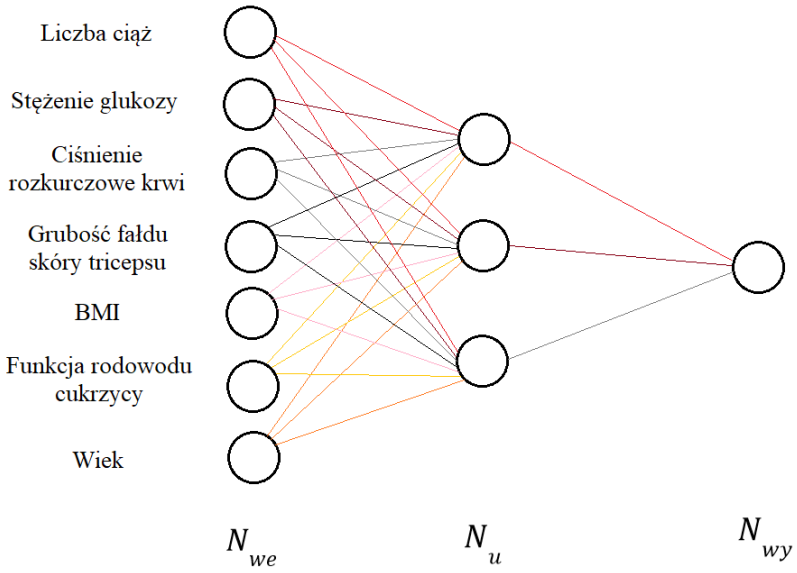
SPIS TREŚCI

1. **Struktura sieci …… 2**
2. **Algorytm uczenia…… 3**
3. **Wykres procesu uczenia sieci …… 3**
4. **Wstępne testy…….. 4**
5. **Sensitivity and Specificity…….. 4**

1. Struktura sieci



Rysunek 1: Uproszczony schemat sieci

Struktura sieci jest przedstawiona na Rys.1. Liczba neuronów w warstwie wejściowej wynosi 7. Warstwa ukryta zawiera od 3 do 5 neuronów. Warstwa wyjściowa ma tylko jeden neuron, wartość którego mówi o stanie zdrowia tej osoby, której dane zostały podane na wejście sieci. W sieci mamy tylko jedną warstwą ukrytą.

Liczbę neuronów w warstwie ukrytej można próbować oszacować wzorem:

$$N_u = \sqrt{N_{we} * N_{wy}}$$

Co w naszym przypadku oznacza $N_u \approx 3$

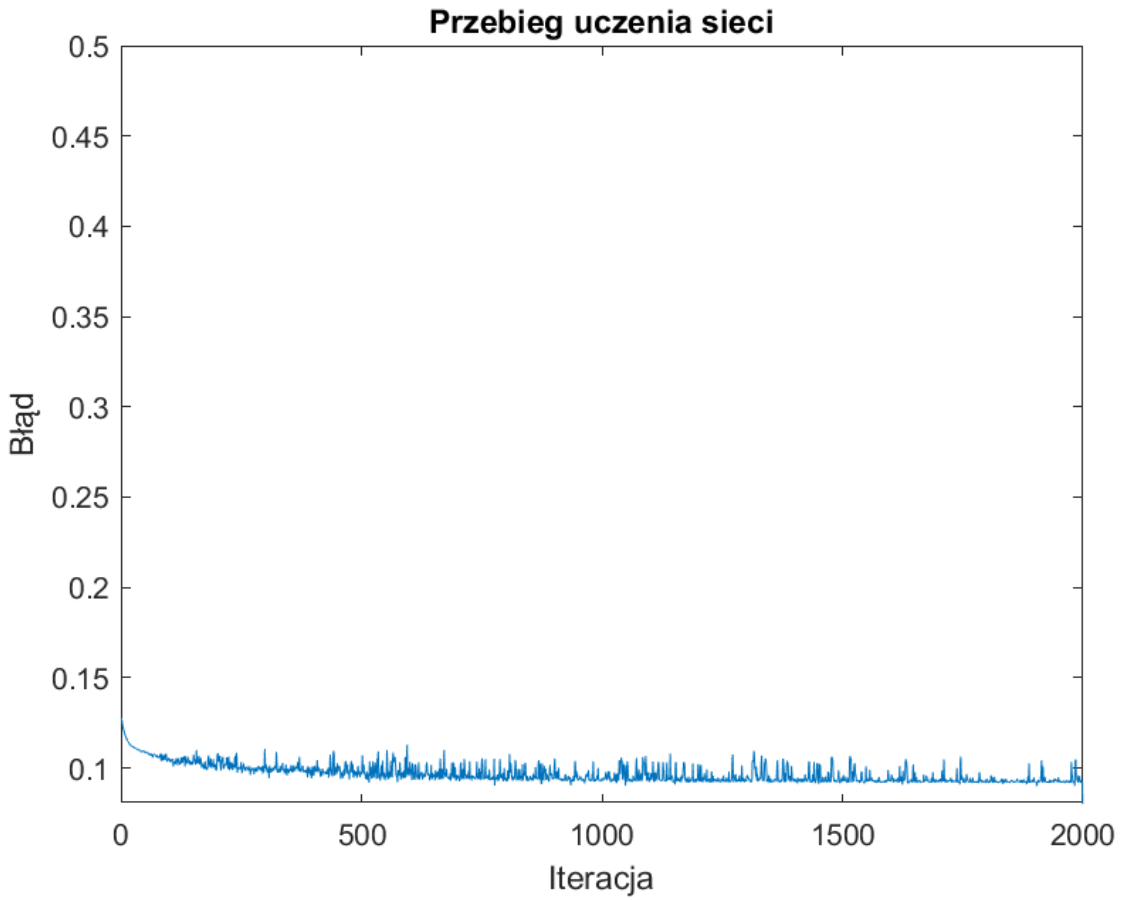
Warstwa	**Liczba neuronów**
Wejściowa	7
Ukryta	3-5
Wyjściowa	1

