Podsumowanie podstaw sztucznej inteligencji

Spis treści

[Krótki wstęp teoretyczny do podstawowych pojęć 2](#_Toc472015000)

[Opis problemu 4](#_Toc472015001)

[Zagadnienia 5](#_Toc472015002)

[Zagadnienie I 5](#_Toc472015003)

[Zagadnienie 2 9](#_Toc472015004)

[Zagadnienie 3 10](#_Toc472015005)

[Zagadnienie 4 11](#_Toc472015006)

[Zagadnienie 5 12](#_Toc472015007)

[Zagadnienie 6 14](#_Toc472015008)

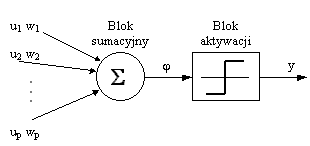
[Programy 14](#_Toc472015009)

[Porównanie 14](#_Toc472015010)

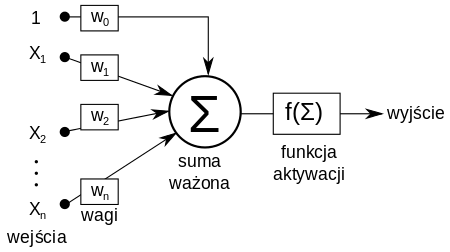
# Krótki wstęp teoretyczny do podstawowych pojęć

Sztuczna inteligencja obejmuje wiele dziedzin nauki takich jak logika rozmyta, sieci neuronowe, robotykę i informatyka. Ostatni dział, który zajmuje się inteligencją próbuje stworzyć modele zachowań maszyny jak żyjących istot. Ten termin został pierwszy raz zaproponowany przez Johna McCarthy’ego w 1955r. i nazwał sztuczną inteligencję „konstruowaniem maszyn, o których działaniu dałoby się powiedzieć, że są podobne do ludzkich przejawów inteligencji.” Sztuczną inteligencję wykorzystuje się w sieciach neuronowych, maszynowych tłumaczeniach tekstu, uczeniu się maszyn, eksploracji danych, rozpoznawaniu danych, mowy, obrazów i pisma.

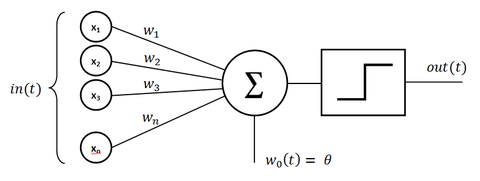
Sztuczny neuron- próba odwzorowania ludzkiego neuronu, którego zadaniem jest przetwarzanie sygnałów. Biologiczny neuron składa się z dendrytów i aksonu, dzięki temu odbiera, przewodzi, przetwarza i wytwarza bodźce. Sztuczny neuron otrzymuje wartości na wejście i przetwarza je w jedną wartość wyjściową.



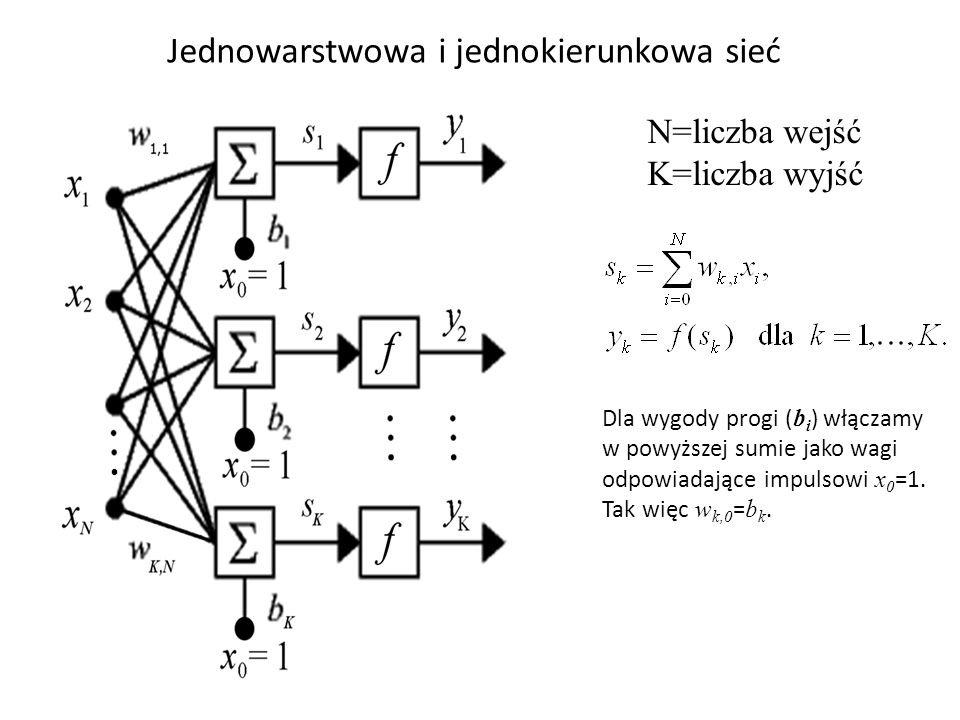
Neuron McCullocha-Pittsa- jeden z typów matematycznych modeli neuronów. Posiada wiele wejść z wagami(liczbami) i tylko jedno wyjście. Ono jest obliczane w następujący sposób: liczona jest suma ważona z wejść, a następnie wartość funkcji aktywacji dla otrzymanej sumy.



