Alicja Kapiszka, Informatyka N1, gr. 30C

**Sieci neuronowe – sprawozdanie nr 2**

**Zadanie 1.**

Przygotować dane i dobrać najprostszą sieć realizującą działanie bramki logicznej XOR.

* Dane wejściowe i wyjściowe:

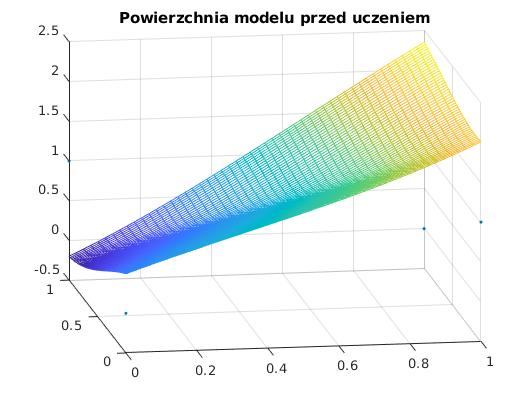
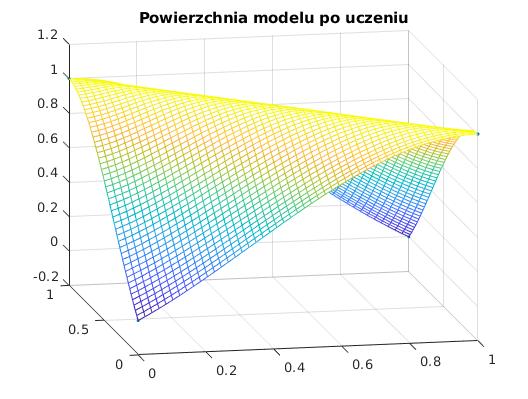
we = [0 0; 0 1; 1 0; 1 1]';

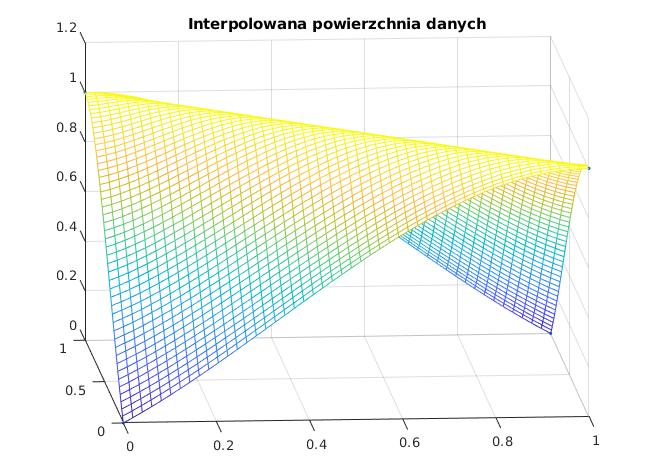
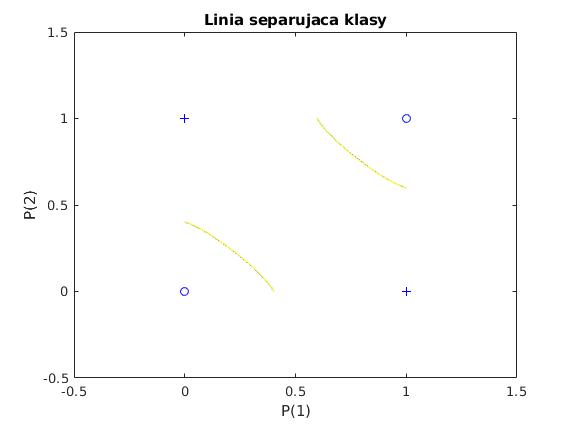
wy = [0; 1; 1; 0]';

* Najprotsza sieć realizująca bramkę:

net = newff(minmax(we),[3 1],{'tansig', 'purelin'});

* Jest to sieć dwuwarstwowa mająca 3 neurony na warstwie ukrytej I 1 neuron na warstwie wyjściowej, przy czym na warstwie ukrytej neurony mają funkcję aktywacji typu tangens hiperboliczny, a na wyjściowej – funkcję liniową.
* Poniżej przedstawiono porównanie powierzchni modelu przed uczeniem oraz po, a także linię separującą dane.



**Zadanie 2.**

Dla wybranych 4 plików z danymi należało: zapoznać się z danymi, dobrać strukturę sieci neuronowej i przeprowadzić uczenie oraz wyznaczyć błąd.

Wytłuszczoną czcionką zaznaczyłam najlepszą według mnie opcję jeśli chodzi o złożoność i wydajność.

