

---

# ĆWICZENIE 1

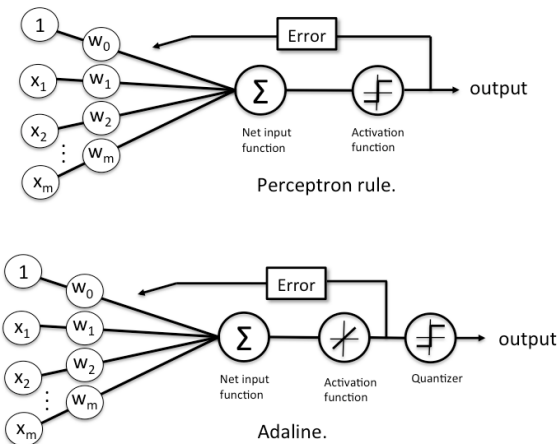
## Perceptron prosty i Adaline

---

Joanna Chlebus  
242505  
Październik 2019

### 1 Plan eksperymentów

Celem przeprowadzonych eksperymentów było poznanie podstawowych metod uczenia wykonywanych przez pojedynczy neuron, obserwacja jego zachowania przy różnych funkcjach przejścia oraz określenie hiperparametrów mających wpływ na szybkość uczenia. W ćwiczeniu rozpatrywano i porównano dwa modele neuronu - **perceptron prosty** oraz **Adaline**.



Rysunek 1: Modele perceptronu prostego i Adaline

Proces uczenia neuronów prostych polega na aktualizacji wag, które początkowo należy zainicjalizować losowymi wartościami z przyjętego przedziału. Wzór na aktualizację wag dla:

a) perceptronu prostego:

$$w(t+1) = w(t) + \alpha x_i (y^d - y), \quad (1)$$

b) Adaline:

$$w(t+1) = w(t) + 2\mu \epsilon_i x_i = w(t) + 2\mu (y^d - y) x_i \quad (2)$$

gdzie  $w(t+1)$  - wartość wag w kolejnej iteracji,  $w$  - wartość wag w aktualnym kroku,  
 $\alpha, \mu$  - współczynnik uczenia,  $x_i$  - i-ty wektor sygnałów wejściowych,  
 $y^d$  - zadany sygnał wyjściowy,  $y$  - predykowane przez neuron wyjście,  
 $\epsilon$  - błąd występujący po predykcji i-tego wektora.

Do wykonania eksperymentów zaimplementowano program, który symulował działanie neuronu w obu powyższych wersjach realizujących **logiczną funkcję AND**.

Wpływ poszczególnych hiperparametrów na szybkość uczenia neuronu zbadano poprzez powtarzalne wywołanie procedury uczenia na zbiorze treningowym. Porównanie dwóch różnych funkcji przejścia (progowej unipolarnej i progowej bipolarnej) przeprowadzono dla perceptronu prostego, w przypadku Adaline użyto wyłącznie funkcji bipolarnej. Szybkość uczenia neuronu określono w liczbie epok, po których proces uczenia osiąga warunek stopu:

- dla perceptronu prostego - brak aktualizacji wag po epoce, tzn. kiedy predykcja dla wszystkich wektorów sygnałów wejściowych będzie zgodna z zadanymi sygnałami wyjściowymi;
- dla Adaline - błąd średniokwadratowy między zadanym wektorem wyjść a całym pobudzeniem neuronu dla wektora wejść mniejszy od przyjętej wartości.

## 2 Opis aplikacji wykorzystywanej do badań

Symulację zachowania neuronów prostych zaimplementowano w języku Python w środowisku PyCharm. Struktura projektu przedstawia się następująco:

```
perceptron
├── data
│   ├── bipolar_train.json
│   ├── bipolar_ext_train.json
│   ├── unipolar_train.json
│   └── unipolar_ext_train.json
├── model
│   └── perceptron.py
├── notebooks
│   └── experiments.ipynb
├── experiments.py
└── utils.py
```

W pliku `model/perceptron.py` zaimplementowano strukturę obu modeli neuronów, wykorzystując mechanizm dziedziczenia ze względu na wiele podobieństw między perceptronem prostym a Adaline. Obiekty neuronów tworzy się poprzez podanie następujących parametrów do ich konstruktorów:

- a) `input_size` - wymiar danych wejściowych;
- b) `activation` - rodzaj funkcji przejścia (dopuszczalne wartości: `'bipolar'` lub `'unipolar'`);
- c) `use_bias` - flaga mówiąca o użyciu bloku przesunięcia *bias*, w obu modelach neuronów ustawiony na `True`;
- d) `theta` - wartość progowa do funkcji przejścia, przyjmuje wartość 0;
- e) `error_threshold` - przyjęty maksymalny dopuszczalny błąd dla modelu Adaline.

Główną funkcją wywoływaną dla klasy perceptronu jest metoda **train**:

```
perceptron.train(x_train, y_train, alpha_lr, epochs),
```

gdzie `x_train` - wektor sygnałów wejściowych, `y_train` - wektor zadanych sygnałów wyjściowych, `alpha_lr` - współczynnik uczenia, `epochs` - maksymalna liczba epok nauki w przypadku nieosiągnięcia warunku stopu procedury uczącej do tej liczby epok.

