



Katedra Technologii Postaci Leku i Biofarmacji
Wydziału Farmaceutycznego
Collegium Medicum
Uniwersytetu Jagiellońskiego
w Krakowie



**SZTUCZNE SIECI NEURONOWE JAKO UNIWERSALNE
NARZĘDZIA DO MODELOWANIA W TECHNOLOGII
POSTACI LEKU I BIOFARMACJI**

PRACA DOKTORSKA

Aleksander Mendyk

Promotor

Prof. UJ, dr hab. Renata Jachowicz

Kraków 2003

*Składam podziękowania Pani Prof. Renacie
Jachowicz za cenne rady i wszelkie wsparcie
udzielone podczas pisania niniejszej pracy*

I. Wstęp	3
1. SZTUCZNE SIECI NEURONOWE	3
1.1. SIECI UCZONE Z NAUCZYCIELEM	8
1.2. SIECI UCZONE BEZ NAUCZYCIELA	17
1.3. SIECI REKURENCYJNE.....	21
1.4. SIECI ROZMYTE	22
2. MODELOWANIE KLASYCZNE A NEURONOWE	25
3. ZASTOSOWANIA SZTUCZNYCH SIECI NEURONOWYCH W TECHNOLOGII POSTACI LEKU	28
3.1. STAŁE POSTACIE LEKU	29
3.2. EMULSJE	33
3.3. MIKROEMULSJE	34
3.4. HYDROŻELE	35
3.5. TRANSDERMALNE SYSTEMY TERAPEUTYCZNE (TTS)	36
3.6. BADANIA PREFORMULACYJNE.....	37
3.7. KORELACJA IN VITRO-IN VIVO.....	37
3.8. APARATURA	38
II. Założenia i cel pracy	39
III. Część doświadczalna	41
1. METODYKA	41
1.1. MODELE NEURONOWE	41
1.2. WERYFIKACJA MODELI NEURONOWYCH	42
1.3. NOTACJA	43
2. OPRZYRZĄDOWANIE	44
2.1. SYMULATOR SIECI NEURONOWYCH	45
2.2. OPROGRAMOWANIE NARZĘDZIOWE	48
2.3. SPRZĘT KOMPUTEROWY	51
3. MODELOWANIE NEURONOWE	52
3.1. STAŁE ROZPROSZENIA.....	52
3.1.1. Zakres badań	53
3.1.2. Zdefiniowanie funkcji systemu neuronowego	57
3.1.3. Przygotowanie danych stanowiących bazę wiedzy	57

3.1.4. Budowa maszyny wnioskującej	62
3.1.5. Budowa interfejsu użytkownika.....	64
3.1.6. Wyniki badań.....	65
3.2. MIKROEMULSJE	80
3.2.1. Zakres badań	81
3.2.2. Zdefiniowanie funkcji systemu neuronowego	81
3.2.3. Przygotowanie danych stanowiących bazę wiedzy	84
3.2.4. Budowa maszyny wnioskującej	88
3.2.5. Budowa interfejsu użytkownika.....	91
3.2.6. Wyniki badań.....	91
3.3. APROKSYMACJA FUNKCJI STĘŻENIE-CZAS W BADANIACH <i>IN VITRO</i> I <i>IN VIVO</i>	107
3.3.1. Zakres badań	107
3.3.2. Zdefiniowanie funkcji systemu neuronowego	107
3.3.3. Przygotowanie danych stanowiących bazę wiedzy	107
3.3.4. Budowa maszyny wnioskującej	108
3.3.5. Budowa interfejsu użytkownika.....	110
3.3.6. Wyniki badań.....	111
IV. Omówienie i dyskusja wyników	116
1. STAŁE ROZPROSZENIA.....	116
2. MIKROEMULSJE	123
3. APROKSYMACJA FUNKCJI STĘŻENIE-CZAS W BADANIACH <i>IN VITRO</i> I <i>IN VIVO</i>..	133
V. Wnioski	139
VI. Streszczenie	141
VII. Symbole.....	144
VIII. Piśmiennictwo	145
IX. Oprogramowanie.....	159
1. ALGORYTM BUDOWY SYSTEMU <i>SD_EXPERT</i>.....	159
2. ZRZUTY EKRAKOWE NAPISANYCH PROGRAMÓW	160

I. WSTĘP

Klasyczne modelowanie polega na identyfikacji analitycznej postaci funkcji wiążącej wektor parametrów układu z wektorem jego stanów. Z uwagi na duży stopień komplikacji niektóre zagadnienia trudno jest analizować w sposób klasyczny, co zmusza do poszukiwania alternatywnych narzędzi analizy danych jak np. sztuczne sieci neuronowe.

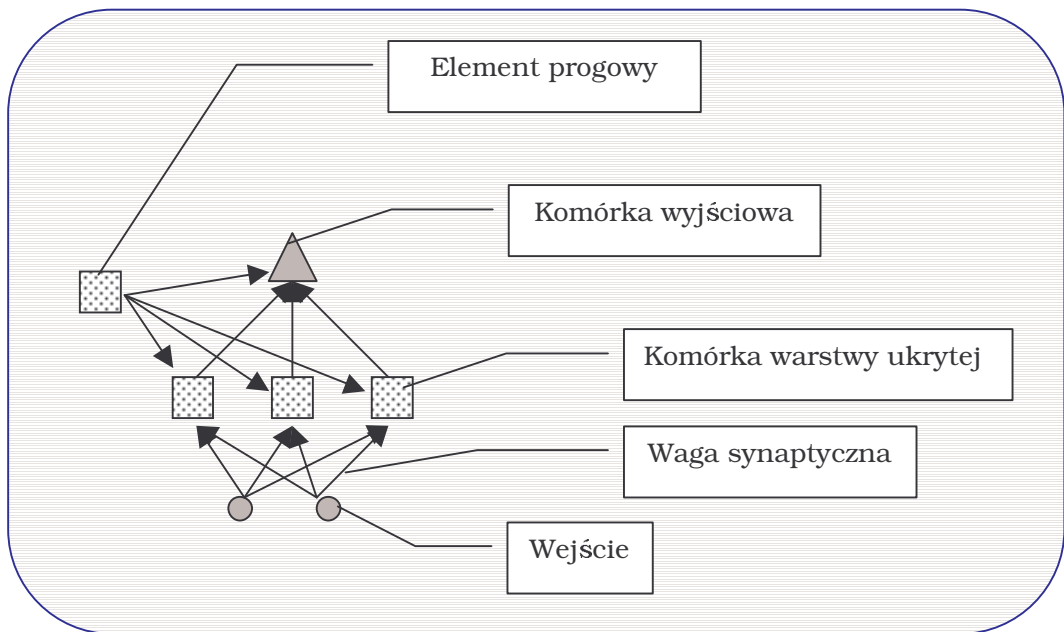
1. SZTUCZNE SIECI NEURONOWE

Sztuczna sieć neuronowa (sieć neuronowa) to system połączonych ze sobą, prostych jednostek przetwarzających informację i zorganizowanych w hierarchiczne struktury [76]. Ta ogólna definicja Kohonena obejmuje wiele typów systemów analizy danych, których wspólną cechą jest zdolność do uczenia się na danych empirycznych i brak wstępnych założeń co do postaci otrzymanego modelu.

Zarówno strukturalnie jak i funkcjonalnie sztuczne sieci neuronowe są pewnym przybliżeniem aktualnych teorii na temat budowy i funkcjonowania biologicznych układów nerwowych. Pierwsze badania nad sieciami neuronowymi datuje się na lata 40 ubiegłego wieku. Zakładano wtedy, że w ciągu niedługiego okresu czasu uda się modelować elektronicznie procesy zachodzące w ludzkim mózgu. Stąd też sztuczne sieci neuronowe są często zaliczane do szeroko rozumianej sztucznej inteligencji. Ponieważ szacunkowa liczba neuronów w mózgu oceniana jest na ok. 10^{11} , nie jest na razie możliwe spełnienie wstępnych założeń badań nad sieciami neuronowymi. Jednak fakt poprawnego funkcjonowania sztucznych sieci neuronowych będących tzw. systemami konekcyjnymi przyczynił się do stworzenia obowiązującego dziś modelu pracy mózgu, który określa wiedzę jako funkcjonalny i topologiczny stan połączeń pomiędzy neuronami [52, 53, 77, 94, 114, 136].

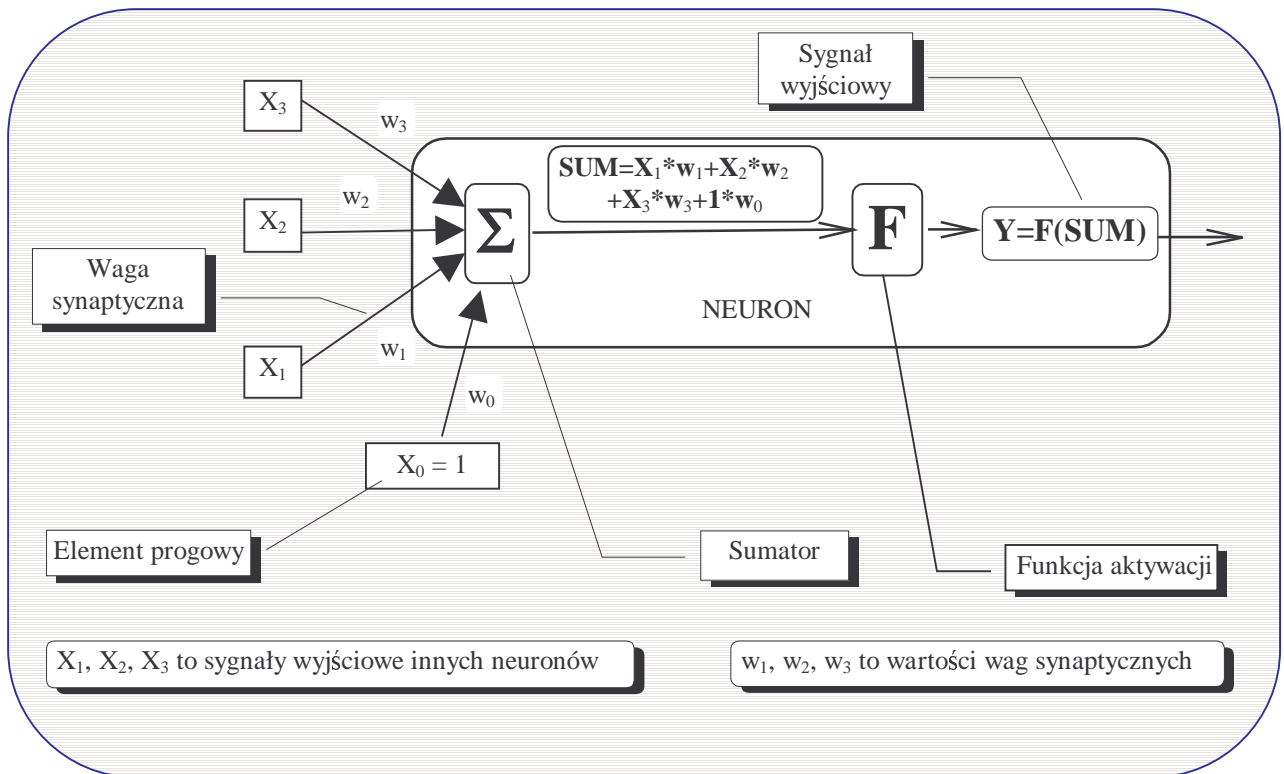
Rozpatrując przytoczoną powyżej definicję Kohonena należy zdefiniować sieci neuronowe jako zbiór sztucznych neuronów będących

jednostkami przetwarzającymi informację i połączonych ze sobą tzw. wagami synaptycznymi symulującymi synapsy biologiczne. Wagi synaptyczne są zazwyczaj reprezentowane przez pojedynczą liczbę, której znak i wartość określają charakter (pobudzający lub hamujący) i siłę przenoszonego pobudzenia. Sygnał z reguły przenoszony jest w jednym kierunku, chociaż znane są wagi dwukierunkowe (Rycina 1).