* dane\_3D\_kapitan – 3 wejścia (prędkości statku), 1 wyjście (ocena stopnia niebezpieczeństwa: 0, 0.5, 1), aproksymacja:

|  |  |  |
| --- | --- | --- |
| Struktura sieci | Rodzaje funkcji aktywacji | Błąd danych uczących |
| [10 10 1] | tansig, tansig, purelin | 0,00450 |
| [10 10 1] | logsig, logsig, purelin | 0,00463 |
| **[10 10 1]** | **logsig, tansig, purelin** | **0,00410** |
| [7 7 1] | logsig, tansig, purelin | 0,0141 |

* dane\_5D\_parity – 5 wejść, 1 wyjście (określające parzystość sumy wejść: 0 lub 1), klasyfikacja:

|  |  |  |
| --- | --- | --- |
| Struktura sieci | Rodzaje funkcji aktywacji | Błąd danych uczących |
| [10 1] | tansig, purelin | 1,42 \* 10-22 |
| [10 1] | logsig, purelin | 7,48 \* 10-19 |
| [5 1] | tansig, purelin | 0,134 |
| **[5 1]** | **logsig, purelin** | **1,53 \* 10-7** |
| [2 1] | tansig, purelin | 0,186 |

* dane\_9D\_glass – 9 wejść (parametry fizyko-chemiczne szkła), 6 wyjść (klasa szkła), klasyfikacja:

|  |  |  |
| --- | --- | --- |
| Struktura sieci | Rodzaje funkcji aktywacji | Błąd danych uczących |
| [8 6] | tansig, logsig | 0,0164 |
| [8 8 6] | tansig, tansig, purelin | 0,0183 |
| **[8 8 6]** | **tansig, tansig, logsig** | **0,00454** |
| [8 8 6] | logsig, logsig, purelin | 0,0146 |
| [8 8 6] | logsig, tansig, purelin | 0,0112 |
| [8 8 6] | logsig, tansig, logsig | 0,0106 |
| [5 5 6] | logsig, tansig, purelin | 0,0344 |

* dane\_35D\_heart – 35 wejść (dane o pacjentach), 1 wyjście (ryzyko zachorowania na choroby serca), klasyfikacja (binarna):

|  |  |  |
| --- | --- | --- |
| Struktura sieci | Rodzaje funkcji aktywacji | Błąd danych uczących |
| [10 10 1] | tansig, tansig, logsig | 0,0141 |
| [8 8 1] | tansig, tansig, logsig | 0,0192 |
| **[8 8 1]** | **logsig, tansig, logsig** | **0,0130** |
| [8 8 1] | logsig, logsig, logsig | 0,0159 |
| [5 5 1] | tansig, tansig, logsig | 0,0380 |

**Zadanie 3.**

Opracować skrypt, który podzieli dane na część uczącą i testującą w zadanej proporcji oraz dobierze optymalną ilość neuronów na warstwie ukrytej.

close all;

clc;

% loading and random sorting of the data

load dane\_5D\_parity\_i.txt

load dane\_5D\_parity\_o.txt

we = dane\_5D\_parity\_i;

wy = dane\_5D\_parity\_o;

sort = randperm(length(wy));

we = we(sort,:)';

wy = wy(sort)';

% dividing the data to the learning and test data

in\_learn = [];

in\_test = [];

out\_learn = [];

out\_test = [];

procent = 20;

ratio = procent/100 \* length(wy);

for i = 1:length(wy)

if i <= ratio

in\_test = [in\_test we(:,1)];

out\_test = [out\_test wy(i)];

else

in\_learn = [in\_learn we(:,1)];

out\_learn = [out\_learn wy(i)];

end

end

max = 10; % max neurons amount

v = []; % errors vector

% learning and testing the net

for n = 1:max

net = newff(in\_learn, out\_learn, [n], {'tansig', 'purelin'});

net = init(net);

net.trainParam.epochs = 100;

net = train(net, in\_learn, out\_learn);

y = sim(net, in\_test);

d = 0;

for i = 1:length(out\_test)

d = d + (out\_test(i) - y(i))^2;

end

v = [v d];

end

% finding the best structure of the net

[min\_d, neu] = min(v);

neu % the amount of neurons on the hidden layer

figure;

hold on;

plot(1:max, v); % error plot

Działanie skryptu dla danych „5D\_parity”:

* pliki zawierają 32 zestawy danych wejściowych i wyjściowych, z czego po 20% (czyli 6 zestawów) zostało przeznaczone do testowania, a pozostała część – do nauki neuronów
* wyznaczona optymalna ilość neuronów na warstwie ukrytej – 4
* wykres zależności obliczonego błędu od ilości neuronów na warstwie ukrytej:

