

29. Diagnostyka raka piersi za pomocą sieci MLP

Justyna Budzyńska

Katarzyna Latos

Warszawa, dn. 14.05.2022

| | |
|-------------------------------------|----|
| Opis implementacji sieci neuronowej | 2 |
| Struktura sieci | 2 |
| Algorytm uczenia sieci | 4 |
| Wstępne testy | 5 |
| Listing kodu | 23 |
| Bibliografia | 25 |

Opis implementacji sieci neuronowej

Na początku naszej pracy z implementacją sieci neuronowej załadowaliśmy bazę danych korzystając z wbudowanej funkcji programu MATLAB, jaką jest *Import Data* (przy okazji generując kod dla tego procesu).

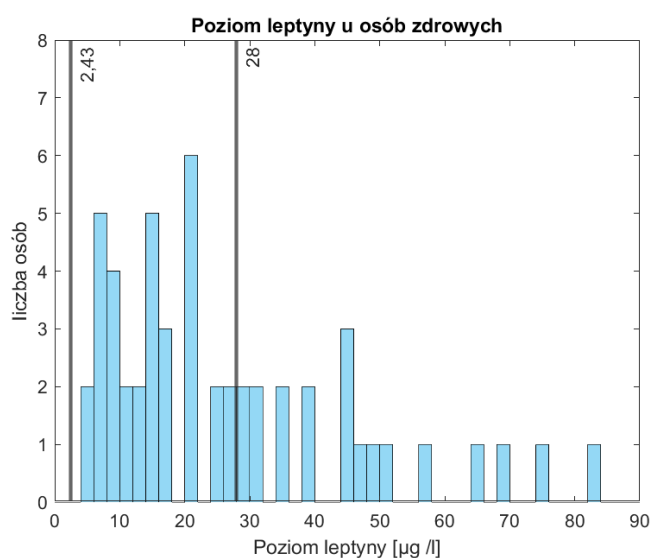
Ponieważ postanowiliśmy pracować na oddzielnym pliku (nie na tym, który posłużył nam do analizy danych podczas etapu I) należało ponownie przeprowadzić normalizację danych. Ten zabieg stosuje się w celu ujednolicenia wartości danych, inaczej mówiąc pozbycia się dużych różnic w zakresach wartości poszczególnych cech. W tym celu wykorzystaliśmy funkcję wbudowaną, jaką jest funkcja *normalize*. Dane po normalizacji przyjmują wartości z zakresu od 0 do 1.

Następnie należało podzielić dane na zbiór danych trenujących i testujących. Wykorzystaliśmy do tego funkcję *cvpartition*. Postanowiliśmy sprawdzić wyniki uczenia sieci dla różnych podziałów danych. Szczegółowiej opiszemy to w rozdziale *Wstępne testy*.

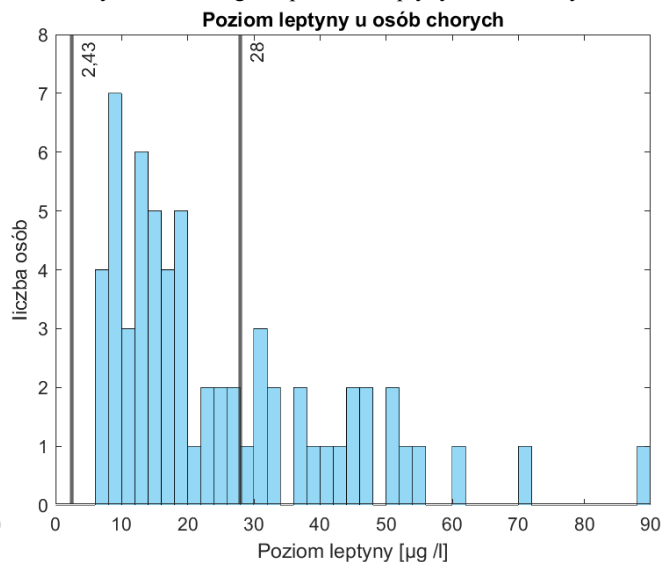
Struktura sieci

W porównaniu do etapu I została zmieniona koncepcja naszej sieci - ograniczyliśmy liczbę warstw ukrytych do jednej. Decyzja ta została podjęta, aby przyspieszyć proces uczenia naszej sieci. Przy implementacji zrezygnowaliśmy z dwóch cech, jakimi są leptyna i adiponektyna. Zgodnie z tym, co zostało zauważone podczas analizy danych w pierwszym etapie, cechy te nie mają dużego wpływu na diagnozę. W celu przypomnienia poniżej zostały umieszczone histogramy obu tych cech zarówno dla osób chorych jak i zdrowych.

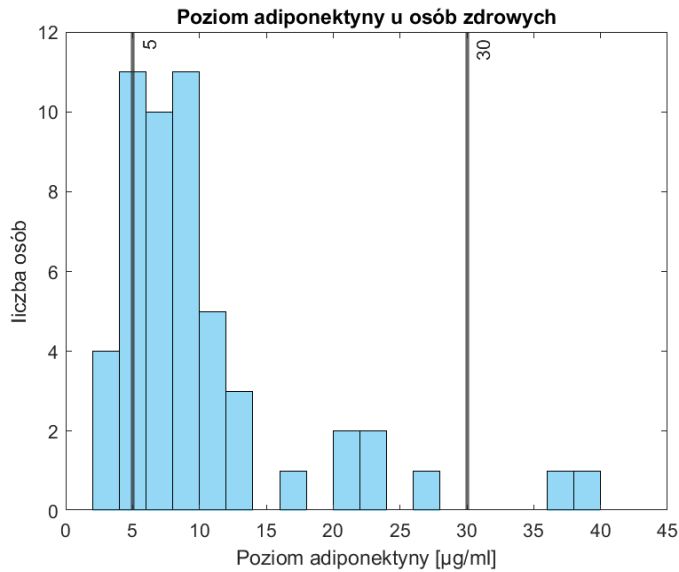
Wykres 1. Histogram poziomu leptyny u osób zdrowych



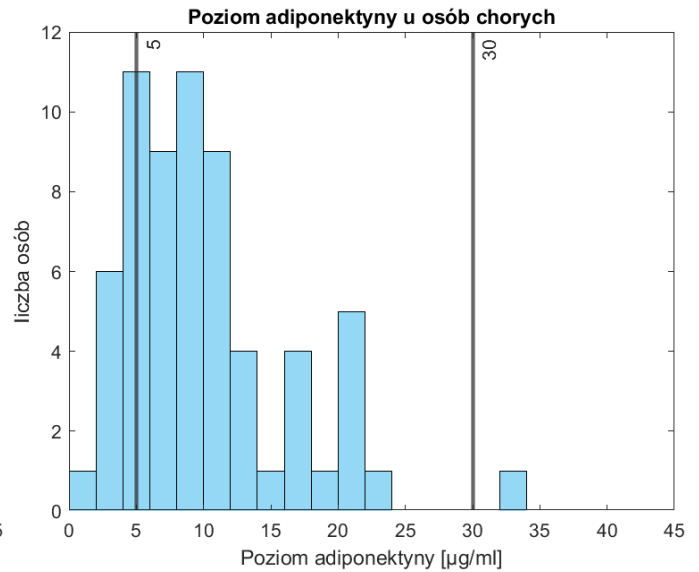
Wykres 2. Histogram poziomu leptyny u osób chorych



Wykres 3. Histogram poziomu adiponektyny u osób zdrowych



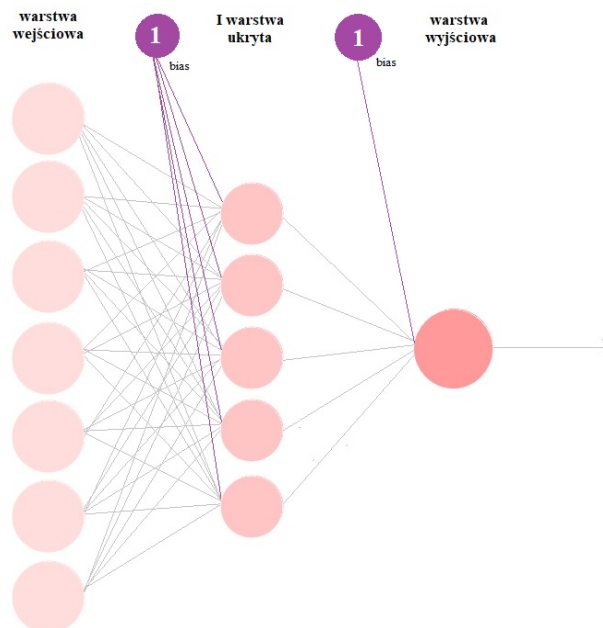
Wykres 4. Histogram poziomu adiponektyny u osób chorych



Tak więc wektor wejściowy będą stanowiły cechy takie jak: wiek, BMI, poziom glukozy, poziom insuliny, wielkość współczynnika HOMA, poziom rezystyny oraz poziom MCP-1. W warstwie ukrytej liczba neuronów uczących będzie wynosiła 5, a cechy, które się w niej znajdują będą dobierane losowo. Zarówno do warstwy wejściowej jak i ukrytej dodana zostanie wartość progowa BIAS przyjmująca wartość 1.

