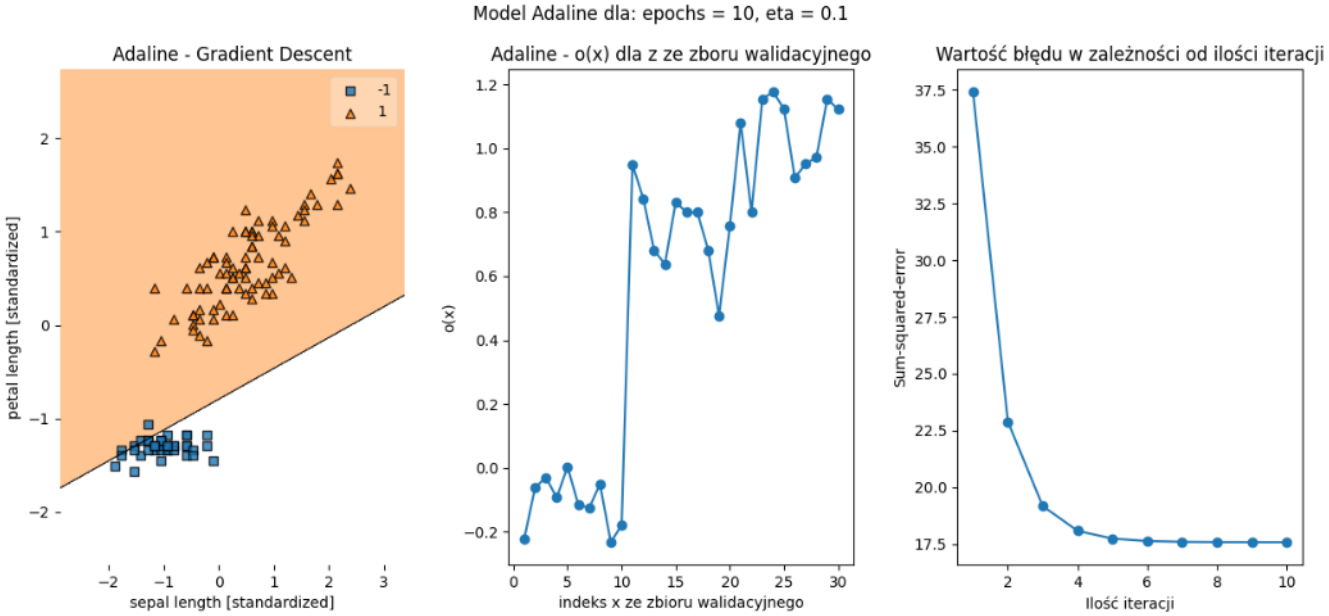


Rys. 8: Perceptron - klasyfkacja Iris Versicolor.

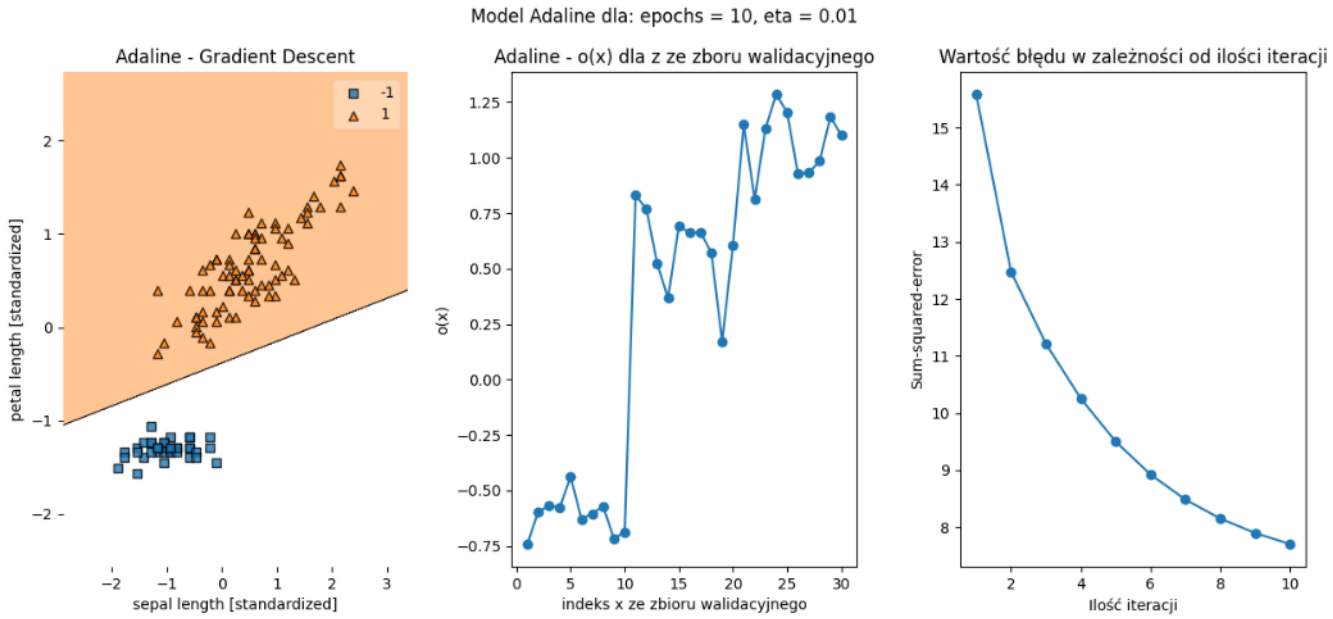


Rys. 9: Perceptron - klasyfkacja Iris Versicolor - większa liczba epok.

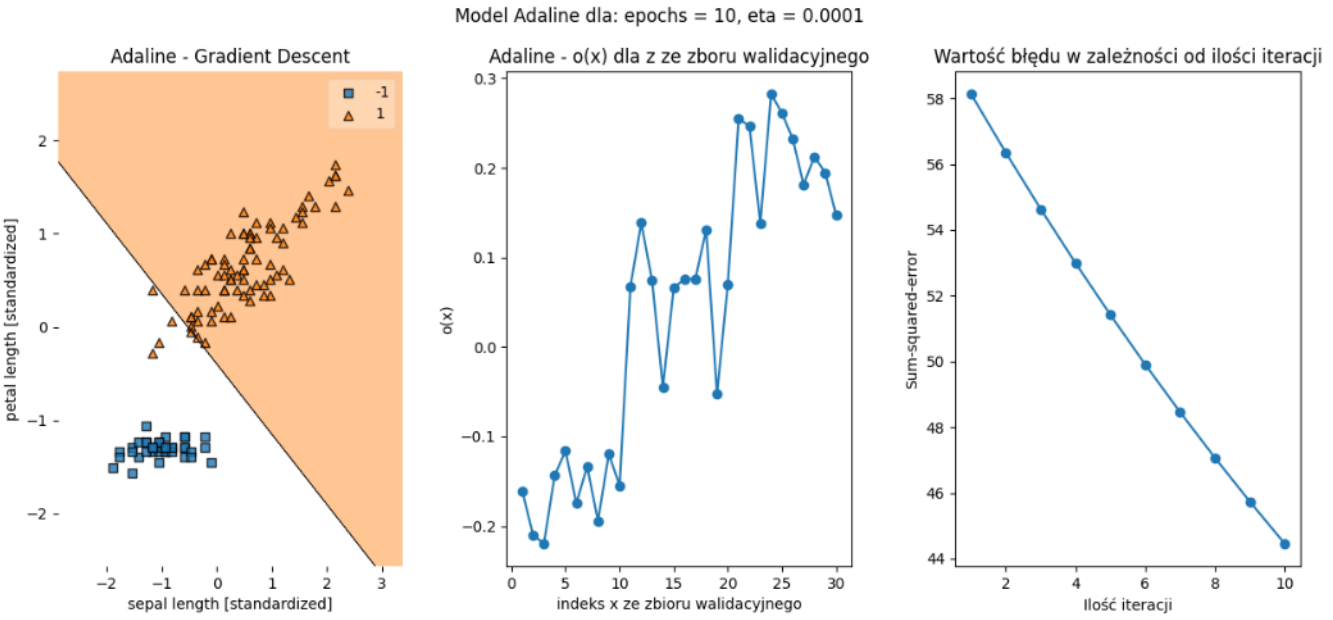
3.5. AdalineSGD - klasy liniowo separowalne



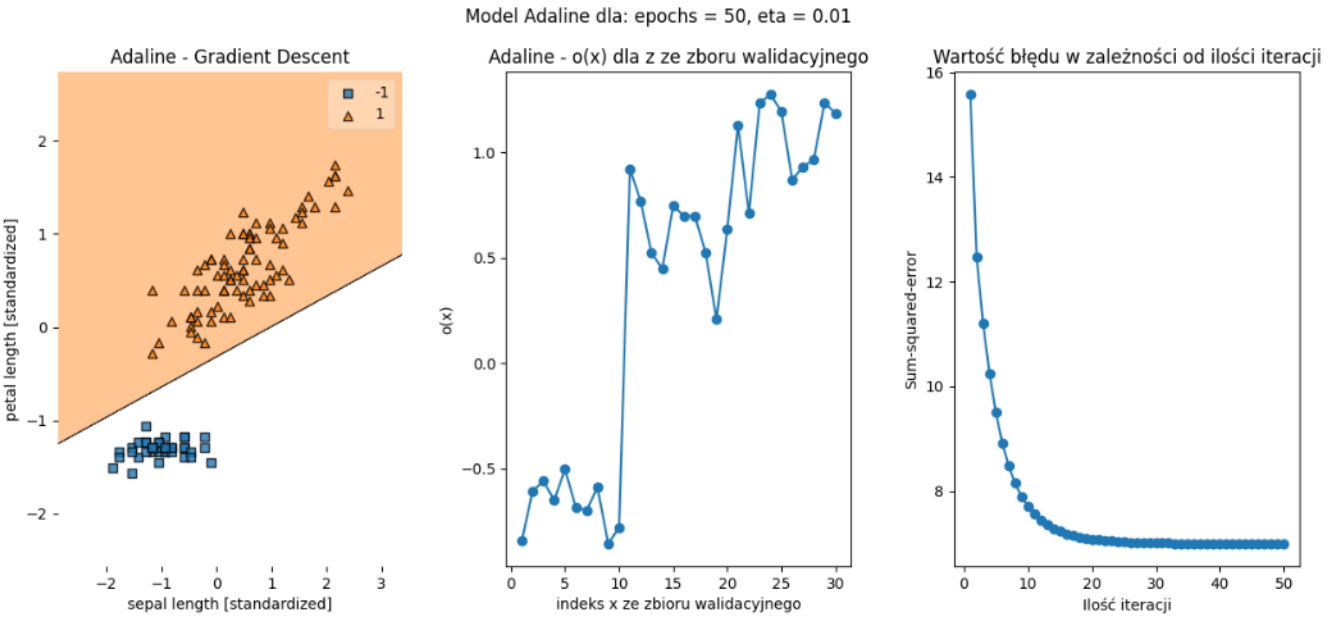
Rys. 10: Adaline - klasyfikacja Iris Setosa - za mała prędkość uczenia.



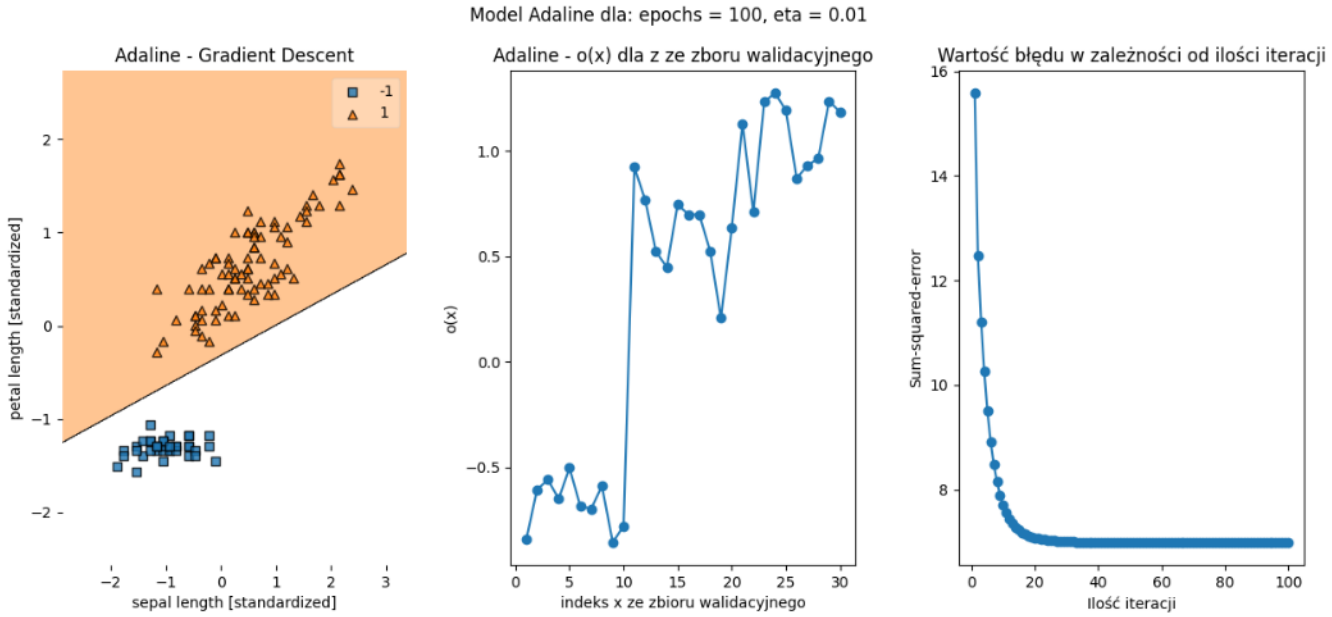
Rys. 11: Adaline - klasyfikacja Iris Setosa - większa predkość uczenia.



Rys. 12: Adaline - klasyfikacja Iris Setosa - za duża prędkość uczenia.

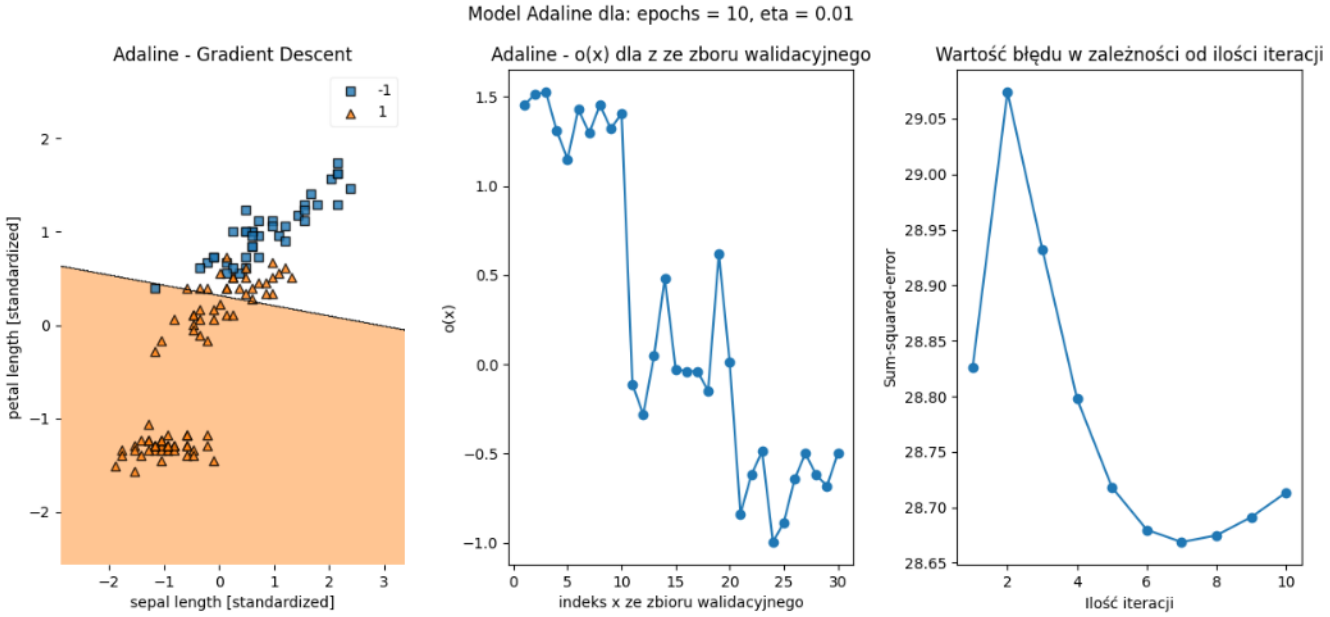


Rys. 13: Adaline - klasyfikacja Iris Setosa - większa ilość epok.

