

# Sieci neuronowe i neurokomputery (SNN)

## Projekt klasyfikatora neuronowego dla dwóch klas

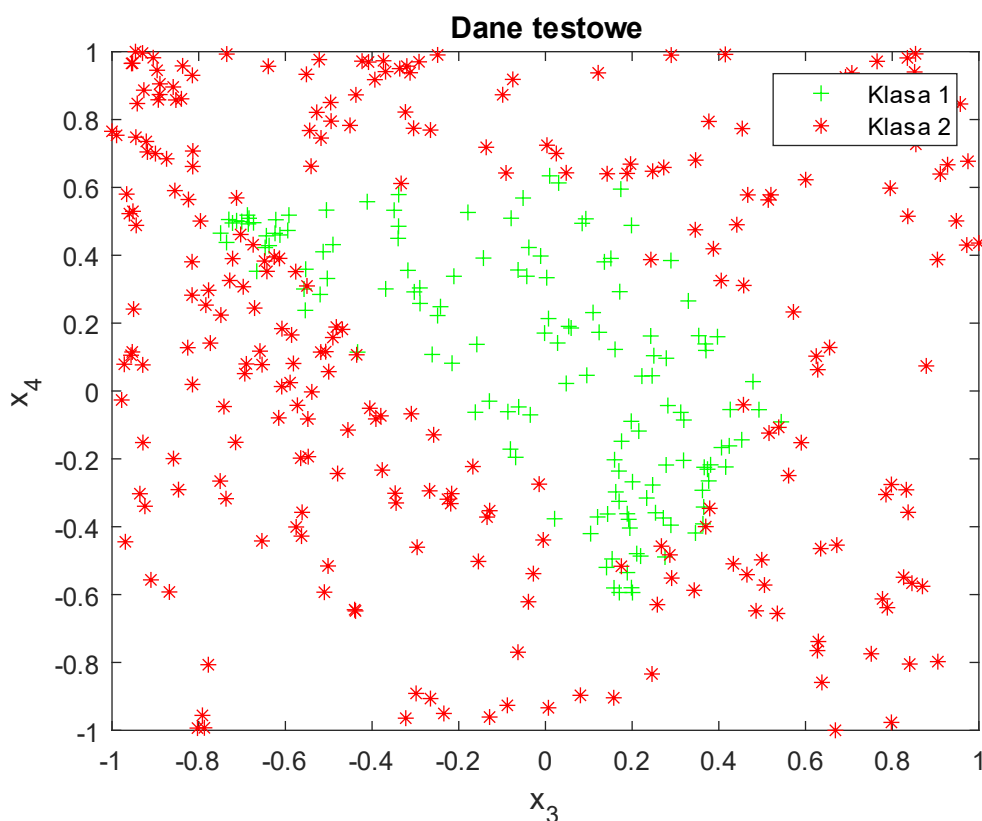
Imię i nazwisko studenta	Nr indeksu	Ocena
Mateusz Hryciów	283365	

### 1. Cel projektu.

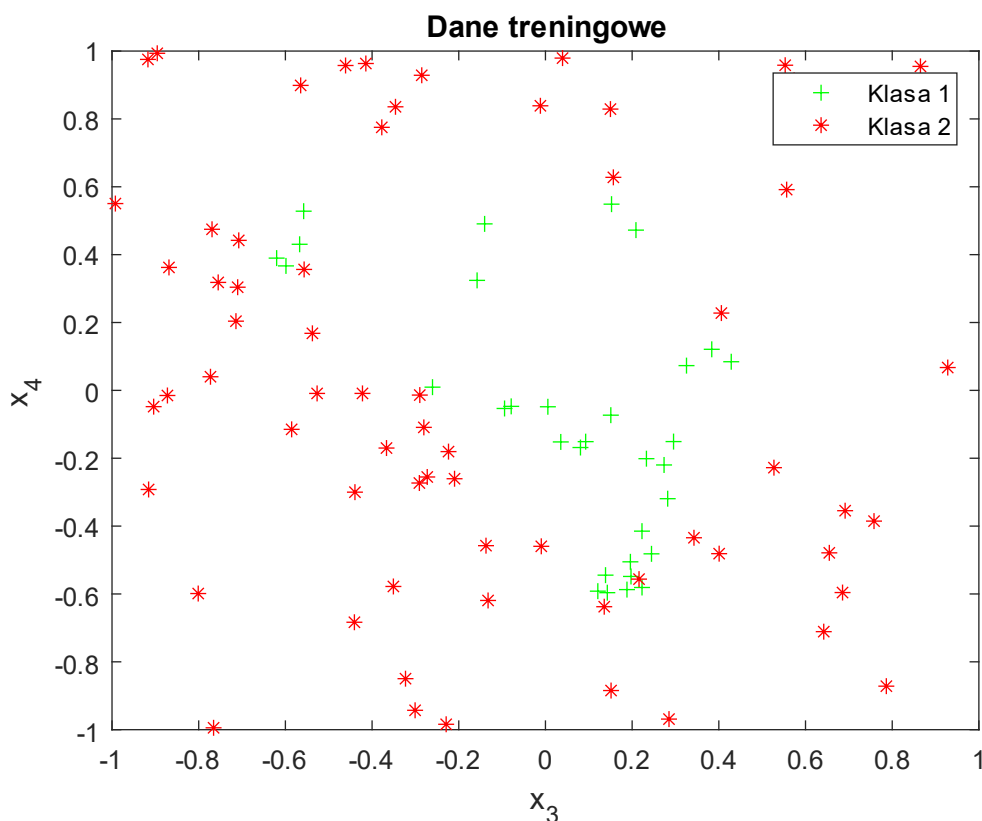
Celem projektu było zaprojektowanie klasyfikatora neuronowego dla danych ze zbioru uczącego, który będzie rozpoznawał dwie klasy.