Rycina 1. Ogólny model sieci neuronowej.

Podstawowy model sztucznego neuronu zaproponowany przez McCullocha i Pittsa [82] zakłada istnienie w sztucznym neuronie tzw. bloku sumatora agregującego wszystkie sygnały przychodzące do sztucznego neuronu i bloku funkcji aktywacji przekształcającej zsumowany sygnał na sygnał wyjściowy (Rycina 2).



Rycina 2. Model sztucznego neuronu McCullocha i Pittsa.

Sztuczne neurony są zorganizowane w danej sieci w warstwy, które mogą się komunikować na zasadzie „każdy z każdym” lub tylko pomiędzy przylegającymi do siebie warstwami. Element progowy jest pojedynczą jednostką, która z kolei jest połączona ze wszystkimi neuronami w sieci i symuluje próg pobudzenia komórki nerwowej. Posiada stałą, jednostkową wartość. Istnieje wiele wariantów tzw. architektur sieci neuronowych, czyli topologicznego ułożenia warstw sztucznych neuronów i ich wzajemnych połączeń. Niestety nie są znane obiektywne metody doboru właściwej architektury sieci neuronowej dla konkretnego problemu. Wiele architektur wzorowanych jest na biologicznych pierwowzorach jak np. pola recepcyjne siatkówki oka. Istnieje wiele algorytmów automatycznego doboru architektury. Generalnie dzieli się je na trzy grupy:

- a) algorytmy konstrukcyjne
- b) algorytmy destrukcyjne
- c) algorytmy optymalizacji dyskretnej [91]

Algorytmy konstrukcyjne polegają na budowie sieci neuronowej od najmniejszej liczby neuronów do największej, optymalnej liczby neuronów i wag. Najczęściej proces uczenia jest powtarzany podczas stopniowego budowania struktury sieci neuronowej. Przykładami algorytmów konstrukcyjnych mogą być:

- a) algorytm kafelkowy
- b) algorytm upstart Freana
- c) wsteczna propagacja ze zmienną liczbą neuronów
- d) dynamiczna kreacja neuronów
- e) metoda formy kanonicznej
- f) kaskadowa korelacja Fahlmana [91, 94]

Algorytmy destrukcyjne działają na odwrotnej zasadzie: najpierw trenowana jest duża sieć neuronowa, po czym usuwane są z niej niepotrzebne elementy. Podział algorytmów destrukcyjnych ze względu na metodę redukcji sieci może wyglądać następująco:

- a) wrażliwościowe
 - algorytm OBD (*Optimal Brain Destruction*)
 - algorytm OBS (*Optimal Brain Surgery*)
- b) funkcji kary
- c) analizy kowariancji [91, 94]

Algorytmy optymalizacji dyskretnej traktują zbiór architektur sieci neuronowych jako nieskończoną przestrzeń dyskretną. W celu znalezienia optymalnej architektury stosowane są następujące podejścia:

- a) algorytmy genetyczne
- b) symulowane wyżarzanie
- c) algorytm A* [91]

Zarówno konstrukcyjne, jak i destrukcyjne podejście do automatycznej budowy sieci neuronowej nie pozwala na niezawodną optymalizację jej architektury. Podstawowymi problemami są:

- a) koszt obliczeniowy

- b) brak kryteriów określenia optymalnej architektury sieci neuronowej
- c) specyficzność danego algorytmu dla danego typu zagadnienia (np. klasyfikacja, aproksymacja) [91]

Z uwagi na to dobór architektury przeprowadza się często na drodze żmudnych eksperymentów numerycznych, które polegają na trenowaniu zespołu architektur i ręcznej ocenie ich sprawności, a następnie na decyzji użytkownika co do modyfikacji architektur do dalszych eksperymentów [52, 91, 94, 114, 136].

Szczegółowy opis struktury sieci neuronowych jest nierozdzielnie związany z ich cechami funkcjonalnymi, wśród których za podstawową uznano zdolność do uczenia się na danych empirycznych. Algorytmy uczące sieci neuronowych dzielą się na dwie podstawowe grupy:

- a) uczenie z nauczycielem
- a) uczenie bez nauczyciela [52, 94]

Różnica pomiędzy obydwoma grupami algorytmów uczących polega na dostępności danych opisujących stan badanego układu (S), odpowiadający danym parametrom układu (X). W algorytmach z nauczycielem, znane są w czasie uczenia stany układu, przedstawiane jako oczekiwane wyjścia sieci neuronowej w odpowiedzi na dane wejścia. Różnica pomiędzy aktualnym a zadaniem wyjściem sieci używana jest do zmiany wartości wag, co stanowi właściwy proces uczenia się sieci neuronowej. Sieci takie nazywane są perceptronami. Nazwa ta posiada historyczne uwarunkowania. Rosenblatt stworzył w latach pięćdziesiątych ubiegłego stulecia pierwszy funkcjonujący model sieci neuronowej, który określił nazwą „PERCEPTRON” [108]. Odmienna sytuacja występuje w sieciach uczonych bez nadzoru, gdzie nie są znane zadane wyjścia sieci. Algorytm pracuje na zasadzie identyfikacji podobieństw pomiędzy danymi wejściowymi grupując je w klasy. Uczenie sieci neuronowej może być prowadzone w oparciu o różne systemy logiczne. Logika boolowska, albo inaczej dwuwartościowa jest podstawą klasycznej matematyki. Można jednak użyć innych, alternatywnych systemów logicznych, wśród których dużym zainteresowaniem cieszy się tzw. logika rozmyta [100, 131].

1.1. SIECI UCZONE Z NAUCZYCIELEM

Podczas uczenia z nauczycielem znane są stany badanego układu (S) w odpowiedzi na konkretne jego parametry (X). Wiedza ta pozwala uczyć się dzięki znajomości jej aktualnego błędu, który jest podstawą do zmiany wag. Wektor X stanowi wejścia sieci neuronowej, a wektor stanów S jej oczekiwane wyjścia. Najbardziej znanym algorytmem wykorzystywanym w tym celu jest tzw. propagacja wsteczna (ang. *backpropagation* – BP). Jest to metoda gradientowej optymalizacji parametrów swobodnych systemu nieliniowego (rzadziej liniowego), które w przypadku sieci neuronowej są jej wagami synaptycznymi. Istotą działania algorytmu jest przepuszczanie sygnału zmieniającego wartość wag wstecz w stosunku do właściwego kierunku przechodzenia sygnału przez sieć neuronową. Sygnał uczący, będący aktualnym błędem sieci neuronowej, pozwala na modyfikację wartości wag w celu znalezienia najlepszego ich zestawu odpowiadającego optymalnemu modelowi badanego zagadnienia. Równanie 1 przedstawia ogólny zapis matematyczny zmian wag synaptycznych w zależności od aktualnej wielkości błędu sieci [94].

$$W_{t+1} = W_t - \eta \nabla E_t \quad (1)$$

gdzie :

W – wektor wag synaptycznych

∇E – gradient funkcji kosztu

η – współczynnik szybkości uczenia

t – krok uczenia

W zapisie skalarnym równanie dla zmiany pojedynczej wagi wygląda następująco:

$$w_{ij}^{t+1} = w_{ij}^t - \eta \frac{\delta E}{\delta w_{ij}^t} \quad (2)$$

gdzie:

w_{ij}^t - waga synaptyczna łącząca i-tą komórkę z j-tą komórką innej warstwy

$-\eta \frac{\partial E}{\partial w_{ij}^t}$ - wartość zmiany wagi w kroku uczenia t będąca pochodną cząstkową funkcji kosztu względem danej wagi synaptycznej

Oznaczając:

$$\Delta w_{ij}^t = -\eta \frac{\partial E}{\partial w_{ij}^t}$$

można przepisać równanie 2 do postaci równania 3.

$$w_{ij}^{t+1} = w_{ij}^t + \Delta w_{ij}^t \quad (3)$$

Funkcja kosztu E może przyjmować różne postacie, jednak najczęściej używana jest funkcja kwadratowa (równanie 4) [52].

$$E = \frac{1}{2} \sum_{i=1}^N (s_i - s_i^{\text{wzorzec}})^2 \quad (4)$$

gdzie:

i – numer wyjścia

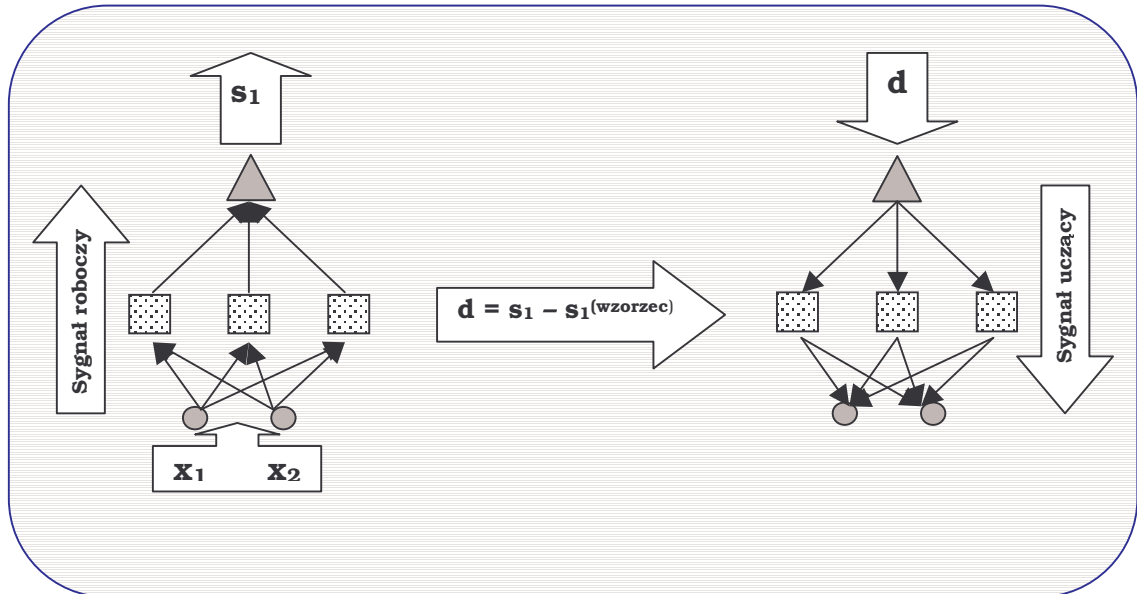
N – całkowita liczba neuronów wyjściowych

s_i – aktualna wartość wyjścia

s_i^{wzorzec} – zadana wartość wyjścia

Schemat uczenia metodą propagacji wstecznej przedstawiono na rycinie 3. W pierwszym etapie następuje przepuszczenie sygnału roboczego sieci wytworzonego po podaniu na wejścia sieci parametrów układu \mathbf{x}_1 i \mathbf{x}_2 , w wyniku czego uzyskuje się na wyjściu sieci stan układu \mathbf{s}_1 . Następnie obliczany jest błąd sieci \mathbf{d} jako różnica aktualnego wyjścia sieci \mathbf{s}_1 , a wzorcowym wyjściem $\mathbf{s}_1^{\text{wzorzec}}$. Błąd ten jest propagowany wstecz jako sygnał

uczący w drugim etapie uczenia sieci. Sygnał uczący służy do zmiany wartości wag, które kodują całą „wiedzę” sieci na temat opracowywanego problemu.



Rycina 3. Schemat uczenia propagacją wsteczną.

Proces uczenia powtarzany jest iteracyjnie aż do osiągnięcia zadanego uprzednio minimum błędu [52]. Popularność tego algorytmu wiąże się z jego matematyczną prostotą i uniwersalnością. Dzięki temu można nauczyć dowolną sieć neuronową wykonywania praktycznie dowolnych czynności: od prostego sumowania dwóch liczb, po przewidywanie kursów akcji na giełdzie. Ceną uniwersalności propagacji wstecznej jest długotrwałość procesu uczenia. Z uwagi na to powstało wiele modyfikacji propagacji wstecznej, wśród których należy wymienić:

- a) technikę momentu
- b) zmienne współczynniki uczenia i momentu
 - zmniejszanie wykładnicze
 - zmniejszanie lub zwiększanie o wartość stałą w zależności od kroku uczenia lub wartości funkcji kosztu
 - gradientowe metody doboru
 - o algorytm Quickprop
 - o algorytm RPROP

c) alternatywne funkcje kosztu [52, 77, 81, 91, 94, 114, 136]

TECHNIKA MOMENTU

Modyfikacja ta ma na celu przyspieszenie procesu uczenia poprzez uodpornienie sieci neuronowej na przypadkowe fluktuacje występujące w danych uczących. Z równania 5 wynika, że zmiana wag w kroku uczenia t zależy już nie tylko od aktualnej wartości gradientu błędu sieci, ale też i od jej wartości poprzednich $t-1$, co wprowadza w uczeniu wag pewną ich bezwładność pozwalającą na to, aby kierunek uczenia nie zmieniał się zgodnie z przypadkowymi wartościami gradientu, ale aby reprezentował pewien trend w danych uczących.

$$w_{ij}^{t+1} = w_{ij}^t + \Delta w_{ij}^t + \alpha \Delta w_{ij}^{t-1} \quad (5)$$

gdzie:

Δw_{ij}^{t-1} zmiana wektora wag w poprzednim kroku uczenia

α – współczynnik momentu

DOBÓR WSPÓŁCZYNNIKÓW UCZENIA

Współczynniki α i η określają szybkość zmian wag synaptycznych. Ich wartości powinny być dobrane w sposób optymalny dla danego problemu. Niestety nie są znane żadne ogólne reguły doboru współczynników uczenia, a jedynie heurystyczne, bazujące na empirycznych obserwacjach:

- a) stałe wartości α i η
- b) stopniowe zmniejszanie α i η w czasie procesu uczenia
- c) zmiana wartości α i η w zależności od aktualnych i poprzednich wartości błędu sieci neuronowej.

Dwie pierwsze metody zakładają jednakowe współczynniki uczenia dla całej sieci neuronowej lub przynajmniej jednego neuronu. Ostatnia z metod pozwala na niezależne przypisywanie wartości współczynników uczenia dla każdej wagi synaptycznej z osobna. Dokonuje się tego na podstawie analizy kierunku zmian funkcji kosztu w poprzedzających krokach uczenia –

procedura aktualizacji wykorzystuje często wartości gradientu funkcji kosztu. Zaletą metod gradientowych jest możliwość zmniejszania, jak też i zwiększania wartości współczynników uczenia w trakcie procesu uczenia w razie zaistnienia takiej potrzeby. Opisana powyżej procedura jest podstawą algorytmów *delta-bar-delta* (DBD) i *extended-delta-bar-delta* (EDBD) stanowiących jedną z najskuteczniejszych modyfikacji propagacji wstecznej.

ALGORYTM QUICKPROP

Jest to zaproponowany przez Fahlmana algorytm heurystyczny [91, 94]. Równanie na zmianę wag jest w ogólnym zarysie identyczne z równaniem 4. Modyfikacji uległ człon Δw_{ij}^t oraz sposób dobierania współczynnika momentu α . Równanie 6 przedstawia wzór dla członu Δw_{ij}^t z równania 5. Element γw_{ij}^t zapobiega utykaniu sieci w minimum lokalnym poprzez utrzymywanie niezerowej wartości gradientu w momencie, kiedy podczas uczenia sieć neuronowa dotrze do płaskiego obszaru („doliny”) funkcji celu.

$$\Delta w_{ij}^t = -\eta \left[\frac{\partial E}{\partial w_{ij}^t} + \gamma w_{ij}^t \right] \quad (6)$$

gdzie:

γ - stała z zakresu (0,1)

Stać γ przyjmuje zwykle małe wartości rzędu 10^{-4} . Dodatkowym atutem algorytmu Quickprop jest adaptacyjny dobór współczynnika momentu α indywidualnie dla każdej wagi. Zależy on zarówno od aktualnej jak też i poprzedniej wartości gradientu funkcji celu (równanie 7).

$$\alpha_{ij}^t = \begin{cases} \alpha_{\max}; (\beta_{ij}^t > \alpha_{\max}) \cup (S_{ij}^t \cdot \Delta w_{ij}^{t-1} \cdot \beta_{ij}^t < 0) \\ \beta_{ij}^t; \text{w innych przypadkach} \end{cases} \quad (7)$$

gdzie:

$$S_{ij}^t = \frac{\delta E}{\delta w_{ij}^t} + \gamma w_{ij}^t$$

$$\beta_{ij}^t = \frac{S_{ij}^t}{S_{ij}^{t-1} - S_{ij}^t}$$

α_{\max} – maksymalna wartość stałej $\alpha = 1,75$

Mimo swojej komplikacji, algorytm Quickprop oceniany jest jako kilka do kilkuset razy szybszy od klasycznej postaci propagacji wstecznej. Istnieją też jego wersje uproszczone dające porównywalne wyniki z uzyskiwanymi przez oryginalną postać.

ALTERNATYWNE FUNKCJE KOSZTU

Funkcja kwadratowa mimo swej prostoty matematycznej jest przyczyną spowolnienia uczenia w terminalnej fazie treningu sieci neuronowej. Poza tym nie istnieje tutaj żaden mechanizm tłumienia błędów występujących w danych uczących. W związku z tym poszukuje się alternatywnych funkcji kosztu posiadających lepsze własności od funkcji kwadratowej.

FUNKCJA KARAYANNISA

Równanie 8 przedstawia definicję funkcji kosztu przedstawioną przez Karayannisa [91, 94].

$$E = \frac{1}{2} \lambda \sum_{i=1}^N (s - s^{\text{wzorzec}})^2 + (1 - \lambda) \sum_{i=1}^N \varphi(s - s^{\text{wzorzec}}) \quad (8)$$

gdzie:

λ – dodatnia stała z zakresu (0;1)

φ - ciągła, dodatnio określona, różniczkowalna funkcja błędu

Równanie 9 przedstawia proponowaną postać funkcji błędu.

$$\varphi(x) = \frac{1}{\beta} \ln[\cosh(\beta \cdot x)] \quad (9)$$

gdzie:

β – dodatnia stała

Zaproponowano również zależność współczynnika od aktualnej wielkości błędu (równanie 10) [91].

$$\lambda = \exp \left(- \frac{n}{\left(\sum_{i=1}^N (s - s^{\text{wzorzec}})^2 \right)^n} \right) \quad (10)$$

FUNKCJA LOGARYTMICZNA

W celu tłumienia dużych wartości błędów zaproponowano funkcję kosztu w postaci funkcji logarytmicznej przedstawionej równaniem 11 [94].

$$E = \sum_{i=1}^N \log \left[1 + \frac{1}{2} (s - s^{\text{wzorzec}})^2 \right] \quad (11)$$

FUNKCJA HAMPPELA

Funkcja kosztu, która wykorzystuje funkcję Hampela przedstawiana jest jako jedno z lepszych rozwiązań proponowanych w celu tłumienia przypadkowych błędów w danych.

$$E = \sum_{i=1}^N F_i^H \quad (12)$$

gdzie:

F_i^H - funkcja Hampela

Funkcja Hampela definiowana jest jako złożenie trzech funkcji: kwadratowej, tangensa hiperbolicznego i funkcji stałej [94]. Jej pochodną przedstawiono w równaniu 13.

$$\frac{\partial F_i^H}{\partial d_i} = \begin{cases} d_i & \text{dla } |d_i| \leq a \\ c_1 \cdot \tanh[c_2 \cdot (b - |d_i|)] \cdot \text{sgn}(d_i) & \text{dla } a \leq |d_i| \leq b \\ 0 & \text{dla } |d_i| > b \end{cases} \quad (13)$$

gdzie:

$d_i = s - s^{\text{wzorzec}}$ - aktualny błąs w i-tej iteracji

c_1 i c_2 – pewne stałe

a i b – zakresy funkcji Hampela

INNE METODY UCZENIA Z NAUCZYCIELEM

Oprócz propagacji wstecznej znane są również i inne metody uczenia z nauczycielem, wśród których można wymienić:

a) metody gradientowe

- gradienty sprzężone (*conjugated gradient* – CG)
- skalowane gradienty sprzężone (*scaled conjugated gradient* – SCG)

b) metody bezgradientowe

- symulowane wyżarzanie (*simulated annealing* – SA)
- optymalizacja genetyczna (*genetic algorithm* – GA) [52, 81, 91, 94]

Wymienione wyżej algorytmy uczenia nie wyczerpują wszystkich możliwości uczenia z nauczycielem. Istnieje wiele systemów hybrydowych łączących różne algorytmy w celu uzyskania lepszej sprawności systemu. Przykładem

może być łączenie systemów opartych na optymalizacji genetycznej z propagacją wsteczną [51]. W tym przypadku wykorzystuje się zdolność algorytmu genetycznego do przeszukiwania przestrzeni rozwiązań w sposób globalny i precyzję lokalnego dopasowywania się sieci neuronowej opartej na propagacji wstecznej.

TRUDNOŚCI ZWIĄZANE Z UCZENIEM Z NAUCZYCIELEM

Mimo tak szerokiej gamy algorytmów uczących, uczenie z nauczycielem ciągle jeszcze napotyka na wiele problemów, tj.

- a) długi czas uczenia
- b) brak optymalnych reguł doboru współczynników uczenia
- c) brak optymalnych kryteriów zatrzymania procesu uczenia
- d) niemożność analizy modelu neuronowego (model „czarnej skrzynki”)

Spośród wymienionych, brak optymalnych kryteriów zatrzymania procesu uczenia należy uznać za szczególnie trudne zagadnienie, zwłaszcza w przypadku, gdy nie dysponujemy wystarczającą ilością danych, aby użyć osobnego zbioru do weryfikacji procesu uczenia. Zbyt długie uczenie sieci neuronowej może powodować jej przeuczenie i brak zdolności generalizacji. Zbyt krótkie uczenie nie pozwoli zaś na otrzymanie reprezentatywnego modelu badanego zagadnienia. W związku z powyższym, problem ten jest rozwiązywany na drodze eksperymentów numerycznych.

Jedną z największych trudności związanych z modelowaniem neuronowym jest niemożność analizy modelu neuronowego. Z tego powodu sieci neuronowe typu MLP są niekiedy określane mianem „czarnej skrzynki”. Klasyczny model matematyczny można objaśnić obliczając znaczenie poszczególnych jego składowych, ponieważ znana jest jego analityczna postać. W przypadku sieci neuronowych nie jest to możliwe – działają one w sposób heurystyczny, co w tym przypadku oznacza, że tworzą model w postaci niejawnej. Aby zapewnić wiarygodność modelu neuronowego przeprowadza się wyczerpującą jego weryfikację, której przykładem może być metoda k-krotnego wzajemnego sprawdzania. Trwają również prace nad

pozbyciem się problemu „czarnej skrzynki” w sieciach MLP. Przykładem może być sieć MLP2LR [91].

WERYFIKACJA SIECI NEURONOWYCH METODĄ K-KROTNEGO WZAJEMNEGO SPRAWDZANIA

Metoda k-krotnego wzajemnego sprawdzania polega na k-krotnym (najczęściej $k=10$) uczeniu tej samej architektury na różnych parach zbiorów uczący-testowy. Pary te powstają przez k-krotny podział bazowego zbioru danych na zbiór uczący i testowy. Liczby rekordów zbioru testowego i bazowego obliczane są wg równań 14 i 15. Każda procedura podziału prowadzona jest w taki sposób, że rekordy danych w zbiorach testowych są za każdym razem inne. Pozwala to na ocenę błędu generalizacji danego systemu neuronowego na całym, bazowym zbiorze danych.

$$L_U = \left(1 - \frac{1}{k}\right) \cdot L_B \quad (14)$$

$$L_T = \frac{1}{k} \cdot L_B \quad (15)$$

gdzie:

L_B – liczba rekordów zbioru bazowego

L_T – liczba rekordów zbioru testowego

L_U – liczba rekordów zbioru uczącego

k – ilość podziałów zbioru bazowego

1.2. SIECI UCZONE BEZ NAUCZYCIELA

Jeżeli nie są znane stany badanego układu, można użyć sieci uczonych bez nadzoru do znalezienia pewnych związków pomiędzy wektorami parametrów układu.

Tego typu sieci bazują na swojej zdolności do samoorganizacji, co pozwala na wykonywanie następujących zadań:

- a) znajdowanie podobieństwa i określanie jego stopnia w stosunku do typowych lub uśrednionych wektorów parametrów układu
- b) klasyfikacja z podziałem na znaną lub też nieznaną liczbę klas
- c) określanie prototypów na zasadzie pamięci asocjacyjnej
- d) odwzorowanie cech istotnych
- e) kodowanie i kompresja danych

Do algorytmów uczących bez nadzoru należy zaliczyć reguły:

- a) Hebba
- b) Oji
- c) Grossberga
- d) Carpentera i Grossberga (sieci ART)
- e) Kohonena [94]

REGUŁA HEBBA

Jest to najprostsza i historycznie najstarsza reguła uczenia stosowana zarówno do uczenia z nadzorem jak i bez nadzoru. Zmiana wagi jest proporcjonalna do iloczynu sygnałów wejściowego i wyjściowego komórek połączonych przez daną wagę. Równanie 16 przedstawia zmianę wagi w wersji uczenia bez nadzoru [52, 94].

$$\Delta w_{ij}^t = \eta \cdot x_i^t \cdot y_j^t \quad (16)$$

gdzie:

y_j^t i x_i^t - sygnały wyjściowe komórek połączonych daną wagą

REGUŁA OJI

Ponieważ reguła Hebba może prowadzić do niekontrolowanego wzrostu wartości wag Oja wprowadził modyfikację reguły Hebba, która pozwala ograniczyć wartości wag (równanie 17) [52]

$$\Delta w_{ij}^t = \eta \cdot y_j^t (x_i^t - y_j^t \cdot w_{ij}^t) \quad (17)$$

REGUŁA GROSSBERGA

Grossberg zaproponował klasyczne uczenie konkurencyjne typu „zwycięzca bierze wszystko” (*Winner Takes All - WTA*) [52]. Uczenie to polega na tym, że każdy neuron warstwy ukrytej konkuruje o wektor wejściowy starając się do niego jak najbardziej dopasować. Operacja ta jest realizowana poprzez porównywanie wektora wejściowego z wektorem wag danej komórki warstwy ukrytej będącą równocześnie jednostką wyjściową. Za zwycięzcę uznaje się i-tą komórkę, której wektor wag jest najbliższy wektora wejściowego. Równanie 18 przedstawia matematyczny zapis tego procesu.

$$|W_i^* - X| \leq |W_i - X| \quad \text{dla każdego } i \quad (18)$$

gdzie:

W_i - wektor wag dowolnej komórki wyjściowej

W_i^* - wektor wag komórki-zwycięzcy

X - wektor wejściowy (parametrów układu)

i - liczba komórek wyjściowych

Równanie 19 przedstawia zmianę wagi dla komórki-zwycięzcy.

$$\Delta w_{ij} = \eta (x_j - w_{ij}) \quad (19)$$

gdzie:

w_{ij} - waga pomiędzy i-tym wyjściem i j-tym wejściem

x_j - j-ta składowa wektora parametrów układu X

REGUŁA KOHONENA

Sieć Kohonena podlega również uczeniu konkurencyjnemu. W odróżnieniu od reguły Grossberga pozwala na uczenie nie tylko komórek-zwycięzcy ale też i komórek z nią sąsiadujących na podstawie określenia tzw. funkcji sąsiedztwa wyrażanej w postaci funkcji gaussowskiej (równanie 20).

$$\Lambda(i,i^*) = \exp(-|W_i - W_i^*|^2 / 2\sigma^2) \quad (20)$$

gdzie:

$\Lambda(i,i^*)$ - funkcja sąsiedztwa

σ - szerokość funkcji sąsiedztwa

Ostatecznie reguła uczenia Kohonena dla konkretnej wagi pomiędzy i-tym wyjściem a j-tym wejściem w_{ij} wygląda następująco:

$$\Delta w_{ij} = \eta \Lambda(i,i^*) (x_j - w_{ij}) \quad (21)$$

gdzie:

w_{ij} - waga pomiędzy i-tym wyjściem i j-tym wejściem

x_j - j-ta składowa wektora parametrów układu X

Z równania 21 wynika, iż algorytm Kohonena jest tzw. algorytmem miękkiej selekcji typu „zwycięzca bierze większość” (*Winner Takes Most - WTM*).

Sieci Kohonena są używane do odwzorowywania cech istotnych tworząc tzw. mapy topologiczne. Mogą też być wykorzystywane jako składnik systemów hybrydowych, gdzie jedna z warstw jest uczona bez a druga pod nadzorem. Model takiej sieci to np. sieć z kontrapropagacją (*counterpropagation - CP*).

SIECI ART

Nazwa sieci pochodzi od *Adaptive Resonance Theory* – teorii rezonansu adaptacyjnego [52, 114]. Są to sieci przeznaczone głównie do klasyfikacji i rozpoznawania obrazów. Ich działanie polega na uczeniu konkurencyjnym. Sieć posiada w swej strukturze dwie warstwy – wyjściową i wejściową. Istotną różnicą jest tutaj obecność wag prowadzących od warstwy wyjściowej do wejściowej, co pozwala na wprowadzenie sieci w stan rezonansu, który odpowiada za prawidłowe nauczenie sieci. Istnieją podtypy ART1 i ART2 różniące się zakresem stosowalności: sieci ART1 stosuje się dla danych binarnych, zaś sieci ART2 dla danych ciągłych.

1.3. SIECI REKURENCYJNE

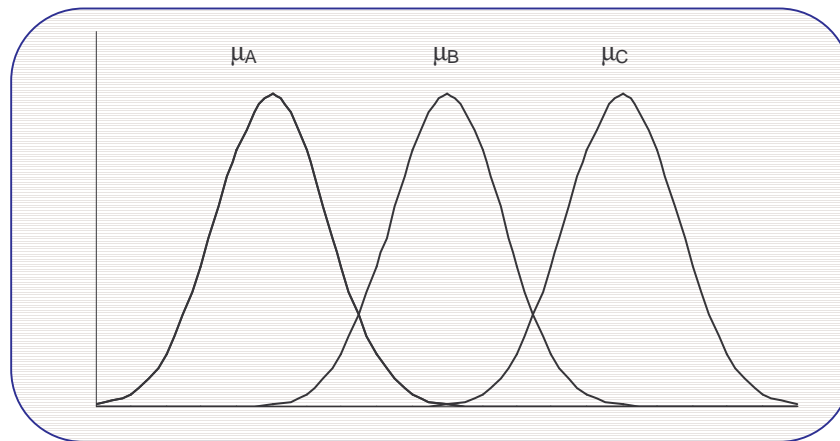
Odrębną grupą sieci neuronowych uczonych zarówno pod nadzorem jak i bez nadzoru są tzw. sieci rekurencyjne. Ich odmienność wynika z istnienia sprzężenia zwrotnego pomiędzy wyjściem i wejściem. Sygnał roboczy sieci przenoszony jest więc w dwóch kierunkach aż do ustalenia się dynamicznej równowagi. Pozwala to na traktowanie sieci neuronowej jako układu dynamicznego, w którym zmiany stanu jednego neuronu powodują zmianę stanów wszystkich neuronów w sieci. Zastosowania sieci rekurencyjnych koncentrują się w obrębie zagadnień związanych z klasyfikacją, rozpoznawaniem wzorców, identyfikacją nieliniowych układów dynamicznych.

Do sieci rekurencyjnych zaliczyć można:

- a) autoasocjacyjną sieć Hopfielda
- b) maszynę Boltzmanna
- c) sieć Hamminga
- d) sieć BAM (*Bidirectional Associative Memory*)
- e) sieci Elmana
- f) sieci uczone za pomocą rekurencyjnej propagacji wstecznej [94]

1.4. SIECI ROZMYTE

Dotychczas omawiano algorytmy uczenia na przykładzie systemów opartych na logice boolowskiej, gdzie podstawą obliczeń były klasyczne liczby (rzeczywiste, naturalne itd.). Jednak ostatnio tzw. logika rozmyta (*fuzzy logic*), stanowiąca odrębny system logiczny, staje się coraz częściej używanym narzędziem z powodu swoich specyficznych właściwości, a także precyzji opartych na niej systemów obliczeniowych. Logika rozmyta jest systemem logicznym, w którym liczby nie są ściśle określone, jak w logice klasycznej, ale wyrażane są jako pewne prawdopodobieństwa, nazywane funkcjami przynależności, że wartość liczby zawiera się w danym zakresie [94, 100, 109, 131]. Funkcje przynależności mogą przyjmować różne postacie, wśród których jedną z częściej używanych jest funkcja gaussowska (Rycina 4, równanie 22).



Rycina 4. Gaussowskie funkcje przynależności.

$$\mu(s) = \exp \left[- \left(\frac{x_i^t - \bar{x}_i^t}{\sigma} \right)^2 \right] \quad (22)$$

gdzie:

$\mu(s)$ - funkcja przynależności

\bar{x}_i^t - centrum funkcji przynależności

x_i^t - aktualna wartość i-tej składowej wektora parametrów (X)

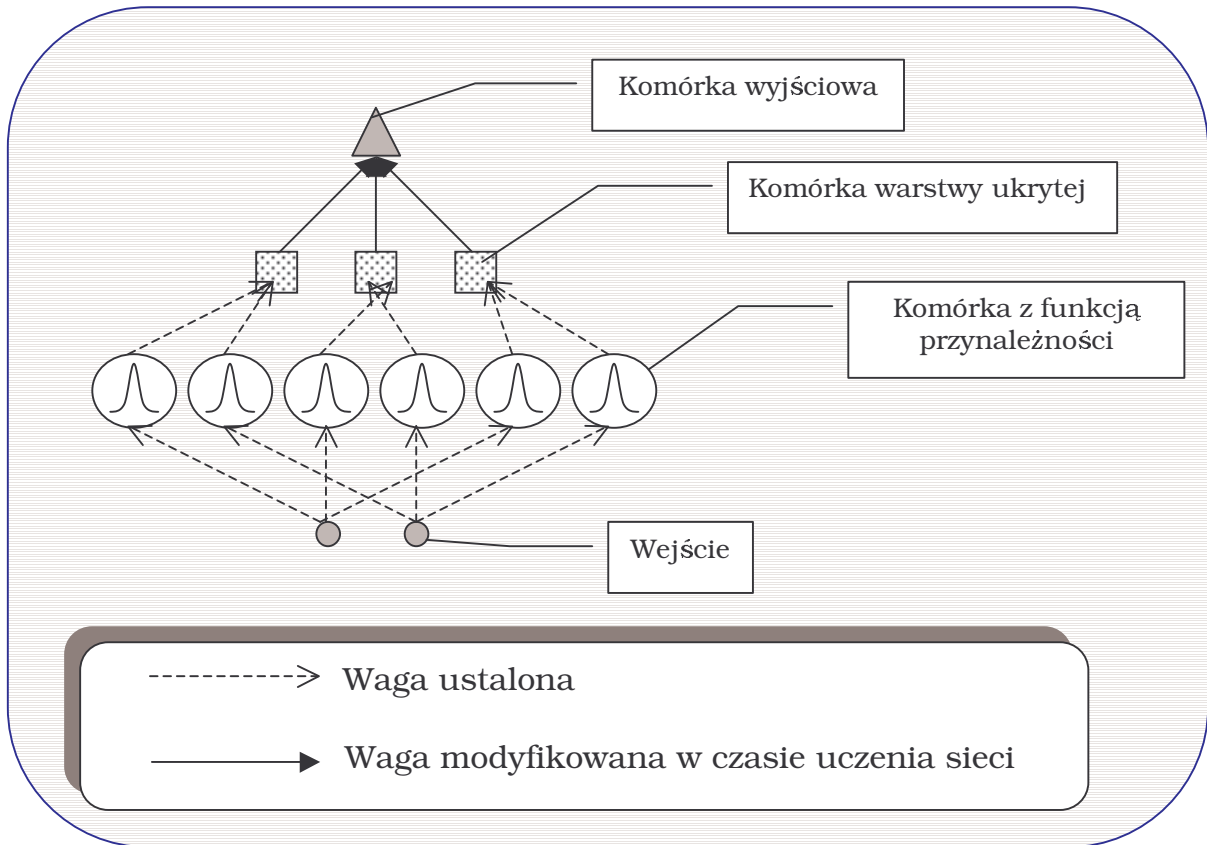
σ - szerokość funkcji przynależności

Twórcą logiki rozmytej jest Lotfi Zadeh, który w swojej pracy określił wstępnie możliwość użycia takiego systemu do obliczeń [135]. Problemy matematyczne związane z używaniem liczb rozmytych zostały ostatecznie pokonane. Mimo to charakterystyczny jest znaczny stopień komplikacji systemów rozmytych. Zaletą liczb rozmytych jest możliwość prowadzenia obliczeń nie tylko w oparciu o wartości liczbowe, ale również wielkości jakościowe. Pozwala to na implementowanie w systemie komputerowym wiedzy eksperta w sposób lingwistyczny tj. oparty na pojęciach typu: „mało”, „średnio”, „dużo”. Możliwość pracy na tak nieprecyzyjnie określonych pojęciach z równoczesną zdolnością do otrzymywania dokładnych rezultatów była do tej pory zarezerwowana tylko dla biologicznych układów nerwowych. Co więcej, systemy rozmyte mają zazwyczaj charakter regułowy, łatwy do interpretacji przez człowieka [94, 100, 109, 131]. Równanie 23 przedstawia zapis matematyczny i interpretację logiczną hipotetycznej reguły systemu rozmytego.

Jeżeli $a \in A$ i $b \in B$ to $z \in Z$ (23)

Jeżeli a jest *Małe* i b jest *Duże* to z jest *Małe*

W trakcie badań nad systemami rozmytymi powstał problem doboru bazy reguł tak, aby można było uzyskać reprezentatywny model badanego zagadnienia. Z pomocą przyszły sieci neuronowe, które okazały się wygodnym narzędziem doboru reguł na podstawie danych empirycznych. Powstały niejako hybrydowe systemy neuronowo-rozmyte, nazywane również sieciami rozmytymi (*neuro-fuzzy systems*). Sieci te, w swej najprostszej wersji modelu Mamdaniego posiadają jedną warstwę ukrytą i jedno liniowe wyjście – tzw. model MISO (*Multiple Input Single Output*) [109]. Strukturę modelu Mamdaniego sieci rozmytej przedstawiono na rycinie 5.



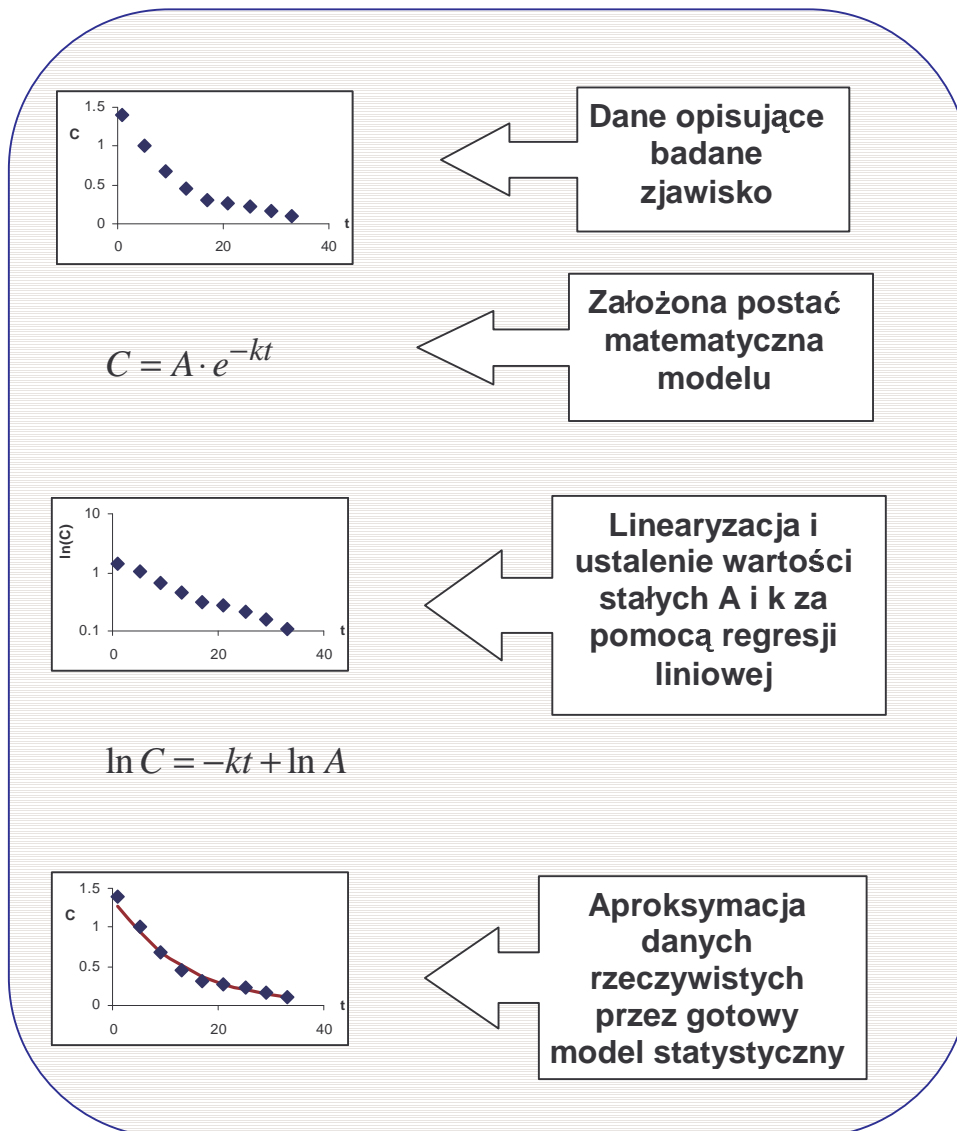
Rycina 5. Struktura modelu Mamdaniego sieci rozmytej.

Każda komórka warstwy ukrytej posiada dodatkowe podjednostki wykonujące operacje logiczne za pomocą funkcji przynależności. Liczba podjednostek jest zawsze równa liczbie wejść sieci neuronowej. Model Mamdaniego sieci rozmytych uczony jest pod nadzorem z wykorzystaniem gradientowych metod optymalizacji z propagacją wsteczną na czele. Uczeniu podlegają zarówno wagi warstwy wyjściowej jak też i parametry funkcji przynależności: centra i szerokości. Komórki warstwy ukrytej są interpretowane jako całe reguły systemu rozmytego. Dzięki tej interpretacji sieć rozmyta nie jest już „czarną skrzynką”, co było uznawane za podstawową wadę sieci MLP opartych na logice klasycznej. Można więc niejako „czytać” bezpośrednio z nauczonej sieci rozmytej reguły, które stworzyła do opisu badanego zagadnienia. Nauczona sieć może być też interpretowana w kategoriach lingwistycznych („dużo”, „mało”...), co daje intuicyjny pogląd na charakter modelu [94, 100, 109, 131].

Znane są też i inne modele sieci rozmytych jak np. TSK, uczone bez nadzoru, czy też będące hybrydami z algorytmami genetycznymi oraz sieciami opartymi na logice klasycznej [109].

2. MODELOWANIE KLASYCZNE A NEURONOWE

Zakładając, iż badany układ jest opisywany przez dwa wektory: parametrów (X) i stanów układu (S) można określić proces modelowania jako próbę znalezienia związku korelacyjnego pomiędzy wektorami X i S. Klasyczne modelowanie polega na identyfikacji analitycznej postaci funkcji wiążącej wektor parametrów układu z wektorem jego stanów.



Rycina 6. Tok postępowania w czasie konstrukcji przykładowego modelu statystycznego.

Dokonuje się tego znajdując funkcję $f(X)$ przedstawioną w równaniu 24.

$$S = f(X) \quad (24)$$

Funkcję $f(X)$ znajduje się w jej jawnej postaci, która zależy od charakteru opisywanego zagadnienia i może przyjmować różne postacie: funkcji liniowej, logarytmicznej, wykładniczej, wielomianów różnych stopni itd. O ile więc dostępne są jakiegokolwiek przesłanki co do postaci funkcji $f(X)$, jej różniczkowalności i monotoniczności, to można ją łatwo znaleźć w konwencjonalny sposób. Wyrażenie $f(X)$ w postaci funkcji liniowej pozwala na szczególnie łatwe znalezienie parametrów modelu z uwagi na dobrze opracowane metody dzisiejszej algebry liniowej. Jeżeli jednak któryś z wymienionych wcześniej warunków nie zostaje spełniony, poszukiwanie modelu metodami klasycznymi staje się uciążliwe i niewygodne, a w wypadku kiedy analizowanego systemu nie można w ogóle opisać za pomocą jakiegokolwiek funkcji - wręcz niemożliwe. Szczególnie trudne są przypadki dużych rozmiarów wektorów X i S - wielowymiarowa analiza jest z matematycznego punktu widzenia niezwykle skomplikowana. Problemy o nieliniowej charakterystyce sprawiają również duże trudności matematyczne. W takich przypadkach byłoby korzystnie, żeby samo narzędzie modelujące przejmowało jak najwięcej obciążenia związanego z tworzeniem modelu - w szczególności zaś dobieranie postaci modelu w sposób automatyczny byłoby istotnym ułatwieniem. Automatyzm uczenia sztucznych sieci neuronowych jest cechą predestynującą je do modelowania w tak trudnych przypadkach. W oparciu o dane empiryczne są one w stanie same identyfikować postać modelu najlepiej opisującego dane zagadnienie. Istotne jest również, że jedna architektura sieci jest w stanie modelować całkowicie odmienne zagadnienia. Wynika z tego, że mimo iż wybór architektury często decyduje o jakości modelu neuronowego, to nie jest on jednak tak krytyczny w swoim znaczeniu jak wybór postaci funkcji opisującej klasyczny model statystyczny. Wektor parametrów układu (X) stanowi wejścia sieci neuronowej, a wektor stanów (S) jej oczekiwane wyjścia - używane do uczenia sieci pod nadzorem np. algorytmem propagacji wstecznej (Rycina 3).

Należy zaznaczyć, że tak otrzymany model stanowi zależność wektora S od X w postaci niejawnej (z wyjątkiem np. sieci rozmytych) tzn., że na podstawie sieci neuronowej nie da się bezpośrednio przedstawić jego analitycznej postaci tj. ściślej, matematycznej formuły wiążącej S z X . Z tego powodu modelowanie za pomocą sieci neuronowych nazywa się modelowaniem heurystycznym, a modele heurystycznymi lub empirycznymi. Opisany problem tzw. „czarnej skrzynki” utrudnia analizę neuronową ponieważ o charakterze $f(X)$ można wnioskować tylko pośrednio na podstawie obserwacji zachowania systemu w reakcji na konkretne wzorce wejściowe (wektory X). Otrzymanie niejawnej postaci skomplikowanego, nieliniowego modelu wiąże się zwykle ze znacznie mniejszym kosztem obliczeniowym. Dodatkowo, jeżeli istnieje możliwość weryfikacji modelu neuronowego, to jego stosowanie jest zazwyczaj tak samo skuteczne jak klasycznego. Wszystko to przemawia za stosowaniem takiego podejścia, które może być zrealizowane w postaci sztucznej sieci neuronowej.

Drugi etap budowania modelu neuronowego, tj. etap testowania bazuje na charakterystycznej właściwości sieci neuronowych jaką jest zdolność do uogólniania (tj. generalizacji). Warunkiem prawidłowej generalizacji jest skutecznie przeprowadzona faza treningu. Uogólnianie może mieć charakter zarówno interpolacji jak też i ekstrapolacji, z tym że ekstrapolacja jest zazwyczaj zadaniem trudniejszym do wykonania. Właściwa generalizacja to zwykle podstawowy cel, którego osiągnięcie może gwarantować poprawność modelu.

Uniwersalność zastosowań sieci neuronowych stanowi zasadniczą ich cechę. Jeden system może być użyty do modelowania bardzo różnych układów. Potwierdza to różnorodność dziedzin, w których zastosowano sieci typu perceptronu uczone algorytmem propagacji wstecznej (MLP).

Cechą sieci neuronowych predestynującą je do zastosowań medycznych jest odporność sieci na szum w danych. Jedną z technik poprawiania generalizacji jest wtrącanie szumu o małej amplitudzie do wzorców uczących. Umożliwia to korzystanie z danych niepewnych, często obarczonych błędami np. losowymi, systematycznymi. Mimo to możliwe jest stworzenie maksymalnie reprezentatywnego modelu opisywanego zjawiska.

Powinny być jednak spełnione pewne statystyczne założenia co do charakteru szumu, czy niepewności danych, ale mogą być one znacznie mniej rygorystyczne niż dla modelowania klasycznego.

3. ZASTOSOWANIA SZTUCZNYCH SIECI NEURONOWYCH W TECHNOLOGII POSTACI LEKU

Sieci neuronowe jako narzędzia analizy danych znalazły już zastosowanie w wielu dziedzinach życia. Ich zdolność do rozpoznawania i klasyfikacji wzorców oraz do przewidywania w czasie jest wykorzystywana wszędzie tam, gdzie klasyczne metody zawodzą z uwagi na swoją nieefektywność. Sieci neuronowe są stosowane w szerszym zakresie w:

- ekonomii - prognoza szeregów czasowych, np. kursów walut, akcji itp. [91], ocena ryzyka kredytowego i inwestycyjnego na podstawie standardowych danych o klientach banku [4]
- technice - filtrowanie sygnału od szumu, prognozy obciążeń sieci energetycznej, sterowanie ruchem lotniczym i metra [91, 94]
- informatyce - rozpoznawanie wzorców (*pattern recognition*): rozpoznawanie mowy, pisma (w tym ręcznego), wydobywanie wiedzy z baz danych (*data-mining*), systemy ekspertowe [52, 81, 89, 91, 94, 114, 136]
- chemii - identyfikacja związków chemicznych na podstawie ich widm IR, NMR [7], modelowanie przestrzenne [112], ocena interakcji substancja-enzym [91]
- medycynie - rozpoznawanie wzorców wykorzystywane w diagnostyce: EEG, EKG, NMR, mammografia itp. [28, 91], modelowanie właściwości ludzkiego umysłu: pamięci, zaburzeń psychicznych [91]

W naukach farmaceutycznych odnotowuje się stopniowy wzrost zainteresowania alternatywnymi narzędziami modelowania [46, 84, 86].

Podjęto próby zastosowania sztucznych sieci neuronowych m.in. do: modelowania farmakokinetyczno-farmakodynamicznego (PK/PD) [17, 18, 19, 125, 126], analizy danych w farmakokinetyce populacyjnej [27], poszukiwania nowych substancji leczniczych [91], klasyfikacji farmakologicznej [20, 104], przewidywania parametrów fizykochemicznych [61, 120] i farmakokinetycznych substancji leczniczych [47] na podstawie ich budowy molekularnej, korelacji badań *in vitro* i *in vivo* [33], allometrii [58], analiz farmakoeconomicznych i epidemiologicznych [101, 102, 103] oraz optymalizacji składu i technologii sporządzania postaci leku [5, 13, 14, 15, 16, 22, 44, 67, 83, 85, 87, 88, 107, 113, 115, 116, 117, 124].

W zakresie prac dotyczących technologii postaci leku dominują dwa kierunki badań, tj. optymalizacja składu postaci leku oraz sposobu ich sporządzania. Jak wynika z piśmiennictwa większość prac koncentruje się wokół zagadnień związanych ze sporządzaniem stałych, doustnych postaci leku [13, 14, 15, 16, 22, 83, 85, 87, 107, 115, 124]. Pozostałe prace dotyczą m.in. emulsji [44], mikroemulsji [5, 88], żeli [116] oraz transdermalnych systemów terapeutycznych [67].

3.1. STAŁE POSTACIE LEKU

Hussain i wsp. [60] jako jedni z pierwszych przedstawili koncepcję użycia neuronowego systemu ekspertowego do przewidywania szybkości uwalniania maleinianu chlorfeniraminu z kapsułek. Substancja lecznicza rozproszona była w hydrofilowej matrycy, którą tworzyły pochodne celulozy: karboksymetyloceluloza, hydroksypropyloceluloza, hydroksypropylometyloceluloza lub hydroksyetyloceluloza. W celu pozyskania danych uczących przygotowano 15 formułacji kapsułek. Jako zmienne wejściowe wybrano jakościowy i ilościowy skład matrycy w postaci czteroelementowego wektora, którego każdy element wyrażał zawartość jednego ze składników matrycy. Wytrenowana sieć miała za zadanie przewidzieć wykładnik równania kinetycznego opisującego proces uwalniania (N) i czas, po którym uwolniło

się 50% substancji leczniczej. Sprawność modelu neuronowego została porównana z klasyczną metodą RSM (*response surface methodology*). Wykazano wyższość modelowania neuronowego nad metodą RSM, a błędy generalizacji popełnione przez sieć neuronową zaklasyfikowano jako dopuszczalne.

W kolejnych badaniach [59] zastosowano sieć neuronową do przewidywania profilu uwalniania substancji leczniczej z 17 formuacji tabletek, których hydroksypropyloceluloza tworzyła matrycę. W przypadku 15 formuacji błąd RMSE osiągnął wartość poniżej zakładanego maksimum $RMSE = 10$. Autorzy nazwali sieć neuronową systemem komputerowego wspomaganie projektowania formuacji (*Computer Aided Formulation Design*) podkreślając zdolność sieci neuronowych do wielowymiarowej analizy danych i znajdowania skomplikowanych zależności opisujących postać leku.

Takahara i wsp. [115] użyli sieci neuronowej do optymalizacji składu tabletek o przedłużonym działaniu zawierających jako substancję leczniczą trapidil. Do badań sporządzono 18 formuacji tabletek z trapidilem, które stały się podstawą zbioru uczącego oraz 9 dodatkowych formuacji jako zbiór testujący poprawność nauczania sieci neuronowej. Wektor wejściowy sieci neuronowej był 3-elementowy i obejmował wejścia opisujące zawartość hydroksypropylometylocelulozy, mikrokryształicznej celulozy w tabletkce oraz siłę nacisku stempla tabletkarki. Wektor wyjściowy zawierał 4 elementy, które opisywały rząd reakcji i stałą szybkości uwalniania trapidilu z tabletek do płynów o pH 1,2 i 6,8. Sieci neuronowe uczone były za pomocą rozszerzonego filtra Kalmana. Zadaniem wytrenowanej sieci neuronowej było znalezienie optymalnej formuacji, z której uwalnianie substancji leczniczej byłoby zgodne z zakładaną charakterystyką profilu uwalniania. Autorzy wykazali dobrą zgodność przewidywanego z obserwowanym profilem uwalniania substancji leczniczej z formuacji wskazanej przez sieć neuronową.

Chen i wsp. [22] zastosowali sztuczną sieć neuronową do zaprojektowania formuacji tabletek o kontrolowanym uwalnianiu substancji leczniczej. Sieć neuronowa posiadała 10 wejść opisujących jakościowy i ilościowy skład tabletek, ich twardość, zawartość wilgoci, a także wielkość

cząstek granulatu użytego do tabletkowania. Sporządzono i przebadano 22 formułacje tabletek, w celu utworzenia bazy danych. Sieć neuronowa miała za zadanie przewidzieć ilość uwolnionej substancji leczniczej po określonym czasie rozpuszczania. Optymalizacja składu tabletek przeprowadzana była na podstawie zadanego przebiegu zmian stężeń leku modelowego i obejmowała 4 etapy:

1. symulacja 22 krzywych opisujących przebieg zmian stężeń substancji leczniczej w osoczu przy użyciu systemu statystycznego Stella™ w oparciu o dane doświadczalne profili uwalniania substancji leczniczej *in vitro* z 22 formułacji tabletek,
2. określenie optymalnego dla substancji leczniczej, przebiegu zmian stężenia w osoczu na podstawie ww. danych,
3. obliczenie metodą dekonwolucji optymalnego profilu uwalniania substancji leczniczej *in vitro* z tabletek na podstawie drugiego etapu
4. ustalenie składu tabletek przez wytrenowaną sieć neuronową

Na podstawie danych otrzymanych przy użyciu sieci neuronowej wytypowano 4 formułacje jako najlepsze. Spośród nich w przypadku 3 formułacji uzyskano zadowalającą zbieżność przewidywań systemu neuronowego z danymi doświadczalnymi. Sieć neuronowa została użyta jako część hybrydowego systemu neuronowo-statystycznego z powodzeniem służącego optymalizacji składu postaci leku.

Sieci typu perceptronu z jedną warstwą ukrytą uczone metodą gradientów sprzężonych posłużyły do przewidywania szybkości uwalniania teofiliny z 6 formułacji peletek sporządzonych metodą ekstruzji i sferonizacji [98]. Peletki zawierały jednakową ilość teofiliny w stosunku do matryc o różnym składzie. Kryterium oceny sprawności modelu neuronowego stanowiły wartości tzw. współczynnika podobieństwa (f_2), wyliczone na podstawie badań szybkości rozpuszczania (wg USP 23). Wyniki badań wykazały zgodność przebiegu krzywych będących efektem badań eksperymentalnych, jak i przewidywanej przez sieć neuronową, o czym świadczą wartości współczynnika $f_2 > 50$, tj. w granicach od 63,2 do 86,8.

Prace Goha i wsp. [48] dotyczyły również możliwości przewidywania uwalniania teofiliny z peletek. Jednak użyto w tym przypadku bardziej skomplikowanych sieci neuronowych, tzw. sieci Jordana-Elmana z warstwami kontekstu. Dodatkowo zastosowano zmodyfikowaną technikę *AdaBoost*, która polega na użyciu zespołu specyficznie trenowanych sieci neuronowych do przewidywania procesu uwalniania substancji leczniczej z danej formulacji. System przypomina komitet ekspertów, z tym że kolejne sieci uczone są na zmodyfikowanych zbiorach uczących. Analizie poddano dane doświadczalne zawarte w omawianej powyżej pracy Peha [98]. Mimo zastosowania zmodyfikowanego systemu sieci neuronowych uzyskano wyniki o większym rozrzucie wartości współczynnika f_2 tj. od 60,5 do 94,6. Wykonano dodatkowo 6 formulacji peletek o jednakowym składzie matrycy, różniących się zawartością teofiliny. Skład matrycy został opracowany we wcześniejszej pracy [98]. Wyniki generalizacji wahały się w zakresie wartości współczynnika f_2 od 70,9 do 92. Sprawność sieci neuronowej została porównana z klasyczną wielowymiarową regresją. Wykazano wyższość systemu neuronowego nad statystycznym.

Bourquin i wsp [14, 15, 16] rozpatrywali możliwość użycia sztucznych sieci neuronowych do optymalizacji składu i technologii sporządzania tabletek produkowanych metodą tabletkowania bezpośredniego. Jako zmienne wejściowe modelu neuronowego brano pod uwagę takie parametry jak: zawartość substancji pomocniczych tj. krzemionki, karboksymetylocelulozy, stearynianu magnezu oraz mikrokrystalicznej celulozy, a także siłę nacisku stempla tabletkarki oraz czas kompresji. Zmienne wyjściowe stanowiły: siła wypchnięcia tabletek z matrycy, ich parametry mechaniczne: twardość i ścieralność oraz szybkość uwalniania substancji leczniczej z tabletek. Zbiór uczący liczył 205 formulacji. Do modelowania neuronowego użyto specyficznej architektury sieci neuronowej (*generalised feed forward multilayer perceptron network*), posiadającej bezpośrednie połączenia pomiędzy warstwą wejściową a wyjściową. Porównano modelowanie neuronowe z klasyczną metodą RSM. Stwierdzono, że w przypadku badań, które nie zostały zaplanowane zgodnie z zasadami statystycznego planowania eksperymentów sztuczne sieci neuronowe są lepszymi narzędziami niż

klasyczna metoda RSM. Zidentyfikowano nieliniowe zależności pomiędzy zmiennymi wyjściowymi i wejściowymi, a ponadto wykazano skuteczną identyfikację tych zależności przez sieci neuronowe. Autorzy podkreślają kluczową rolę organizacji danych uczących. Stwierdzono, iż obecność rekordów znacznie odbiegających od reszty danych (*outliers*) może, w przypadku małej ilości danych, powodować złe uwarunkowanie modelu neuronowego, a co za tym idzie, również gorsze jego własności predykcyjne w stosunku do metod statystycznych. Zalecane jest więc wcześniejsze statystyczne opracowanie danych uczących, co znajduje również potwierdzenie w pracach o innym charakterze [81, 91]. Zastosowanie sztucznych sieci neuronowych pozwoliło na zidentyfikowanie zawartości krzemionki i stearynianu magnezu jako kluczowych czynników wpływających na siłę niezbędną do wypchnięcia tabletek z matrycy.

3.2. EMULSJE

Opracowano neuronowy model, za pomocą którego określono wpływ składu emulsji W/O na jej trwałość [44]. Zmiany reologicznego parametru ($tn\delta$) posłużyły do określenia trwałości danej formułacji. Model neuronowy porównano z modelami liniowymi i wielomianowymi. Zastosowano 3 zmienne wejściowe: zawartość fazy olejowej, wodnej i silikonowego tenzydu oraz 9 zmiennych wyjściowych, które stanowiły wartości reologicznego parametru ($tn\delta$) po różnym okresie przechowywania emulsji, w przedziałach czasowych od 1 do 12 tygodni. Wykazano przewagę modelu neuronowego nad klasycznymi, gdyż sieci neuronowe zdołały prawidłowo przewidzieć zmiany parametru reologicznego ($tn\delta$) we wszystkich przedziałach czasowych. Wskazano na dobre zdolności generalizacji, które mogą być wykorzystane w projektowaniu formułacji emulsji W/O. Generalizacja została potwierdzona na przykładzie dwóch formułacji emulsji nieznanymi sieci podczas treningu.

W toku dalszych badań [45] autorzy stworzyli hybrydowy system neuronowo-statystyczny w celu wyjaśnienia wpływu składu emulsji na lepkość układu. Wejścia sieci neuronowej były identyczne jak w powyżej

omawianej pracy, natomiast wyjścia były tylko dwa i reprezentowały współczynniki modelu liniowego opisującego zależność lepkości emulsji od szybkości ścinania. Baza danych została utworzona w oparciu o skład 9 formułacji. Wykazano wyższość sieci neuronowych nad wielomianami oraz modelami liniowymi. W niektórych przypadkach błędy predykcji dla sieci neuronowych były dwa razy mniejsze niż dla modeli statystycznych. Użycie hybrydowego systemu neuronowo-statystycznego pozwoliło z jednej strony uzyskać dobre wyniki predykcji dzięki elastyczności sieci neuronowych, a z drugiej strony uniknąć problemu „czarnej skrzynki” poprzez możliwość analizy modelu liniowego. Wykazano brak wpływu stężenia tenzydu na lepkość emulsji, natomiast istotny wpływ stosunku ilościowego fazy olejowej do wodnej.

3.3. MIKROEMULSJE

Sieć neuronowa okazała się skutecznym narzędziem do przewidywania przejść fazowych [6]. Użyto dwuwarstwowej sieci neuronowej uczonej propagacją wsteczną do klasyfikacji układów dwóch tenzydów, fazy olejowej i wodnej. Układy fazowe klasyfikowano jako mikroemulsje, ciekłe kryształy, emulsje W/O i emulsje O/W na podstawie badań mikroskopowych i konduktometrycznych. Zbiór uczący liczył zaledwie 40 rekordów a testowy 180 rekordów. W wyniku badań uzyskano ok. 90,5% prawidłowych klasyfikacji.

Agatonovic-Kustrin i wsp. [5] kontynuując badania nad przewidywaniem za pomocą sieci neuronowych składu mikroemulsji użyli hybrydowego systemu neuronowo-genetycznego, w którym algorytm genetyczny został użyty do redukcji wektora wejściowego z 18 do 9 wejść. Jako wejść użyto obok składu ilościowego m.in. deskryptorów chemicznej struktury kotenzydów, a zbiór uczący liczył 684 rekordy obejmujące układy złożone z dwóch rodzajów tenzydów, 8 rodzajów kotenzydów, jednego rodzaju fazy olejowej i wodnej. Zastosowano analogiczny jak poprzednio,

czterowartościowy system klasyfikacji układów fazowych. Najlepsze wyniki generalizacji sieci neuronowej wahały się w granicach 82% do 91% w zależności od rozpatrywanej klasy układów fazowych. Udowodniono w ten sposób możliwość stworzenia neuronowego modelu dla układów izotropowych w oparciu o parametry fizykochemiczne poszczególnych składników.

3.4. HYDROŻELE

Takayama i wsp. [116] porównali sieć neuronową z metodą RSM opartą na wielomianach drugiego rzędu na przykładzie optymalizacji składu hydrożelu z ketoprofenem do podania na skórę. Użyto specyficznej sieci neuronowej, w której warstwa ukryta nie jest połączona na zasadzie „każdy z każdym” z warstwą wyjściową, ale posiada pola recepcyjne dla konkretnych wyjść sieci. Pozwoliło to na stworzenie modelu z dwoma wejściami i trzema wyjściami. Jako wejścia i równocześnie zmienne sterujące wybrano zawartości promotorów wchłaniania: etanolu i O-etylomentolu. Wyjściami sieci były: szybkość penetracji ketoprofenu przez skórę szczura (R_p) oraz czas opóźnienia (*lag time* t_L), a także punktacja testu naskórnego (*Total Irritation Score* – *TIS*) zmierzone *in vivo* na modelach zwierzęcych (szczury). Sieci neuronowe uczone były za pomocą algorytmu rozszerzonego filtra Kalmana. Skład 12 formułacji hydrożelu z ketoprofenem stanowił bazę danych. Obliczone uprzednio parametry charakteryzujące optymalną formułację (R_p , t_L i *TIS*) zostały użyte jako wartości docelowe dla wytrenowanego modelu neuronowego w celu określenia optymalnej zawartości O-etylomentolu i etanolu. Model neuronowy wykazał dobrą zgodność z danymi doświadczalnymi uzyskanymi dla optymalnej formułacji hydrożelu z ketoprofenem. Błędy predykcji osiągnięte przez sieć neuronową były mniejsze w stosunku do metody RSM. Zwrócono uwagę na nieliniowy charakter sieci neuronowych, które dzięki temu są narzędziami odpowiednimi do identyfikacji nieliniowych zależności występujących w układach biologicznych w przeciwieństwie do wielomianów drugiego rzędu

użytych w metodzie RSM, które nie były w stanie we właściwy sposób odwzorowywać tych zależności.

3.5. TRANSDERMALNE SYSTEMY TERAPEUTYCZNE (TTS)

Kandimalla i wsp. [67] zwrócili uwagę na użyteczność sieci neuronowych i metody RSM w analizie wpływu promotorów wchłaniania obecnych w transdermalnych systemach terapeutycznych na szybkość przenikania melatoniny przez skórę szczura w warunkach *in vitro*. Przygotowano 10 rodzajów transdermalnych systemów terapeutycznych zawierających dwa lub trzy promotory wchłaniania, tj. wodę, etanol lub glikol propylenowy. Rodzaj promotora oraz jego zawartość w transdermalnym systemie terapeutycznym stanowiły wektor wejściowy modelu neuronowego (3 wejścia), zaś na wyjściu sieć przewidywała szybkość penetracji melatoniny przez membranę z uwzględnieniem czasu opóźnienia. Model neuronowy uczono na wszystkich 10 formulacjach z użyciem algorytmu propagacji wstecznej z techniką momentu. Przeprowadzono również modelowanie metodą RSM. Model neuronowy został zastosowany do zidentyfikowania optymalnego zestawu promotorów wchłaniania ze względu na zadaną szybkość penetracji melatoniny przez skórę w warunkach *in vitro*. Wykazano, że zastosowanie trzech promotorów wchłaniania było bardziej efektywne niż dowolna kombinacja dwóch substancji pomocniczych. Zwrócono uwagę na komplementarność systemów neuronowych i metody RSM, która wyrażała się zbliżonymi wynikami symulacji uzyskanymi przez oba systemy. Podkreślono również zdolność sztucznych sieci neuronowych do analizy wielowymiarowych danych, co jest dużo trudniejsze do przeprowadzenia za pomocą klasycznej metody RSM.

3.6. BADANIA PREFORMULACYJNE

Z uwagi na brak matematycznych modeli opisujących możliwość wystąpienia interakcji substancji pomocniczych w postaci leku Ebube i wsp. [34] użyli sztucznych sieci neuronowych. Wejścia modelu neuronowego opisywały ilościowy i jakościowy skład dwu- lub trójskładnikowych mieszanin polimerów: hydroksypropylometylocelulozy (HPMC), karagenu, alginianu sodu, poliwinylpirolidonu (PVP). Wyjścia sieci neuronowej stanowiły: szybkość pęcznienia polimerów, temperatura przejścia szklistego i lepkość mieszaniny polimerów. Baza wiedzy systemu neuronowego zawierała dane dotyczące 15 formułacji, a zbiór testowy 9 formułacji. Autorzy wykazali przydatność sieci neuronowych do analizy danych w badaniach preformulacyjnych o czym świadczy zgodność wyników ekperymentalnych z danymi uzyskanymi w procesie symulacji. Zwrócono uwagę na problem dokładności ekstrapolacji, która jest odwrotnie proporcjonalna do dystansu badanego rekordu od maksymalnych (lub minimalnych) wartości danych uczących. Jest to zgodne z wynikami badań Bourquina [16] dotyczącymi również dokładności ekstrapolacji systemów neuronowych.

3.7. KORELACJA IN VITRO-IN VIVO

Dowell i wsp. [33] podjęli prace nad opracowaniem metodyki użycia sztucznych sieci neuronowych do korelacji *in vitro* - *in vivo*. Dane dotyczyły trzech formułacji tabletek, dla których oznaczono profil uwalniania substancji leczniczej *in vitro* i przebieg zmian stężenia substancji leczniczej w osoczu krwi. Sieci neuronowe zostały wykorzystane do przewidywania parametrów farmakokinetycznych leku na podstawie ilości uwolnionej substancji leczniczej *in vitro*. Na wejściach podawane były ilości uwolnionej substancji leczniczej z tabletek po określonym czasie, na wyjściu zaś jej stężenia w osoczu. Autorzy przedstawili szereg możliwych do zastosowania sposobów organizacji danych wejściowych i wyjściowych. Zaprezentowano

również różne typy sieci neuronowych i ich architektur, np. sieci GRNN, charakteryzujące się krótkim czasem uczenia. Zwrócono uwagę na konieczność poszerzenia wektora wejściowego o np. dane charakteryzujące skład formulacji – model taki mógłby również służyć optymalizacji składu formulacji na podstawie zadanej charakterystyki farmakokinetycznej leku.