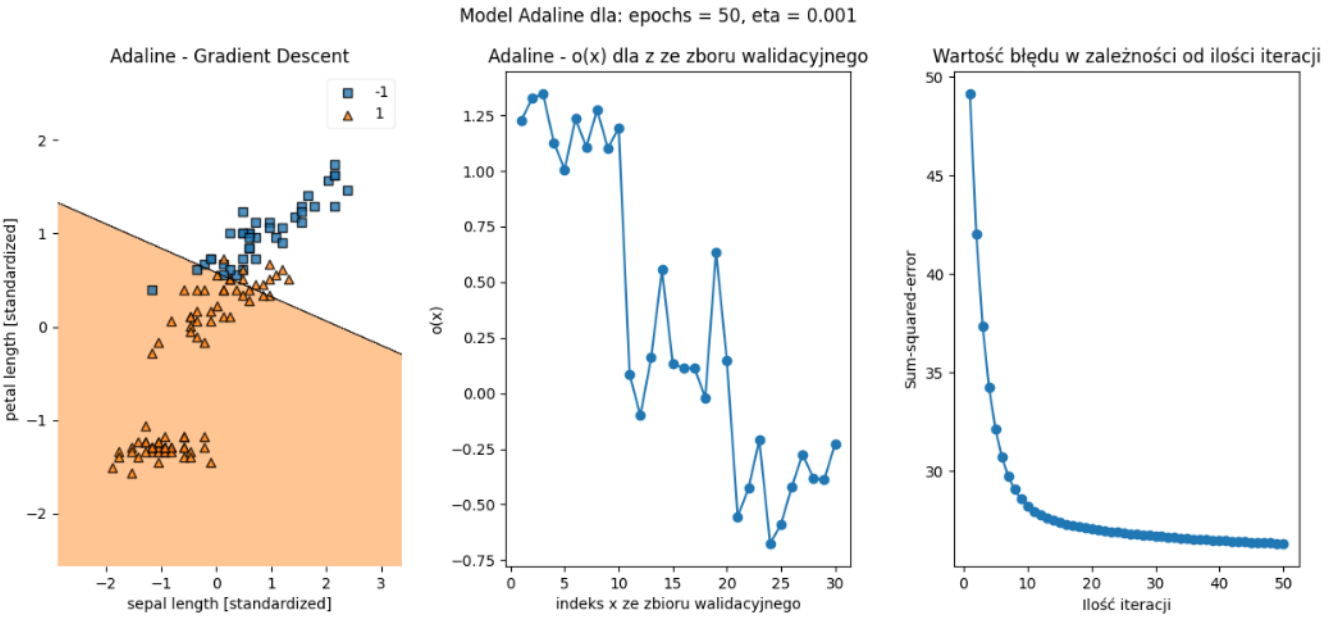


Rys. 14: Adaline - klasyfikacja Iris Setosa dla 100 epok.

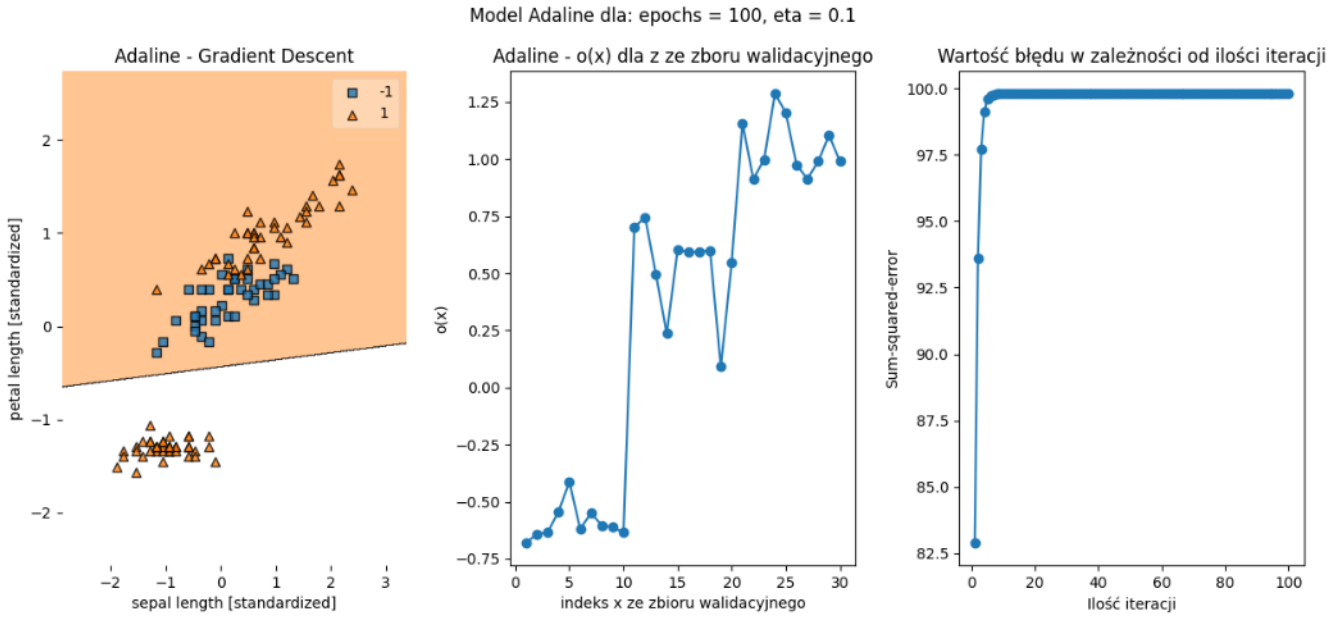
3.6. Adaline - klasy nieliniowo separowalne