### 2. Opis i wizualizacja danych

Każda próbka danych była opisana przez cztery różne zmienne oraz posiadała etykietę klasy, do której należy. W celu uczenia i weryfikacji sieci dane zostały podzielone w stosunku 1:4. Mniejszą część (100 próbek) stanowiły – dane treningowe, natomiast większą (400 próbek) – dane testowe. Używając opisanych w punkcie 3. danych wejściowych dokonano wizualizacji danych.



Rysunek 1. Dane testowe wykorzystane do weryfikacji sieci neuronowej.



Rysunek 2. Dane treningowe użyte do uczenia sieci neuronowej.

Liczebności poszczególnych klas w każdym ze zbiorów przedstawiono w tabeli 1.

Tabela 1. Liczebności poszczególnych zbiorów

	Klasa I (zielone krzyżyki)	Klasa II (czerwone gwiazdki)
Zbiór testowy	141	259
Zbiór treningowy	32	68

W każdym ze zbiorów przeważają dane dotyczące klasy II. Analizując przedstawione wykresy można zauważyć, że w dużej mierze dane są separowalne. Jednakże nie da się tego zrobić za pomocą pojedynczej linii, co oznacza, że liczba neuronów ukrytych będzie musiała być większa niż jeden. Jeżeli każdy neuron ukryty potraktujemy jako linię prostą na powyższych wykresach, to można przyjąć, że ich minimalna ilość niezbędna do zadowalającej separacji wynosi cztery. Jednakże można zauważyć, że istnieją pewne obszary, na których występują obydwie klasy. Są to części wspólne. Z tego względu niemożliwe jest zaprojektowanie klasyfikatora, który pozwoli na bezbłędną klasyfikację dwóch klas.

### 3. Selekcja zmiennych wejściowych

Jak wspomniano każda próbka była opisana przez cztery różne zmienne. Zadaniem projektowym był natomiast wybór dwóch z nich, które są najbardziej istotne. W tym celu wykorzystano zaproponowaną procedurę Grama-Schmidta. Zastosowaną ją do danych testowych oraz treningowych równocześnie. Ostatecznie otrzymano, że najbardziej istotne są zmienne **3.** i **4.** Wszystkie zmienne zostały zcentralizowane oraz znormalizowane do przedziału  $< -1; 1 >$ .

## 4. Zastosowany algorytm uczenia sieci neuronowej i opis zastosowanego programu komputerowego

Każdorazowo uczenie sieci składało się z dwóch etapów. W pierwszym wykorzystywano algorytm wstecznej propagacji błędów. Jako parametr stopu przyjęto wartość błędów na poziomie 0,01 oraz liczbę epok równą 100. W drugim etapie wykorzystano algorytm Lavenberga-Marquardta. Przyjęto taki sam poziom błędów, natomiast liczbę epok uczenia równą 200.

Inicjalizacji wag dokonano za pomocą domyślnego algorytmu Nguyena-Widrowa. Funkcją aktywacji neuronów ukrytych był tangens hiperboliczny, natomiast neuronu wyjściowego funkcja liniowa.

Uczenia sieci dokonywana za pomocą funkcji znajdującej się w pliku *train\_net.m*. Wszystkie obliczenia dokonywane były w wykorzystaniu środowiska MATLAB R2020a.

## 5. Udokumentowanie wyboru architektury sieci

Celem projektu było stworzenie dwóch rodzajów klasyfikatorów. Pierwszy z nich miał się składać z dwóch niezależnych klasyfikatorów binarnych z jednym neuronem wejściowym. Natomiast drugi posiadał dwa neurony wyjściowe. Dla każdej z zaproponowanych sieci liczba neuronów wejściowych wynosiła 2, gdyż taka była założona liczba najbardziej istotnych zmiennych. Celem badania było określenie jaka liczba neuronów w warstwie ukrytej pozwala na uzyskanie najlepszej jakości klasyfikacji. W tym celu zbadano 10 różnych architektur sieci neuronowych o liczbie neuronów ukrytych od 2 do 20. Jak wcześniej wspomniano, minimalna liczba neuronów ukrytych powinna wynosić cztery. Z tego powodu oczekuje się, że dla tylko dwóch neuronów uzyskane wyniki będą najgorsze.

Zaprezentowane wyniki przedstawiają wartości w procentach poprawnie oraz błędnie rozpoznanych klas. Dodatkowo przyjęto wartości progu dyskryminacji od 0,4 do 0,6, dla których próbka uznawana jest za niesklasyfikowaną.

Zgodnie z poleceniem dla każdej architektury przeprowadzono symulacje dla 20 różnych wartości wag początkowych. Obliczone wartości procentowe następnie uśredniono. Wykorzystany skrypt dołączono w **Załączniku 1**. Dla każdego z klasyfikatorów uzyskano następujące wyniki.

- dwa niezależne klasyfikatory binarne

Tabela 2. Uśrednione wyniki dla klasyfikatora binarnego klasy I.

Numer sieci	Liczba neuronów ukrytych	Procent poprawnych klasyfikacji [%]	Procent niepoprawnych klasyfikacji [%]	Procent niesklasyfikowanych danych [%]
1	2	68,8	14,2	17
2	4	80,4	8	11,6
3	6	84,05	6,9	9,05
4	8	84,75	6,35	8,9
5	10	86,65	4,75	8,6
6	12	86,4	5,7	7,9
7	14	85,6	4,45	9,95
8	16	85,3	5,8	8,9
9	18	83,95	5,6	10,45
10	20	84,65	5,45	9,9

Tabela 3. Uśrednione wyniki dla klasyfikatora binarnego klasy II.

Numer sieci	Liczba neuronów ukrytych	Procent poprawnych klasyfikacji [%]	Procent niepoprawnych klasyfikacji [%]	Procent niesklasyfikowanych danych [%]
1	2	63,4	17,15	19,45
2	4	79,35	6,7	13,95
3	6	82,8	6,1	11,1
4	8	84,3	6,3	9,4
5	10	84,85	5,7	9,45
6	12	85,85	4,65	9,5
7	14	85,85	4,55	9,6
8	16	83,9	4,15	11,95
9	18	82,9	6,05	11,05
10	20	83,15	6,1	10,75

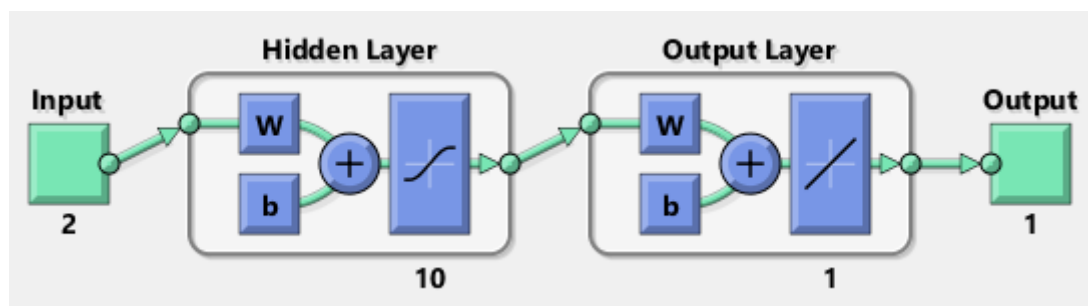
- klasyfikator o dwóch neuronach wyjściowych

Tabela 4. Uśrednione wyniki dla klasyfikatora o dwóch neuronach wyjściowych

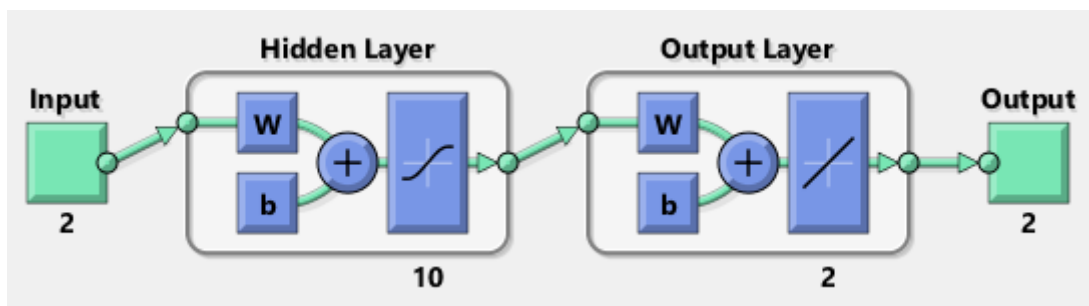
Numer sieci	Liczba neuronów ukrytych	Procent poprawnych klasyfikacji [%]	Procent niepoprawnych klasyfikacji [%]	Procent niesklasyfikowanych danych [%]
1	2	69,85	15,45	14,7
2	4	82,95	6,4	10,65
3	6	83,45	5,65	10,9
4	8	84,65	5,2	10,15
5	10	86,9	4,55	8,55
6	12	84,95	4,9	10,15
7	14	84,3	5,75	9,95
8	16	85,7	5,25	9,05
9	18	86,3	4,3	9,4
10	20	86,8	3,65	9,55

W powyższych tabelach kolorem zielonym zaznaczono sieci o liczbie neuronów ukrytych, które pozwoliły na uzyskanie największej poprawności klasyfikacji. Zgodnie z przewidywaniami dla każdego rodzaju klasyfikatora najgorsze wyniki uzyskano dla sieci o dwóch neuronach ukrytych. Wraz ze wzrostem ich liczby skuteczność sieci również rosła. Największą poprawność klasyfikacji osiągnięto dla 10-12 neuronów ukrytych (ok. 86 – 87 %). Dalsze zwiększanie ich liczby nie prowadziło do poprawy jakości klasyfikacji. Zdecydowano zatem, że zarówno dla dwóch niezależnych klasyfikatorów binarnych, jak i klasyfikatora o dwóch neuronach wyjściowych najlepsza architektura sieci opiera się na 10 neuronach ukrytych. Dla tej wartości przeprowadzono dalsze badania.

Ostatecznie uzyskane architektury wyglądają następująco:



Rysunek 3. Architektura sieci dla każdego z dwóch niezależnych klasyfikatorów binarnych.



Rysunek 4. Architektura sieci dla sieci o dwóch neuronach wyjściowych.

## 6. Tabele wartości wag klasyfikatora neuronowego

Dla każdego rodzaju klasyfikatora o 10 neuronach ukrytych przedstawiono wartości wag dla połączeń między neuronami wejściowymi i ukrytymi oraz bias. Jak wcześniej wspomniano, dla każdej architektury, proces uczenia prowadzono 20 razy dla różnych, losowych wartości wag początkowych. Z tego powodu prezentowane wyniki będą dotyczyć sieci neuronowych o największej wartości poprawnych klasyfikacji.

- dwa niezależne klasyfikatory binarne

Tabela 5. Wagi połączeń oraz bias dla neuronów ukrytych klasyfikatora binarnego klasy I.

Numer neuronu warstwy ukrytej	Wejście nr 1	Wejście nr 2	Bias
1	1,408	-3,142	-6,509
2	5,034	13,079	-4,544
3	21,904	0,431	-8,056
4	-16,156	-0,381	6,044
5	2,579	7,013	-1,725
6	5,148	-2,073	1,807
7	34,214	22,704	8,357
8	-5,370	11,817	-8,877
9	-8,350	19,265	-14,221
10	12,210	2,323	6,293

Tabela 6. Wagi połączeń oraz bias dla neuronów wyjściowych klasyfikatora binarnego klasy I.

Numer neuronu warstwy ukrytej	1	2	3	4	5	6	7	8	9	10	Bias
Wyjście nr 1	0,263	-2,401	-4,228	-3,252	2,391	0,278	1,005	2,710	-3,388	-0,293	-1,423

Tabela 7. Wagi wejściowe oraz bias dla neuronów ukrytych klasyfikatora binarnego klasy II.

Numer neuronu warstwy ukrytej	Wejście nr 1	Wejście nr 2	Bias
1	-0,240	-5,552	3,292
2	4,295	1,840	-4,518
3	-7,743	5,072	5,252
4	-1,716	-1,682	0,115
5	-3,238	4,021	-0,826
6	2,971	5,210	0,041
7	7,770	1,861	0,283
8	3,854	2,562	1,732
9	3,111	3,593	3,459
10	6,451	4,696	3,372

Tabela 8. Wagi połączeń oraz bias dla neuronów wyjściowych klasyfikatora binarnego klasy II.

Numer neuronu warstwy ukrytej	1	2	3	4	5	6	7	8	9	10	Bias
Wyjście nr 1	-0,985	-0,397	-0,789	-2,450	-0,449	-1,001	-0,389	-4,590	0,638	2,497	1,497

- klasyfikator o dwóch neuronach wyjściowych

Tabela 9. Wagi wejściowe oraz bias dla neuronów ukrytych klasyfikatora o dwóch neuronach wyjściowych

Numer neuronu warstwy ukrytej	Wejście nr 1	Wejście nr 2	Bias
1	-4,105	2,938	3,260
2	-7,600	3,932	4,853
3	2,227	3,007	1,142
4	0,009	7,462	-0,271
5	-1,430	-4,305	-0,783
6	4,512	2,264	-0,655
7	-14,155	-10,256	-3,812
8	2,268	-4,194	4,523
9	1,571	11,514	-7,050
10	4,038	-1,769	5,728

Tabela 10. Wagi połączeń oraz bias dla neuronów wyjściowych klasyfikatora o dwóch neuronach wyjściowych.