Inną “szkołą” wyznaczania warstwy ukrytej, jest zastosowanie wzoru $N_u = \frac{N_{we}}{2} + N_{wy}$, co w naszym przypadku da liczbę 4.5 (zaokrągloną do 5.).
Podczas implementacji naszej sieci, zostawiłyśmy 5 neuronów w warstwie ukrytej, bo w takiej konfiguracji sieć dawała lepsze wyniki.

2. **Algorytm uczenia**
   Funkcja aktywacji, która została użyta na warstwie ukrytej - tangens hiperboliczny.
   Funkcja aktywacja na warstwie wyjściowej - funkcja sigmoidalna.
   Do uczenia sieci zostanie zastosowany algorytm wstecznej propagacji błędów.

   Ustalona liczba epok - 2800;
   Utalony learning rate - 0.05;
   Szczegółowa implementacja algorytmu z opisem każdego kroku jest zawarta w pliku matlabowym.

3. **Wykres procesu uczenia sieci**



4. Wstępne testy

Po wstępnym testowaniu sieci uzyskujemy wartości rzeczywiste z przedziału (0,1). W celu kontynuacji przeprowadzenia dalszej klasyfikacji - wartości poniżej 0.45 przybliżyłyśmy do 0, a wartości równe i powyżej 0.45 do 1.

```
%% funkcja, która przerzuca wartości wektora wyjściowego na 0 i 1
function t = Transform(out)
   prog = 0.45;                              %ustawienie progu
   t = zeros(size(out,1), size(out, 2));          %prealokacja pamieci na wektor skadający się z 0 i 1
   for n = 1 : size(out,2)
      if out(1, n) < prog
         t(1, n)=0;
      elseif out(1, n) > prog
         t(1, n)=1;
      end
  end
end
```

Za obliczenie skuteczności sieci jest odpowiedzialny o ten kawałek kodu:

```
%% oblicznie skuteczności sieci
count = 0;
for n = 1 : N
   if transformed(1, n) == dataTestY(1, n)          %jeśli wartość wyjścia sieci jest równa wartości prawdziwej to dodaj 1
       count = count + 1;
   end
end
precent = (count/N)*100;                            %procentowa poprawność wykrycia choroby
```

Skuteczność przez cały czas utrzymuje się na poziomie **około 70%.**

5. Sensitivity and Specificity

Czułość jest wyznaczana ze wzoru $\frac{TP}{TP+FN}$, a specyficzność $\frac{TN}{TN+FP}$.
W naszym projekcie będzie to:
True Positive (TP) - kobieta chora zaklasyfikowana jako chora.
True Negative (TN) - kobieta zdrowa zaklasyfikowana jako zdrowa.
False Positive (FP) - kobieta zdrowa zaklasyfikowana jako chora.
False Negative (FN) - kobieta chora zaklasyfikowana jako zdrowa.

Do wyznaczenia ww. parametrów posłużyłyśmy się klasyfikatorem ‘classperf’ z toolboxa Bioinformatics.

Przykładowe otrzymane wartości:

```
Sensitivity: 0.6147
Specificity: 0.7444
```

Czułość wynosi 62%
Specyficzność wynosi 74%