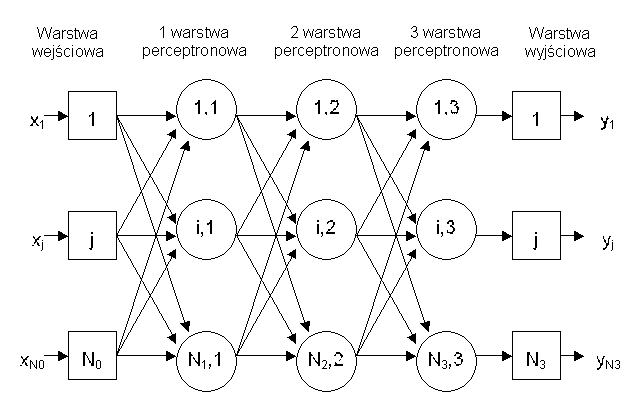
Perceptron- podstawowa sieć neuronowa, zbudowana z jednego lub więcej neuronów MCP. Zachodzi uczenie nadzorowane. Zbiór(liniowo separowalny) jest dzielony na dwie klasy na wyjściu. Osiągamy wymagane wyniki poprzez zmianę wag.



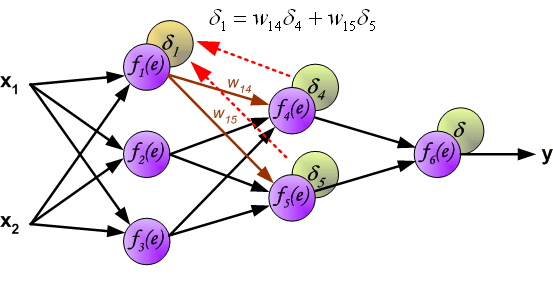
Sieć jednowarstwowa- występuje kilka wejść, określona ilość neuronów na jednym poziomie(jedna warstwa), jedno wyjście określające wynik.



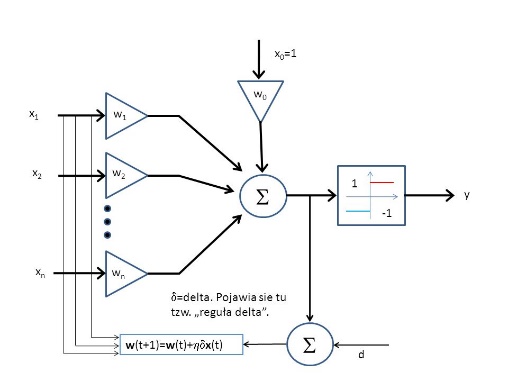
Sieć wielowarstwowa- składa się z warstwy wejściowej(kilka neuronów), warstw ukrytych( nie ma potrzeby używania więcej niż 3 warstw), które zawierają odpowiednią ilość neuronów( bardzo ważne jest ustalenie tej ilości aby poprawnie program działał) oraz warstwy wyjściowej dająca wynik.