Metody do przeprowadzania eksperymentów znajdują się w pliku `experiments.py`. Wszelkie funkcje pomocnicze umieszczono w pliku `utils.py`, w którym można znaleźć m.in. metodę do ładowania danych wejściowych ze słownika oraz metody do rysowania wykresów. W folderze `data` zapisano dane treningowe w formacie `.json` odpowiednio według nazwy dla logiki unipolarnej oraz bipolarnej.

Prezentacja całej procedury eksperymentalnej znajduje się w notebook'u `notebooks/experiments.ipynb`, gdzie wywołano metody eksperymentalne oraz przeprowadzono wizualizację ich wyników na wykresach.

### 3 Charakterystyka zbiorów danych użytych do badań

W ćwiczeniu użyto zbiorów danych treningowych dla logiki unipolarnej oraz bipolarnej.

Tabela 1: Zbiór danych dla logiki unipolarnej

x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 2: Zbiór danych dla logiki bipolarnej

x1	x2	y
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

Do uczenia modelu perceptronu prostego użyto obu zbiorów, natomiast model Adaline rozpatrywano tylko dla funkcji bipolarnej. W niektórych eksperymentach użyto również rozszerzonego podstawowego datasetu o kilka dodatkowych przykładów, w których podano przybliżone wartości sygnałów wejściowych odbiegające maksymalnie o 0.1 od wartości logicznej prawdy bądź fałszu dla każdej z logik.

### 4 Wpływ wartości hiperparametrów na szybkość uczenia perceptronu prostego

#### 4.1 Eksperyment 1. Zbadanie szybkości uczenia w zależności od losowanych wag

*Założenia:*

Stałe parametry uczenia:

wsp. uczenia = 0.05

funkcje przejścia - unipolarna, bipolarna

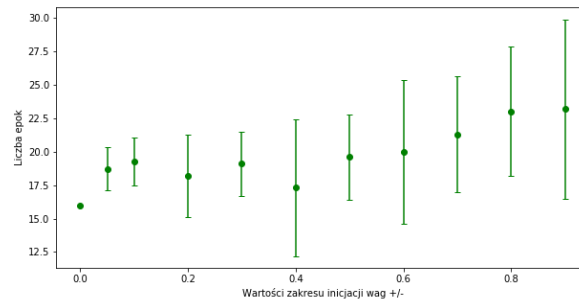
zbiór danych - podstawowy odpowiednio dla f. unipolarnej i bipolarnej

Zmianom podlegał zakres dolny i górny wartości inicjacji wag do procesu uczenia, które na podstawie podanych wartości skrajnych były losowane z rozkładu jednostajnego ciągłego. Rozpoczęto eksperyment dla wartości z przedziału  $<-1, 1>$ , a następnie w sposób krokowy zawężano przedział losowania poprzez zmniejszanie wartości odpowiadającej za logiczną prawdę oraz zwiększanie wartości logicznego fałszu co 0.1 aż do przedziału  $<-0.1, 0.1>$ . Następnie analogicznie zmniejszano dalej przedział o wartości co  $\pm 0.05$  do osiągnięcia  $<0.0, 0.0>$ .

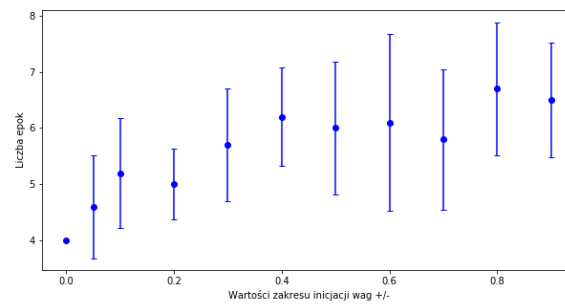
*Przebieg eksperymentu:*

Badano wpływ wartości inicjujących wag na szybkość uczenia neuronu prostego. Wywoływano procedurę uczącą perceptronu prostego z unipolarną i bipolarną funkcją aktywacji dla wybranych zakresów wag początkowych. Za każdym razem uczono neuron aż do osiągnięcia warunku stopu, a szybkość uczenia mierzono liczbą epok potrzebną do jego uzyskania. Eksperyment dla każdego przedziału wag powtórzono 10, 50 oraz 100 razy, a uzyskane wyniki (liczba epok) uśredniono i wyliczono z nich odchylenie standardowe.

Wyniki:

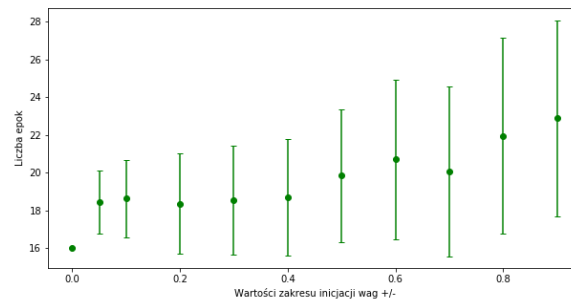


(a) logika unipolarna

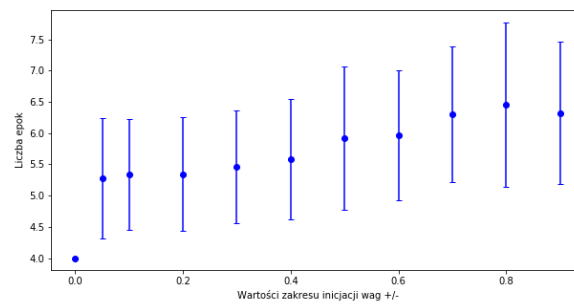


(b) logika bipolarna

Rysunek 2: Liczba epok uczenia w zależności od zakresu początkowych wag po 10 powtórzeniach