Na wyjściu naszej sieci będzie tylko jedna cecha, jaką jest diagnoza, czyli to czy u pacjentki wykryto raka piersi (wartość 1) czy też nie (wartość 0).

Graf 1. Graf przedstawiający strukturę sieci neuronowej



Algorytm uczenia sieci

Po zdefiniowaniu wektora cech wejściowych i wyjściowych zbioru trenującego, zdefiniowałyśmy wstępną liczbę epok ($epoch = 18\ 000$), wstępny współczynnik uczenia η ($eta = 0,04$), liczbę neuronów w warstwie ukrytej ($neurons = 5$), wartość progową BIAS jako wektor o długości równej liczbie instancji zbioru uczącego składający się z samych 1. Wartości określone jako wartości wstępne będziemy zmieniać, analizować i dostosowywać w rozdziale *Wstępne testy*. Następnie zainicjalizowałyśmy macierz wag dla cech wchodzących do pierwszej warstwy ukrytej i wyjściowej. Dobór wag początkowych dokonywany jest w sposób losowy. Wykorzystałyśmy do tego funkcję wbudowaną `rand`, która tworzy macierz wartości z zakresu od 0 do 1 o zadanych w nawiasach wymiarach.

W celu trenowania naszej sieci zastosowałyśmy *pętlę for*, która będzie uczyć naszą sieć $epoch$ liczbę razy. Algorytmem uczenia sieci na jaki się zdecydowałyśmy jest algorytm wstecznej propagacji błędów. Polega on na modyfikowaniu wektora wag w odwrotnym kierunku niż gradient funkcji błędu. Oznacza to, że najpierw modyfikowana będzie waga warstwy wyjściowej, a dopiero później waga warstwy ukrytej. Modyfikacja wag ma na celu znalezienie takiej macierzy wag, aby błąd na wyjściu sieci był jak najmniejszy.

Naukę naszej sieci rozpoczniemy od obliczenia pobudzenia neuronów warstwy ukrytej zgodnie ze wzorem:

$$v^k = \sum_{n=1}^N w_{l,n} x_n^k,$$

gdzie: w - waga neuronu.

x - neuron uczący

Jako funkcję aktywacji wybrałyśmy funkcję sigmoidalną, którą opisuje się następującym wzorem:

$$f(v) = \frac{1}{1+(e^{-v})}$$

Do jej wyznaczenia skorzystałyśmy z funkcji *logsig*, która zawarta jest w *Neural Network Toolbox*.

Obliczyłyśmy stan przewidywanych wyjść neuronów warstwy wyjściowej:

$$\xi_l^k = f(v_l^k)$$

Określiłyśmy poprawne wyjście neuronów warstwy wyjściowej dla konkretnej pacjentki. Dzięki czemu mogliśmy przystąpić do obliczenia błędu.

Sygnał błędu dla warstwy wyjściowej obliczyliśmy wykorzystując wzór:

$$\delta_m^{(o)k} = (d_m^k - y_m^k),$$

gdzie: d - poprawne wyjście neuronu warstwy wyjściowej,
 y - estymowane wyjście neuronu warstwy wyjściowej

Następnie obliczyliśmy sygnał błędu warstwy ukrytej zgodnie ze wzorem:

$$\delta_l^{(h)k} = f'(v_l^k) * \sum_{m=1}^M (\omega_{m,l} * \delta_m^{(o)k}),$$

gdzie: ω - waga neuronu

Do obliczenia pochodnej funkcji aktywacji wykorzystaliśmy funkcję *dlogsig* z pakietu *Neural Network Toolbox*.

Kolejnym krokiem była modyfikacja wagi warstwy wyjściowej:

$$\omega_{m,l}(t + 1) = \omega_{m,l}(t) + \eta * \delta_m^{(o)k} * \xi_l^k$$

Po zmodyfikowaniu wagi warstwy wyjściowej należało zmodyfikować wagę warstwy ukrytej:

$$w_{l,n}(t + 1) = w_{l,n}(t) + \eta * \delta_l^{(h)k} * x_n^k$$

Następnym krokiem było obliczenie błędu średniokwadratowego MSE:

$$MSE = \frac{1}{K} \cdot \sum_{k=1}^K (d^k - y^k)^2,$$

gdzie K - liczba wektorów wejściowych zbioru uczącego

Wstępne testy

Testowanie naszej sieci będzie polegało na obserwacji zmian zachodzących w procesie uczenia dla różnych podziałów danych na uczące i testujące. Chcemy sprawdzić działanie dla podziałów w stosunku 6 : 4, 7 : 3 oraz 8 : 2 (odpowiednio dla danych uczących i testujących). Testować będziemy również różne wartości współczynnika uczenia η , a także liczbę iteracji.

Wyzaczyliśmy także tzw. tablicę pomyłek, która posłuży nam do oceny jakości uczenia sieci neuronowej.

Opisuje ona cztery przypadki, 2 dla zgodności (zaznaczono kolorem zielonym) i 2 dla niezgodności (zaznaczono kolorem czerwonym) prognozy ze stanem faktycznym.

Tabela 1. Tablica pomyłek

| | | Stan faktyczny | |
|----------|-------------------------|---------------------------|---------------------------|
| | | dodatni (pacjentka chora) | ujemny (pacjentka zdrowa) |
| Prognoza | dodatnia (nowotwór) | prawdziwie dodatni (TP) | falszywie dodatni (FP) |
| | ujemna (brak nowotworu) | falszywie ujemny (FN) | prawdziwie ujemny (TN) |

Wykorzystując powyższą tabelę obliczyliśmy czułość i swoistość testu.

Czułość testu diagnostycznego jest to stosunek wyników prawdziwie dodatnich (TP) do sumy wyników prawdziwie dodatnich (TP) i fałszywie ujemnych (FN). Interpretowana jest ona jako zdolność testu do prawidłowego rozpoznania choroby tam, gdzie ona występuje.

$$sens = \frac{TP}{TP + FN}$$

Swoistość testu diagnostycznego jest to stosunek wyników prawdziwie ujemnych (TN) do sumy wyników prawdziwie ujemnych (TN) i fałszywie dodatnich (FP).

$$spec = \frac{TN}{TN + FP}$$

Dodatkowo policzyliśmy wartości takie jak:

- wartość predykcyjną dodatnią (PPV), czyli prawdopodobieństwo, że chora pacjentka otrzyma pozytywny wynik testu

$$PPV = \frac{TP}{TP + FP}$$

- wartość predykcyjną ujemną (NPV), czyli prawdopodobieństwo, że zdrowa pacjentka otrzyma negatywny wynik testu

$$NPV = \frac{TN}{TN + FN}$$

- dokładność (ACC), czyli prawdopodobieństwo prawidłowej diagnozy z wykorzystaniem sieci neuronowej

$$ACC = \frac{TP + TN}{n},$$

gdzie n - liczba instancji w zbiorze testującym

Na początku chcieliśmy sprawdzić jakie zmiany zachodzą w procesie uczenia dla różnych stosunków danych trenujących do danych testujących. Zdecydowaliśmy się na przetestowanie trzech stosunków 6:4, 7:3 oraz 8:2. Test ten przeprowadziliśmy dla dwóch różnych par parametrów (liczby epok oraz współczynnika uczenia).

Zdecydowaliśmy się na test tylko dla jednej próby (jeden losowy podział danych) dla poszczególnych parametrów. Z tego względu wyniki mają charakter bardziej poglądowy. Chcieliśmy jednak sprawdzić, czy i jak duże znaczenie ma stosunek podziału zbioru.

Wyniki uzyskane dla 18 000 liczby epok oraz współczynnika uczenia wynoszącego 0,04:

Tabela 2. Tabela wykresów dla różnych podziałów danych dla epoch = 18 000 i $\eta = 0,04$

| | | wykresy | |
|--------------------------|-----|--|--|
| | | wykres błędu | wykres poprawności diagnozy |
| stosunek podziału zbioru | 6:4 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |
| | 7:3 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |

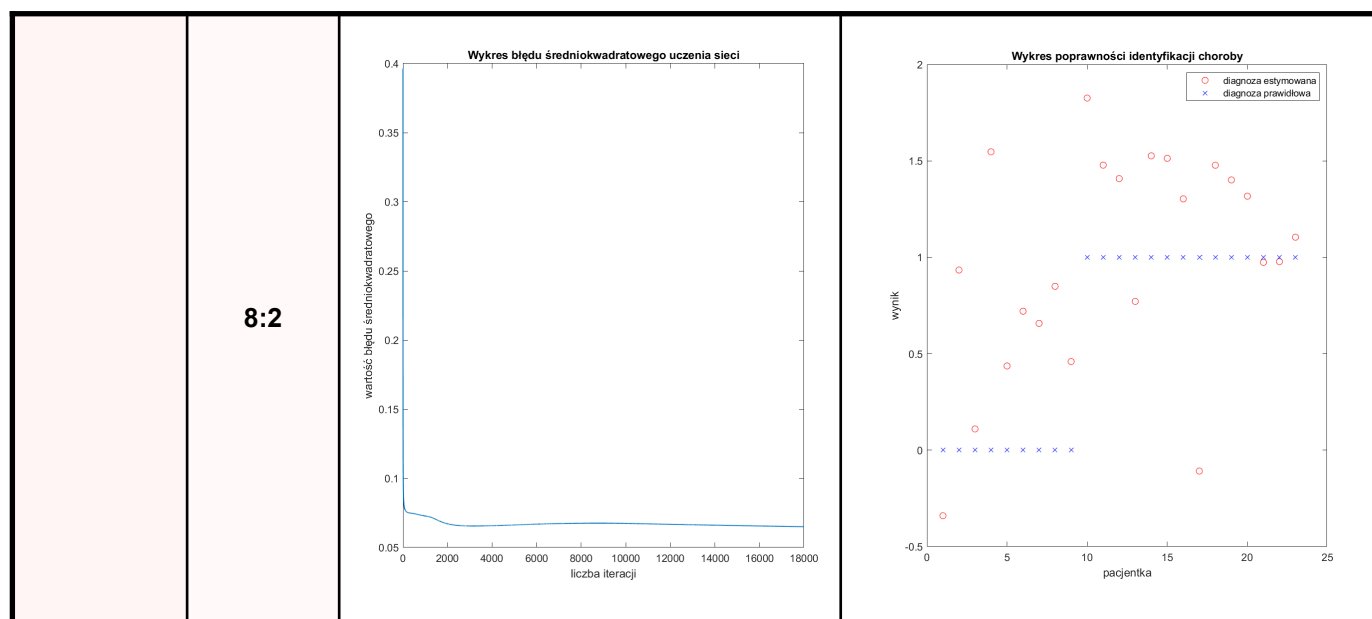


Tabela 3. Tabela wartości miar

| | Podział grupy | | |
|----------------------------------|---------------|-------|-------|
| | 6:4 | 7:3 | 8:2 |
| czułość [%] | 96,15 | 83,33 | 92,86 |
| swoistość [%] | 45 | 50,00 | 44,44 |
| PPV [%] | 69,44 | 65,20 | 72,20 |
| NPV [%] | 90 | 72,70 | 80,00 |
| ACC [%] | 73,19 | 67,60 | 73,90 |
| <i>epoch = 18000, eta = 0,04</i> | | | |

Wyniki uzyskane dla 12 000 liczby epok oraz współczynnika uczenia wynoszącego 0,05:

Tabela 4. Tabela wykresów dla różnych podziałów danych dla epoch = 12 000 i $\eta = 0,05$

| | | wykresy | |
|--------------------------|-----|--|--|
| | | wykres błędu | wykres poprawności diagnozy |
| stosunek podziału zbioru | 6:4 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |
| | 7:3 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |

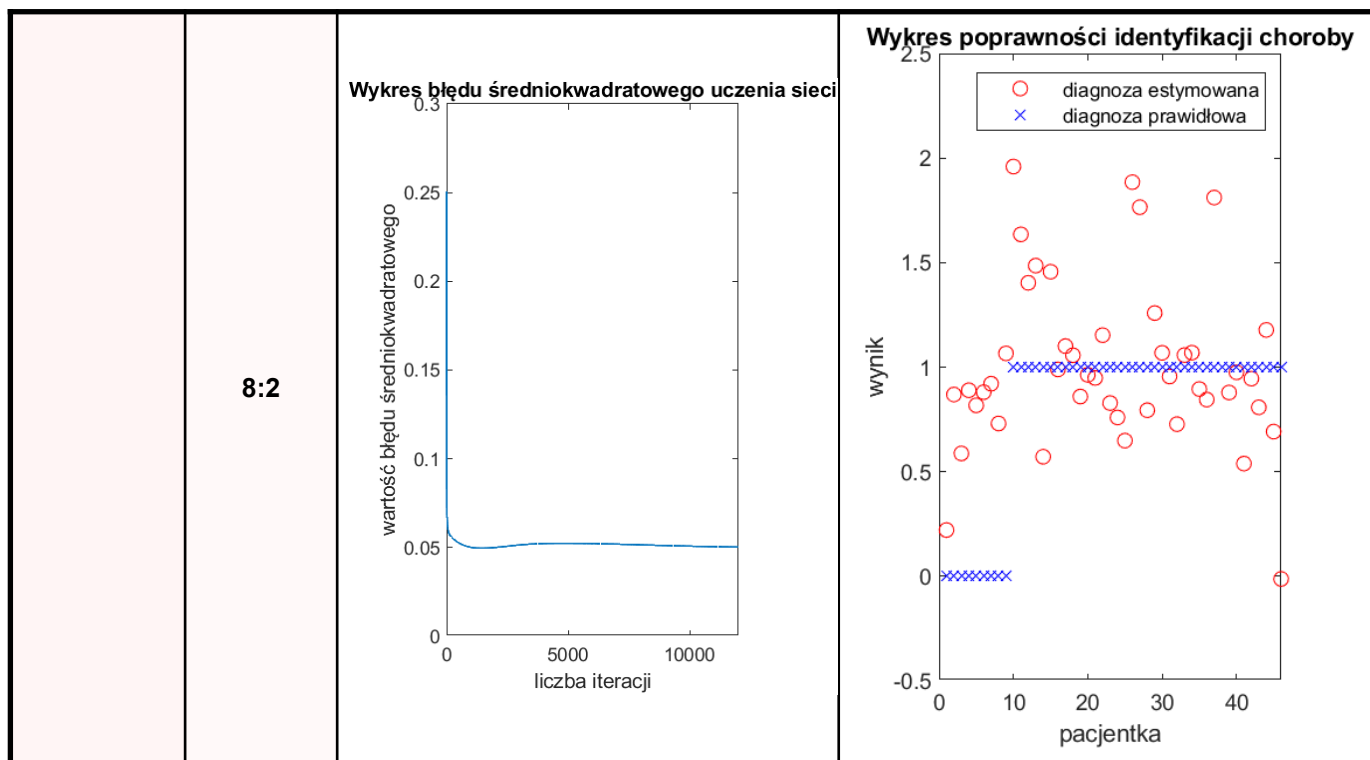


Tabela 5. Tabela wartości miar

| | Podział grupy | | |
|---------------------------|---------------|-------|-------|
| | 6:4 | 7:3 | 8:2 |
| czułość [%] | 86,21 | 86,67 | 97,30 |
| swoistość [%] | 35,29 | 37,50 | 11,11 |
| PPV [%] | 69,44 | 72,22 | 81,82 |
| NPV [%] | 60 | 61 | 50 |
| ACC [%] | 67,39 | 94,12 | 75,27 |
| epoch = 12000, eta = 0,05 | | | |

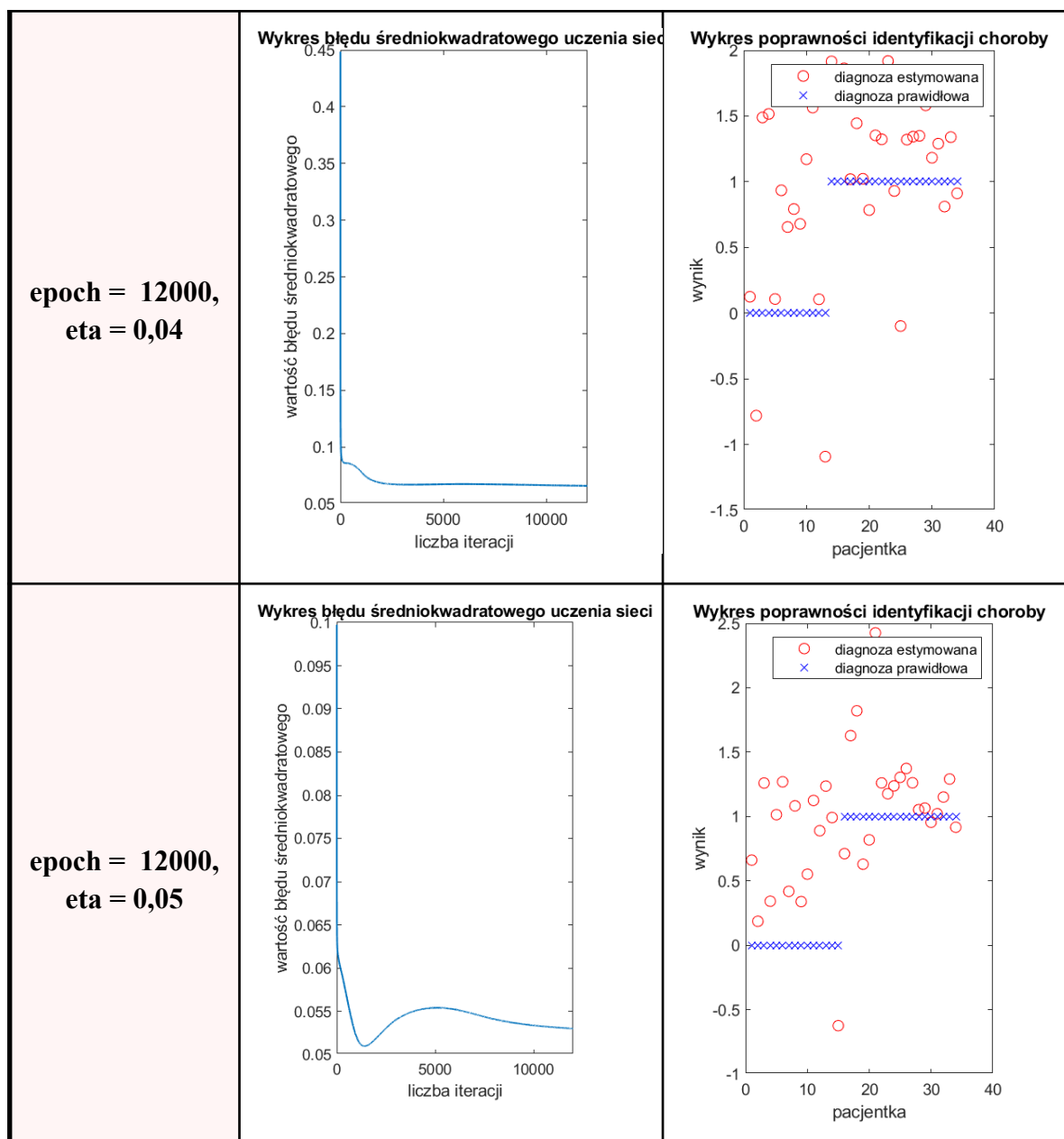
Po przeanalizowaniu powyższych danych doszliśmy do wniosku, że najlepszym podziałem jest podział 7:3. W przypadku 8:2 dla większości pomiarów otrzymano bardzo niski wynik współczynnika dokładności (ACC), co oznacza niskie prawdopodobieństwo prawidłowej diagnozy. Zaletą podziału 8:2 jest niewątpliwie bardzo wysoki poziom wartości predykcyjnej ujemnej (NPV), jednakże pozostałe wartości (niski poziom wartości predykcyjnej dodatniej (PPV) oraz niska swoistość) spowodowały, że zdecydowaliśmy się na jego odrzucenie. Dla podziału 6:4 dla *epoch*=18000 oraz *eta*=0,04 otrzymano bardzo dobre wyniki swoistości, czułości oraz wartości predykcyjnej ujemnej (NPV). Jednakże przy zmianie wartości *eta* i *epoch* uzyskano zupełnie inne, gorsze wartości. Z tego powodu ostatecznie wyznaczyliśmy

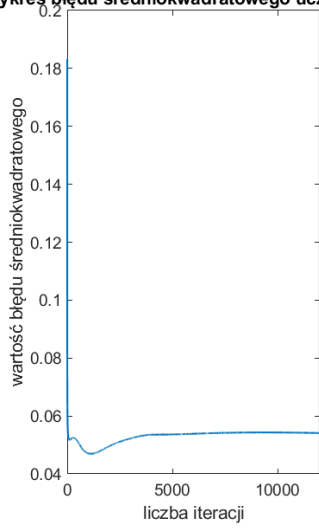
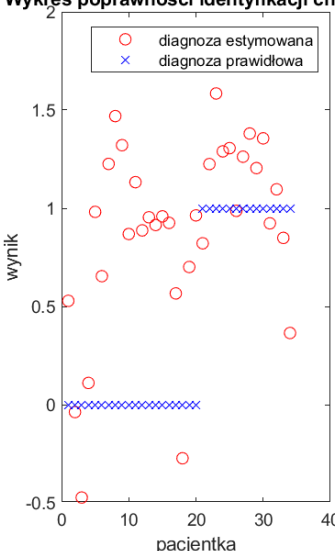
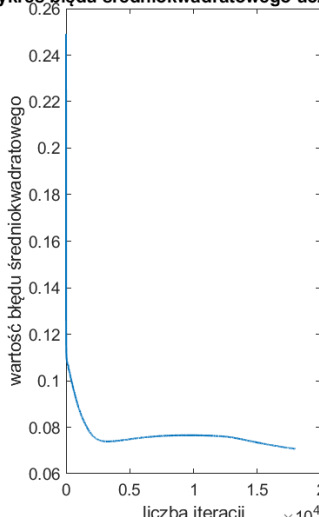
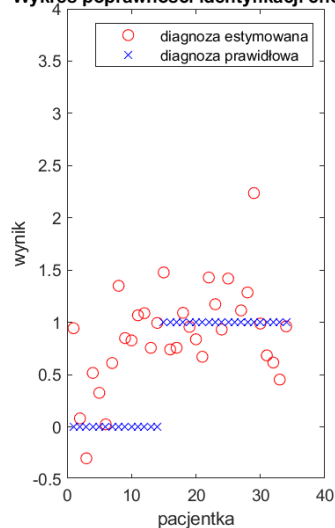
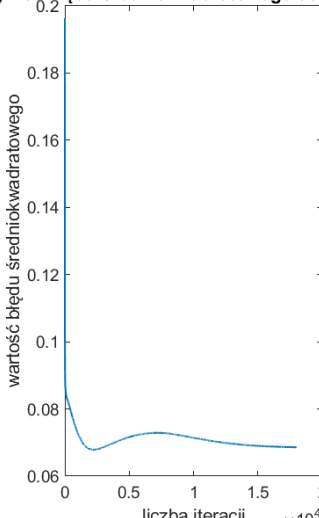
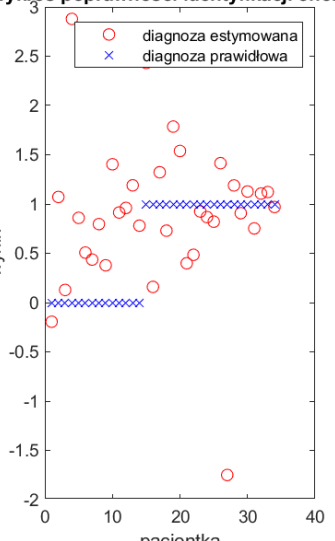
podział 7:3, ponieważ zmiana *epoch* i *eta* nie powoduje znacznej zmiany w wynikach, które są w większości przypadków dobre. Poziom współczynnika dokładności (ACC) jest zawsze wysoki, tak samo jak wartości predykcyjnej ujemnej (NPV) i dodatniej (PPV). Dodatkowo swoistość testu przy podziale 7:3 jest bliska 100%, co także jest zaletą.

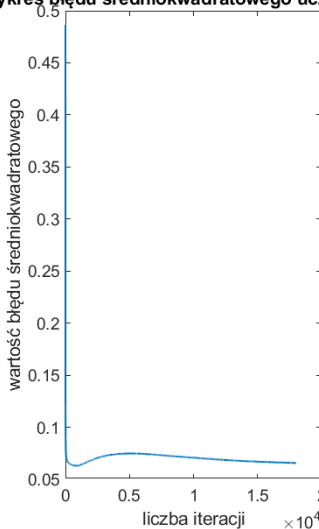
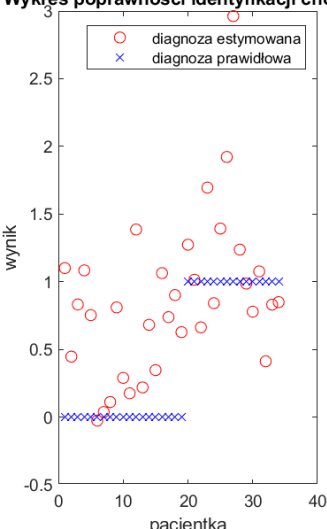
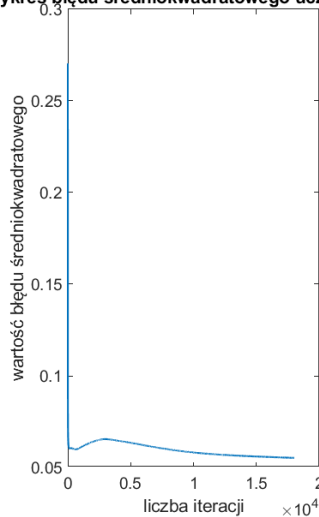
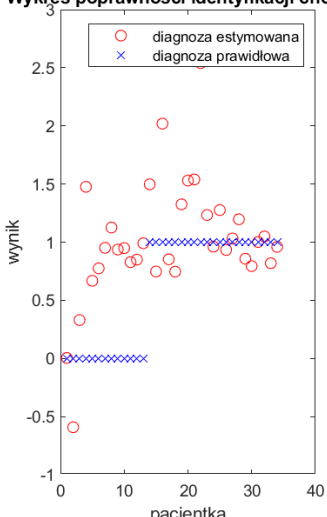
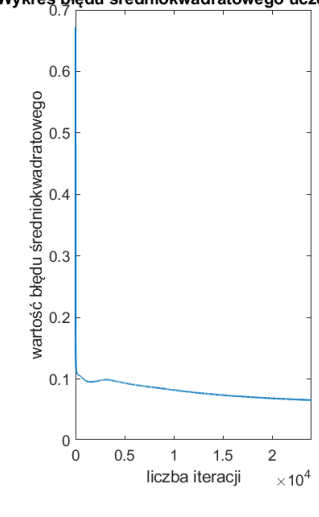
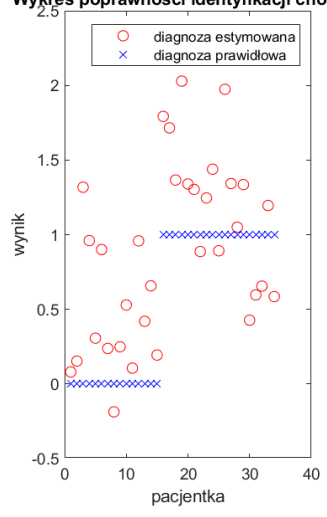
Dalsze testy wykonywano tylko dla podziału grup trenującej i testującej w stosunku 7:3. W trakcie testowania zmieniano zarówno wartość *epoch* jak i *eta*. Poniżej przedstawiono kilka otrzymanych rezultatów:

Tabela 6. Tabela wykresów MSE i poprawności diagnozy dla różnych wartości *epoch* i *eta*

| | Wykres błędu średniokwadratowego MSE | Wykres poprawności diagnozy |
|--------------------------------------|---|---|
| epoch = 12000, eta = 0,03 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |



| | | |
|---|--|---|
| <p>epoch = 12000, eta = 0,07</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 18000, eta = 0,03</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 18000, eta = 0,04</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |

| | | |
|---|---|---|
| <p>epoch = 18000, eta = 0,05</p> | <p>Wykres błęd średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 18000, eta = 0,07</p> | <p>Wykres błęd średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 24000, eta = 0,03</p> | <p>Wykres błęd średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |

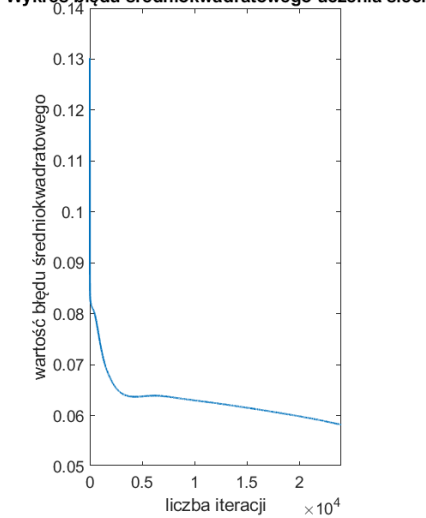
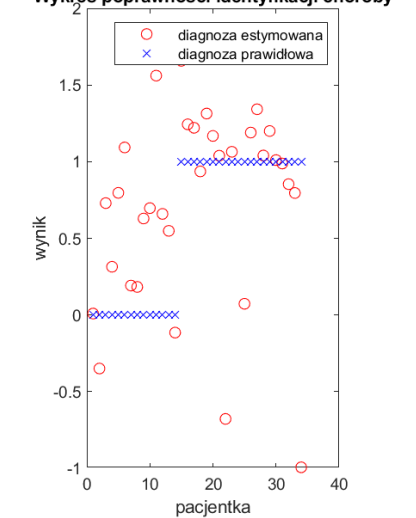
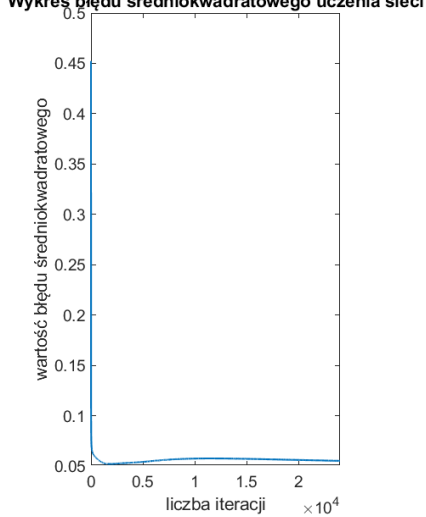
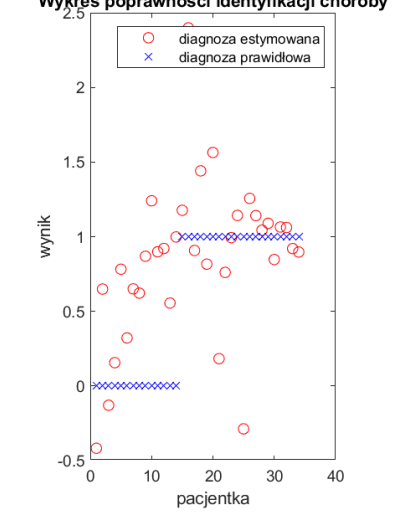
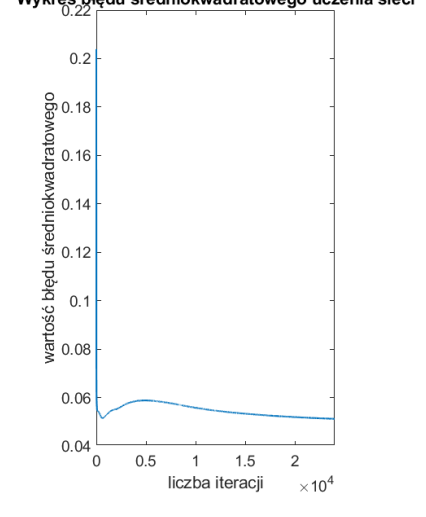
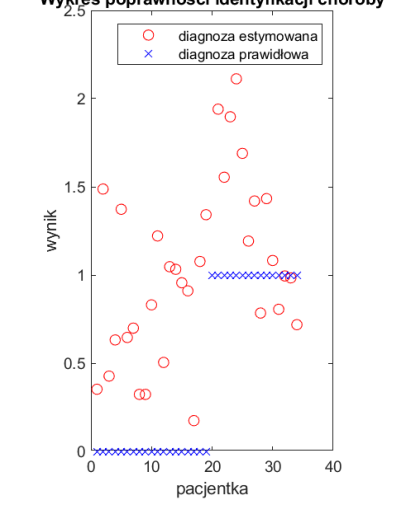
| | | |
|---|--|---|
| <p>epoch = 24000, eta = 0,04</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 24000, eta = 0,05</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |
| <p>epoch = 24000, eta = 0,07</p> | <p>Wykres błędu średniokwadratowego uczenia sieci</p>  | <p>Wykres poprawności identyfikacji choroby</p>  |

Tabela 7. Tabela wartości miar dla różnych wartości *epoch* i *eta*

| | | czułość [%] | swoistość [%] | PPV [%] | NPV [%] | ACC [%] |
|----------------------|-------------------|-------------|---------------|---------|---------|---------|
| epoch = 12000 | eta = 0,03 | 100 | 25 | 60 | 100 | 64,71 |
| | eta = 0,04 | 95,24 | 38,46 | 71,43 | 83,33 | 73,53 |
| | eta = 0,05 | 100 | 33,33 | 65,52 | 100 | 70,59 |
| | eta = 0,07 | 92,86 | 20 | 44,83 | 80 | 50 |
| epoch = 18000 | eta = 0,03 | 95 | 28,57 | 65,52 | 80 | 67,65 |
| | eta = 0,04 | 80 | 28,57 | 61,54 | 50 | 58,82 |
| | eta = 0,05 | 93,33 | 42,11 | 56 | 88,89 | 64,71 |
| | eta = 0,07 | 100 | 23,08 | 67,74 | 100 | 70,59 |
| epoch = 24000 | eta = 0,03 | 94,74 | 60 | 75 | 90 | 79,41 |
| | eta = 0,04 | 85 | 42,86 | 68 | 66,67 | 67,65 |
| | eta = 0,05 | 90 | 28,57 | 64,29 | 66,67 | 64,71 |
| | eta = 0,07 | 100 | 26,32 | 51,72 | 100 | 58,82 |

Analizując powyższą tabelę chcieliśmy znaleźć takie wartości *eta* i *epoch*, żeby zarówno czułość jak i swoistość była jak najwyższa. Miary te (odpowiednio) mówią nam o zdolności sieci do wykrywania osób rzeczywiście chorych i rzeczywiście zdrowych. Zdecydowanie bardziej zależy nam na dobrym wyniku czułości, gdyż dużo większe znaczenie ma wykrycie nowotworu u chorej pacjentki niż prawidłowe rozpoznanie braku nowotworu u pacjentki zdrowej. Innymi słowy, lepiej by sieć myliła się wskazując, że zdrowa pacjentka ma nowotwór (niska wartość swoistości), niż by wskazywała, że chora pacjentka jest zdrowa (niska wartość czułości).

Dla większości wykonanych przez nas testów wartość czułości przyjęła wysokie wartości (średnia wyniosła 93%). Przyglądając się wartościom swoistości zauważamy, że wartości te są zdecydowanie niższe od czułości, średnia dla tej miary wynosi 33%.

Najlepsza wartość swoistości osiągnięta jest dla *eta* = 0,03 oraz *epoch* = 24 000 (aż 60%).

Czułość dla tych parametrów przyjmuje wartość ponad 94%. Dokładność przekracza wartość 79%, co jest najlepszym wynikiem jaki osiągnięto w przeprowadzonych testach. Reszta miar dla tych parametrów również osiąga dobre wartości (wyższe niż średnia wszystkich pomiarów). Błąd średniokwadratowy MSE dla kolejnych iteracji maleje. Dla 24 000 iteracji osiąga on wartość zbliżoną do wartości 0,07.

Z powyższych względów decydujemy się na wybór tych parametrów i dalsze testy dla nich.

Kolejnym krokiem było sprawdzenie średniej dla poszczególnych miar dla 5 losowych podziałów danych.

Tabela 8. Tabela wykresów MSE i poprawności diagnozy dla $epoch = 24\ 000$ i $eta = 0,03$

| | Wykres błędu średniokwadratowego MSE | Wykres poprawności diagnozy |
|----|---|---|
| 1. | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |
| 2. | <p>Wykres błędu średniokwadratowego uczenia sieci</p> | <p>Wykres poprawności identyfikacji choroby</p> |

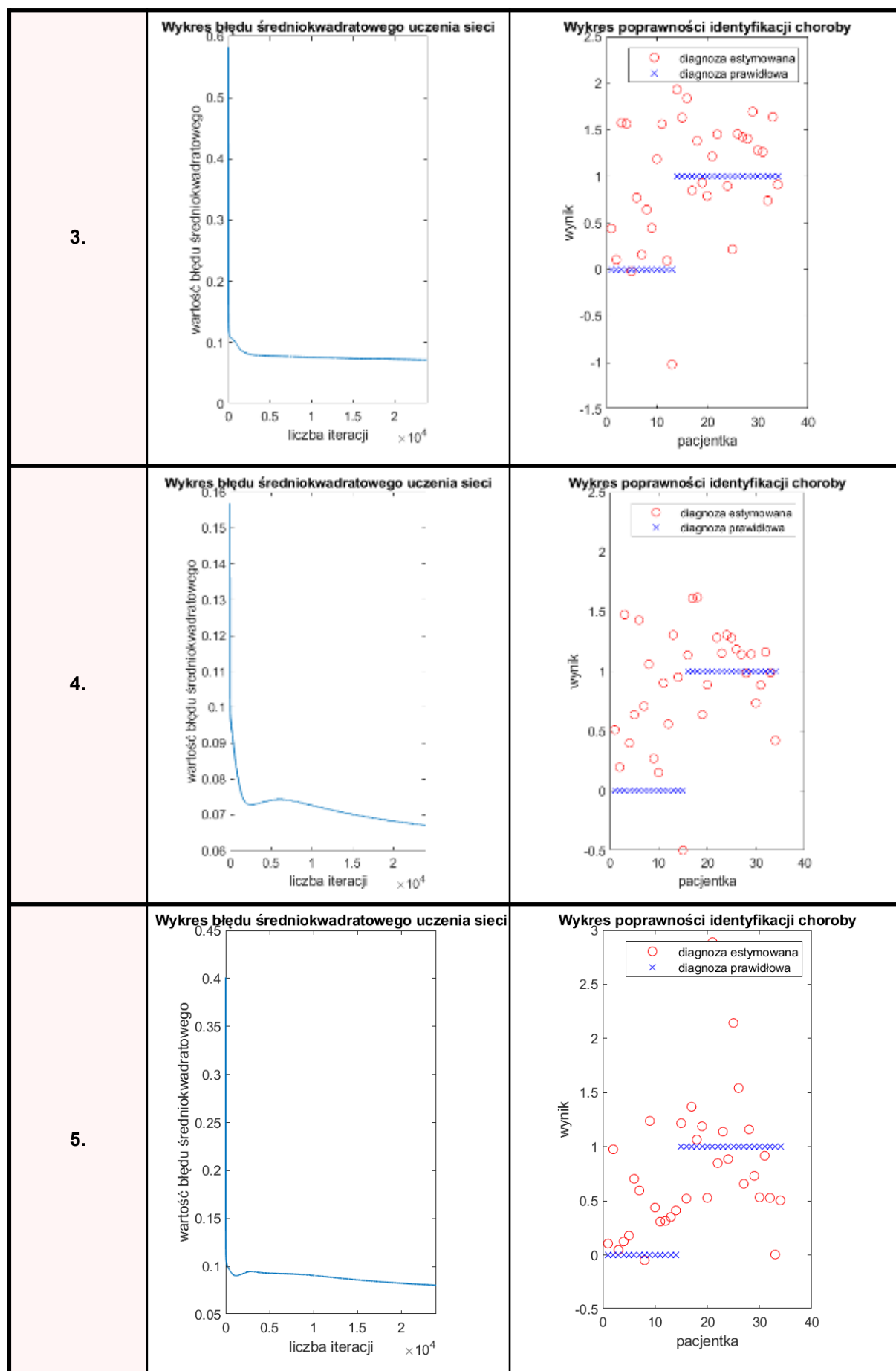


Tabela 9. Tabela wartości miar dla $epoch = 24\ 000$ i $eta = 0,03$

| | czułość [%] | swoistość [%] | PPV [%] | NPV [%] | ACC [%] |
|----------------|--------------------|----------------------|----------------|----------------|----------------|
| 1. | 94,44 | 63,5 | 73,91 | 90,91 | 79,41 |
| 2. | 82,61 | 36,36 | 73,08 | 50 | 67,65 |
| 3. | 92,24 | 53,85 | 76,92 | 87,5 | 79,41 |
| 4. | 94,74 | 33,33 | 64,29 | 83,33 | 67,65 |
| 5. | 95 | 71,43 | 82,61 | 90,91 | 85,29 |
| średnia | 91,81 | 51,69 | 74,16 | 80,53 | 75,88 |

Powyższy test potwierdził słuszność naszego wyboru. Dane jakie otrzymaliśmy w poprzednim teście nie są przypadkowe. Czułość sieci jest względnie wysoka, jej średnia wartość przekracza 90%, swoistość osiąga znacznie gorsze wyniki, ale tak jak opisywaliśmy w poprzednim punkcie, jest to dla nas mniej istotna miara. Dokładność naszej sieci wynosi prawie 75%, co oznacza, że 75% badanych osób uzyskałoby prawidłową diagnozę.