Algorytm wstecznej propagacji błędów- na początek należy podzielić zbiór na uczący i sprawdzający. Przetwarza się zbiór uczący i nauczyciel sprawdza czy wyniki są odpowiednie. Jeśli nie to określa błąd. Błąd jest przesyłany do sieci, ale od wyjściowej do wejściowej. Przetwarza się kolejny wektor wejściowy i powtarza się algorytm. Działa się tak aż to uzyskania błędu mniejszego niż założenie. Potem sprawdza wyniki zbiór sprawdzający- zlicza poprawne wyniki.



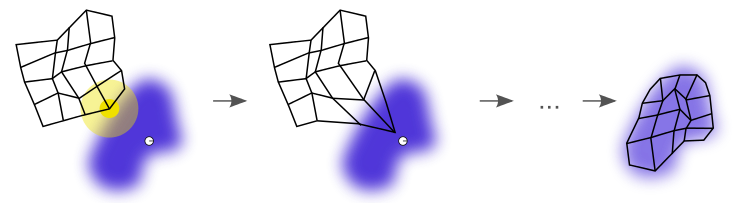
Reguła Hebba- wyróżniamy uczenie z i bez nauczyciela, wagi są zmieniane po każdej iteracji. Stosowany jest współczynnik zapominania, gdyż wagi rosną bardzo szybko i spada wydajność. Uczenie bez nauczyciela polega na tym, że nie mamy ostatniego elementu wiersza, który odpowiada za wynik.



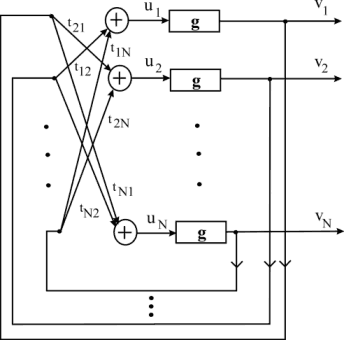
Reguła Oji- ulepszona wersja reguły Hebba, przypomina regułę wstecznej propagacji przez to, że do zmiany wagi jest wykorzystywany sygnał wsteczny.

WTA- zwycięzca bierze wszystko. Neuron z najwyższa sumą wag zostaje zwycięzcą i może wykonać zmianę. Istnieje możliwość powstania neuronów martwych gdy ani raz nie wygrają rywalizacji.

WTM- ulepszenie WTA. Sąsiedzi zwycięzcy też wykonują zmianę, ale w mniejszym stopniu. Dzięki temu zachowany jest kształt siatki.



Sieć rekurencyjna- neurony są tak połączone, że dana wejściowa jest wprowadzana do układu, aż nie nastąpi stabilizacja wag. Gdy to się stanie otrzymany wynik jest poprawny. Przykładem jest sieć Hopfielda. Pracują one na podstawie skojarzeń z wcześniejszymi wynikami. Modeluje się nimi pamięć skojarzeniową.



# Opis problemu

Problemem, który był rozwiązywany podanymi metodami było określenie gatunku rośliny- Irysa na podstawie wymiarów. Baza składa się z 100 elementów, nie ma zawsze podziału liniowego. Budowa bazy:

Długość działki kielicha

Szerokość działki kielicha

Długość płatka

Szerokość płatka

Gatunek

Przykładowy wiersz:

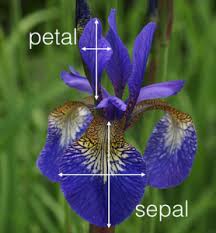
5.1,3.5,1.4,0.2,Iris-setosa

7.0,3.2,4.7,1.4,Iris-versicolor

Odnośniki

[uczące](Baza/iris.txt) [walidujące](Baza/iris2.txt)





Danymi wejściowymi były cztery wymiary kwiatu, daną wyjściową było 1 lub 0, przypisane do danego gatunku. Nie wyróżniono podproblemu, gdyż projekt polegał na porównywaniu metod na jednym problemie.

Przykładowe zastosowanie

Napisane metody rozwiązywania problemu mogłyby posłużyć do stworzenia programu, który by rozróżniał rośliny. Często zdarza się, że cebulki roślin są pomieszane w sklepie lub podczas przenoszenia do nowego miejsca w ogrodzie zapomina się jaki to był kwiat. Po pojawieniu się kwiatu można by podać wymiaru płatków i kielicha i dzięki temu będzie wiadomo co to za roślina bez szukania danych w książkach. Jest to dobre rozwiązanie dla początkujących ogrodników, którzy mogą nie zauważyć subtelnych różnych między różnymi gatunkami tej samej rośliny. Irysy na pierwszy rzut oka są bardzo do siebie podobne. Drugim pomysłem jest szukanie krzyżówek. Sieci SOM na podstawie jakiś cech podobieństwa dzielą dane na zbiory. Gdyby pobrać dane z grup mniejszych i wygenerować jakby ten kwiat wyglądał moglibyśmy otrzymać ciekawszą odmianę rośliny.

# Zagadnienia

Dane wejściowe to 70 danych uczących i 30 danych walidujących. Wyjście to określenie gatunku metodą zero jedynkową. Wykresy, dane w tabelkach i logi znajdują się w załączonych plikach. Każde uczenie odbyło się sto razy.

# Zagadnienie I

„Sztuczny neuron – idea, budowa, model McCulloch-Pitts, perceptron (działanie, klasyfikacja z wykorzystaniem perceptronu), uczenie (perceptronu i dowolnego neuronu), zastosowanie, przykłady (proste układy funkcji logicznych)”

Wyniki

Fragment logów

[logiMCP](1/MCP%20logi.txt) [logiPerceptron](1/Percepton%20logi.txt)

**MCP**

powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.018947675893152494 Ilosc błędów uczace: 70 MSE 0.018384768180797133

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.018763532069634305 Ilosc błędów uczace: 70 MSE 0.018198556303537006

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.01858119632160717 Ilosc błędów uczace: 70 MSE 0.018014211412393823

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.018400657627036607 Ilosc błędów uczace: 70 MSE 0.017831722112577494

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.01822190477144817 Ilosc błędów uczace: 70 MSE 0.017651076799138757

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.018044926357970715 Ilosc błędów uczace: 70 MSE 0.017472263667825356

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.017869710817172042 Ilosc błędów uczace: 70 MSE 0.017295270725716187

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.017696246416685237 Ilosc błędów uczace: 70 MSE 0.01712008580163178

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.017524521270624968 Ilosc błędów uczace: 70 MSE 0.01694669655631979

.

.

.

Epoka nr 53dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.011395307614495812 Ilosc błędów uczace: 70 MSE 0.010819024646777049

Epoka nr 54dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.01128939050265833 Ilosc błędów uczace: 70 MSE 0.010713262621990413

Epoka nr 55dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.011184611930918381 Ilosc błędów uczace: 70 MSE 0.010608669271336073

Epoka nr 56dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.011080959860681567 Ilosc błędów uczace: 70 MSE 0.010505232011119336

Epoka nr 57dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010978422334962455 Ilosc błędów uczace: 70 MSE 0.010402938343776598

Epoka nr 58dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.0108769874797562 Ilosc błędów uczace: 70 MSE 0.010301775859362086

Epoka nr 59dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010776643505321861 Ilosc błędów uczace: 70 MSE 0.010201732236938791

Epoka nr 60dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010677378707380237 Ilosc błędów uczace: 70 MSE 0.010102795245876438

Epoka nr 61dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010579181468228756 Ilosc błędów uczace: 70 MSE 0.010004952747059485

Epoka nr 62dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010482040257776189 Ilosc błędów uczace: 70 MSE 0.009908192694007962

Epoka nr 63dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 30 MSE 0.010385943634499626 Ilosc błędów uczace: 70 MSE 0.009812503133914022

**Perceptron**

powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

.

.

.

powtorzenie nr 99

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Przykładowe wagi

**Percepton**

-4.441013332237791

-3.3872155036841662

-0.5499440335959289

0.6589755872962715

-4.200703425553582

-2.9653070916011117

-1.2289936631652103

0.760467436013694

-4.115857688466552

-3.375326795148541

-0.9183534770626091

0.03569997563395849

**MCP**

-0.2889063716112288

-0.13710708020718348

0.12362875232183089

0.47780620928764983

-0.46408766944668733

0.04634290022743179

0.2930077884444111

0.5818154682443515

-0.11534258847430366

-0.28264782743036576

0.08928911523683634

0.5738626064193628

[Wyniki](1/wyniki%20psi%201.xlsx)

Wnioski

Na pierwszy rzut oka widać różnicę w wynikach. MCP nie nauczyło się, ani nie wykonało postępu. Błąd jest równy 100%, a MSE różne od zera. Dla tej bazy danych ta metoda uczenia jest nieodpowiednia. Nie otrzymaliśmy żadnych rezultatów. Natomiast perceptron poradził sobie doskonale z zadaniem. Błędy od razu wynoszą zero. Uczenie się udało. W obu przypadkach czas nauki jest bliski zera. Jest to spowodowane tym, że danych jest tylko 100(Dla 1000 danych czas wahał się od 46ms do 1). Baza została zmniejszona, aby sprawdzić czy będzie można zauważyć różnice. Dla większej bazy MCP nauczyło się natychmiast. Perceptron działał podobnie.

# Zagadnienie 2

„Sztuczna siec neuronowa – architektura (siec jedno- i wielo- warstwowa), uczenie (reguła delta z pochodną, algorytm wstecznej propagacji błędu), zastosowanie, przykłady realizacji prostych układów logicznych, praktyczne uwagi uczenia sieci, problemy uczenia sieci.”

Fragment logów

**Adaline**

[LogiAdaline](2/Adaline%20logi.txt)

powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

.

.

.

powtorzenie nr 99

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

Epoka nr 9dlugosc uczenia epoki: 0 Ilosc błędów walidujace: 0 MSE 0.0 Ilosc błędów uczace: 0 MSE 0.0

[Wyniki](2/wyniki%202.xlsx)

Wnioski

Wyniki pokazują, że dla konfiguracji błędy, MSE wynoszą zero. Oznacza to, że ADALINE nie ma problemów z nauką na danych rekordach. Być może dane w określonej grupie są bardzo różne od drugiej grupy. Czas jest bliski zero. Gdy danych jest 700 i 300 błędy i MSE nie zmieniły się, nadal uczy się poprawnie od razu, ale zwiększył się czas do nawet 109ms. Ta zmiana spowodowana jest ilością danych, które należy przetworzyć.

# Zagadnienie 3

„Sztuczna siec neuronowa – architektura (siec jedno- i wielo- warstwowa), uczenie (reguła delta z pochodną, algorytm wstecznej propagacji błędu), zastosowanie, przykłady realizacji prostych układów logicznych, praktyczne uwagi uczenia sieci, problemy uczenia sieci.”

Przykładowe logi

**Sieć**

**Dwie warstwy**

[logi](3/Siec%20logi.txt)

powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 1 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 1 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

.

.

.

powtorzenie nr 99

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 7dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 8dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 9dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

[Wyniki](3/wyniki%203.xlsx)

Wnioski

Dla wersji z danymi 70 i 30 czas ponownie jest bliski 0 z kilkoma skokami do 1ms. Sieć nauczyła się od razu poprawnych rozwiązań wyboru gatunku kwiatu. Gdy danych jest 1000 zmienia się tylko czas działania- około 6ms i występują tak jak w wcześniejszej wersji skoki na poziomie 1ms i 2ms.

# Zagadnienie 4

„Uczenie sieci z i bez nauczyciela, reguła Hebba, zastosowanie, przykłady.”

Logi

**Hebba bez nauczyciela**

[logi](4/logi%20Hebba%20bez%20nauczyciela%20.txt)

Powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

**Hebba bez nauczyciela z współczynnikiem zapominania**

[logi](4/logi%20Hebba%20bez%20nauczycielaZzapominaniem%20.txt)

Powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

**Oji bez nauczyciela**

[logi](4/logi%20OjiBezNauczyciela%20.txt)

Powtorzenie nr 0

Epoka nr 0dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

[Wyniki](4/wyniki%204.xlsx) [wyniki](4/wyniki%20dla%20700%20300.xlsx)

Wnioski

Dla każdej reguły czas równa się prawie zero jak i także błąd. Dla wersji 1000 danych zmienia się tylko długość pracy. W danym zagadnieniu omawialiśmy metodę Hebba i Oji oraz różne ich warianty. Odbywało się uczenie bez i z nauczycielem. W każdym przypadku neurony nauczyły się od razu. Może to być spowodowane tym, że w danych istnieje przedział, który nie należy do żadnego z zbiorów. Być może problem jest za łatwy. Porównując czasy widać niewielkie zmiany, ale tendencja jest taka sama. Na początku jest najwyższy czas, potem się stabilizuje. Żadna z metod nie "wybija" się nad inne. Przyczyną może być mała baza danych albo podobieństwo metod(różnią się tylko funkcją korekcji wag).