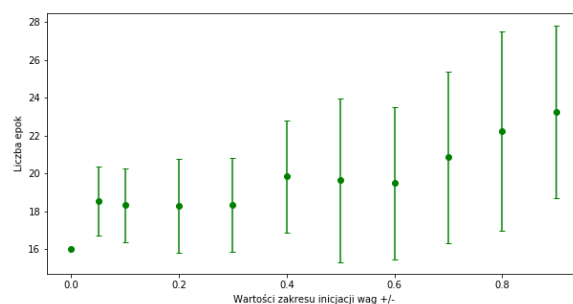


(a) logika unipolarna

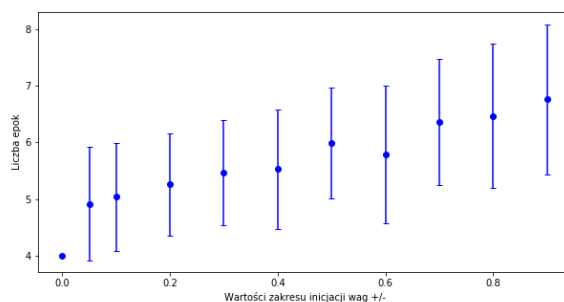


(b) logika bipolarna

Rysunek 3: Liczba epok uczenia w zależności od zakresu początkowych wag po 50 powtórzeniach



(a) logika unipolarna



(b) logika bipolarna

Rysunek 4: Liczba epok uczenia w zależności od zakresu początkowych wag po 100 powtórzeniach

*Komentarz:*

Z wykresów zaobserwowano, że im mniejszy był zakres losowania wag początkowych, tym perceptron uczył się szybciej przy zachowaniu stałego współczynnika uczenia. Wraz ze wzrostem przedziału wartości początkowych wag rosło także odchylenie standardowe od średniej liczby epok potrzebnej do osiągnięcia warunku stopu. Można więc wywnioskować, że optymalny zakres inicjacji wag dla rozwiązywanego problemu jest bardzo mały - bliski wartości 0. Odnosi się to zarówno do perceptronu z unipolarną jak i bipolarną funkcją przejścia. Należy też wspomnieć, że wartości optymalne wag są uzależnione od rodzaju rozwiązywanego przez neuron problemu - dla różnych problemów mogą się różnić.

Im większe były wartości skrajne, tym początkowe wartości wag mogły bardziej odbiegać od wartości optymalnych. Tym samym potrzebna była większa liczba iteracji procedury uczącej, aby uzyskać warunek stopu i uaktualnić wartości wag do odpowiednich. Wzrost odchylenia standardowego przy większym przedziale inicjacji wag świadczy również o tym, że wyuczone wartości wag (przy tym samym współczynniku uczenia) były bardziej losowe niż w przypadku mniejszych skrajnych wartości inicjujących wag.

## 4.2 Eksperyment 2. Określenie wpływu wartości współczynnika uczenia na szybkość uczenia

*Założenia:*

Stałe parametry uczenia:

zakres inicjacji wag =  $[-0.1, 0.1]$

funkcje przejścia - unipolarna, bipolarna

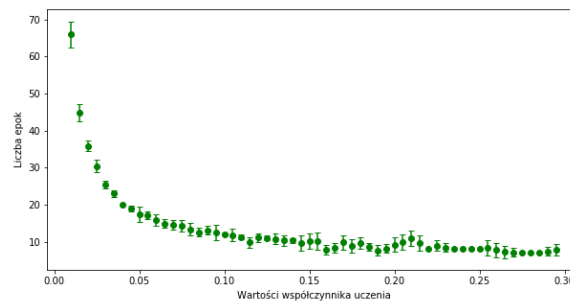
zbiór danych - podstawowy odpowiednio dla f. unipolarnej i bipolarnej

Zmianom podlegała wartość współczynnika uczenia. Przyjęto wartości tego hiperparametru z przedziału  $[0.01, 0.3]$  z odstępem co 0.005.

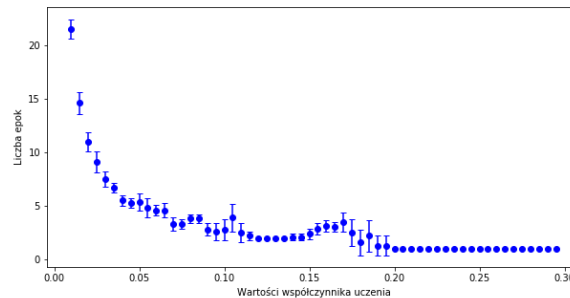
*Przebieg eksperymentu:*

Zbadano wpływ różnych wartości współczynnika uczenia na szybkość uczenia perceptronu prostego, której miarą była liczba epok procedury uczącej neuronu. Eksperyment dla każdej wartości badanego hiperparametru powtórzono 10, 50 oraz 100 razy, za każdym razem trenując neuron do osiągnięcia warunku stopu. Wyniki uśredniono oraz wyliczono z nich odchylenie standardowe.

*Wyniki:*

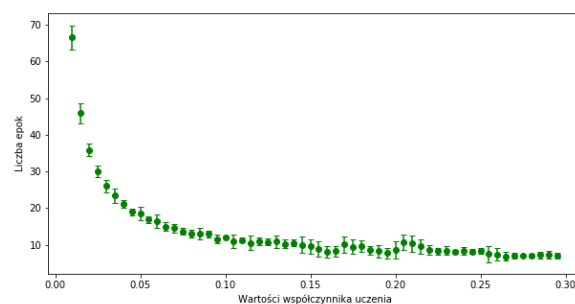


(a) logika unipolarna

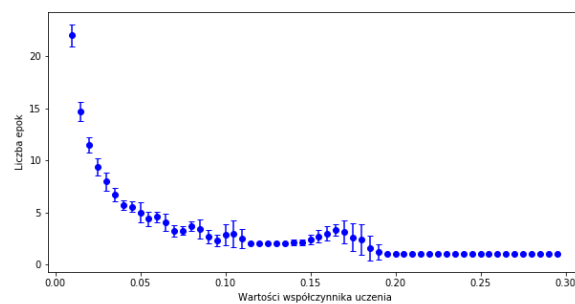


(b) logika bipolarna

Rysunek 5: Liczba epok uczenia w zależności od zakresu początkowych wag po 10 powtórzeniach

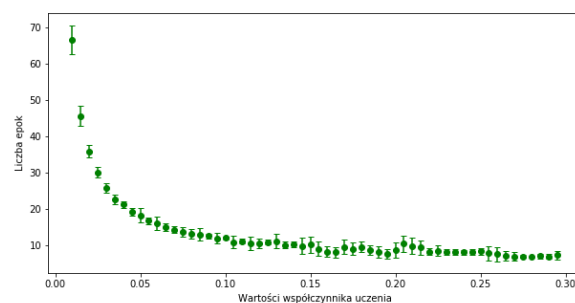


(a) logika unipolarna

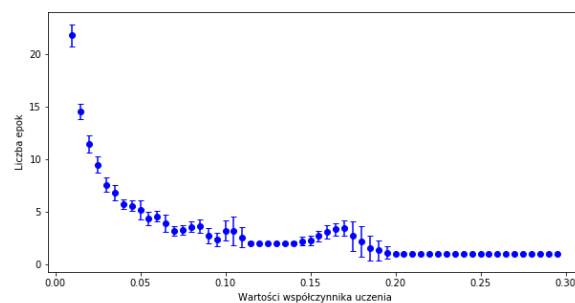


(b) logika bipolarna

Rysunek 6: Liczba epok uczenia w zależności od zakresu początkowych wag po 50 powtórzeniach



(a) logika unipolarna



(b) logika bipolarna

Rysunek 7: Liczba epok uczenia w zależności od zakresu początkowych wag po 100 powtórzeniach

*Komentarz:*

Analizując wyniki przedstawione na wykresie, można stwierdzić, że przy stałym zakresie losowania wag początkowych wzrost wartości współczynnika uczenia powoduje przyspieszenie procesu uczenia perceptronu. Średnia liczba epok potrzebna do osiągnięcia warunku stopu procedury uczącej przy zadanej większej wartości tego hiperparametru jest mniejsza niż przy mniejszej jego wartości - regułę tę można zauważyć dla obu funkcji przejścia (unipolarnej i bipolarnej). Jest to spowodowane tym, że zainicjalizowane na początku uczenia wagi aktualizowane są o większą wartość z każdą kolejną iteracją. Skutkuje to szybszym zbieganiem ich wartości do optymalnych, a tym samym mniejszą liczbą epok wymaganą do wyuczenia neuronu rozwiązywania zadanych problemów.

Powyżej określonej wartości współczynnika  $\alpha$  szybkość uczenia perceptronu utrzymuje się na stałym poziomie - dalszy wzrost wartości hiperparametru  $\alpha$  nie powoduje zmiany w liczbie epok wymaganych do uzyskania warunku stopu uczenia.

#### 4.3 Eksperyment 3. Zbadanie szybkości uczenia perceptronu prostego w zależności od funkcji aktywacji

*Założenia:*

- a) dla **Rysunku 8** - jak w **Eksperymentcie 1.** dla perceptronu prostego
- b) dla **Rysunku 9** - jak w **Eksperymentcie 2.** dla perceptronu prostego

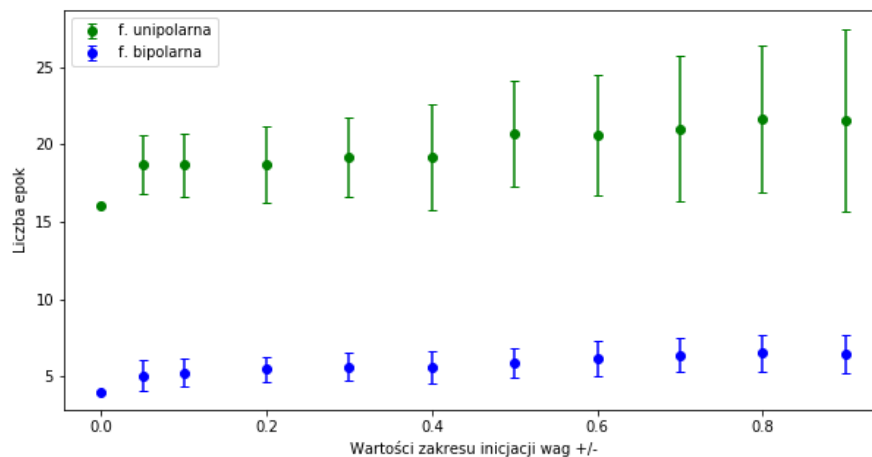
*Przebieg eksperymentu:*

Zbadano szybkość uczenia neuronu prostego przy różnych funkcjach przejścia, unipolarnej i bipolarnej, dla zmiennego przedziału losowanych wag początkowych (sposób przebiegu eksperymentu analogiczny do **Eksperymentu 1.**) przy zachowaniu pozostałych stałych parametrów.

Różnice w szybkości uczenia dla funkcji przejść porównano także dla zmiennej wartości współczynnika uczenia (sposób przebiegu eksperymentu analogiczny o **Eksperymentu 2.**), pozostałe parametry uczenia pozostały stałe.

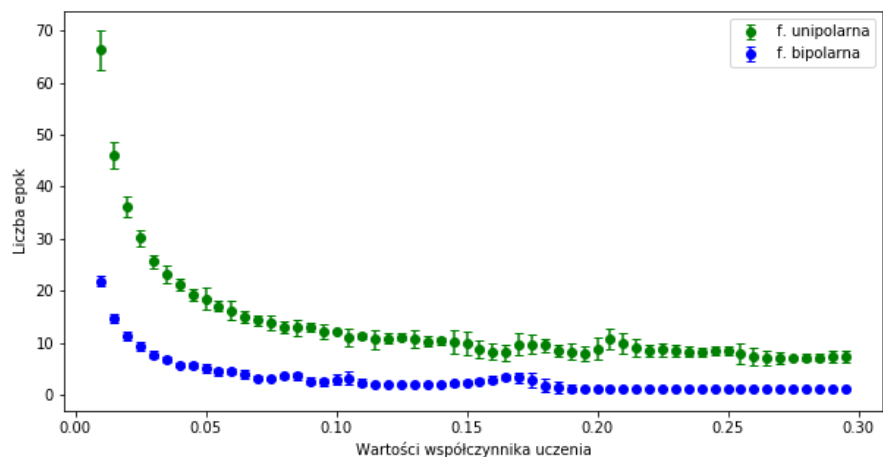
Każdy eksperyment powtórzono 100-krotnie aż do uzyskania warunku stopu przez procedurę uczącą. Wyniki zestawiono na wspólnych wykresach w celu łatwiejszego zobrazowania zależności szybkości uczenia od funkcji aktywacji perceptronu.

*Wyniki:*



Rysunek 8: Liczba epok uczenia w zależności od różnych zakresów losowania początkowych wag dla dwóch funkcji





Rysunek 9: Liczba epok uczenia w zależności od wartości hiperparametru  $\alpha$  dla dwóch różnych funkcji przejścia

#### Komentarz:

Na podstawie wyników przedstawionych na wykresach widać, że perceptron prosty uczy się szybciej przy bipolarnej funkcji przejścia niż przy unipolarnej, niezależnie czy zmieniamy zakres skrajnych wartości losowania wag początkowych czy wartość współczynnika uczenia. Powodem jest tutaj inna wartość wyliczanego błędu do korekcji wag w każdej iteracji procedury uczącej. Dla funkcji unipolarnej wartości błędu mogą wynieść -1, 0 lub 1, natomiast dla funkcji bipolarnej -2, 0, 2.

W przypadku złej predykcji neuronu w stosunku do zadanej jej prawdziwej wartości aktualizacja wag dla bipolarnej funkcji aktywacji będzie większa ze względu na większą wartość bezwzględną błędu w porównaniu z błędem uzyskanym przy funkcji unipolarnej. Wagi będą więc szybciej zbiegały w kierunku swoich wartości optymalnych, przez co zmniejszy się liczba epok wymagana do uzyskania warunku stopu uczenia.

## 5 Wpływ wartości hiperparametrów na szybkość uczenia perceptronu Adaline

### 5.1 Eksperyment 4. Zbadanie szybkości uczenia od przedziału losowanych wag

#### Założenia:

Parametry stałe uczenia:

wsp. uczenia  $\mu = 0.05$

dopuszczalny błąd = 0.26

logika - bipolarna

zbiór danych - podstawowy dla funkcji bipolarnej

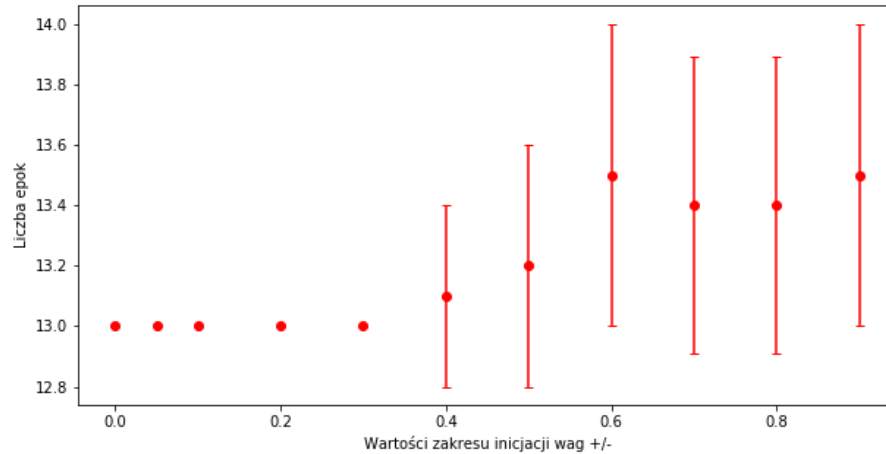
Dopuszczalny maksymalny błąd przyjęto na poziomie 0.26, czyli nieznacznie więcej niż minimalny błąd średniokwadratowy możliwy do uzyskania dla problemu logicznej funkcji AND (ok. 0.25).

Zmianom podlegał zakres dolny i górny wartości inicjacji wag do procesu uczenia, które na podstawie podanych wartości skrajnych były losowane z rozkładu jednostajnego ciągłego. Rozpoczęto eksperyment dla wartości z przedziału  $\langle -1, 1 \rangle$ , a następnie w sposób krokowy zawężano przedział losowania poprzez zmniejszanie wartości odpowiadającej za logiczną prawdę oraz zwiększanie wartości logicznego fałszu co 0.1 aż do przedziału  $\langle -0.1, 0.1 \rangle$ . Następnie analogicznie zmniejszano dalej przedział o wartości co  $\pm 0.05$  do osiągnięcia  $\langle 0.0, 0.0 \rangle$ . Rozpatrzono również przedziały inicjacji wag od  $\langle -1, 1 \rangle$  do  $\langle -4, 4 \rangle$  poprzez zmianę wartości skrajnych co 0.1.

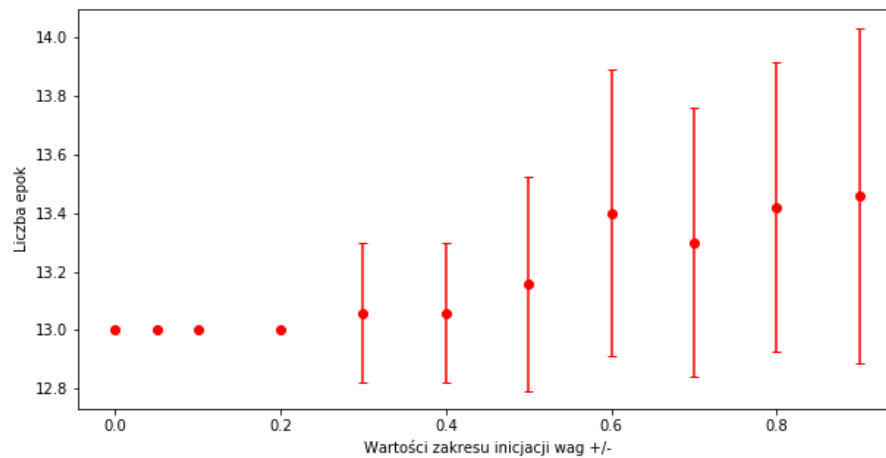
### Przebieg eksperymentu:

Badano wpływ zakresu losowania wag początkowych na szybkość uczenia neuronu Adaline, której miarą była liczba epok potrzebna do osiągnięcia warunku stopu procedury uczącej. Wywoływano metodę odpowiadającą za uczenie określoną liczbę razy dla każdego przedziału losowanych wag. Eksperyment dla każdego zakresu powtórzono 10, 50 oraz 100 razy, wyniki uśredniono i wyliczono z nich odchylenie standardowe.

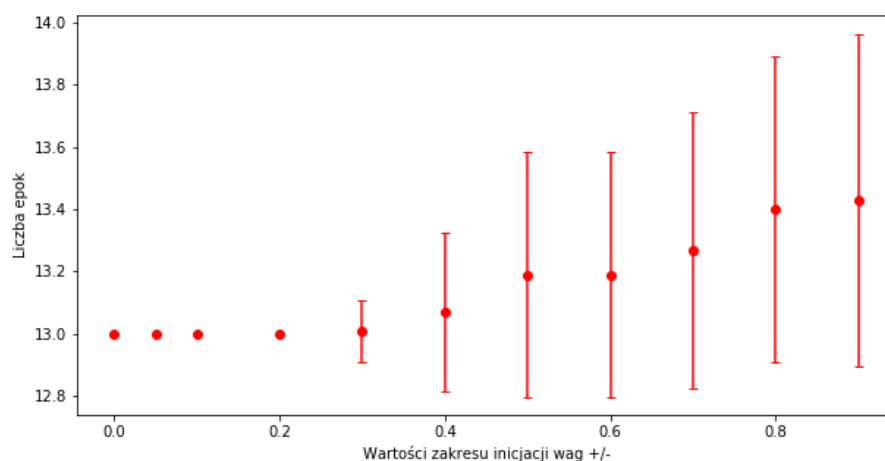
### Wyniki:



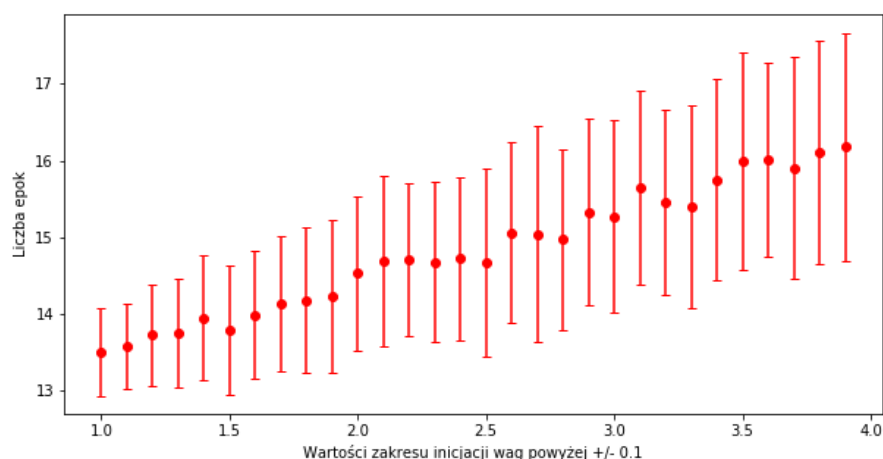
Rysunek 10: Liczba epok uczenia w zależności od wartości skrajnych losowania wag +/- poniżej 1.0 dla 10 powtórzeń



Rysunek 11: Liczba epok uczenia w zależności od wartości skrajnych losowania wag +/- poniżej 1.0 dla 50 powtórzeń



Rysunek 12: Liczba epok uczenia w zależności od wartości skrajnych losowania wag +/- poniżej 1.0 dla 100 powtórzeń



Rysunek 13: Liczba epok uczenia w zależności od wartości skrajnych losowania wag powyżej +/- 1.0 dla 100 powtórzeń

#### Komentarz:

Można zauważyć, że analogicznie jak w przypadku perceptronu prostego wraz ze zwiększaniem się przedziału losowanych wag zwiększa się uśredniona liczba epok potrzebnych do zatrzymania się uczenia Adaline (osiągnięcie błędu mniejszego bądź równego dopuszczalnemu). Wzrost ten jednak jest bardzo mały, po zaokrągleniu do części całkowitych otrzymalibyśmy praktycznie stałą liczbę epok dla każdego badanego w eksperymencie zakresu wag mieszczącego się w przedziale  $<-1.0, 1.0>$ . Większą różnicę w liczbie epok oraz w odchyleniu standardowym od jej średniej dla 100 powtórzeń procedury zaobserwowano dopiero dla zakresów losowych wag powyżej przedziału  $<-1.0, 1.0>$ .

Wartość inicjalizacji wag w przypadku Adaline nie ma tak dużego wpływu na wzrost odchylenia standardowego śr. liczby epok uczenia (odchylenie zaokrąglone do wartości całkowitych to 0 lub 1 epoka), jak to się działo w przypadku perceptronu prostego. W tym modelu aktualizujemy wagi o wartości rzeczywiste - różnicę między zadaniem prawdziwym wyjściem a całym pobudzeniem neuronu (liniowa funkcja aktywacji, model ciągły), w przeciwieństwie do perceptronu prostego, gdzie wykonywaliśmy kroki o stałe całkowite wartości (model dyskretny). Wartość błędu jest więc lepiej dopasowana do rzeczywistego wyjścia sieci.

## 5.2 Eksperyment 5. Zbadanie szybkości uczenia w zależności od wartości współczynnika uczenia

*Założenia:*

Parametry stałe uczenia:

zakres losowych wag początkowych:  $<-0.1, 0.1>$

logika - bipolarna

zbiór danych - podstawowy dla funkcji bipolarnej

dopuszczalny błąd = 0.26

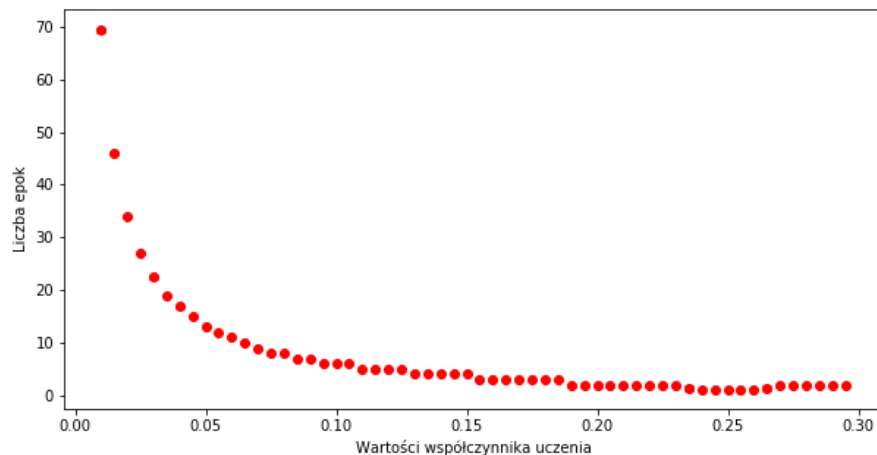
Zmianie podlegała wartość współczynnika uczenia  $\mu$ . Przyjęto wartości tego hiperparametru z przedziału  $<0.01, 0.3>$  z odstępem co 0.005 dla pierwszej części eksperymentu (**Rysunek 14**) oraz z takim samym odstępem wartości z przedziału  $<0.3, 0.5>$  dla drugiej części (**Rysunek 15**).

*Przebieg eksperymentu:*

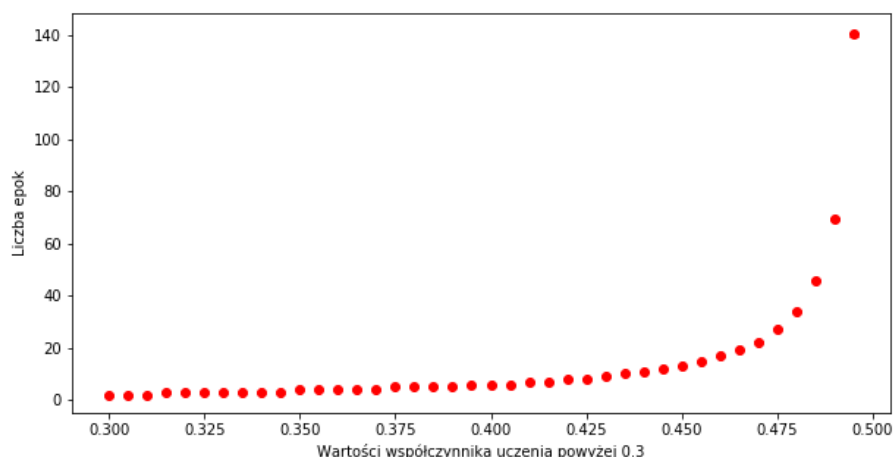
Badano wpływ wartości współczynnika  $\mu$  na szybkość uczenia się neuronu Adaline, przyjmując jako kryterium liczbę epok potrzebną do osiągnięcia warunku stopu procedury uczącej. Na początku zbadano zachowanie się neuronu w trakcie uczenia dla wartości współczynnika uczenia z przedziału  $<0.01, 0.3>$  powtarzając 10, 50 oraz 100-krotnie wywołanie metody `train` dla każdej wybranej wartości  $\mu$ . Następnie analogicznie wykonano drugą część eksperymentu dla wartości tego hiperparametru z zakresu  $<0.3, 0.5>$ , tym razem jednak powtarzając wywołanie procedury uczącej tylko 100-krotnie.

*Wyniki:*

Zamieszczono wykres dotyczący jedynie 100 powtórzeń eksperymentu dla wartości hiperparametru  $\mu$  nieprzekraczającej 0.3, ponieważ nie różnił się on znacząco od wykresów dla 10 oraz 50 powtórzeń.



Rysunek 14: Liczba epok uczenia w zależności od współczynnika uczenia o wartościach poniżej 0.3 dla 100 powtórzeń



Rysunek 15: Liczba epok uczenia w zależności od współczynnika uczenia o wartościach powyżej 0.3 dla 100 powtórzeń

#### Komentarz:

Można zaobserwować, że przy bardzo małych wartościach współczynnika uczenia neuron potrzebuje większej liczby epok do wyuczenia się optymalnych wag dla zadanego problemu. Aktualizacja wag w pojedynczym kroku jest mała, przez co musi zostać wykonana większa liczba iteracji procedury uczącej niż w przypadku większych wartości  $\mu$ . Dodatkowo istnieje możliwość, że metoda gradientu prostego wpadnie w lokalne minimum funkcji kosztu (tutaj jest nią funkcja błędu średniokwadratowego) i w nim pozostanie.

Po przekroczeniu pewnej wartości badanego hiperparametru (tutaj wartości większe od ok. 0.4) liczba epok wymaganych do osiągnięcia warunku stopu uczenia Adaline zaczęła gwałtownie rosnąć. Dla wartości  $\mu$  powyżej 0.5 nie można już było uzyskać wyników eksperymentalnych, gdyż wartość błędu średniokwadratowego wyliczanego do korekcji wag uzyskiwała zbyt duże wartości i wykraczała poza zakres liczby zmiennoprzecinkowej. Dzieje się tak dlatego, że dla wysokich wartości współczynnika uczenia algorytm szukania minimum funkcji kosztu może zbyt dużym krokiem w kierunku przeciwnym do gradientu ominąć minimum funkcji kosztu i tym samym nie będzie w stanie już znaleźć wartości optymalnych wag.

## 6 Podsumowanie wyników badań eksperymentalnych

Przeprowadzono badania wpływu różnych czynników na szybkość uczenia perceptronu prostego i Adaline. Każdy z nich odgrywał ważną rolę podczas uczenia - różne ich wartości mogły powodować przyspieszenie bądź spowolnienie procedury szukania optymalnych wag. Dobranie odpowiednich hiperparametrów jest więc ważną częścią decydującą o usprawnieniu czasu i szybkości uczenia neuronu. Nie jest to jednak łatwe zadanie, ponieważ wartości hiperparametrów są ściśle związane z rodzajem rozwiązywanego problemu oraz dodatkowo są od siebie wzajemnie zależne.

Porównując dwa modele neuronów prostych, można stwierdzić, że Adaline jest usprawnieniem perceptronu prostego. W regule uczenia perceptronu prostego bierzemy pod uwagę wartości dyskretne do aktualizacji wag, ponieważ funkcja aktywacji przyjmuje jedynie dwie wartości całkowitoliczbowe. Korekcja wag odbywa się więc na podstawie informacji, czy wystąpił błąd predykcji czy też nie. Jeśli model zaklasyfikował dany wektor sygnałów wejściowych poprawnie zmiana wag nie zachodzi. Natomiast w Adaline jako funkcję aktywacji zastosowano liniową funkcję tożsamościową. Aktualizacja wag zachodzi już nie o wartości dyskretne a rzeczywiste, tzn. bierze pod uwagę różnicę między zadanym wyjściem a całkowitym pobudzeniem neuronu. Daje to możliwość korekcji wag o realny błąd między zadanym wyjściem a predykcją modelu, nawet w przypadku gdy kwantyzator nałożony na predykcję modelu wskazywałby wynik zgodny z prawdą. Wadą obu modeli jest jednak

fakt, że mogą one wyuczyć się jedynie rozwiązywania problemów separowalnych liniowo (np. nie będą w stanie realizować funkcji XOR, która jest nieseparowalna liniowo).