Poniżej zamieszczono również przykładową tablicę pomyłek wygenerowaną w programie:

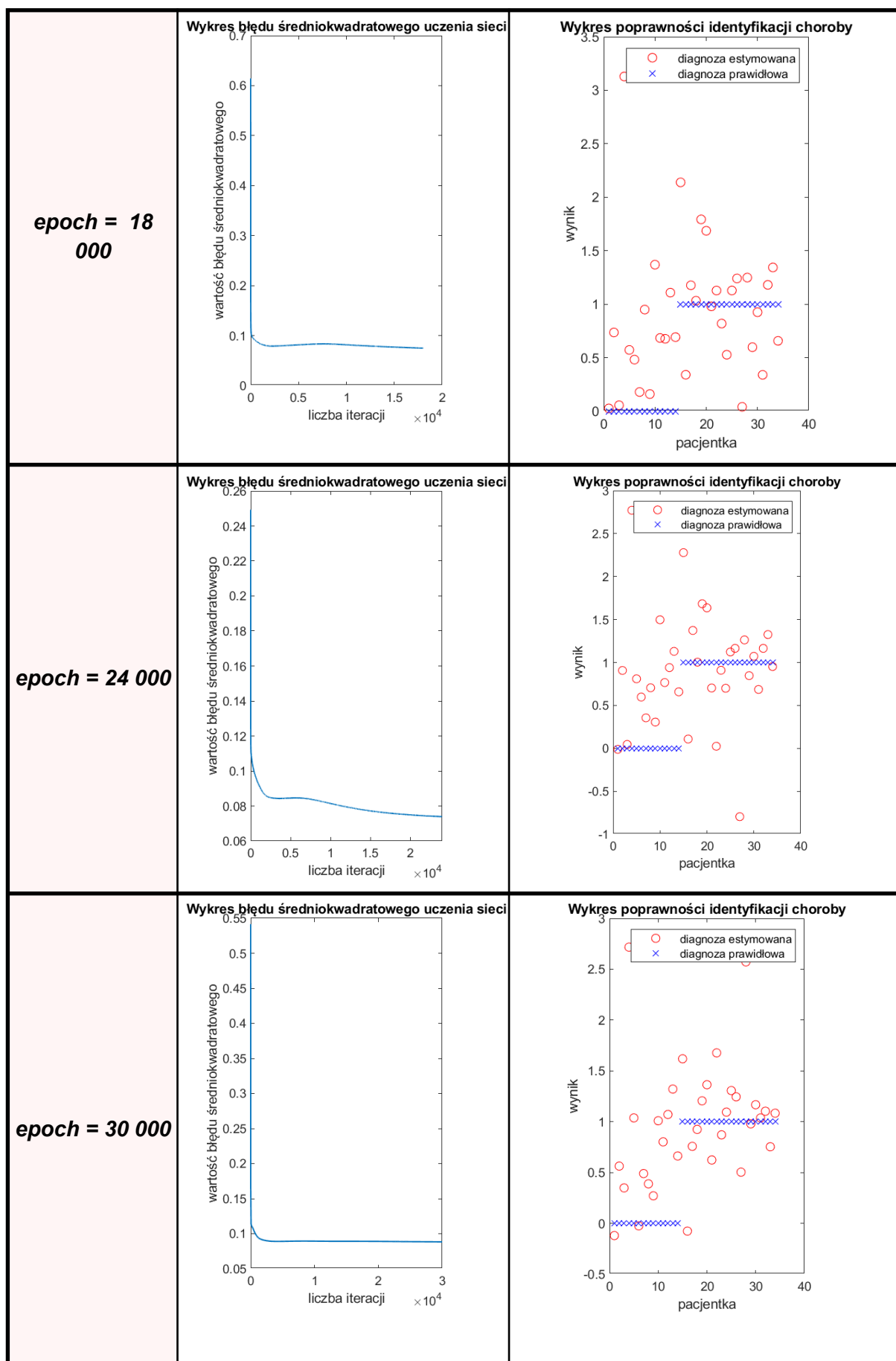
| | dodatni | ujemny |
|----------------|----------------|---------------|
| dodatni | 19.00 | 4.00 |
| ujemny | 1.00 | 10.00 |

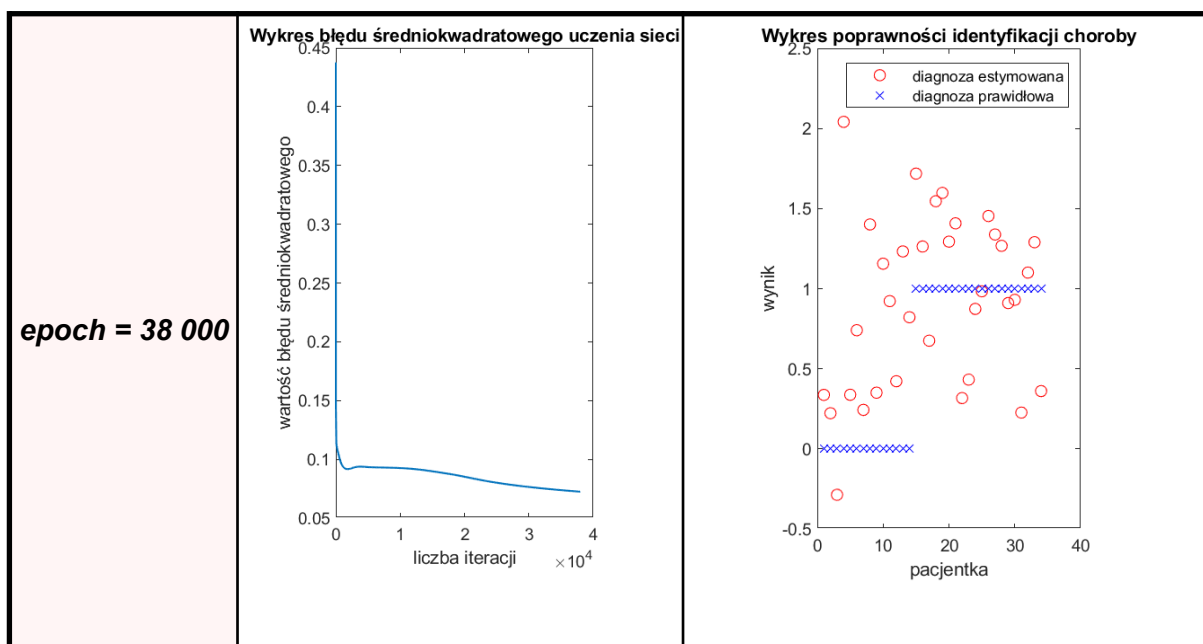
Rys. 1. Przykładowa tablica pomyłek

Chciałyśmy również sprawdzić zmianę błędu MSE w zależności od liczby iteracji (*epoch*) dla jednego współczynnika uczenia.

