Klaudia Pakos

IO, rok 3, 267561

PODSTAWY SZTUCZNEJ INTELIGENCJI

SPRAWOZDANIE NR 2

BUDOWA I DZIAŁANIE SIECI JEDNOWARSTWOWEJ

1. Cel ćwiczenia

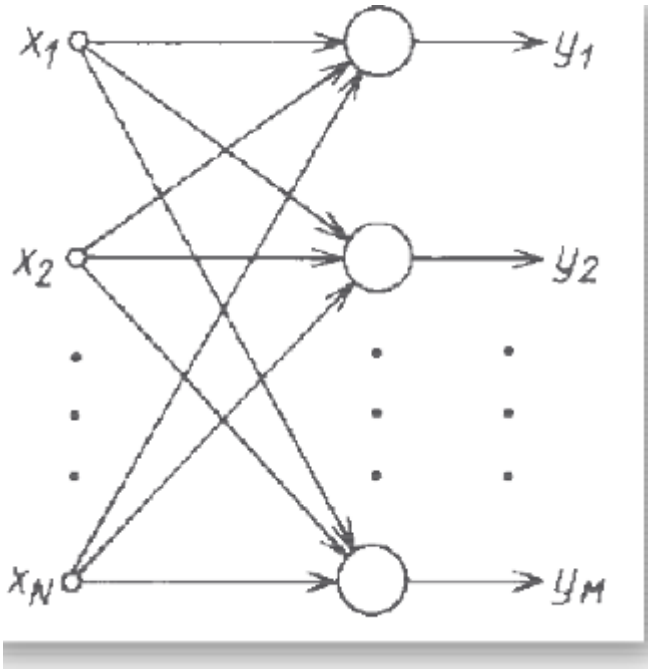
Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

2. Zadania do wykonania

a) Wygenerowanie danych uczących i testujących, zawierających 10 dużych i 10 małych liter dowolnie wybranego alfabetu w postaci dwuwymiarowej tablicy
b) Przygotowanie (implementacja lub wykorzystanie gotowych narzędzi) dwóch jednowarstwowych sieci - każda wg. innego algorytmu podanego na wykładzie.
c) Uczenie sieci dla przy różnych współczynnikach uczenia.
d) Testowanie sieci.

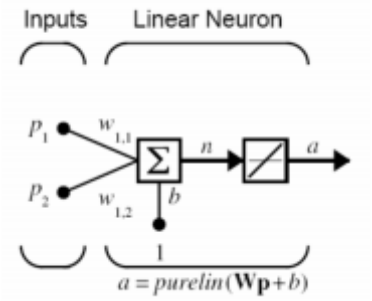
3. Opis sieci jednowarstwowej

Sieć neuronowa to zbiór neuronów realizujących różne cele. W przypadku sztucznych sieci neuronowych struktura ta jest sztuczna, jednak jej zadaniem jest modelować działanie naturalnego układu nerwowego (mózgu).
Sieć jednokierunkowa składa się zaś z neuronów ułożonych w taki sposób, że istnieje jeden kierunek przepływu.
Sieć jednowarstwowa posiada zaś tylko jedną warstwę neuronów, zatem ich zastosowanie pozwala na rozwiązanie niewielu problemów. Na węzłach wchodzących nie znajdują się warstwy neuronów, gdyż nie zachodzi w nich żaden proces obliczeniowy. Dobór wag następuje w procesie uczenia, czyli dopasowaniu sygnałów wyjściowych do wartości oczekiwanych.

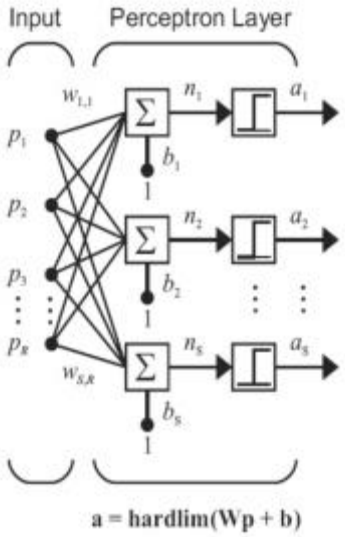


4. Przebieg zadania

Wykorzystano dwa różne algorytmy.
Pierwszy to funkcja newlin, gdzie utworzony zostaje neuron liniowy.



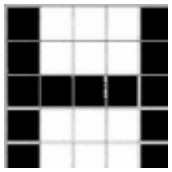
Drugi, to newp(PR,S), gdzie utworzony zostaje perceptron.



Wybrano 10 pierwszych liter alfabetu- dużych i małych (łącznie 20), które mieściły się w tablicy 5x5 oraz wprowadzono je to programu w celu umożliwienia testowania działania sieci.

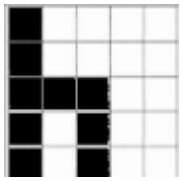
Umieszczanie liter w tablicy polegało na zinterpretowaniu danego pola przez sieć. Tablica składa się z 25 pól, które odpowiednio są czarne - 1, lub białe - 0. Czarne pole oznaczało, że w danym miejscu występował fragment litery, np. dla dużej i małej litery H wygląda to następująco:

1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1



Następnie dla litery H wygenerowana interpretacja jest zapisywana w postaci ciągu zer i jedynek:
H = [1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1];

1 0 0 0 0
1 0 0 0 0
1 1 1 0 0
1 0 1 0 0
1 0 1 0 0



Oraz dla litery h wygenerowana interpretacja jest zapisywana w postaci ciągu zer i jedynek:
h = [1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 0 0];

Następnie nauczono sieć rozpoznawać duże i małe litery, w wyniku czego mogliśmy otrzymać komunikat o wielkości testowanej litery.

Kod programu wraz komentarzem przedstawiono poniżej:

```
close all; clear all; clc;

%Możliwe wartości początkowe- 0 lub 1; 25 wejsc do
sieci( bo litery 5x5)
start = [0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
0 1; 0 1; 0 1; 0 1;
         0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
0 1; 0 1; 0 1;];

%maksymalna ilosc wyjsc
exit = 1;

%metoda 1
%net = newlin(start, exit);

%metoda 2
net = newp(start, exit);

%kolumnowa reprezentacja binarna pierwszych 10 liter
alfabetu dla tablicy 5x5
     %A a B b C c D d E e F f H h I i K k L l
IN = [0 0 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1;
```

```
        1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0;
        1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0;
        1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;

         1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1;
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
         0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0;
         1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;

         1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1;
         1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 0 0 0;
         1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 0 1 0 0;
         1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0;
         1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;

         1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0;
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0;
         0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
         1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;

         1 0 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1;
         0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1;
         0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 1 1;
         0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0;
         1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;];
         %A a B b C c D d E e F f H h I i K k L l
OUT =  [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0];

%parametry treningu sieci
net.trainParam.epochs = 10000;            %max ilosc
epok
net.trainParam.goal = 0.01;               %blad
net.trainParam.mu = 0.5;                 %wspolczynnik
uczenia

%uczenie sieci
net = train(net, IN, OUT);

%dane do testow
A_test = [0; 1; 1; 1; 0;
          1; 0; 0; 0; 1;
          1; 1; 1; 1; 1;
          1; 0; 0; 0; 1;
          1; 0; 0; 0; 1];

a_test = [0; 1; 1; 0; 0;
          0; 0; 0; 1; 0;
          0; 1; 1; 1; 0;
```

```
          1; 0; 0; 1; 0;
          0; 1; 1; 1; 1];

B_test = [1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0];

b_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0];

C_test = [0; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 1; 0];

c_test = [0; 0; 0; 0; 0;
          0; 0; 0; 0; 0;
          0; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 0; 0];

D_test = [1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 0; 0; 1; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0];

d_test = [0; 0; 0; 1; 0;
          0; 0; 0; 1; 0;
          0; 1; 1; 1; 0;
          1; 0; 0; 1; 0;
          0; 1; 1; 1; 0];

E_test = [1; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 1; 0];

e_test = [0; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 0; 0];
```

```
F_test = [1; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];

f_test = [0; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];

H_test = [1; 0; 0; 0; 1;
          1; 0; 0; 0; 1;
          1; 1; 1; 1; 1;
          1; 0; 0; 0; 1;
          1; 0; 0; 0; 1];

h_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 1; 0; 0;
          1; 0; 1; 0; 0];

I_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];

i_test = [1; 0; 0; 0; 0;
          0; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];

K_test = [1; 0; 0; 1; 0;
          1; 0; 1; 0; 0;
          1; 1; 0; 0; 0;
          1; 0; 1; 0; 0;
          1; 0; 0; 1; 0];

k_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 1; 0; 0;
          1; 1; 0; 0; 0;
          1; 0; 1; 0; 0];

L_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
```

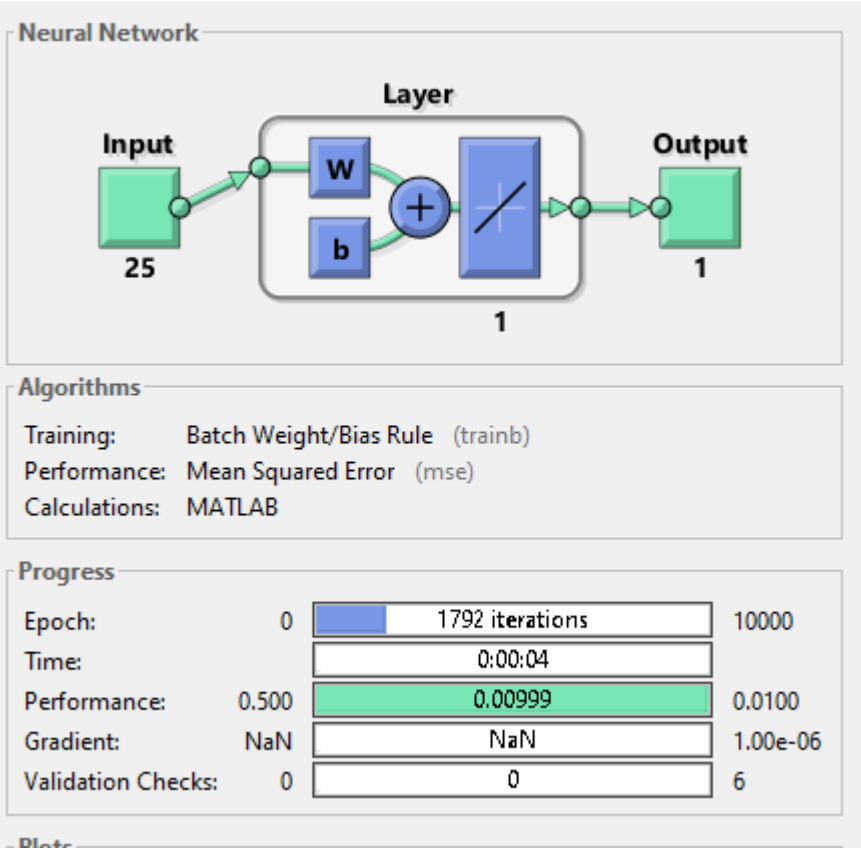
```
                1; 0; 0; 0; 0;
                1; 1; 1; 1; 0];

l_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0];

%testowanie dzialania
efect = [sim(net, a_test); sim(net, C_test); sim(net,
d_test); sim(net, F_test); sim(net, h_test); sim(net,
H_test); sim(net, i_test); sim(net, k_test); sim(net,
L_test); sim(net, l_test)]

%wynik
if round(efect) == 0
    disp('Litera mala');
else
    disp('Litera duza');
end
```

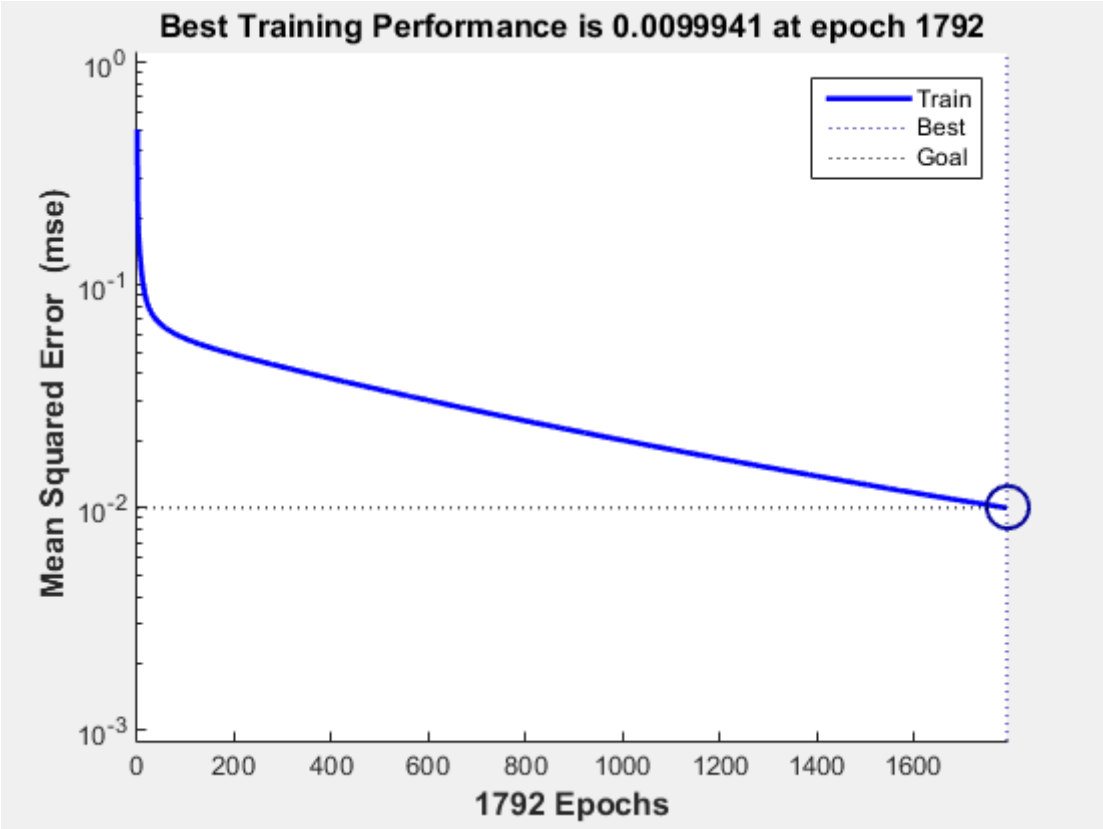
5. Przedstawienie wyników

Przeprowadzono testy dla różnych wartości współczynnika uczenia się (WU) dla 10 przykładowych danych testowych dla obu algorytmów. Otrzymane wyniki przedstawiono poniżej. Program na podstawie otrzymanych wartości klasyfikował litery. Dla wartości <0,5 – litery małe, dla >0,5 litery duże

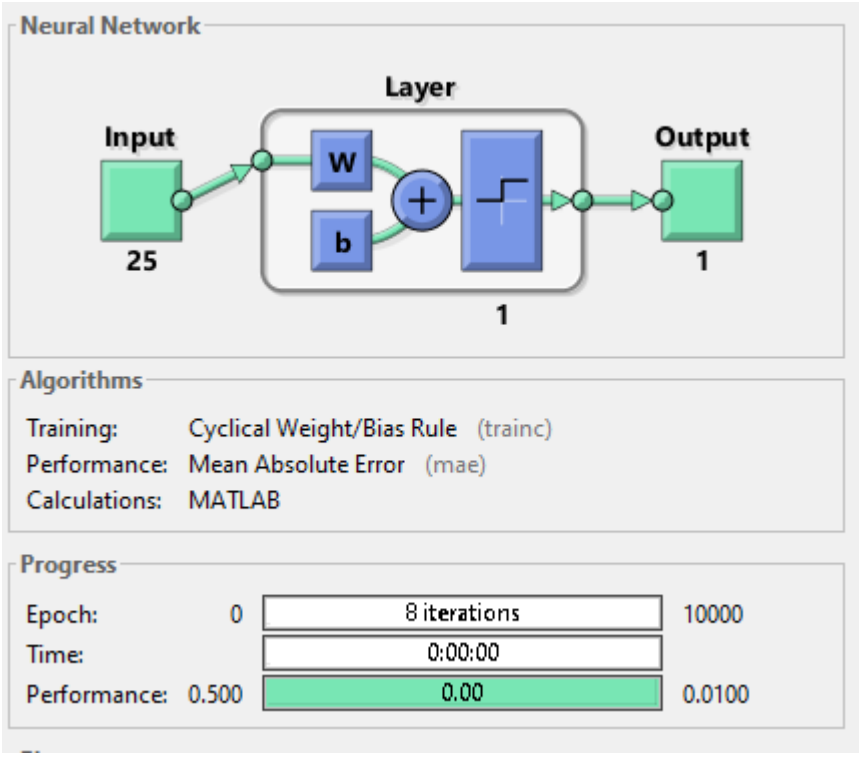
L.p.	Litera	WU=0,005		WU=0,05		WU=0,5	
		newlin	newp	newlin	newp	newlin	newp
1	a	0,0628	0	0,0628	0	0,0628	0
2	C	0,9878	1	0,9878	1	0,9878	1
3	d	0,0588	0	0,0588	0	0,0588	0
4	F	1,0065	1	1,0065	1	1,0065	1
5	h	-0,0089	0	-0,0089	0	-0,0089	0
6	H	1,0243	1	1,0243	1	1,0243	1
7	i	0,0479	0	0,0479	0	0,0479	0
8	k	0,0011	0	0,0011	0	0,0011	0
9	L	0,8603	1	0,8603	1	0,8603	1
10	l	0,31	0	0,31	0	0,31	0

Ponadto zauważono, że liczba epok niezależnie od wartości współczynnika uczenia się była stała dla każdego z algorytmów i wynosiła dla newlin 1792 iteracje, a dla newp 8 iteracji. Poniżej przedstawiono otrzymane wyniki z programu Neural Network Training (nntraintool) Dla newlin:


Neural Network
Layer
Input
W
Output
25
b
1
1
Algorithms
Training: Batch Weight/Bias Rule (trainb)
Performance: Mean Squared Error (mse)
Calculations: MATLAB
Progress
Epoch: 0 1792 iterations 10000
Time: 0:00:04
Performance: 0.500 0.00999 0.0100
Gradient: NaN NaN 1.00e-06
Validation Checks: 0 0 6


Best Training Performance is 0.0099941 at epoch 1792
Train
Best
Goal
Mean Squared Error (mse)
10^0
10^-1
10^-2
10^-3
0
200
400
600
800
1000
1200
1400
1600
1792 Epochs

Dla newp:


Neural Network
Layer
Input
W
b
Output
25
1
1
Algorithms
Training: Cyclical Weight/Bias Rule (trainc)
Performance: Mean Absolute Error (mae)
Calculations: MATLAB
Progress
Epoch: 0 8 iterations 10000
Time: 0:00:00
Performance: 0.500 0.00 0.0100


Best Training Performance is 0.5 at epoch 0
Train
Best
Goal
Mean Absolute Error (mae)
8 Epochs

6. Wnioski

Na podstawie wykonanych testów stwierdzono, że współczynnik uczenia nie miał w tym przypadku wpływu na wynik- były one takie same dla każdego ze współczynników. Nie miał on również wpływu na ilość wykonanych epok. Może to być spowodowane wyborem współczynników uczenia. Łatwo zauważyć za to, że oba algorytmy zwracały poprawne wartości. Przy czym funkcja newline potrzebowała dużo razy więcej epok niż funkcja newp. Zatem czas działania funkcji newline jest dłuższy niż newp. Ponadto zaletą funkcji newp jest jej jednoznaczność, w jej wyniku otrzymać możemy jedynie wartości 0 lub 1. Zatem rekomendowanym algorytmem do rozwiązania tego problemu jest newp z wykorzystaniem perceptronu, ponieważ używając mniej epok daje poprawne wyniki.