Numer neuronu warstwy ukrytej	1	2	3	4	5	6	7	8	9	10	Bias
Wyjście nr 1	-2,182	2,546	-0,611	-0,301	-0,803	-0,338	-1,151	0,447	-0,688	-0,549	-1,238
Wyjście nr 2	2,177	-2,542	0,612	0,301	0,803	0,339	1,153	-0,450	0,688	0,555	1,236

## 7. Wyniki klasyfikacji: wstępna ocena dla zbioru uczącego i testowego

Dla określonych w punkcie 6. sieci neuronowych przeprowadzono klasyfikację dla zbioru uczącego oraz testowego. Uzyskano wyniki:

- dwa niezależne klasyfikatory binarne

Tabela 11. Wyniki klasyfikacji na zbiorze uczącym dla dwóch klasyfikatorów binarnych.

Klasa	Zbiór uczący					
	Poprawna klasyfikacja		Niepoprawna klasyfikacja		Niesklasyfikowane	
	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]
1	96,88	31	0,00	0	3,13	1
2	91,18	62	4,41	3	4,41	3
Suma	94,03	93	2,21	3	3,77	4

Tabela 12. Wyniki klasyfikacji na zbiorze testowym dla dwóch klasyfikatorów binarnych.

Klasa	Zbiór testowy					
	Poprawna klasyfikacja		Niepoprawna klasyfikacja		Niesklasyfikowane	
	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]
1	72,34	102	21,28	30	6,38	9
2	88,03	228	4,25	11	7,72	20
Suma	80,19	330	12,77	41	7,06	29

- klasyfikator o dwóch neuronach wyjściowych

Tabela 13. Wyniki klasyfikacji na zbiorze uczącym dla klasyfikatora o dwóch neuronach wyjściowych.

Zbiór uczący						
Klasa	Poprawna klasyfikacja		Niepoprawna klasyfikacja		Niesklasyfikowane	
	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]
1	87,50	28	6,25	2	6,25	2
2	95,59	65	2,94	2	1,47	1
Suma	91,55	93	4,60	4	3,86	3

Tabela 14. Wyniki klasyfikacji na zbiorze testowym dla klasyfikatora o dwóch neuronach wyjściowych.

Zbiór testowy						
Klasa	Poprawna klasyfikacja		Niepoprawna klasyfikacja		Niesklasyfikowane	
	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]	Procent [%]	Liczba [–]
1	77,31	109	17,02	24	5,67	8
2	93,82	243	4,25	11	1,93	5
Suma	85,57	352	10,64	35	3,8	13

Można zaobserwować, że zarówno dla dwóch klasyfikatorów, jak i klasyfikatora o dwóch wyjściach procent poprawnie sklasyfikowanych danych maleje dla zbioru testowego. Dodatkowo z większą dokładnością określana jest klasa 2.

## 8. Wyniki klasyfikacji: ocena jakości klasyfikatora.

Poszczególne wyniki klasyfikacji otrzymano poprzez zastosowanie nomenklatury związanej z tablicą pomyłek. Przy czym jako klasą pozytywną przyjęto klasę I, z jako klasę negatywną – II. Otrzymano wyniki:

- dwa niezależne klasyfikatory binarne

Tabela 15. Wartości parametrów dotyczących jakości dla dwóch klasyfikatorów binarnych dla zbioru uczącego.

Zbiór uczący	
Parametr	Wartość procentowa [%]
Kompletność	96,86
Fałszywy alarm	0
Precyzja	100
Dokładność	93
Specyficzność	100

Tabela 16. Wartości parametrów dotyczących jakości dla dwóch klasyfikatorów binarnych dla zbioru testowego.

Zbiór testowy	
Parametr	Wartość procentowa [%]
Kompletność	72,34
Fałszywy alarm	11,58
Precyzja	77,27
Dokładność	82,50
Specyficzność	88,37

- klasyfikator o dwóch neuronach wyjściowych

Tabela 17. Wartości parametrów dotyczących jakości dla klasyfikatora o dwóch neuronach wyjściowych dla zbioru uczącego.

Zbiór uczący	
Parametr	Wartość procentowa [%]
Kompletność	87,50
Fałszywy alarm	2,94
Precyzja	93,33
Dokładność	93,00
Specyficzność	97,01

Tabela 18. Wartości parametrów dotyczących jakości klasyfikatora o dwóch neuronach wyjściowych dla zbioru testowego.

Zbiór testowy	
Parametr	Wartość procentowa [%]
Kompletność	77,30
Fałszywy alarm	9,27
Precyzja	81,95
Dokładność	88,00
Specyficzność	91,01

## 9. Wizualizacja otrzymanych wyników

Wyniki klasyfikacji przedstawiono na poniższych wykresach. Linie oddzielające klasy uzyskano za pomocą równania funkcji liniowej [3]:

$$y = ax + b$$

gdzie parametry wynoszą:

$$a = \frac{-w_{1,i}}{w_{2,i}}$$

$$b = \frac{-bias_i}{w_{2,i}}$$

gdzie:

$w_{1,i}$  – waga połączenia pierwszego neuronu wejściowego z i-tym neuronem ukrytym,

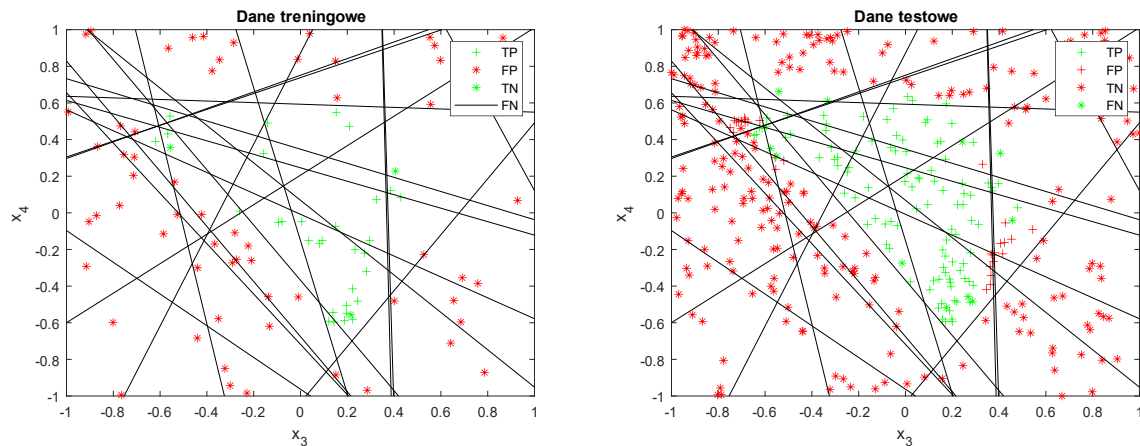
$w_{2,i}$  – waga połączenia drugiego neuronu wejściowego z i-tym neuronem ukrytym,

$bias_i$  – bias i-tego neuronu ukrytego.



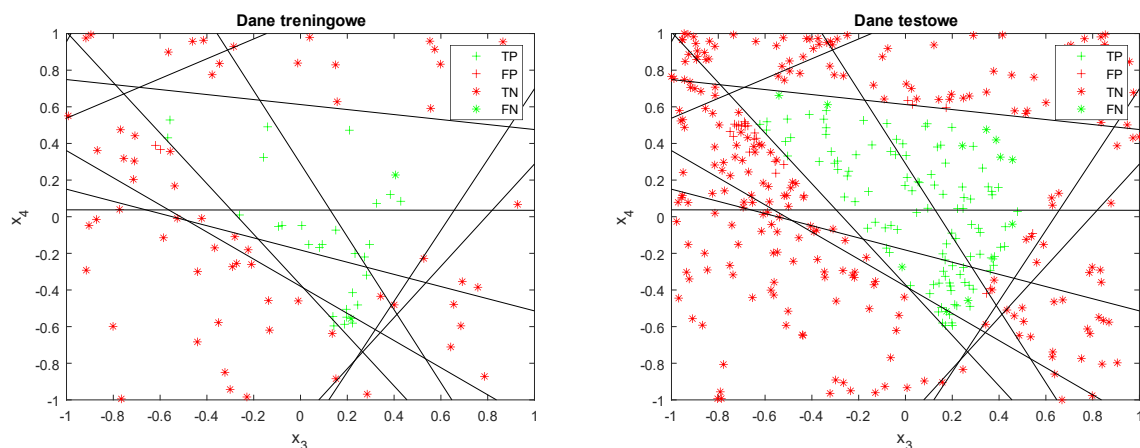
Ostatecznie otrzymano wykresy:

- dwa niezależne klasyfikatory binarne



Rysunek 5. Wizualizacja wyników klasyfikacji dla dwóch klasyfikatorów binarnych.

- klasyfikator o dwóch neuronach wyjściowych



Rysunek 6. Wizualizacja wyników klasyfikacji dla klasyfikatora o dwóch neuronach wyjściowych.

Na rysunkach widoczne jest, że zastosowano nadmiarową liczbę neuronów ukrytych. Możliwe jest dokonanie klasyfikacji o podobnej jakości, przy ich mniejszej liczbie. Jednakże większość przykładów należących do obydwu klas została sklasyfikowana poprawnie.

## 10. Wnioski końcowe

Na podstawie zaprezentowanych tabel można zaobserwować, że sieci neuronowe z większą skutecznością klasyfikowały przypadki należące do klasy 2. Jest to spowodowane faktem, że ich liczebność była zdecydowanie większa niż klasy 1. Duża liczba danych z pewnej klasy pozwala sieci na lepsze określenie granic pomiędzy grupami. Analogicznie, jeżeli liczba danych jest zbyt mała, to klasyfikator określa granice z mniejszą skutecznością.

Dla zbadanych sieci oraz zbiorów treningowego oraz testowego osiągnięto wartość dokładności ok. 90%. Parametr ten opisuje jaka część wszystkich danych została sklasyfikowana prawidłowo. Można zatem przyjąć, że nauczona sieć neuronowa działa w poprawny sposób. Jednakże zarówno w przypadku klasyfikatorów binarnych, jak i klasyfikatora o dwóch neuronach wyjściowych zaobserwowano pogorszenie wartości parametrów opisujących działanie sieci dla zbioru testowego. Oznacza to, że sieć

mogła za bardzo dopasować się do danych treningowych. Jednakże zdolność generalizacji wciąż jest na zadowalającym poziomie.

Porównując dwa niezależne klasyfikatory binarne oraz klasyfikator o dwóch neuronach wyjściowych można stwierdzić, że skuteczniejszy jest ten drugi. Dla zbiorów testowych, które pozwalają na lepszą ocenę, uzyskano dla niego większą dokładność działania. Cechą wspólną obydwu klasyfikatorów jest lepsza klasyfikacja przykładów należących do klasy 2. W przypadku zastosowania dwóch niezależnych klasyfikatorów binarnych, w niektórych przypadkach, istniała pewna trudność z przydzieleniem przykładu do danej klasy. Z tego powodu dla tego klasyfikatora większy odsetek stanowią dane niesklasyfikowane. Zatem można ostatecznie stwierdzić, że lepszy jest klasyfikator o dwóch neuronach wyjściowych.

Analizując rysunki przedstawiające finalne wyniki klasyfikacji można zaobserwować znaczącą liczbę linii symbolizujących działanie neuronów ukrytych. Pewna ich część nie wnosi dużo informacji do klasyfikatora. Z tego powodu można przypuszczać, że można by zmniejszyć liczbę neuronów ukrytych, co nie pogorszyłoby jakości działania klasyfikatora.

Na przedstawionych rysunkach widoczne jest, że nie wszystkie dane są separowalne oraz, że istnieją pewne części wspólne obydwu klas. Z tego powodu niemożliwe jest uzyskanie bezbłędnego klasyfikatora. Mimo to uzyskane wyniki są zadowalające. Nie oznacza to jednak, że nie można ich już poprawić. W celu otrzymania klasyfikatora o lepszych parametrach należałoby modyfikować również parametry związane z procesem uczenia, takie jak: liczba epok, czy kryterium stopu.

## 11. Źródła oraz zastosowane programy

[1] Z. Wawrzyniak: Materiały do wykładu i projektu

[2] Dokumentacja programu MATLAB

[3]<https://medium.com/@thomascourtz/calculate-the-decision-boundary-of-a-single-perceptron-visualizing-linear-separability-c4d77099ef38>

### Załącznik 1

```
clc;
clear;
close all
% Import danych z pliku tekstowego
dane_ucz=importdata('data_train.txt');
dane_test=importdata('data_test.txt');
% Opis tablicy 'dane':
% kolumny 1,2 - współrzędne punktów do klasyfikacji
% kolumna 3 - etykieta punktu {-1,1}

prog_dolny = 0.4;
prog_gorny = 0.6;
N = 30; % Liczba iteracji do usrednienia
liczba_neuronow_ukrytych=10;

wart_kl_poz=1; %wartości etykiet używane przy uczeniu
wart_kl_neg=0;
idx_poz=find(dane_ucz(:,3)==1); %indeksy punktów klasy pozytywnej
idx_neg=find(dane_ucz(:,3)~=1); %indeksy punktów klasy negatywnej
dane_ucz(idx_poz,3)=wart_kl_poz; %zmiana wartości etykiet klasy poz.
dane_ucz(idx_neg,3)=wart_kl_neg; %zmiana wartości etykiet klasy neg.

best_proc_1 = 0;
for j = 1:N
    [net]=train_net(dane_ucz(:,1:2),dane_ucz(:,3),liczba_neuronow_ukrytych);
    for i=1:size(dane_ucz,1)
        y(i) = net(dane_ucz(i,[1,2]))';
    end
end
```

```

ind_poz = find(y > prog_gorny);
ind_neg = find(y < prog_dolny);
ind_brak = find(y>0.4 & y<0.6);

proc_dobrze =
(numel(find(ismember(ind_poz,idx_poz)))+numel(find(ismember(ind_neg,idx_neg))))/size(dane_ucz,
1)*100;
    if proc_dobrze > best_proc_1
        best_proc_1 = proc_dobrze;
        best_net_1 = net;
        w_1 = net.IW{1}';
        bin_1 = net.b{1};
        v_1 = net.LW{2,1}';
        bout_1 = net.b{2};
    end
end

wart_kl_poz=1;                %wartości etykiet używane przy uczeniu
wart_kl_neg=0;
idx_poz_2=find(dane_ucz(:,3)~=1);    %indeksy punktów klasy pozytywnej
idx_neg_2=find(dane_ucz(:,3)==1);    %indeksy punktów klasy negatywnej
dane_ucz(idx_poz_2,3)=wart_kl_poz;    %zmiana wartości etykiet klasy poz.
dane_ucz(idx_neg_2,3)=wart_kl_neg;    %zmiana wartości etykiet klasy neg.

best_proc_2 = 0;
for j = 1:N
    [net]=train_net(dane_ucz(:,1:2),dane_ucz(:,3),liczba_neuronow_ukrytych);
    for i=1:size(dane_ucz,1)
        y(i) = net(dane_ucz(i,[1,2]))';
    end

    ind_poz = find(y > prog_gorny);
    ind_neg = find(y < prog_dolny);
    ind_brak = find(y>0.4 & y<0.6);

    proc_dobrze =
(numel(find(ismember(ind_poz,idx_poz_2)))+numel(find(ismember(ind_neg,idx_neg_2))))/size(dane_
ucz,1)*100;
    if proc_dobrze > best_proc_2
        best_proc_2 = proc_dobrze;
        best_net_2 = net;
        w_2 = net.IW{1}';
        bin_2 = net.b{1};
        v_2 = net.LW{2,1}';
        bout_2 = net.b{2};
    end
end

idx_poz=find(dane_test(:,3)==1);    %indeksy punktów klasy pozytywnej
idx_neg=find(dane_test(:,3)~=1);    %indeksy punktów klasy negatywnej
idx_poz_2=find(dane_test(:,3)~=1);    %indeksy punktów klasy pozytywnej
idx_neg_2=find(dane_test(:,3)==1);    %indeksy punktów klasy negatywnej

% Wynik klasyfikacji sieci na zbiorze uczącym i testowym
for i=1:size(dane_test,1)
    y_1(i) = best_net_1(dane_test(i,[1,2]))';
    y_2(i) = best_net_2(dane_test(i,[1,2]))';
end
ind_poz_1 = find(y_1 > prog_gorny);
ind_neg_1 = find(y_1 < prog_dolny);
ind_brak_1 = find(y_1>=0.4 & y_1<=0.6);
TP = numel(find(ismember(ind_poz_1,idx_poz)));
FP = numel(find(ismember(ind_neg_1,idx_poz)));
proc_dobrze_1 = TP/numel(idx_poz)*100;
proc_zle_1 = FP/numel(idx_poz)*100;
proc_brak_1 = numel(find(ismember(ind_brak_1,idx_poz)))/numel(idx_poz)*100;

ind_poz_2 = find(y_2 > prog_gorny);
ind_neg_2 = find(y_2 < prog_dolny);
ind_brak_2 = find(y_2>=0.4 & y_2<=0.6);
TN = numel(find(ismember(ind_poz_2,idx_poz_2)));
FN = numel(find(ismember(ind_neg_2,idx_poz_2)));
proc_dobrze_2 = TN/numel(idx_poz_2)*100;
proc_zle_2 = FN/numel(idx_poz_2)*100;

```

```

proc_brak_2 = numel(find(ismember(ind_brak_2,idx_poz_2)))/numel(idx_poz_2)*100;

wyniki_ucz = [proc_dobrze_1, TP, proc_zle_1, FP, proc_brak_1,
numel(find(ismember(ind_brak_1,idx_poz_1)))+...
proc_dobrze_2, TN,proc_zle_2, FN,proc_brak_2,
numel(find(ismember(ind_brak_2,idx_poz_2)))]];

komp = TP/numel(idx_poz);
f_a = FP/numel(idx_neg);
prec = TP/(TP+FP);
dokl = (TP+TN)/(numel(idx_poz)+numel(idx_poz_2));
spec = TN/(FP+TN);

parametry_ucz =[komp; f_a; prec; dokl; spec]*100;

figure(1)
plot(dane_test(ind_poz_1(ismember(ind_poz_1,idx_poz_1)),1),dane_test(ind_poz_1(ismember(ind_poz_1,idx_poz_1)),2),'g+',...

dane_test(ind_neg_1(ismember(ind_neg_1,idx_poz_1)),1),dane_test(ind_neg_1(ismember(ind_neg_1,idx_poz_1)),2),'r+',...

dane_test(ind_poz_2(ismember(ind_poz_2,idx_poz_2)),1),dane_test(ind_poz_2(ismember(ind_poz_2,idx_poz_2)),2),'r*',...

dane_test(ind_neg_2(ismember(ind_neg_2,idx_poz_2)),1),dane_test(ind_neg_2(ismember(ind_neg_2,idx_poz_2)),2),'g*')

hold on
for i=1:size(w_1,2)
    a = (-(bin_1(i)/w_1(2,i))/(bin_1(i)/w_1(1,i)));
    b = (-(bin_1(i)/w_1(2,i)));
    x_line = [-1 1];
    y_line = a*x_line+b;
    plot(x_line,y_line,'k')
    a = (-(bin_2(i)/w_2(2,i))/(bin_2(i)/w_2(1,i)));
    b = (-(bin_2(i)/w_2(2,i)));
    x_line = [-1 1];
    y_line = a*x_line+b;
    plot(x_line,y_line,'k')
end
hold off
xlim([-1 1])
ylim([-1 1])
title('Dane testowe')
xlabel('x_3')
ylabel('x_4')
legend('TP', 'FP', 'TN', 'FN')

```