# Zagadnienie 5

„Sieci Kohonena, uczenie (WTA), odwzorowanie cech istotnych (sieci samoorganizujące - SOMy), uczenie sieci SOM, przykłady, zastosowanie”

**WTA i Hebba**

Logi

[logi](5/Uczenie%20logi.txt)

powtorzenie nr 0

Neuron nr 0

Epoka nr 0dlugosc uczenia epoki: 1 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 1dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 2dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 3dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 4dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 5dlugosc uczenia epoki: 1 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 6dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

.

.

.

Epoka nr 89dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 90dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 91dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 92dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 93dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 94dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 95dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 96dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 97dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 98dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

Epoka nr 99dlugosc uczenia epoki: 0 Ilosc błędów uczace: 0 MSE 0.0 Ilosc błędów walidujace: 0 MSE 0.0

[Wyniki](5/wyniki5.xlsx) [wyniki](5/WTA.xlsx)

Wnioski

W zadaniu neurony grupowały się(klasy) i był szukany zwycięzca. Grupowanie polegało na szukanie wspólnej cechy danych i osadzanie w przestrzeni w odpowiednim miejscu. To powodowało że tworzyły się grupy danych podobnych. W wynikach widać, że uczyły się od razu, bo błędy wynoszą zero. Natomiast zmiany widać podczas przydzielania do klas. Gdy danych było dużo przydzielanie było szybciej poprawne niż w przypadku małej ilości danych. Zdarzały się neurony martwe czyli te, które nigdy nie wygrały i nie pasowały do żadnej grupy. Porównano czas uczenia się tej samej metody z zagadnienia 4 i widać że dzięki klasom uczą się one szybciej.

# Zagadnienie 6

„Sieci Kohonena, uczenie (WTA), odwzorowanie cech istotnych (sieci samoorganizujące - SOMy), uczenie sieci SOM, przykłady, zastosowanie”

**WTM**

[Klasyfikacja](6/Klasyfikacja)

[Wyniki](6/WTM.xlsx)

Wnioski

W sieci SOM wykorzystujemy nauczanie WTM czyli zwycięzca bierze najwięcej. W podziale jest uwzględniany sąsiad, aby zachować kształt siatki. Na załączonych przykładowych mapach widać jak się grupują. Obok neuronu, który zwyciężył najwięcej razy zawsze widzimy sąsiadów z wartością większą od zera. Nie musiały one wygrać, ale były blisko wyniku. Dzięki temu wyniki badań są dokładniejsze i rzetelniejsze, bo nie występuje tylko podział zero jedynkowy.

Podział został zapisany do pliku i można zauważyć, że przykładowo do jednej grupy przydzielono kwiaty o dużej wartości pierwszej i trzeciej miary( długość kielicha i płatka). Są pliki z jednym rekordem, nie pasował on do żadnej podgrupy.

[podział](6/podział)

# Programy

[1](1/kod)

[2](2/kod)

[3](3/kod)

[4](4/kod)

[5](5/kod)

[6](6/kod)

# Porównanie

Podczas zajęć projektowych napisano kilka metod szukania przydzielania do grupy. Projekt polegał na porównaniu nich na jednych danych. Wszystkie metody były na tyle dobre, że znalezienie odpowiedniej konfiguracji wag zajęło im jedną epokę. Błędy złego przypisania i MSE dla danych uczących i walidujących równały się zero. Wyjątkiem jest tutaj MCP(McCulloch-Pitts Neuron), który w ogóle nie nauczył się i MSE wynosi około 0.01. Czasy w każdej metodzie były bliskie zera. Jest to spowodowane małą bazą danych. Gdy zwiększono ilość danych czas trwania uczenia zwiększył wartość nawet do 46ms. Najciekawszą metodą były SOM. Mogliśmy zobaczyć na mapach jak dzielone są dane. Tworzyły się również mniejsze grupy, które mogłyby być wykorzystane do stworzenia nowego gatunku rośliny.