Tabela 10. Tabela wykresów MSE i poprawności diagnozy dla różnych *epoch* i $\eta = 0,03$

| | Wykres błędu średniokwadratowego MSE | Wykres poprawności diagnozy |
|------------------------------|--|--|
| <i>epoch</i> = 8 000 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> <p>Y-axis: wartość błędu średniokwadratowego (0.05 to 0.4) X-axis: liczba iteracji (0 to 8000)</p> | <p>Wykres poprawności identyfikacji choroby</p> <p>Y-axis: wynik (0 to 2) X-axis: pacjentka (0 to 40) Legend: ○ diagnoza estymowana, × diagnoza prawidłowa</p> |
| <i>epoch</i> = 12 000 | <p>Wykres błędu średniokwadratowego uczenia sieci</p> <p>Y-axis: wartość błędu średniokwadratowego (0.08 to 0.26) X-axis: liczba iteracji (0 to 10000)</p> | <p>Wykres poprawności identyfikacji choroby</p> <p>Y-axis: wynik (0 to 2.5) X-axis: pacjentka (0 to 40) Legend: ○ diagnoza estymowana, × diagnoza prawidłowa</p> |





Z powyższych wykresów wynika, że wraz ze zwiększaniem liczby iteracji błąd niekoniecznie musi maleć. Zarówno dla 8 000, 12 000 i 30 000 iteracji błąd dążył do wartości 0,1. Dla $epoch = 12\ 000$, $epoch = 24\ 000$ i $epoch = 38\ 000$ wartość błędu wynosiła około 0,08.

Ostatecznie jako parametry sieci uznano $epoch = 24000$ oraz $eta = 0,03$, ponieważ dla nich osiągnęte są najlepsze wartości miar. W porównaniu do pozostałych testowanych kombinacji wartości $epoch$ i eta , pozwala uzyskać znacząco lepsze wyniki uczenia sieci.

Listing kodu

```
%% Import data from text file
% Script for importing data from the following text file:
%
%      filename: C:\Users\jbudz\Desktop\studia\VI
SEM\SNB\Breast Cancer Coimbra_MLR\dataR2.csv
%
% Auto-generated by MATLAB on 27-Apr-2022
18:14:49

%% Set up the Import Options and import the data
opts = delimitedTextImportOptions("NumVariables",
10);

% Specify range and delimiter
opts.DataLines = [2, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Age", "BMI", "Glucose",
"Insulin", "HOMA", "Leptin", "Adiponectin", "Resistin",
"MCP1", "Classification"];
opts.VariableTypes = ["double", "double", "double",
"double", "double", "double", "double", "double",
"double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Import the data
dataR2 = readtable("C:\Users\jbudz\Desktop\studia\VI
SEM\SNB\Breast Cancer Coimbra_MLR\dataR2.csv",
opts);

%% Clear temporary variables
clear opts

%% Normalizacja danych
dataR2N = normalize(dataR2, 'range');

%% Podział zbioru na trenujące i testujące
c = cvpartition(size(dataR2,1),'Holdout',0.3); % podział
danych w stosunku 7:3
Ptrain = dataR2N(c.training, :); % zbiór trenujący
Ptest = dataR2N(c.test,:); % zbiór testujący

%% training and testing

% training

Xwej = [Ptrain.Age.'; Ptrain.BMI.'; Ptrain.Glucose.';
Ptrain.Insulin.'; ...
Ptrain.HOMA.'; Ptrain.Resistin.';
Ptrain.MCP1.']; % wektor cech wejściowych

OUT = [Ptrain.Classification.']; % wektor cech
wyjściowych

epoch = 24000; % liczba epok sieci (iteracje)
eta = 0.03; % współczynnik trenujący
neurons_wej = size(Xwej,1); % neurony w warstwie
wejściowej
neurons1 = 5; %neurony w warstwie ukrytej

W1 = rand(neurons1, neurons_wej+1); % macierz wag
warstwy ukrytej
Wex = rand(1, neurons1+1); % macierz wag warstwy
wyjściowej

ind = []; % tablica indeksów (kolejnych iteracji
programu)
MSE = []; % tablica błędów średniokwadratowych
sumMSErr = 0; % wartość początkowa błędu
średniokwadratowego
bias = 1; % wartość współczynnika BIAS

N = length(Xwej); % liczba instancji zbioru trenującego

for step = 1:epoch

    for i = 1: N

        % Feedforward
        exFunc1 = W1 * [Xwej(:,i); bias]; %
pobudzenie neuronów I warstwy ukrytej
        actFunc1 = logsig(exFunc1); % funkcja
aktywacji I warstwy ukrytej
        y.est = Wex * [actFunc1; bias]; % estymowane
wyjście neuronów warstwy
                                % wyjściowej

        % Backpropagation
        y.correct = OUT(1,i); % poprawne wyjście
neuronów warstwy wyjściowej
        error = (y.correct - y.est); % błąd dla
konkretnej instancji
        % zapis błędów każdej iteracji do tabel
        Error(i) = error;
        Yest(i) = y.est;
        Ycorrect(i) = y.correct;

        %sygnał błędu warstwy ukrytej
        errorW1 = dlogsig(exFunc1,
actFunc1).*Wex(:,1:neurons1)' * error;

        Wex = Wex + eta * error * [actFunc1; bias]'; %
aktualizacja wagi
```



```

                                % warstwy
wyjściowej
    W1 = W1 + eta * errorW1 * [Xwej(:,i); bias];
% aktualizacja wagi

                                % warstwy
ukrytej

    sumMSErr = sumMSErr + (1/N) * error.^2; %
błąd średniokwadratowy
    ind(step) = step;
    MSE(step) = sumMSErr/step; % średnia
wartość błędu średniokwadratowego
                                % dla poszczególnych
iteracji

    end

end

% testing

Testwej = [Ptest.Age.'; Ptest.BMI.'; Ptest.Glucose.';
Ptest.Insulin.'; ...
    Ptest.HOMA.'; Ptest.Resistin.'; Ptest.MCP1.'];
% wektor cech wejściowych

                                % ze zbioru
testującego

OUT = [Ptest.Classification.']; % wektor poprawnych
wyjść ze zbioru testującego

count = 0; % numer testowanej pacjentki

M = length(Testwej); % długość zbioru testującego

for j = 1:M
    count = count + 1;
    exFuncT1 = W1 * [Testwej(:,j); bias]; %
pobudzenie neuronów warstwy ukrytej
    actFuncT1 = logsig(exFuncT1); % funkcja
aktywacji

    out.est(count) = Wex * [actFuncT1; bias]; %
estymowane wyjście neuronów
                                % warstwy wyjściowej
    out.correct(count) = OUT(1,j); % poprawne
wyjście neuronów warstwy wyjściowej
    inp(count) = j;

    if (out.est(count) < 0.5)
        out_ap.est(count) = 0;
    else
        out_ap.est(count) = 1;
    end
end

```

```

end

% wykresy

subplot(1,2,1);
plot(ind, MSE, "LineWidth", 1);
xlabel("liczba iteracji");
ylabel("wartość błędu średniokwadratowego");
title("Wykres błędu średniokwadratowego uczenia
sieci");
subplot(1,2,2);
plot(inp, out.est, 'o', 'Color', 'r');
hold on;
plot(inp, out.correct, 'x', 'Color', 'b');
xlabel("pacjentka");
ylabel("wynik");
legend("diagnoza estymowana", "diagnoza
prawidłowa");
title("Wykres poprawności identyfikacji choroby");

% tablica pomyłek

TP = numel(find(out.correct == 1 & out_ap.est ==
out.correct));
TN = numel(find(out.correct == 0 & out_ap.est ==
out.correct));

FP = numel(find(out.correct == 0 & out_ap.est ~=
out.correct));
FN = numel(find(out.correct == 1 & out_ap.est ~=
out.correct));

tab = table([TP; FN], [FP; TN], 'RowNames', {'dodatni',
'ujemny'}, 'VariableNames', {'dodatni', 'ujemny'})

% czułość testu - zdolność do wykrywania osób
rzeczywiście chorych
format bank; % wyświetlanie wyników z dokładnością
do dwóch miejsc po przecinku
sens = (TP/(TP + FN)) * 100

% swoistość testu - zdolność testu do prawidłowego
wykluczenia osób bez choroby
spec = (TN/(TN + FP)) * 100

% wartość predykcyjna dodatnia
PPV = (TP/(TP + FP))*100

% wartość predykcyjna ujemna
NPV = (TN/(TN + FN))*100

% dokładność testu
ACC = ((TN + TP)/height(Ptest))*100

```

Bibliografia

Slajdy z wykładów z przedmiotu Sieci neuronowe w zastosowaniach biomedycznych (SNB) - dr inż. Paweł Mazurek

https://pl.wikipedia.org/wiki/Czu%C5%82o%C5%9B%C4%87_i_swoisto%C5%9B%C4%87
https://pqstat.pl/?mod_f=diagnoza