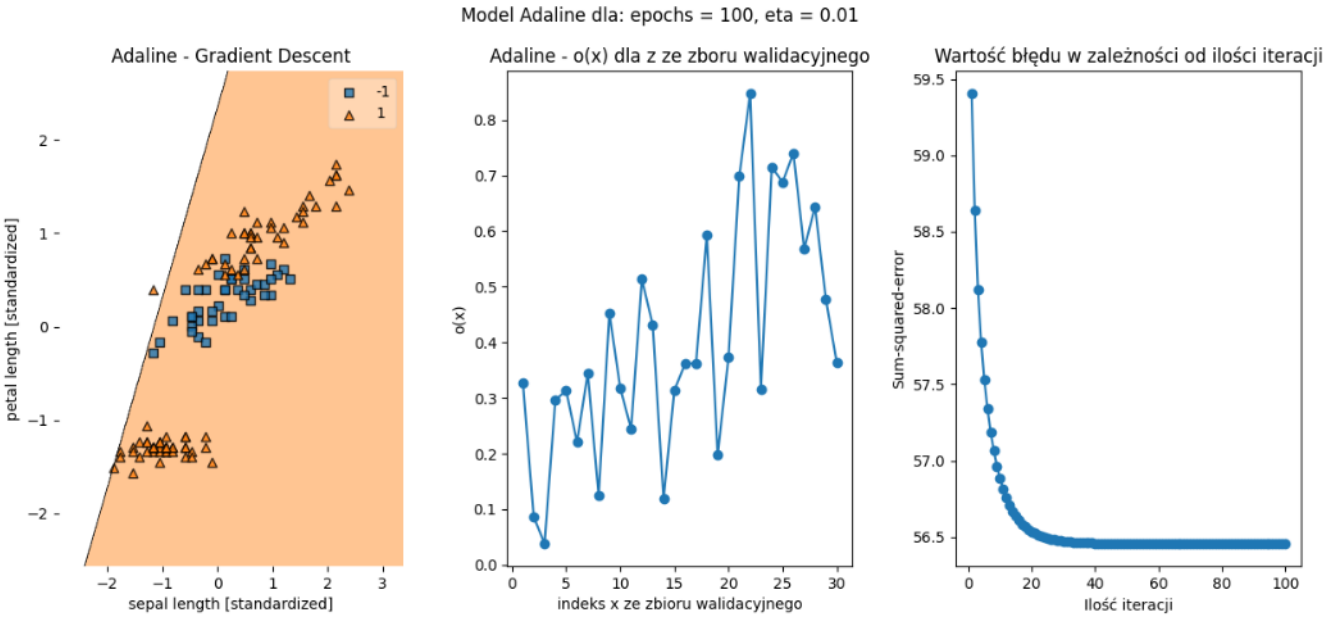
Rys. 15: Adaline - klasyfikacja Iris Virginica.



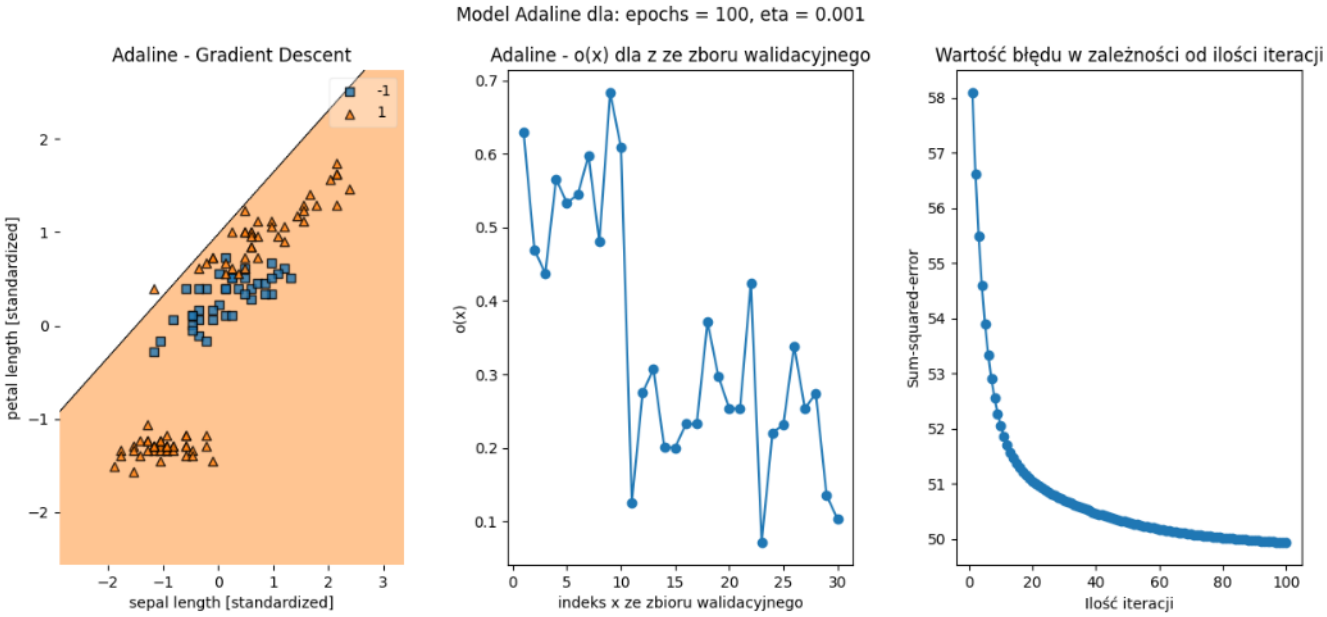
Rys. 16: Adaline - klasyfikacja Iris Virginica - większa liczba epok i większa prędkość uczenia.



Rys. 17: Adaline - klasyfikacja Iris Versicolor.



Rys. 18: Adaline - klasyfikacja Iris Versicolor - większa prędkość uczenia.



Rys. 19: Adaline - klasyfikacja Iris Versicolor - największa prędkość uczenia.

3.7. Klasyfikator 3 klas AdalineSGD

Poniżej przedstawiono najlepsze wyniki działania programu, jakie udało się uzyskać dla AdalineSGD.

```
Co przewidział klasyfikator
Epochs Setosa=  100 / Epochs Versicolor=  100 / Epochs Virginica=  100
Eta Setosa=  0.001 / Eta Versicolor=  0.1 / Eta Virginica=  0.0001
['Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set']
['-', 'Vir', '-', '-', '-', '-', '-', 'Vir', '-', '-']
['Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir']
```

Rys. 20: Wynik działania klasyfikatora dla 3 klas przy użyciu AdalineSGD.

3.8. Klasyfikator 3 klas AdalineGD

Poniżej przedstawiono najlepsze wyniki działania programu, jakie udało się uzyskać dla AdalineGD.

```
Co przewidział klasyfikator
Epochs Setosa=  100 / Epochs Versicolor=  10 / Epochs Virginica=  100
Eta Setosa=  0.001 / Eta Versicolor=  0.01 / Eta Virginica=  0.001
['Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set', 'Set']
['-', 'Vir', '-', 'Ver', '-', '-', '-', '-', 'Ver', '-']
['Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir', 'Vir']
```

Rys. 21: Wynik działania klasyfikatora dla 3 klas przy użyciu AdalineGD.

4. Wnioski

- Jak można zauważyć na wykresach 3 oraz 4, zmiana prędkości uczenia perceptronu dla stałej liczby epok, od pewnego progu ne wpływa na położenie granicy decyzyjnej, podobnie jak liczba epok. Dzieje się tak ponieważ perceptron przestaje aktualizować wektor wag, gdy wszystkie klasy zostaną poprawnie sklasyfikowane.

- Powyższa właściwość przekłada się również na mały margines granicy decyzyjnej. Jak widać na rys11, Adaline charakteryzuje się większym marginesem granicy decyzyjnej niż Perceptron.
- Jak widać na wykresie 3 Perceptron poradził sobie doskonale z klasyfikacją danych liniowo separowalnych - już po 6 iteracjach przy prędkości uczenia 0.1 algorytm sklasyfikował dane poprawnie.
- Zgodnie z przewidywaniami proste neurony nie radzą sobie dobrze z klasyfikacją danych liniowo nieseparowalnych. Wykrycie klasy Virginica jest utrudnione. W perceptronie, aby poprawnie wykryć tę klasę należy zwiększyć liczbę epok, rezultaty widać na rysunkach 6 oraz 7. Jak widać, w najlepszym przypadku neuron poza próbkami poprawnymi, do klasy Virginica zaliczył kilka próbek Versicolor.

  Wynik ostatecznej klasyfikacji mógłby zostać poprawiony gdyby pod uwagę wzięte zostały inne parametry kwiatów, jak szerokość płatka i szerokość kielicha kwiatu.

  Podobnie dla reguły AdalineSGD na rysunkach 15 i 16 przedstawiono próby oddzielenia dwóch liniowo nieseparowalnych klas. AdalineSGD radzi sobie gorzej niż Perceptron. Poza sklasyfikowaniu błędnych próbek jako poprawne, dodatkowo klasyfikuje próbki poprawne, jako błędne.
- Zarówno perceptron, jak i AdalineSGD nie są w stanie sklasyfikować poprawnie danych Versicolor, ponieważ wartości tych danych znajdują się pomiędzy wartościami dwóch pozostałych klas - rysunki 8 i 9 dla perceptronu oraz rysunki 18 i 19 dla AdalineSGD.
- Rysunki 10, 11 i 12 przedstawiają wyniki klasyfikacji klasy Iris Setosa odseparowanej liniowo od pozostałych. Na ich podstawie można zademonstrować wpływ prędkości uczenia na jakość wyników:
  - Jak widać, w przeciwieństwie do Perceptronu, 10 epok nie wystarczy, by nauczony neuron poprawnie sklasyfikował dane przy prędkości uczenia $\eta = 0.1$. Prędkość uczenia jest zbyt mała w porównaniu do ilości epok - algorytm uczy się ze zbioru treningowego, lecz robi to niezbyt dokładnie - wartość błędu ustala się na stałej, lecz dużej wartości.
  - Przy prędkości uczenia $\eta = 0.01$ AdalineSGD klasyfikuje tak samo poprawnie jak perceptron, zachowując przy tym lepszy margines granicy decyzyjnej. Jeśli teraz zwiększy się ilość epok, jak na rysunkach 13 i 14, widać że wykres błędu algorytmu zbiega do stałej lecz bardzo małej wartości.
  - Gdy prędkość uczenia jest zbyt duża, jak na rysunku 12, algorytm nie zdąży nauczyć się poprawnie rozpoznawać próbek - wykres błędu nie jest zbieżny.
- Jak widać na rysunku 20, zaprojektowany klasyfikator, składający się z 3 neuronów AdalineSGD zaklasyfikował poprawnie dwie skrajne klasy - Setosa i Virginica, natomiast klasy Versicolor nie był w stanie zaklasyfikować, co więcej, zaklasyfikował dwa kwiaty błędnie.

  Sprawdzono również działanie tak samo funkcjonującego klasyfikatora, wykorzystującego model AdalineGD. Jak widać na rysunku 21, algorytm poprawnie sklasyfikował 2 skrajne klasy, a dodatkowo wykrył poprawnie 2 próbki klasy środkowej - Versicolor.
- W obu klasyfikatorach dla 3 klas, zarówno SGD jak i GD, przed wybraniem wyniku jako najbardziej pozytywnego wskazania, można było zauważyć, że neuron odpowiadający za wykrywanie klasy środkowej - Versicolor, jako swoją klasę charakteryzował również próbki Iris Setosa. Wykrycie to było jednak obciążone bardzo dużym błędem. Dlatego jako wynik ostateczny, do klasyfikacji pierwszej grupy został użyty neuron rozpoznający Setosy, gdyż jego wskazania jak wspomniano wcześniej, po 6 iteracjach były bezbłędne.