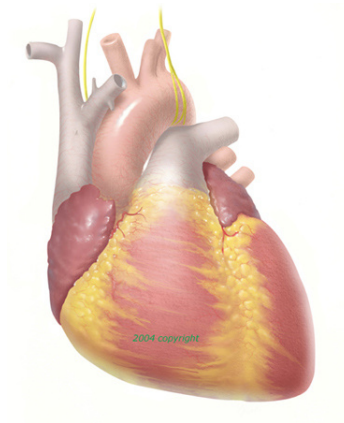


System do wspomaganie stawiania diagnoz chorób serca



Autorzy: Bartłomiej Hyży
Michał Pieróg
Witold Baran
Jakub Jaśkowiec
Mateusz Niesyto

SPIS TREŚCI

1.	Sformułowanie zadania projektowego	3
1.1	Obszar i przedmiot projektowania	3
1.1.1	Dziedzina problemu	3
1.1.2	Obszar modelowania.....	3
1.1.3	Zakres odpowiedzialności systemu	4
1.2	Zwięzła nazwa problemu.....	4
1.3	Cele do osiągnięcia	5
1.3.1	Cele rezultatu	5
1.3.2	Cele projektu	5
2	Opis wymagań	6
2.1	Funkcje systemu z punktu widzenia użytkownika	6
2.2	Przepływy informacyjne doprowadzone do i wyprowadzane z systemu.....	7
2.3	Sygnalizowane specjalne wymagania i ograniczenia	7
3	Sztuczne sieci neuronowe – zastosowany model	8
3.1	Sztuczne sieci neuronowe	8
3.2	Trening sieci neuronowej	10
3.3	Budowa optymalnej sieci neuronowej	12
3.4	Format danych uczących.....	13
3.5	Testy systemu.....	14
3.6	Wnioski	15
4	Analiza systemu– diagramy UML	19
4.1	Diagram przypadków użycia systemu.	19
4.2	Diagramy klas systemu.	20
4.3	Diagramy sekwencji	23
4.4	Diagram najważniejszego stanu systemu	24
4.5	Diagram Komunikacji systemu	24
4.6	Diagram Komponentów systemu.....	25
5	Opis zmian wprowadzonych w zrealizowanym systemie.....	26
5.1	Zmiany na etapie projektowania klas	26
5.2	Zmiany w funkcjonowaniu systemu	26
6	Instrukcja obsługi Systemu	27
6.1	Wymagania.....	27
6.2	Logowanie do systemu	27
6.3	Panel administratora	28
6.4	Panel lekarza.....	30
6.5	Wylogowanie użytkownika z systemu	32
6.6	Pasek stanu	32
7	Bibliografia	33
8	Harmonogram projektu i podział zadań	36

1. Sformułowanie zadania projektowego

1.1 Obszar i przedmiot projektowania

1.1.1 Dziedzina problemu

Projekt dotyczy bardzo przyszłościowej i pożądanej dziedziny jaką jest informatyczne wspomaganie lekarzy przy stawianiu diagnoz. Zakresem obejmuje on choroby serca.

Zgodnie ze statystykami, choroby serca są jedną z najczęstszych przyczyn zgonu na całym świecie. Częstym problemem kardiologii jest fakt, iż syndromy jakie się pojawiają u pacjenta często dotyczą innych organów, co powoduje znaczne opóźnienie w stawianiu właściwej diagnozy. Tym co może przyczynić do poprawy tej sytuacji jest utworzenie systemu, który będzie potrafił zaalarmować lekarza, iż ma do czynienia z chorobą serca. Wszystko to sprawia iż istnieje zapotrzebowanie na realizację systemu wspomagania decyzji w kardiologii i zbudowania modelu, pomagającego w diagnozie chorób serca.

1.1.2 Obszar modelowania

System jest nastawiony przede wszystkim na lekarza i jego zadaniem jest wspomaganie postawienia prawidłowej diagnozy. Celem utworzonego modelu nie jest jednak dostarczenie informacji na temat konkretnych chorób. Jego zadaniem jest za to określić czy pacjent ma chorobę serca czy też nie. Choroby te charakteryzują się tym, że objawy często dotyczą innych organów. Czasem powoduje to sytuację, że lekarz zbyt późno określa iż to serce jest przyczyną złego stanu pacjenta. Dlatego ważne jest by przekazać sygnał ostrzegawczy jak najwcześniej.

1.1.2.1 Opis zastosowanego modelu

System został zbudowany w oparciu o sieci neuronowe. Charakteryzują się one całkiem dobrą poprawnością stawianych diagnoz. W literaturze wskazuje się na wskaźnik powyżej 90%. Niestety wynik ten można uzyskać dopiero przy 2000-3000 próbek. Ponieważ na potrzeby projektu było do dyspozycji około 800 próbek, poprawność diagnoz jaką udało się uzyskać to ok. 60%.

Do diagnozy czy pacjent posiada chorobę serca wykorzystuje się w systemie 13 parametrów o stanie pacjenta, za których dostarczenie odpowiedzialny będzie lekarz lub pielęgniarka robiący badanie. Założeniem systemu jest istnienie centralnej bazy danych w szpitalu/przychodni. Dzięki temu rozproszone aplikacje klienckie obsługiwane przez lekarzy będą mogły korzystać z akumulowanych globalnie danych medycznych równocześnie poszerzając ich zbiór. Proponowany system do diagnozy chorób korzysta właśnie z takiej bazy.

1.1.3 Zakres odpowiedzialności systemu

Wprowadzenie danych w celu przeprowadzenia diagnozy

Program umożliwia edycje istniejących w bazie danych, takich jak:

- podstawowe dane pacjenta (wiek, płeć)
- wywiad z pacjentem
- badania wykonywane na miejscu przez lekarza (tętno, itp)
- specjalistyczne badania wykonywane na zlecenie lekarza

Program zakłada iż za dostarczenie danych do centralnej bazy danych szpitala/przychodni odpowiadają inne systemy, a on z nich tylko korzysta (ewentualnie je aktualizuje).

Przeglądanie bazy danych

Lekarz ma z poziomu aplikacji możliwość przeglądania danych o pacjentach.

Uzyskanie diagnozy

Po wybraniu informacji o pacjencie z dostępnej bazy i uruchomieniu diagnozy, lekarz uzyska informację czy pacjent ma chorobę serca lub nie.

Bieżąca optymalizacja działania systemu

Ponieważ system zbudowany jest w oparciu o sieć neuronową to posiada on mechanizm testowania poprawności modelu oraz jego korekcji w przypadku wystąpienia nieprawidłowości.

1.2 Zwięzła nazwa problemu

Projekt dotyczy systemu do wspomaganie stawiania diagnozy w postaci obecności choroby serca u pacjenta.

1.3 Cele do osiągnięcia

1.3.1 Cele rezultatu

Podstawowym celem rezultatu jest stworzenie wydajnego systemu wspomagającego decyzje lekarzy w zakresie stawiania diagnoz chorób serca. System taki ma mieć zapewnioną wysoką efektywność i dawać możliwości dalszego rozwoju (głównie pod względem ilości rozpoznawanych przypadków).

Celem systemu nie jest zastąpienie pracy lekarza. Autorzy projektu upatrują siłę w synergicznym połączeniu nowoczesnych technologii z doświadczeniem i intuicją lekarską, dlatego tworzony system traktowany jest w kategorii doradcy lekarza, a nie jego zastępcy.

1.3.2 Cele projektu

Cele przedsięwzięcia projektowego dotyczą dwóch aspektów, tj. sposobów wytwarzania oprogramowania oraz poznanie dziedziny wspomagania medycyny informatyką. Pierwsza grupa celów jest zorientowana na poznanie metodyk projektowych i sposobów wydajnego tworzenia systemów informatycznych. Z kolei druga grupa dotyczy głównie możliwości poznania zaawansowanych metod wspomagania podejmowania decyzji (jak sieci neuronowe) i rozpoznania możliwych obszarów dalszego rozwoju naukowego dla członków projektu.

2 Opis wymagań

2.1 Funkcje systemu z punktu widzenia użytkownika

Aplikacja jest wyposażona w mechanizm logowania. W bazie danych możliwe jest umieszczeniu dwóch typów użytkowników, którzy posiadają różne funkcjonalności. Użytkownikami mogą być:

- Lekarze
- Administratorzy

Funkcje dostępne z panelu administratora:

- Włączenie panelu trenowania sieci neuronowych
- Włączenie/wyłączenie procesu trenowania sieci i bieżąca obserwacja jej parametrów:
 - Wykres zmian. Wizualizowane parametry to:
 - training set accuracy,
 - generalization set accuracy,
 - training set MSE,
 - Generalizaton set MSE
 - Panele opisujące stan sieci aktualnej i najlepszej znalezionej podczas procesu trenowania. Widoczne parametry to:
 - training set accuracy,
 - training set MSE,
 - generalization set accuracy,
 - generalization set MSE,
 - validation set accuracy,
 - validation set MSE
 - Panele opisujące strukturę sieci obecnej i najlepsze znalezionej podczas procesu trenowania
- Zapisywanie nowej struktury sieci neuronowych

Funkcje dostępne z panelu lekarza:

- Logowanie do systemu
- Przeszukiwanie bazy danych z postawieniem diagnozy - lekarz może przeszukiwać bazę po różnych kryteriach dotyczących pacjentów, diagnozować ich i edytować ich dane medyczne (po zweryfikowaniu diagnozy można ją zapisać)
- Przeglądanie bazy danych pacjentów bez zweryfikowanej diagnozy - lekarz może pracować z danymi tak jak wyżej, ale ma ograniczone rezultaty wyszukiwania tylko do pacjentów bez zatwierdzonej diagnozy

2.2 Przepływy informacyjne doprowadzone do i wyprowadzane z systemu

- Centrum przetwarzania informacji w systemie jest sieć neuronowa, która jest konstruowana na podstawie danych z bazy bazie historycznych próbek badań i diagnoz.
- WEJŚCIE SIECI: Do sieci wchodzi 13 parametrów pacjenta, które określają jego aktualny stan zdrowia
- WYJŚCIE SIECI: Na wyjściu pojawia się informacja czy pacjent jest chory czy zdrowy



2.3 Sygnalizowane specjalne wymagania i ograniczenia

Do podstawowych ograniczeń w systemie należy zaliczyć:

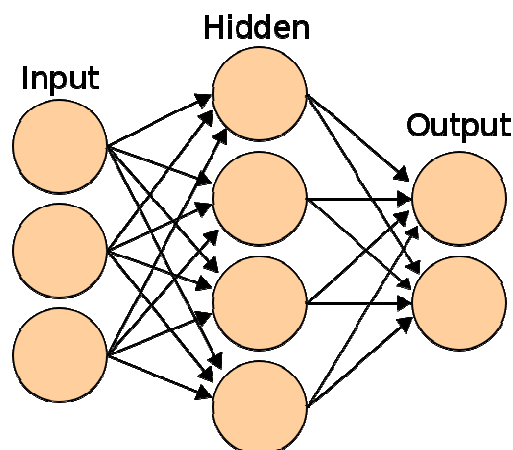
- Skuteczność systemu uwarunkowana jest ilością danych treningowych, stopniem nauki sieci, itp.
- System nie służy do administracji danymi. Nowych użytkowników, pacjentów dodaje inny system (informacje te są przechowywane w centralnej bazie danych szpitala czy przychodni).

3.1 Sztuczne sieci neuronowe

Sztuczna sieć neuronowa (ang. ANN – Artificial Neural Network) to matematyczny model obliczeniowy inspirowany budową i działaniem biologicznych sieci neuronowych, np. tych obecnych w mózgu człowieka. Sieć neuronowa składa się z grup (warstw) sztucznych neuronów połączonych ze sobą kierunkowymi połączeniami o ustalonych wagach. Dane wejściowe podawane na neurony warstwy wejściowej przetwarzane są przez sieć przepływając przez kolejne jej połączenia, docierając w końcu do neuronów wyjściowych, stanowiących wyjście systemu.

Sztuczne sieci neuronowe to systemy adaptacyjne, które zmieniają swoje działanie w zależności od struktury i parametrów określających ich budowę. Dobór tych właściwości odbywa się podczas procesu nazywanego nauką sieci. Współczesne sieci neuronowe stanowią narzędzie nieliniowego, statystycznego modelowania danych i używane są wszędzie tam, gdzie zachodzi konieczność określenia skomplikowanych zależności pomiędzy dwoma grupami pewnych danych – wejściowych i wyjściowych.

Sieci neuronowe mogą posiadać różną budowę, w przypadku zaprezentowanego systemu użyte zostały sieci wielowarstwowe, zbudowane z grup neuronów ułożonych w formie kolejnych warstw, co prezentuje poniższy rysunek:



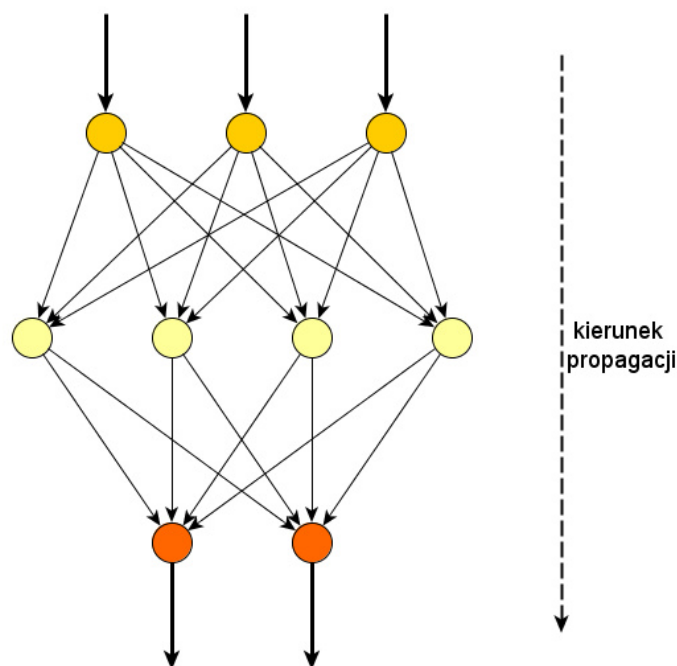
Rysunek 1: schemat budowy sztucznej sieci neuronowej (źródło: Wikipedia)

Wyróżniamy trzy rodzaje warstw neuronów, wchodzących w skład sieci neuronowej:

- **Warstwa wejściowa (ang. input)** – wartościami jej neuronów są kolejne próbki danych uczących, zaś po treningu – danych poddawanych ocenie/klasyfikacji
- **Warstwy ukryte (ang. hidden)** – określają trudne do ręcznego wyspecyfikowania zależności pomiędzy danymi wejściowymi a wyjściowymi; odpowiednie dobranie ich liczby oraz liczby neuronów wchodzących w skład pojedynczej takiej warstwy jest kluczowe dla osiągnięcia zadowalającej skuteczności systemu
- **Warstwa wyjściowa (ang. output)** – wartościami jej neuronów są wyniki przetworzenia kolejnych próbek danych uczących lub poddawanych klasyfikacji/ocenie – stanowią wyjście systemu

Warstwa wejściowa i wyjściowa są obligatoryjne, z kolei liczba i budowa warstw ukrytych jest dowolna, przy czym przydatność modeli nie zawierających żadnych warstw ukrytych jest dyskusyjna.

W zaimplementowanym systemie wykorzystany został model sieci neuronowej typu feed-forward, tj. takiej, w której kierunek przepływu informacji jest stały i ustalony. Najczęściej dane przepływają od neuronów wejściowych, do neuronów warstw ukrytych, po czym trafiają do neuronów wyjściowych:



Rysunek 2: sieć neuronowa typu feed-forward

3.2 Trening sieci neuronowej

Każda sztuczna sieć neuronowa przed użyciem zostać wcześniej wytrenowana. Jako trening/naukę sieci określa się proces wyznaczania optymalnych wag poszczególnych połączeń pomiędzy neuronami tworzącymi sieć, celem maksymalizacji określonego kryterium efektywności. W przypadku utworzonego systemu wspomagania diagnostyki był to procent pacjentów poprawnie zakwalifikowanych do grupy zdrowych lub chorych.

Zaimplementowany system opiera się o model wielowarstwowego perceptrona (ang. MLP – Multi-Layer Perceptron) – sztucznej sieci neuronowej typu feed-forward. Składa się ona z wielu warstw neuronów, połączenia pomiędzy którymi opisane są skierowanym grafem. Każda warstwa połączona jest z warstwą następną, zaś składające się na nią neurony (poza warstwę wejściową) posiadają wejścia połączone z neuronami z warstwy poprzedniej oraz nieliniową funkcję aktywacji (sigmoidalna).

Trening sieci typu MLP opiera się o algorytm wstecznej propagacji (ang. backpropagation). Wagi połączeń neuronowych inicjalizowane są pewnymi losowymi wartościami. Następnie na wejście sieci podawane są kolejne próbki należące do zbioru uczącego, w wyniku czego na wyjściu pojawiają się wartości określające rezultat klasyfikacji. W przypadku zaprojektowanego systemu jest to pojedyncza wartość rzeczywista x :

- $x > 0.7$ – pacjent jest chory na serce
- $x \leq 0.7$ – pacjent ma zdrowe serce

Uzyskane wartości porównywane są z tymi oczekiwanymi, określonymi przez specjalistów odpowiedzialnych za utworzenie bazy danych medycznych. Na podstawie tego porównania określone są gradienty błędów, które w następnych iteracjach treningu używane są do wyznaczenia modyfikacji wag metodą gradientu sprzężonego. Problem nauki sieci neuronowej jest więc w istocie zadaniem optymalizacji, gdzie funkcja celu określona jest jako odchylenie klasyfikacji próbek treningowych od wartości oczekiwanych. Ponieważ przed rozpoczęciem nauki sieci konieczne jest ręczne przygotowanie zbioru treningowego, poprzez przyporządkowanie próbek do odpowiednich klas, metoda ta należy do zbioru algorytmów nadzorowanej (ang. supervised) nauki.

Kluczową kwestią pozwalającą na uzyskanie optymalnej pod względem skuteczności działania sieci neuronowej jest odpowiednie przygotowanie zbioru uczącego. W przypadku opisywanego systemu użyte zostały dane medyczne udostępnione na stronie UCI Machine Learning Repository odnośnie chorób serca, dostępne pod adresem:

<http://archive.ics.uci.edu/ml/datasets/Heart+Disease>.

Wykorzystany został standardowy sposób podziału zbioru źródłowego na podzbiory:

- Zbiór treningowy (ang. training set) – 60%
- Zbiór generalizacyjny (ang. generalization set) – 20%
- Zbiór walidacyjny (ang. validation set) – 20%

Zbiór treningowy wykorzystywany jest w kolejnych iteracjach nauki sieci do określania kierunku modyfikacji wag celem minimalizacji funkcji błędu. Zbiór generalizacyjny jest po każdej iteracji używany do określenia stopnia ogólności aktualnej sieci, tj. jej skuteczności dla wcześniej nie przetwarzanych danych. Teoretycznie istnieje możliwość zakończenia nauki po osiągnięciu określonej skuteczności na tym zbiorze, w praktyce jednak często określa się wymaganą skuteczność na 100% i ogranicza się czas wykonania treningu poprzez określenie maksymalnej ilości iteracji. Ostatni ze zbiorów – walidacyjny – służy do ostatecznego określenia skuteczności uzyskanej sieci.

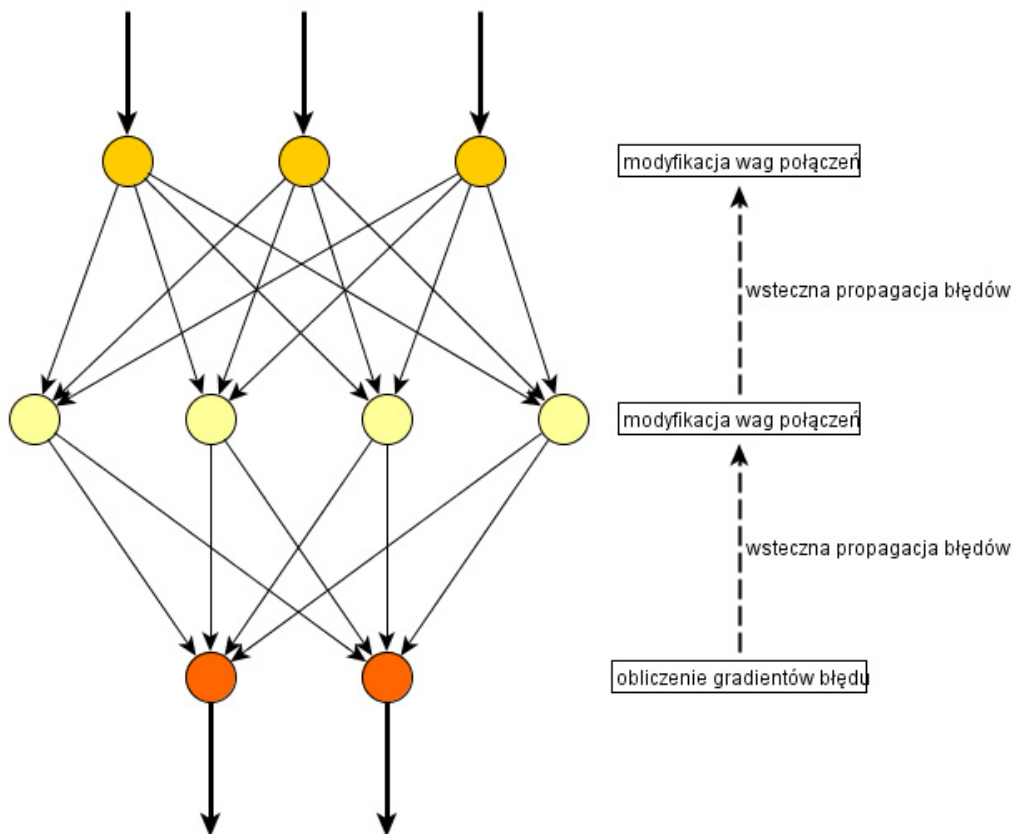
Należy położyć nacisk na fakt, iż wymienione podzbiory danych uczących muszą koniecznie być wzajemnie rozłączne, co gwarantuje, że uzyskana sieć nie „przyzwyczai” się do danych treningowych, osiągając na nich wysokie skuteczności, równocześnie zachowując się bardzo słabo dla danych wychodzących poza ich zbiór.

Celem poprawy zbieżności algorytmu uczącego do wartości optymalnych wprowadzone zostały następujące usprawnienia:

- „learning rate” – współczynnik nauki, określający tempo treningu, czym większy, tym większe zmiany wag w kolejnych iteracjach
- „learning rate adjust” – współczynnik dostosowania tempa nauki związany z błędem średniokwadratowym (ang. MSE – Mean Square Error) zbioru treningowego; czym większy, tym większe zmiany wag wraz ze wzrostem MSE zb. treningowego
- „momentum” – moment pędu nauki, czym większy, tym większe zmiany wag w kolejnych iteracjach; odpowiednio dobrana wartość pozwala na szybkie wyjście z obszaru lokalnie słabych rozwiązań
- „forgetting rate” – współczynnika zapominania, wpływający na stopień zaniku wag wszystkich połączeń w kolejnych iteracjach; pozwala na zapominanie połączeń o małym wpływie na rezultat klasyfikacji; w ogólności zanik połączeń do minimalnej wystarczającej ilości jest zjawiskiem porządnym, gdyż prostsze modele są bardziej ogólne i nie wykazują tendencji do przystosowywania się do konkretnych danych treningowych; nieoptymalne dobranie tego współczynnika może powodować wysoce chaotyczne zachowania algorytmu treningowego

Właściwy trening sieci neuronowej opiera się o algorytm wstecznej propagacji (ang. backpropagation) zademonstrowany na poniższym rysunku. Wykorzystując

schemat propagacji danych przez sieć – feed-forward – próbki treningowe poddawane są klasyfikacji do grup chory/zdrowy na wyjściu sieci. Otrzymane wyniki służą następnie do obliczenia gradientów błędów, które z kolei propagowane są wstecznie do poprzedzających warstw, począwszy od warstwy wyjściowej. Wpływ gradientu błędu dowolnego neurona z warstwy n na wagę jego połączenia z dowolnym neuronem warstwy $n-1$ obliczany jest poprzez wzięcie pod uwagę aktualnej wagi tego połączenia. W rezultacie największe modyfikacje wag połączeń obserwowane są na neuronach warstw „poprzednich”, które posiadają największy wpływ na błędy na neuronach warstw „następnych”.



Rysunek 3: schemat wstecznej propagacji błędów w zastosowanej sieci neuronowej

3.3 Budowa optymalnej sieci neuronowej

Budowa optymalnej sieci neuronowej przeprowadzana jest za pomocą kaskadowego procesu doboru najlepszych parametrów budowy sieci. Za kryterium optymalizacji sieci uznawana jest maksymalizacja procentu poprawnie zaklasyfikowanych próbek pacjentów z bazy danych do grup chory/zdrowy.

Proces optymalizacji sieci polega więc na iteracyjnym sprawdzaniu skuteczności różnych sieci zbudowanych w oparciu o parametry z następujących przedziałów:

- Liczba warstw ukrytych: 1-3
- Liczba neuronów w pojedynczej warstwie ukrytej: 1-20
- „learning rate”: 0.01-0.1
- „learning rate adjust”: 0.0005-0.002
- „momentum”: 0.3-0.8
- „forgetting rate”: 0.000001-0.00002

Konfiguracja sieci o największej skuteczności jest zapamiętywana oraz zapisywana na serwerze na żądanie administratora, dzięki czemu może być potem pobrana przez użytkowników systemu (lekarzy) i wykorzystana do najbardziej optymalnej diagnostyki.

3.4 Format danych uczących

Dane uczące znajdują się w bazie danych MySQL. Jej schemat został stworzony na podstawie źródłowych plików tekstowych z danymi medycznymi pobranymi ze strony UCI Machine Learning Repository, którymi następnie została wypełniona.

Fragment tego zbioru danych uczących przedstawia się następująco:

```
63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0,2.3,3.0,0.0,6.0,0
67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0,1.5,2.0,3.0,3.0,2
67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0,2.6,2.0,2.0,7.0,1
37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0,3.5,3.0,0.0,3.0,0
41.0,0.0,2.0,130.0,204.0,0.0,2.0,172.0,0.0,1.4,1.0,0.0,3.0,0
56.0,1.0,2.0,120.0,236.0,0.0,0.0,178.0,0.0,0.8,1.0,0.0,3.0,0
62.0,0.0,4.0,140.0,268.0,0.0,2.0,160.0,0.0,3.6,3.0,2.0,3.0,3
57.0,0.0,4.0,120.0,354.0,0.0,0.0,163.0,1.0,0.6,1.0,0.0,3.0,0
63.0,1.0,4.0,130.0,254.0,0.0,2.0,147.0,0.0,1.4,2.0,1.0,7.0,2
(...)
```

Każdy rekord znajduje się w osobnej linii i składa się z wartości 14 atrybutów przedzielonych przecinkami. Znaczenie kolejnych atrybutów opisuje poniższa tabela. Celem polepszenia zbieżności treningu sieci neuronowej wartości rekordów po wczytaniu z bazy danych poddawane są przetwarzaniu wstępnemu (ang. preprocessing) oraz normalizacji (ang. normalization). Preprocessing w przypadku stworzonego systemu polega na zastąpieniu wartości pustych (ang. NULL values) w rekordach wartościami średnimi dla danych atrybutów. Normalizacja z kolei polega na sprowadzeniu wartości wyrażonych w różnych skalach (np. 150, 0.4, tak/nie, brak/lekki/dokuczliwy) do znormalizowanych przedziałów, np. 0.0-1.0 w przypadku danych liczbowych lub -0.5/0.0/1.0 w przypadku danych z trzema możliwymi wariantami (dokuczliwość bólu). Dzięki normalizacji danych trening sieci osiąga lepszą zbieżność, z uwagi na to, że

wszystkie atrybuty – a w efekcie również wagi - wyrażone są w podobnych skalach.

#	Atrybut	Rzeczywisty przedział wartości	Znormalizowany przedział wartości
1	Age (wiek)	0-100	0.0-1.0
2	Sex (płeć)	0,1	0,1
3	Cp (typ bólu w klatce piersiowej)	1-4	1-4
4	Trestbps (ciśnienie krwi)	70-190	0.0-1.0
5	Chol (poziom cholesterolu)	100-400	0.0-1.0
6	Fbs (poziom cukru we krwi)	0,1	0,1
7	Restecg (wynik EKG)	0,1,2	0,1,2
8	Thalach (maksymalna tętno)	100-200	0.0-1.0
9	Exang (ból w klatce piers. wymuszony wysiłkiem fizycznym)	0,1	0,1
10	Oldpeak („ST depression”)	0-4	0-4
11	Slope („the slope of the peak exercise ST segment”)	1,2,3	1,2,3
12	Ca (liczba żył głównych)	0-3	0-3
13	Thal	3,6,7	3,6,7
14	Num (chory/zdrowy)	0-4	0,1

3.5 Testy systemu

Celem testowania by wyznaczenie parametrów, dla których sieć wykazywała najlepszą skuteczność. W tym celu ustawialiśmy odpowiednie parametry, odpalaliśmy program dla administratora, uruchamialiśmy testowanie i obserwowaliśmy przebiegi procesu uczenia. W ten sposób określiliśmy parametry jakich będzie używać nasza sieć przy analizie danych medycznych.

W ramach testowania, zmienialiśmy 6 parametrów sieci neuronowej obserwując dla jakich wartości osiąga najlepszą wydajność.

Parametry:

- Learning Rate – szybkość z jaką sieć poszukuje lepszych rozwiązań.
- Learning Rate Adjust - długość skoków jaką wykonuje sieć, jeśli wpadnie w obszar słabszych rozwiązań.
- Momentum - szybkość z jaką sieć przeskakuje pomiędzy lokalnie słabymi rozwiązaniami.
- Forgetting Rate - szybkość z jaką sieć zapomina nieużywane połączenia.
- Ilość warstw ukrytych

- Ilość neuronów w warstwie ukrytej

Dane użyte do testowania znajdują się w bazie danych, w odpowiedniej tabeli. Po załadowaniu zbior danych medycznych został podzielony na 3 rozłączne podzbiory: Training, Generalization, Validation, w proporcjach 60%/20%/20%:

- Training Set - dane na których trenuje się sieć.
- Generalization Set - dane na których sieć sprawdza swoją skuteczność co iterację.
- Validation Set - dane na których sieć sprawdza ostatecznie swoją skuteczność po całym treningu.

Z punktu widzenia wiarygodności najważniejsza jest skuteczność na zbiorze Validation, gdyż dane z tego zbioru w żaden sposób nie brały udziału w nauce sieci, przez co sieć nie jest "przyzwyczajona" do tych danych, przez co daje najbardziej obiektywne dane o skuteczności.

3.6 Wnioski

Pierwsze 4 parametry mają wpływ jedynie na przebieg treningu sieci, nie zaś na skuteczność sieci. Zachowując odpowiedni dla nich rząd wielkości, przy ich zmianie obserwujemy zmiany skuteczności na poziomie 5%. Dlatego pełnią raczej rolę uzupełnienia do pozostałych parametrów. Należy pamiętać, że nieodpowiedni ich ustawienie (nieodpowiedni rząd wielkości – plus minus 50% wartość ustalonych przy testowaniu) może spowodować dużą nieregularność wykresu, co będzie oznaczać kiepskie rezultaty.

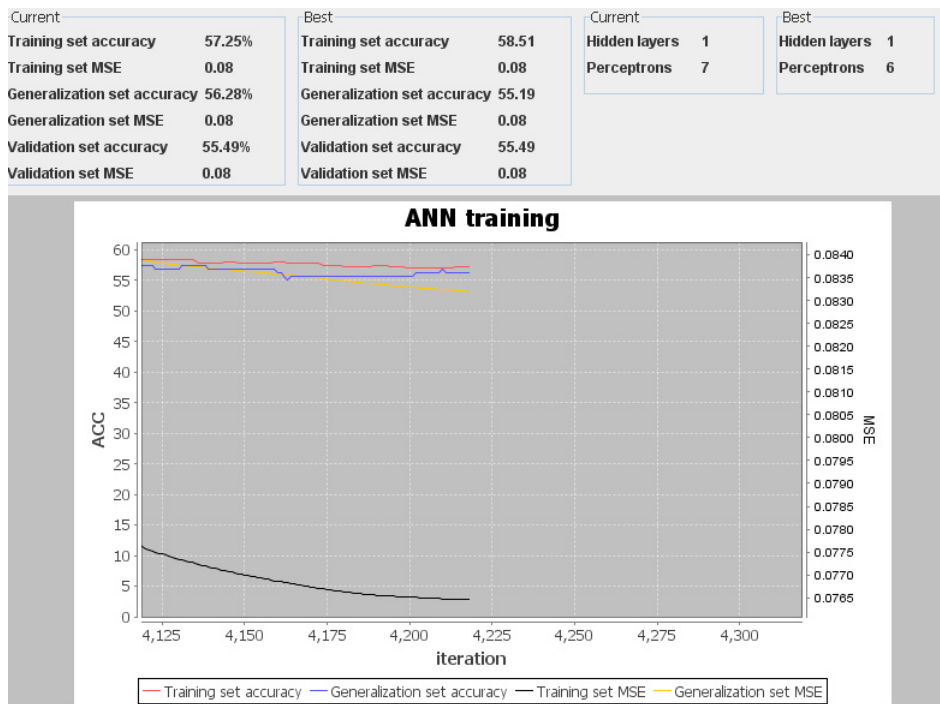
Ostatecznie wybrane zostały:

Learning Rate = 0.03

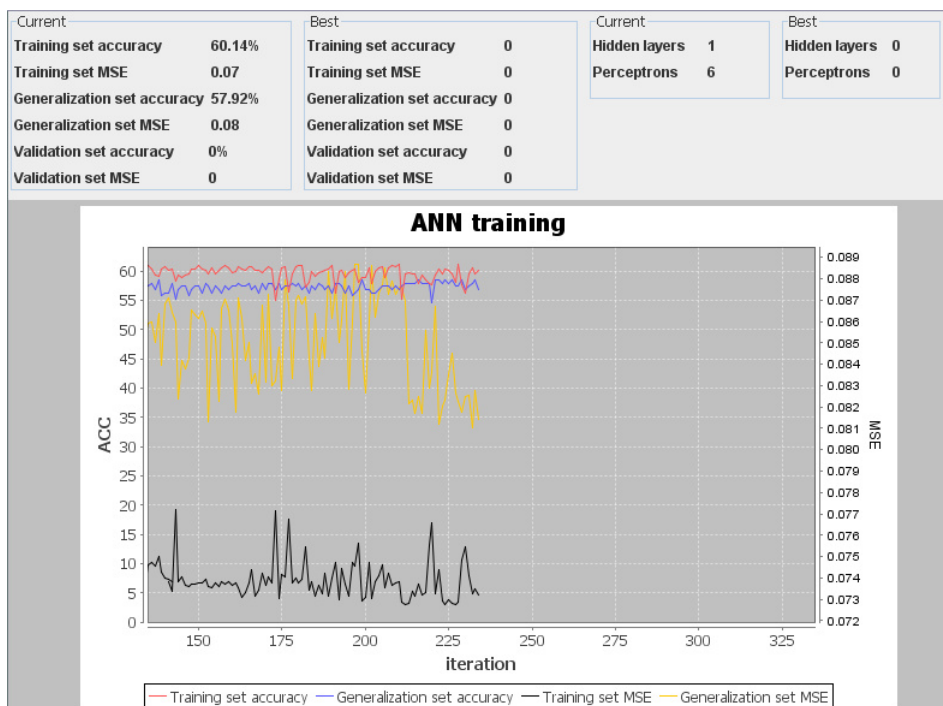
Learning Rate Adjust = 0.001

Momentum = 0.5

Forgetting Rate = 0.000005

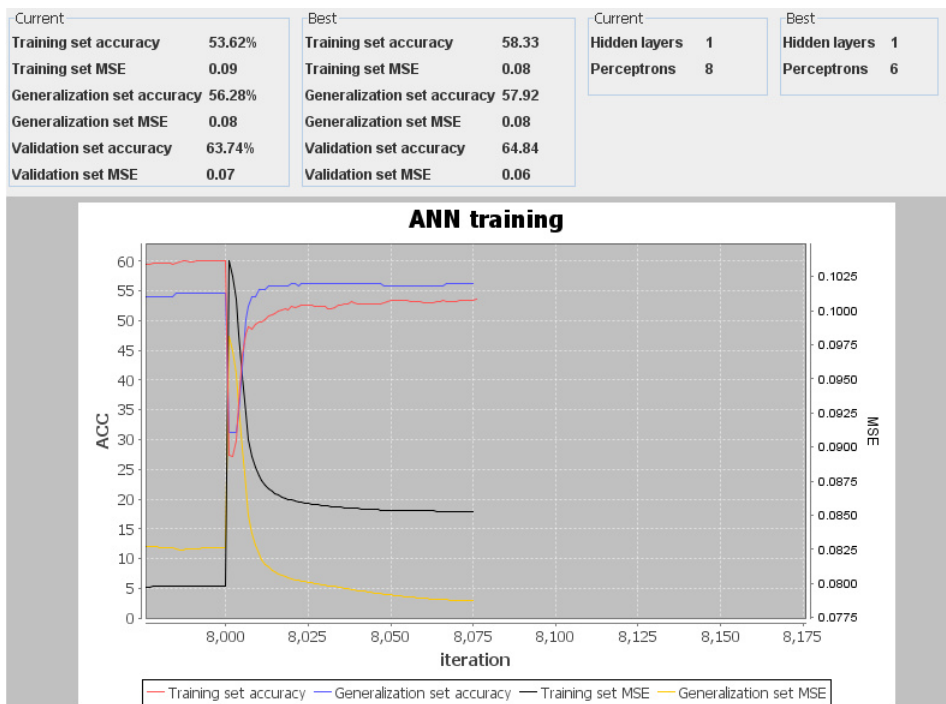


Rysunek 4: Parametry : Learning Rate =0.32 ; Learning Rate Adjust = 0.0005; Momentum =0.6; Forgetting Rate = 0.00001



Rysunek 5: Momentum zwiększone do 0.9

Kluczowe znaczenie mają natomiast liczba warstw ukrytych oraz liczba neuronów w warstwie. Po licznych testach okazało się, że sieć osiąga najlepsze rezultaty przy jednej warstwie oraz liczbie neuronów 6-8.



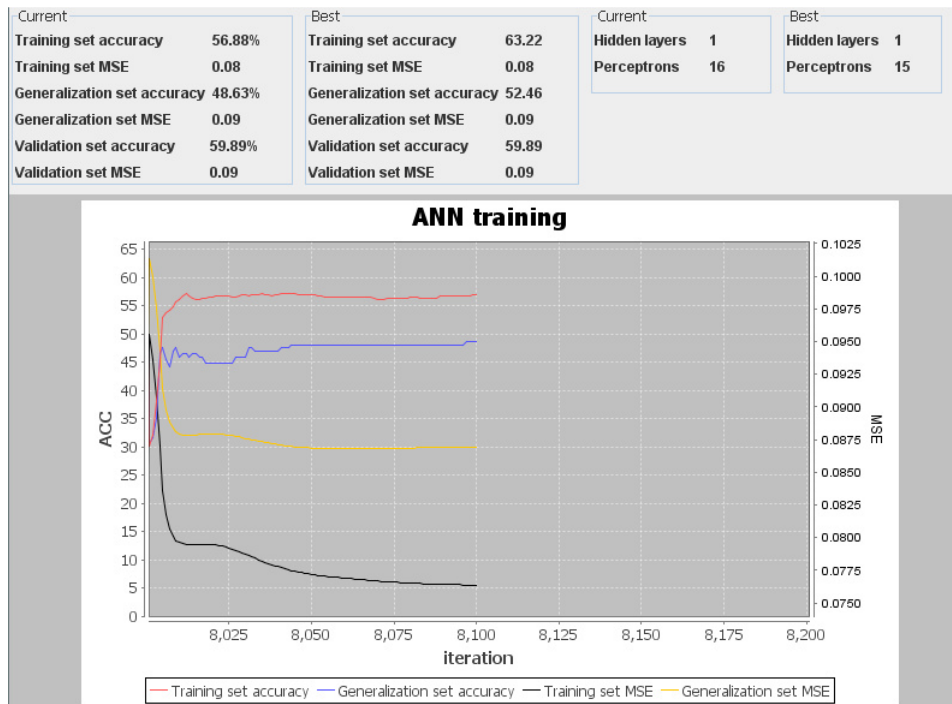
Rysunek 6: Learning Rate = 0.03 Learning Rate Adjust = 0.001, Momentum = 0.5, Forgetting Rate = 0.000005, Warstwy ukryte:1, Neurony w warstwach ukrytych: 7

Zwiększając liczbę warstw ukrytych odnotowaliśmy zdecydowany spadek skuteczności. Jest to spowodowane bardzo szybkim wpadaniem w lokalne minimum, z którego algorytm nie może już wyjść. Na wykresie obserwujemy, że funkcje dokładności i błędu są stałe przez większość testowania. Skuteczność sieci spada do 20-40%.



Rysunek 7: warstwy ukryte: 2, liczba neuronów w tych warstwach: 8

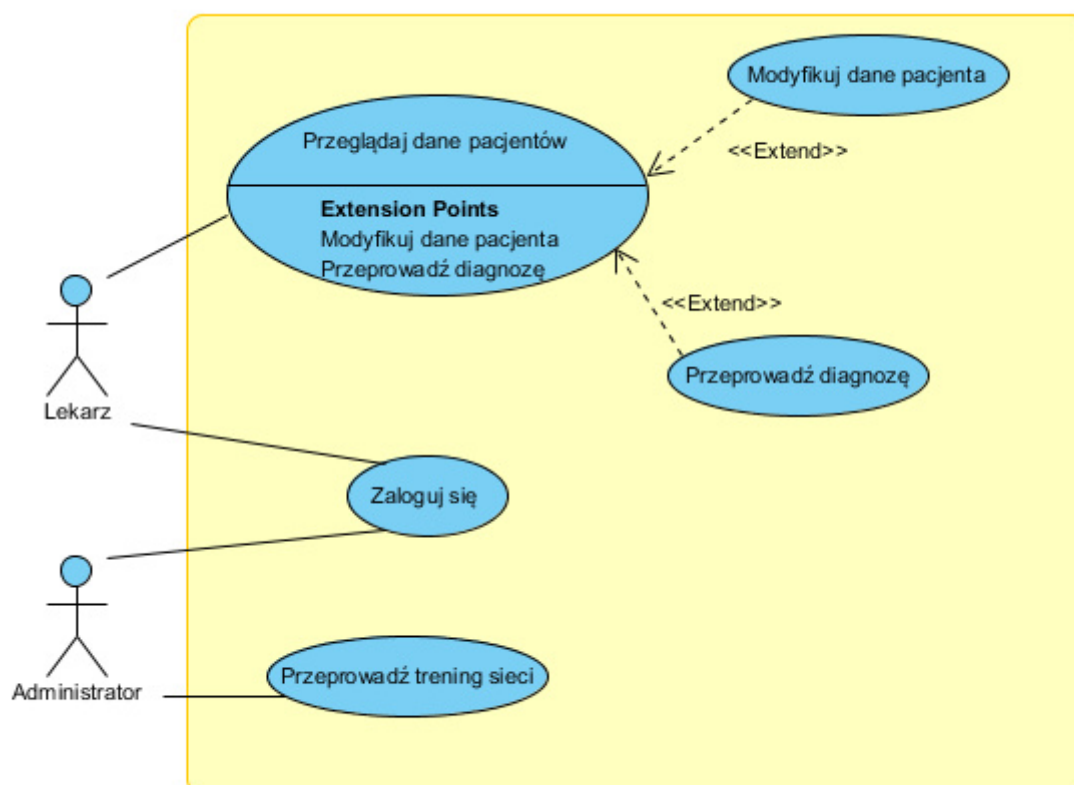
Przy zwiększeniu liczby neuronów, skuteczność zaczyna lekko maleć. Spowodowane było to tym, że przy dużej liczbie neuronów sieć szybko "przyzwyczaja" się do danych testowych.



Rysunek 8: warstwy ukryte: 1, liczba neuronów w tych warstwach: 14

4 Analiza systemu- diagramy UML

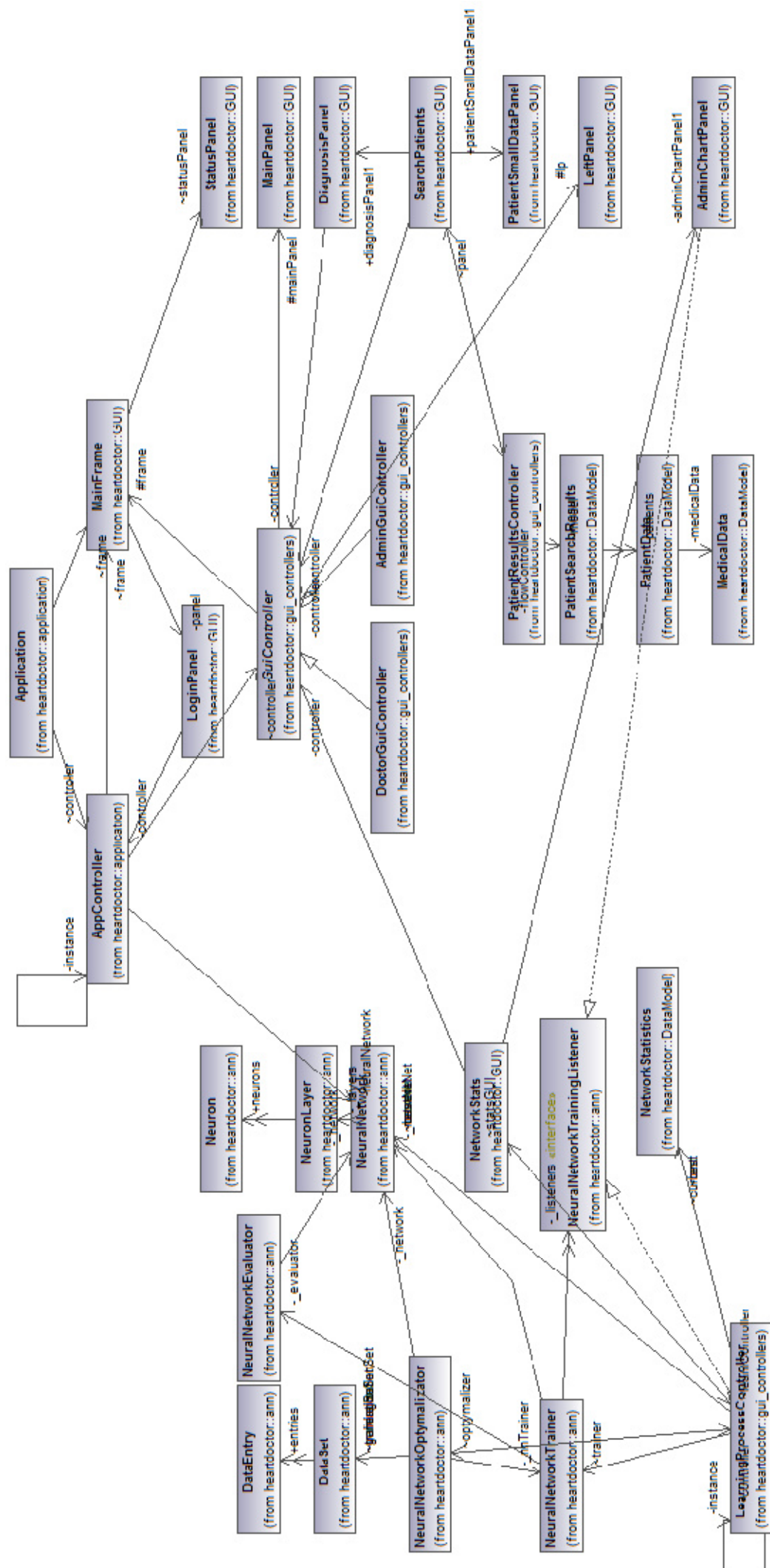
4.1 Diagram przypadków użycia systemu.

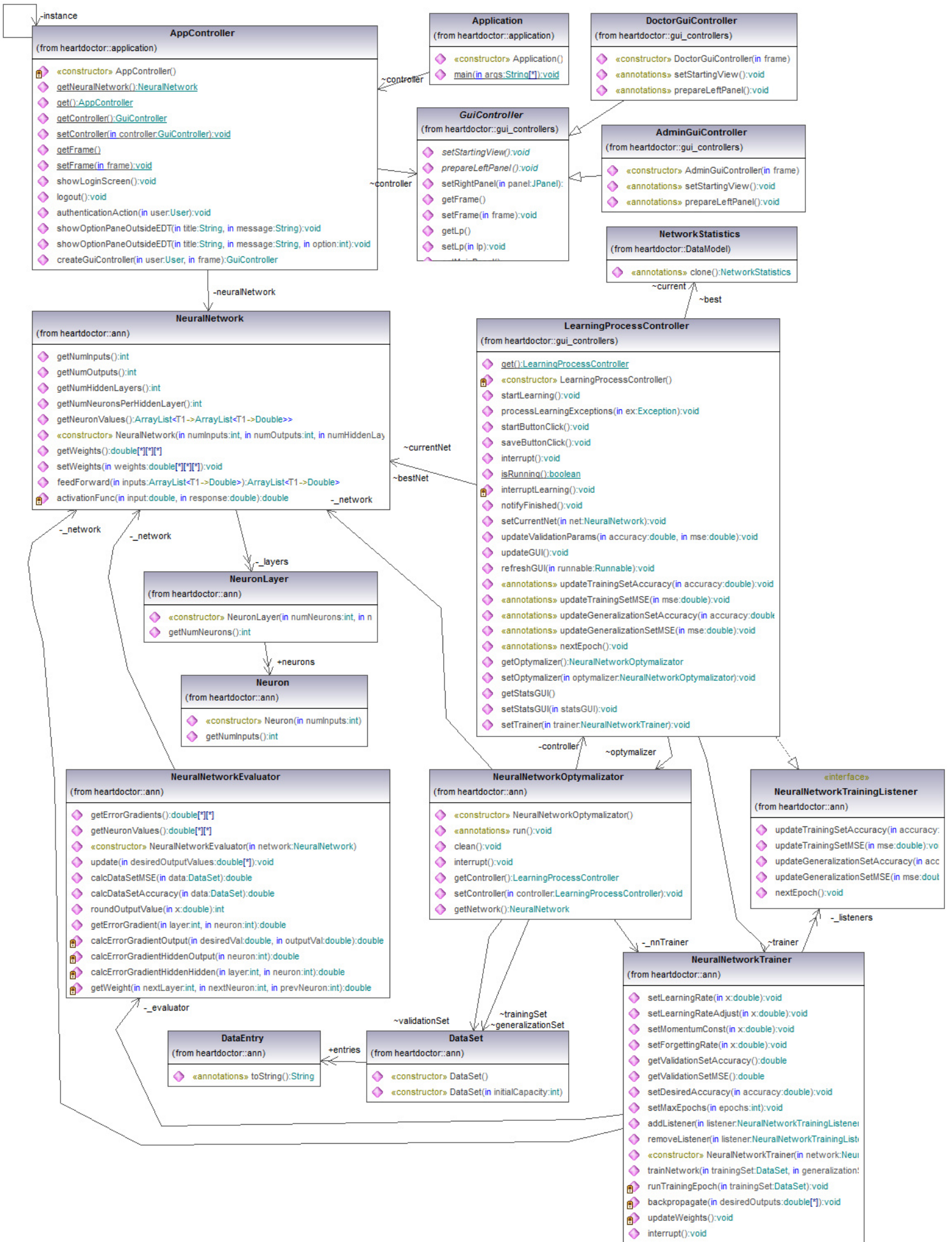


4.2 Diagramy klas systemu.

Struktura systemu jest dość rozbudowana, dlatego celem czytelniejszego jej zobrazowania zaprezentowane są dwa diagramy:

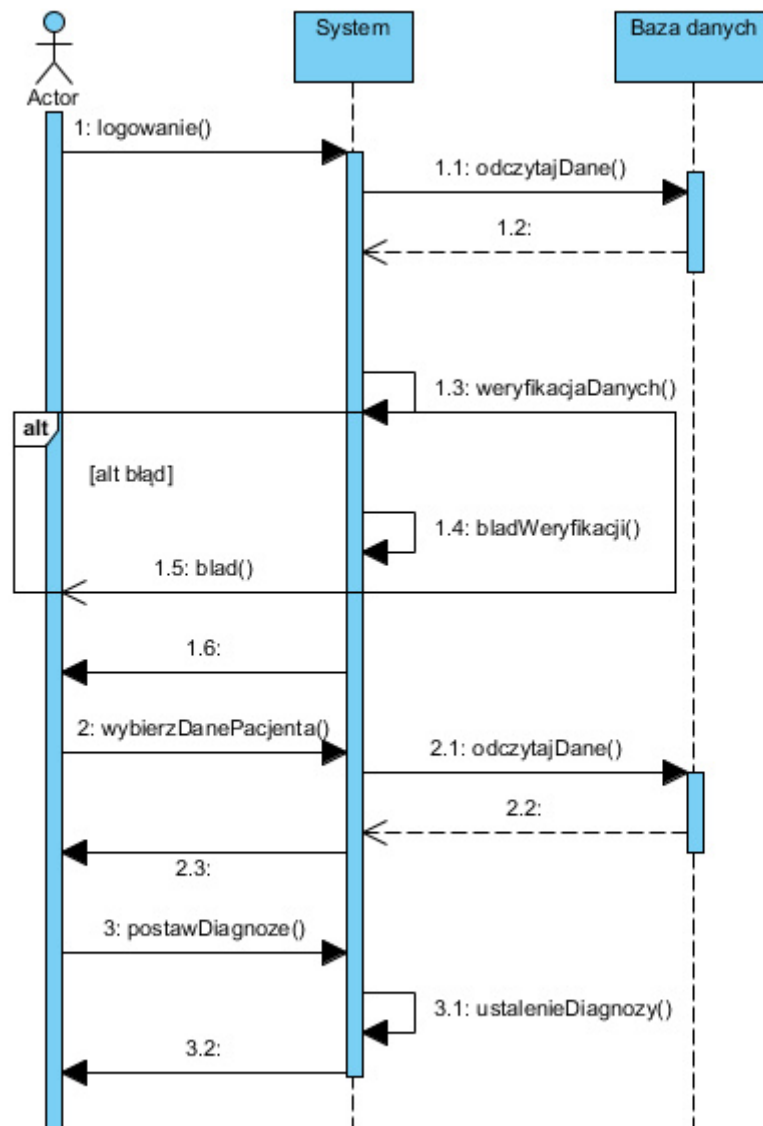
- pierwszy diagram pokazuje pełną strukturę systemu bez szczegółów dotyczących udostępnianych metod
- drugi diagram przedstawia powiązania pomiędzy najważniejszymi klasami, w szczególności tymi dotyczącymi sieci neuronowych



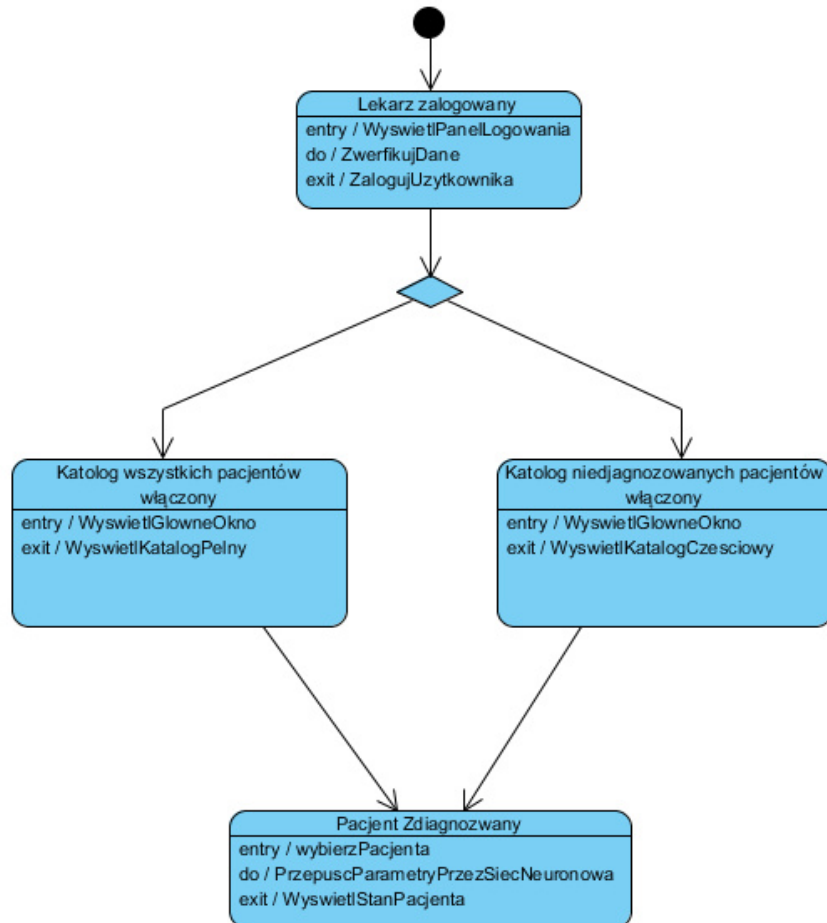


4.3 Diagramy sekwencji

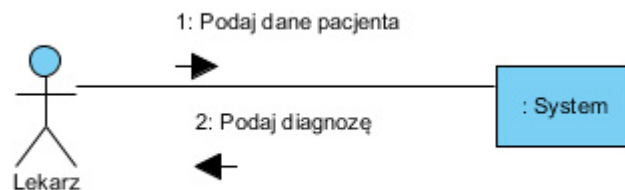
Poniższy diagram przedstawia sekwencję zdarzeń występujących przy dokonywaniu diagnozy.



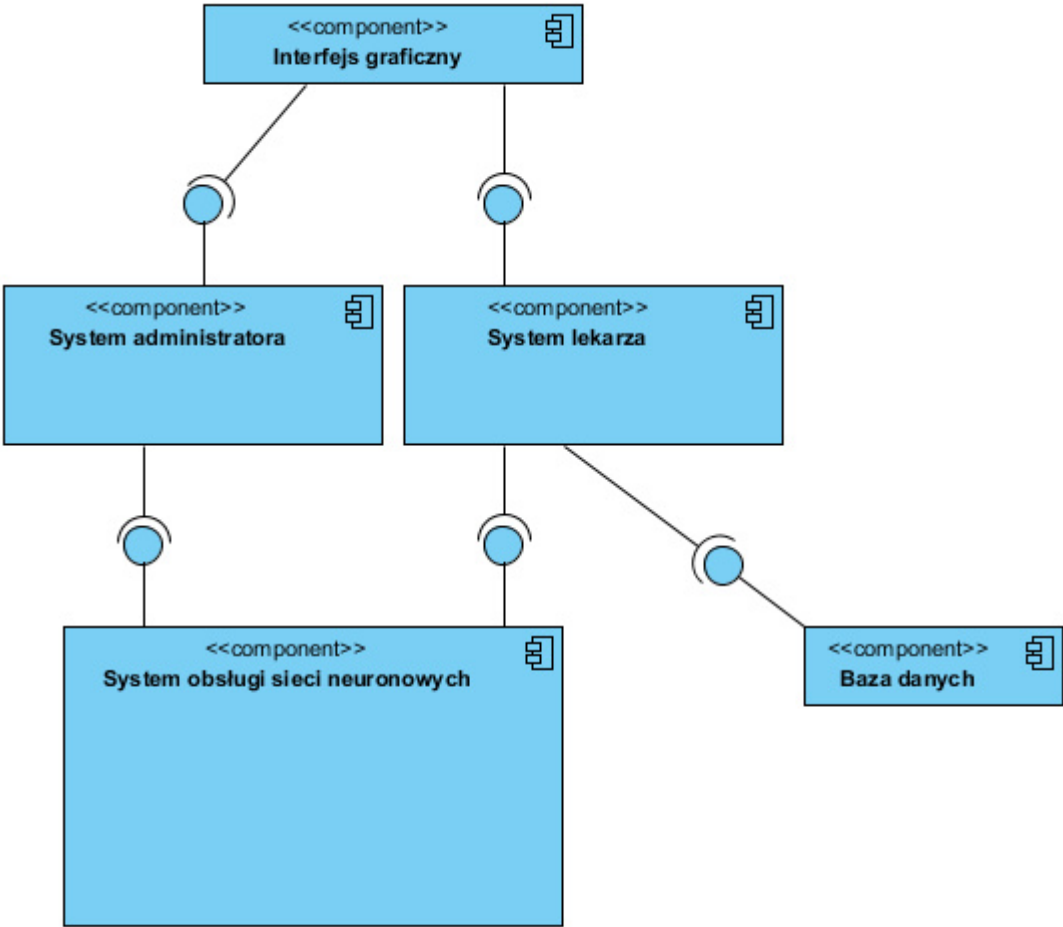
4.4 Diagram najważniejszego stanu systemu



4.5 Diagram Komunikacji systemu



4.6 Diagram Komponentów systemu



5 Opis zmian wprowadzonych w zrealizowanym systemie.

5.1 Zmiany na etapie projektowania klas

Zmiany w zakresie budowanego modelu sieci

Początkowo planowane było utworzenie sieci neuronowej w sposób bardziej uniwersalny niż w ostatecznym rezultacie. Miało to oznaczać m. in.: możliwość zmiany ilości wejść/wyjść do sieci. Czasochłonność takiego rozwiązania spowodowała zmianę podejścia na rzecz rozwiązania bardziej wyspecjalizowanego, nastawionego na rozwiązanie konkretnego problemu. Stąd też w obecnym systemie jest stała liczba parametrów wprowadzanych do sieci oraz jedno wyjście w postaci diagnozy czy pacjent jest chory/zdrowy.

5.2 Zmiany w funkcjonowaniu systemu

Zmiany w zakresie stawianej diagnozy

Początkowym założeniem systemu była diagnoza 5 najczęściej występujących chorób serca. Jednak na etapie analizy dostępnych baz danych medycznych okazało się, iż w zakresie informacji o konkretnych chorobach, do dyspozycji są one jedynie w przypadku części próbek. Reszta z nich oznaczona jest zgodnie ze schematem zdrowy/chory. W związku z tym konieczne było ograniczenie funkcjonalności systemu do informowania wyłącznie na temat występowania/niewystępowania choroby u pacjenta (bez jej identyfikacji).

6 Instrukcja obsługi Systemu

6.1 Wymagania

Przed uruchomieniem aplikacji należy upewnić się w systemie posiadamy zainstalowaną maszynę wirtualną Javy oraz jest dostępne połączenie internetowe (aplikacja łączy się z serwisem db4free.net, gdzie umieszczona jest baza danych).

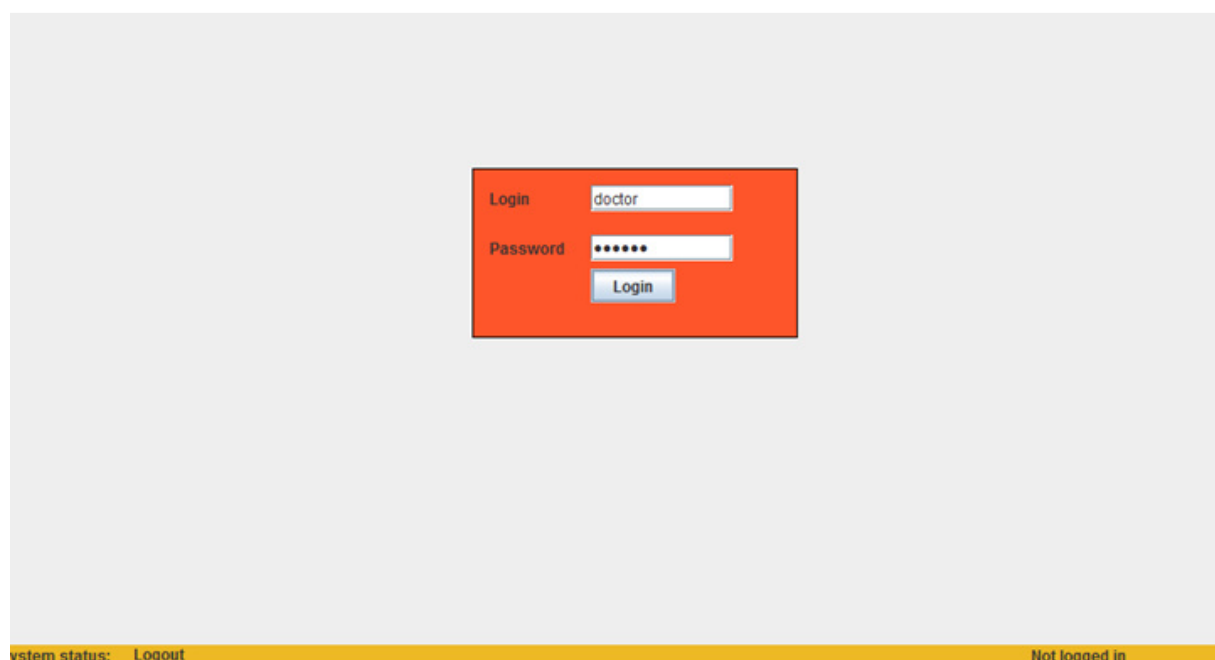
6.2 Logowanie do systemu

W systemie istnieją dwie grupy użytkowników, tj. administratorzy oraz lekarze. W celach testów zostały utworzone konta dla obu grup. Ich dane do logowania to:

ADMINISTRATOR:
Login: admin
Hasło: admin

LEKARZ:
Login: doctor
Hasło: doctor

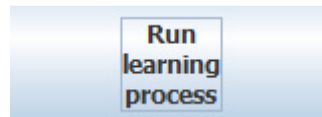
Powyższe informacje należy wpisać na ekranie logowania i kliknąć przycisk 'login'.



system status: Logout Not logged in

6.3 Panel administratora

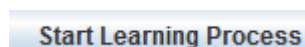
Celem wprowadzenia grupy administratorów jest umożliwienie kontroli jakości dostarczanych przez system diagnoz poprzez dostarczenie mechanizmu do optymalizacji sieci neuronowych. Uruchomienie tej funkcjonalności następuje poprzez kliknięcie przycisku:



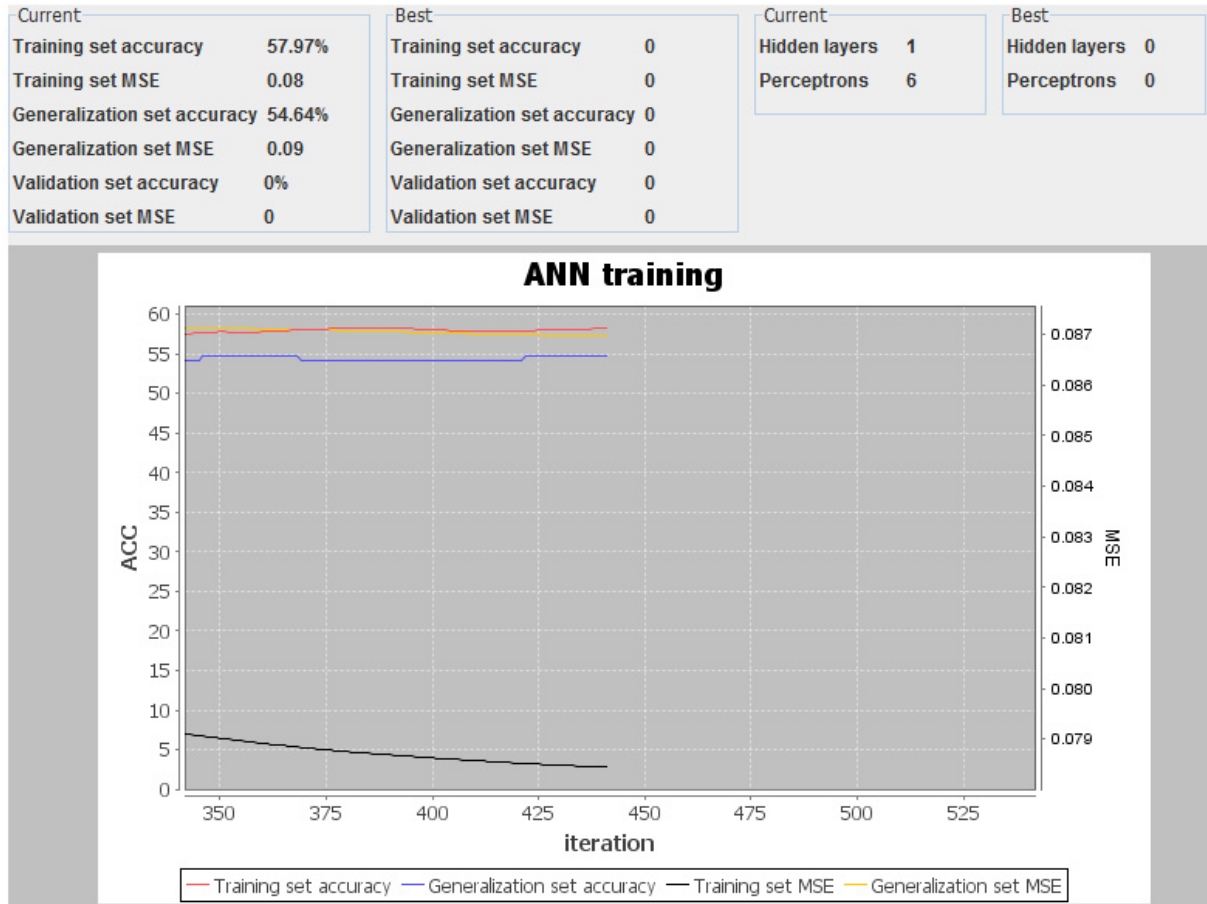
Po kliknięciu pojawia się następujący panel:



Teraz, by uruchomić proces trenowania sieci neuronowych należy kliknąć:

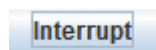


Uruchomi się proces trenowania sieci neuronowych, tak jak na poniższym przykładzie:

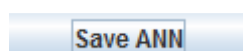


Na wykresie można obserwować jakość przeprowadzanego treningu. Z kolei ponad wykresem znajdują się okienka zawierające informacje w postaci tekstowej. By jakkolwiek trening sieci został wykonany należy poczekać przynajmniej do momentu kiedy pojawi się nowa wartość w panelu 'best' (oczywiście proces trenowania można też przerwać wcześniej jeśli uznamy iż trwa zbyt długo).

Tak więc, jeśli wyniki będą satysfakcjonujące zatrzymujemy proces trenowania klikając:

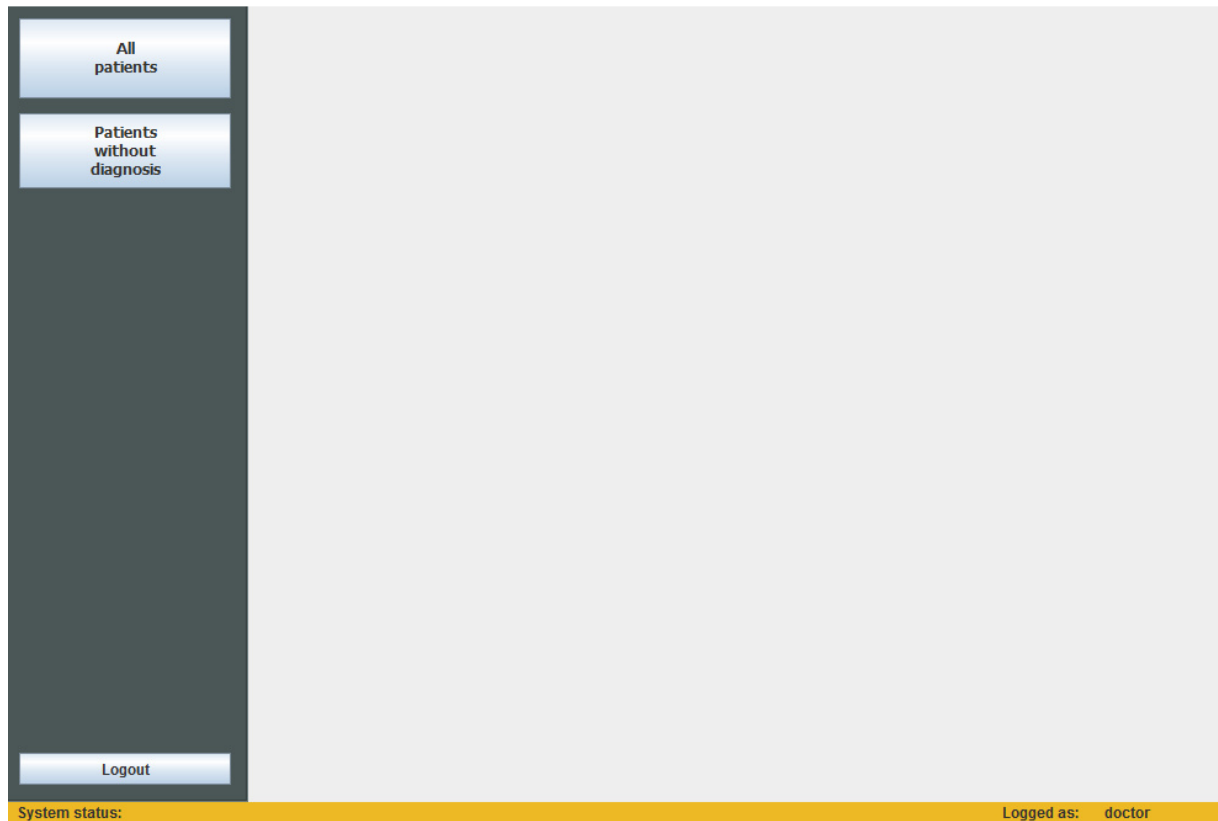


Na koniec, aby zapisać nową strukturę sieci należy potwierdzić ją przyciskiem:



6.4 Panel lekarza

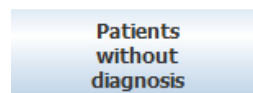
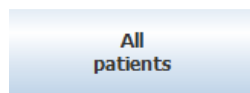
Podstawowym panelem systemu jest panel lekarza, który umożliwia stawianie diagnoz. Po zalogowaniu się na konto typu 'doctor', na ekranie pojawia się następujący ekran:



Panel umożliwia wyświetlenie dwóch katalogów:

- katalogu wszystkich pacjentów
- katalogu pacjentów bez diagnozy

Służą do tego odpowiednio przycisku:



Search by: Condition: Value:

Name = SEARCH

← Leszek Skiba →

Name: Leszek Fryderyk
Last name: Skiba
Pesel: 48050528716

Patient data preview

Resting blood pressure	<input type="text" value="145.0"/>	Chest pain type	<input type="text" value="TypicalAngina"/>
Serum cholestorlal	<input type="text" value="233.0"/>	Slope	<input type="text" value="DownSloping"/>
Fasting blood sugar	<input type="text" value="1.0"/>	Resting ECG result	<input type="text" value="Left Ventricular Hypertrophy"/>
Max heart rate	<input type="text" value="150.0"/>	Vessels colored	<input type="text" value="0"/>
Oldpeak	<input type="text" value="2.3"/>	Thal	<input type="text" value="FixedDefect"/>

Exercise induced angina

Age Sex

Diagnosis Verified diagnosis

Diagnose Save changes

Powyższe zdjęcie przedstawia widok katalogu. Służy on przede wszystkim do przeszukiwania bazy pacjentów, modyfikowania informacji o pacjencie oraz przeprowadzania diagnozy.

Przeszukiwanie bazy:

Search by: Condition: Value:

Parametr do przeszukania: Lastname = Wartość parametru: SKIBA Włączenie przeszukiwania: SEARCH

Results: 3

← Patient data: Leszek Skiba →

Leszek Skiba
Apolinary Skiba
Emil Skiba

Name: Leszek Fryderyk
Last name: Skiba
Pesel: 48050528716

Liczba znalezionych wyników

Modyfikowanie informacji o pacjencie

Resting blood pressure <input type="text" value="145.0"/>	Chest pain type <input type="text" value="TypicalAngina"/>
Serum cholestorol <input type="text" value="233.0"/>	Slope <input type="text" value="Down Sloping"/>
Fasting blood sugar <input type="text" value="1.0"/>	Resting ECG result <input type="text" value="Left Ventricular Hypertrophy"/>
Max heart rate <input type="text" value="150.0"/>	Vessels colored <input type="text" value="0"/>
Oldpeak <input type="text" value="2.3"/>	Thal <input type="text" value="FixedDefect"/>
Exercise induced angina <input type="checkbox"/>	Age <input type="text" value="63.0"/> Sex <input type="text" value="Male"/>
Diagnosis <input type="text" value="Not diagnosed"/>	Verified diagnosis <input type="text" value="Healthy"/>

Po wprowadzeniu zmian należy nacisnąć przycisk:

Save changes

Przeprowadzenie diagnozy

By przeprowadzić diagnozę dla danego pacjenta należy nacisnąć przycisk:

Diagnose

Zapisanie diagnozy następuje po kliknięciu przycisku:

Save changes

6.5 Wylogowanie użytkownika z systemu

By wylogować użytkownika z systemu należy nacisnąć przycisk:

Logout

6.6 Pasek stanu

W celu bieżącego informowania użytkownika o akcjach wykonywanych przez system, utworzono pasek stanu, który znajduje się u dołu aplikacji:

System status: LOGOWANIE...

Not logged in

7 Bibliografia

- **Tytuł:** Neural Network application in diagnosis of patient: A case study
Autorzy: Gharehchopogh, F.S. ; Khalifelu, Z.A
Rok wydania: 2011
Źródło: **Computer Networks and information technology**
- **Tytuł:** The Application of Ontology in a Clinical Decision Support System for Acute Postoperative pain Management
Autorzy: Hazmy Iman Abas, Maryati Mohd. Yusof, Shahrul Azman Mohd Noah
Rok wydania: 2011
Źródło: Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on
- **Tytuł:** Clinical Decision Support Systems: Medical knowledge acquisition and representation methods
Autorzy: Aleksovska-Stojkowska, L.; Loskovska, S.;
Rok wydania: 2010
Źródło: Electro/Information Technology (EIT), 2010 IEEE International Conference on
- **Tytuł:** Decision support system for heart disease based on support vector machine and Artificial Neural Network
Autorzy: Gudadhe-M; Wankhade-K; Dongre-S
Rok wydania: 2010
Źródło: Computers in Biology and Medicine
- **Tytuł:** Modeling and design of evolutionary neural network for heart disease detection
Autorzy: K.S.Kavitha, K.V.Ramakrishnan, Manoj Kumar Singh
Rok wydania: 2010
Źródło: International Journal of Computer Science Issues
- **Tytuł:** Intelligent Diagnosis of Heart Diseases using Neural Network Approach
Autorzy: Ranjana Raut, Dr. S. V. Dudu
Rok wydania: 2010
Źródło: International Journal of Computer Applications
- **Tytuł:** A Proposal for Ontology-based Integration of Heterogeneous Decision Support Systems for Structural Health Monitoring
Autorzy: Stefan Anderlik and Reinhard Bernhard Freudenthaler Martin Fritz
Rok wydania: 2010
Źródło: Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services
- **Tytuł:** Clinical Decision Support Systems: State of the Art"
Autorzy: Eta S. Berner, Ed.D.
Rok wydania: 2009
Źródło: AHRQ Publication

- **Tytuł:** Support Vector Machine for Assistant Clinical Diagnosis of Cardiac Disease
Autorzy: Gong Wei, Wang Shoubin
Rok wydania: 2009
Źródło: Global Congress on Intelligent Systems
- **Tytuł:** Clinical Decision Support System for Fetal Delivery Using Artificial Neural Network
Autorzy: Janghel, R.R. ; Shukla, A. ; Tiwari, R. ; Tiwari, P.
Rok wydania: 2009
Źródło: New Trends in Information and Service Science, 2009. NISS '09. International Conference on
- **Tytuł:** A clinical decision support system using multilayer perceptron neural network to assess well being in diabetes
Autorzy: MR Narasingarao, R Manda, GR Sridhar, K Madhu, AA Rao
Rok wydania: 2009
Źródło: The Journal of the Association of Physicians of India
- **Tytuł:** Effective diagnosis of heart disease through neural networks ensembles
Autorzy: Resul Dasa, Ibrahim Turkoglu, Abdulkadir Sengurb
Rok wydania: 2009
Źródło: Expert Systems with Applications: An International Journal
- **Tytuł:** A clinical decision support system for cancer diseases
Autorzy: André Cid Ferrizzi, Toni Jardini
Rok wydania: 2008
Źródło: Proceedings of the 2008 Euro American Conference on Telematics and Information Systems
- **Tytuł:** Intelligent heart disease prediction system using data mining techniques
Autorzy: Awang, R. Palaniappan, S. ;
Rok wydania: 2008
Źródło: Computers and System Applications 2008 International Conference
- **Tytuł:** Suggested criteria for successful deployment of a Clinical Decision Support System (CDSS)
Autorzy: Frize, M.; Weyand, S.; Barciak, E.;
Rok wydania: 2008
Źródło: Medical Measurements and Applications Proceedings 2010 International Workshop
- **Tytuł:** Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance
Autorzy: Maciej A. Mazurowski, Piotr A. Habas, Jacek M. Zurada, Joseph Y. Lo, Jay A. Baker, Georgia D. Tourassi
Rok wydania: 2008
Źródło: The Official Journal of the International Neural Network Society, European Neural Network Society \& Japanese Neural Network Society
- **Tytuł:** Evolved Bayesian Networks as a Versatile Alternative to Partin Tables for Prostate Cancer Management
Autorzy: Ratiba Kabli, John McCall
Rok wydania: 2008
Źródło: Proceedings of the 10th annual conference on Genetic and evolutionary computation
- **Tytuł:** Decision Support in Health Care via Root Evidence Sampling
Autorzy: Benjamin Perry ; Eli Faulkner ;
Rok wydania: 2007
Źródło: Proceedings of the 40th Hawaii International Conference on System Sciences - 2007

- Tytuł:** Decision Support in Health Care via Root Evidence Sampling
Autorzy: Benjamin Perry ; Eli Faulkner ;
Rok wydania: 2007
Źródło: System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on
- Tytuł:** A multilayer perceptron-based medical decision support system for heart disease
Autorzy: Hongmei Yana, Yingtao Jiangb, Jun Zhenge, Chenglin Pengc, Qinghui Lid
Rok wydania: 2006
Źródło: Expert Systems with Applications
- Tytuł:** Designing an Artificial Neural Network Model for the Prediction of Thrombo-embolic Stroke
Autorzy: D.Shanthi, Dr.G.Sahoo, Dr.N.Saravanan
Rok wydania: 2005
Źródło: International Journals of Biometric and Bioinformatics
- Tytuł:** Clinical Decision Support based on Mobile Telecommunication Systems
Autorzy: Tsumoko Shusaku, Hirano Shoji, Abe Hidenao, Nakakuni Hideaki
Rok wydania: 2005
Źródło: WI '05 Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence
- Tytuł:** Medical diagnosis using neural networks
Autorzy: M. Kamruzzaman, Ahmed Ryadh Hasan, Abu Bakar Siddiquee, Md. Ehsanul Hoque Mazumder
Rok wydania: 2004
Źródło: 3rd International Conference on Electrical \& Computer Engineering
- Tytuł:** Research on medical diagnostic decision-making based on attribute reduction and support vector machines
Autorzy: Zhonghui Hu, Yuangui Li, Yunze Cai, Xiaoming Xu
Rok wydania: 2004
Źródło: Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress
- Tytuł:** "Research on Medical Diagnostic Decision-making Based on Attribute Reduction and Support Vector Machines"
Autorzy: Zhonghui Hu, Yuangui Li, Yunze Cai, Xiaoming Xu
Rok wydania: 2004
Źródło: Proceedings of the 5th World Congress on Intelligent Control and Automation
- Tytuł:** Bayesian Networks and Decision Trees in the Diagnosis of Female Urinary Incontinence
Autorzy: Hunt, M.; von Kinsky, B; Venkatesh, S.; Petros, P.
Rok wydania: 2000
Źródło: Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE

8 Harmonogram projektu i podział zadań

Podział zadań:

- Bartłomiej Hyży:
 - projekt system
 - podział zadań, koordynacja
 - implementacja sztucznej sieci neuronowej i algorytmu uczącego
- Michał Pieróg
 - postawienie i administracja bazą danych medycznych
 - implementacja komunikacji z bazą danych
 - projekt i implementacja interfejsu użytkownika (GUI) aplikacji
 - nagranie filmów instruktażowych do aplikacji
- Witold Baran
 - implementacja treningu sieci w module administrator
 - implementacja części interfejsu użytkownika
- Jakub Jaśkowiec
 - implementacja części interfejsu użytkownika
 - implementacja komunikacji z bazą danych
 - role użytkowników, wyszukiwanie pacjentów w module lekarze
- Mateusz Niesyto
 - pomoc w implementacji treningu sieci w module administrator
 - wykonanie dokumentacji projektowej
 - wykonanie prezentacji projektu

Harmonogram

Poniższy spis przedstawia terminy wykonania kolejnych etapów projektu. W nawiasie zostały zawarte rzeczywiste daty, w których udało się zrealizować dany etap.

1. **Start projektu**
Projektowanie, zbieranie materiałów, tworzenie baz danych dla modelu, itp.
10.11.2011 (10.11.2011)
2. **Implementacja**
Realizacja właściwa projektu, pisanie kodu, wykonywanie testów, itp.
22.11.2011 (26.11.2011)
3. **Dokumentacja**
Pisanie dokumentacji, tworzenie prezentacji, raportu, itp.
31.11.2011 (30.11.2011)