3.8. APARATURA

Praca Inghelbrecht i wsp. [62] poświęcona jest analizie parametrów pracy kompaktora. Jako zmienne sterujące wybrano szybkości obrotowe śruby pionowej i poziomej oraz rolek, a także ciśnienie powietrza w systemie hydraulicznym kompaktora (4 zmienne). Ocenie poddano wpływ tych parametrów na ścieralność sporządzonego granulatu. Sieć neuronowa była bardziej precyzyjna od wielomianów drugiego rzędu (metoda RSM). Wyniki analizy modelu neuronowego pozwoliły określić szybkość obrotową rolek w stosunku do szybkości obrotowej poziomej śruby jako 1:2, co usprawniło pracę kompaktora. Stwierdzono również brak wpływu szybkości pionowej śruby na mechaniczne właściwości granulatu.

Antikainen [8] przedstawił koncepcję kompleksowego systemu kontroli pracy granulatora fluidyzacyjnego za pomocą sieci neuronowej opartej na logice rozmytej. Szybkość przepływu roztworu lepiszcza, ciśnienie pod jakim roztwór był rozpylany i temperatura powietrza wprowadzanego do granulatora zostały wybrane jako parametry kluczowe dla określenia zwiększania wielkości ziaren granulatu. Przedstawiony model wykazał dobrą zgodność parametrów przewidywanych z obserwowanymi podczas prac doświadczalnych. Podkreślono łatwość pracy z systemami neuronowo-rozmytymi oraz ich zdolność do implementowania reguł wyrażanych w naturalnym języku (reguły lingwistyczne). Cechy te predestynują systemy neuronowo-rozmyte do rutynowego zastosowania jako narzędzi kontroli podczas produkcji w przemyśle farmaceutycznym.

II. ZAŁOŻENIA I CEL PRACY

Sztuczne sieci neuronowe pozwalają na analizę zagadnień, które są trudne bądź niemożliwe do analizy metodami statystycznymi. Zastosowania techniczne potwierdziły w praktyce unikalne właściwości tych systemów i stały się bodźcem do użycia sieci neuronowych w szeroko rozumianych naukach medycznych.

W technologii postaci leku nieustannie podejmowane są badania zmierzające do poszukiwania nowych rozwiązań technologicznych w celu uzyskania postaci leku zapewniających odpowiedni efekt terapeutyczny. Wybór substancji pomocniczych, jak i wykorzystanie właściwych procesów technologicznych stanowi podstawę do osiągnięcia zamierzonych celów. Różnorodność proponowanych technik, jak i ogromna ilość wprowadzanych do badań substancji pomocniczych może stwarzać pewne problemy w wyborze optymalnego toku postępowania.

W niniejszej pracy podjęto próbę unifikacji metodyki modelowania neuronowego w zakresie formułowania postaci leku. Założono możliwość użycia modeli neuronowych w charakterze maszyn wnioskujących, jako składników neuronowych systemów ekspertowych.

Celem pracy była ocena przydatności stosowania sztucznych sieci neuronowych jako uniwersalnych narzędzi modelujących w zakresie doboru składu postaci leku i opracowywania wyników badań dostępności farmaceutycznej i biologicznej.

Realizacja celu pracy obejmowała konstrukcję neuronowych systemów spełniających następujące zadania:

- optymalizację składu stałych rozprożeń substancji leczniczych w zależności od ich budowy chemicznej i rozpuszczalności, a także rodzaju użytego nośnika;
- optymalizację składu mikroemulsji zawierających tenzydy o różnych właściwościach fizykochemicznych;

- aproksymację dowolnej funkcji opisującej przebieg krzywych stężenie-czas w badaniach *in vitro* i *in vivo*.

III. CZĘŚĆ DOŚWIADCZALNA

1. METODYKA

1.1. MODELE NEURONOWE

Systemy neuronowe wspomagające formulację stałych rozproszeń i mikroemulsji w swej funkcji odpowiadają tzw. systemom ekspertowym. Ich zadaniem jest wspomaganie procesu decyzyjnego dotyczącego zasadności formułowania postaci leku przy użyciu wybranych substancji pomocniczych, jak też i określonej technologii sporządzania. Z uwagi na to metodyka przygotowania systemów neuronowych była analogiczna jak dla systemów ekspertowych, mimo iż sieci neuronowe nie posiadają wszystkich cech klasycznych systemów ekspertowych. Pewne modyfikacje wynikały ze specyfiki modelowania neuronowego.

Przygotowanie systemu neuronowego:

- 1) zdefiniowanie funkcji systemu neuronowego
- 2) przygotowanie danych stanowiących bazę wiedzy
 - a) akwizycja danych
 - b) wstępne przygotowanie danych (*data preprocessing*)
 - zdefiniowanie wektora wejściowego i wyjściowego
 - skalowanie, normalizacja, redukcja lub multiplikacja liczby rekordów
 - wzbogacanie zbioru danych o reguły zdroworozsądkowe oraz zaszumione rekordy danych jako metody wpływające na poprawę zdolności predykcyjnych modelu
 - c) podział wyjściowego zbioru danych na pary uczący-testowy zgodnie z wymogami metodyki k-krotnego wzajemnego sprawdzania

- 3) budowa maszyny wnioskującej
 - a) użycie modelu neuronowego w charakterze maszyny wnioskującej
 - b) selekcja optymalnej architektury sieci neuronowej w toku eksperymentów numerycznych
 - zdefiniowanie architektury sieci neuronowej
 - trening z zadanymi parametrami uczenia
 - walidacja otrzymanego modelu na zbiorach testowych za pomocą metody k-krotnego wzajemnego sprawdzania
 - modyfikacja wektora wejściowego na podstawie analizy wrażliwościowej [94]
 - c) budowa systemów, w których sieci wyższego rzędu bazują na odpowiedziach sieci uczonych bezpośrednio na danych pochodzących z bazy wiedzy (tzw. komitet ekspertów)
- 4) budowa interfejsu użytkownika
 - a) graficzny interfejs użytkownika projektowany dzięki bibliotekom VCL języka Borland Delphi® 5.0
 - b) automatyzacja pracy

W celu realizacji niniejszej pracy wytrenowano i przetestowano ok. 50 000 sieci neuronowych.

1.2. WERYFIKACJA MODELI NEURONOWYCH

Weryfikację modeli neuronowych przeprowadzano zgodnie z metodą k-krotnego wzajemnego sprawdzania gdzie **k** przyjmowało wartość 10 lub było równe ilości formułacji wchodzących w skład danego zbioru bazowego (metoda *leave-one-out*).

Istotność statystyczną różnicy średnich błędów sieci neuronowych sprawdzano za pomocą testu t-Studenta ($\alpha=0,05$) oraz analizy wariancji.

Prezentowane wyniki modeli predykcyjnych stałych rozproszeń i mikroemulsji są wynikami testowania sieci neuronowych na zbiorach wyłączonych z uczenia, czyli przedstawiają generalizację systemu. W modelach aproksymacyjnych cytowane są wyniki testowania na zbiorach uczących.

1.3. NOTACJA

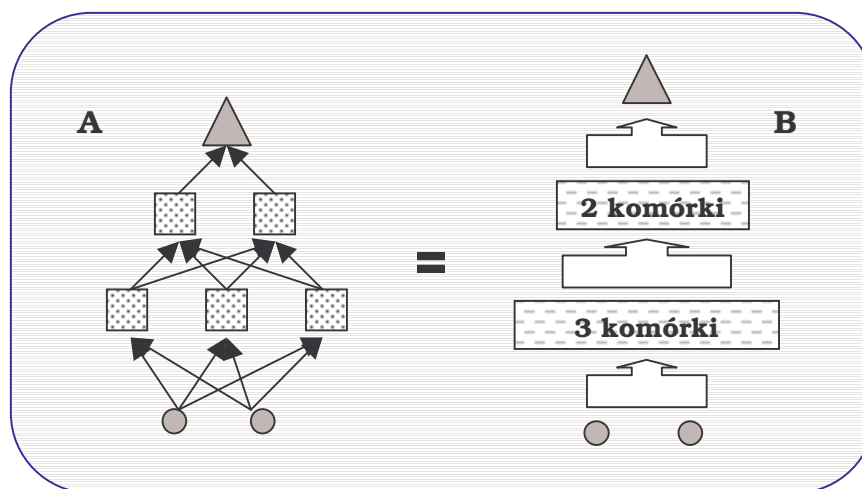
Wprowadzono ujednolicony opis architektur sieci neuronowych według następującego schematu:

*{liczba komórek warstw ukrytych} **hid** {funkcja aktywacji} {inne}*

Zgodnie z tym architekturę przedstawioną na rycinie 7 należy zapisać jako „3_2 hid tanh”, przy założeniu, że sieć neuronowa posiada tangens hiperboliczny jako funkcję aktywacji i jest siecią typu MLP. Architektura ta określana jest mianem sieci trójwarstwowej, ponieważ liczone są obie warstwy ukryte i wyjściowa (bez warstwy wejściowej). Oznaczenie „fuzzy” w miejscu funkcji aktywacji notuje sieć opartą na logice rozmytej. Oznaczenia *{inne}* mogą dotyczyć maksymalnej ilości iteracji, lub metody defuzyfikacji sieci rozmytych (np. M-SLIDE). Sekcja *{inne}* może zawierać również opis metod skanowania użytych do otrzymania zbiorów danych dla modeli mikroemulsji: „_Intelliscan” (algorytm **Intelliscan**) i „_scan_2_2” (skanowanie równomierne co 2%).

W celu uproszczenia schematu graficznego architektury sieci pominięto element progowy. W przypadku dużych sieci użyto schematu przedstawionego na rycinie 7B.

W przedstawianych równaniach matematycznych zastosowano system notacji, w którym duże litery oznaczają wielkości wektorowe, zaś małe litery wielkości skalarne.



Rycina 7. Równoważne schematy architektury 3_2 hid tanh.

2. OPRZYRZĄDOWANIE

Sztuczne sieci neuronowe pracują w sposób równoległy. Ten sposób pracy różni je od krokowego systemu pracy klasycznych komputerów. Obecnie istnieją już specjalizowane komputery neuronowe, jednakże ich wykorzystanie jest jeszcze ograniczone, z uwagi na małą elastyczność i wysoką cenę. W związku z tym stosuje się w dalszym ciągu odpowiednie programy symulujące sztuczne sieci neuronowe na komputerach PC. Niektórzy autorzy z powodzeniem posługują się w badaniach nad sieciami neuronowymi oprogramowaniem własnej konstrukcji [27, 130]. Aktualnie dostępny jest szeroki zakres narzędzi programistycznych typu RAD (*Rapid Application Development*) pozwalających na szybkie i efektywne projektowanie aplikacji. Skłoniło to do zaprojektowania i napisania oprogramowania własnej konstrukcji w postaci symulatora **Nets2003**. Z uwagi na to, że praca z sieciami neuronowymi wymaga obróbki dużych ilości danych, zaprojektowano i napisano zestaw programów narzędziowych ułatwiających zarówno akwizycję jak i późniejszą transformację danych w celu ich prezentacji siecom neuronowym. Język Borland Delphi® 5.0, który wybrano do napisania oprogramowania zapewnił elastyczność i efektywność pracy jak też i dużą wydajność otrzymanych aplikacji. Arkusz kalkulacyjny

MS Excel® 2000 okazał się niewystarczający zarówno pod względem ilości danych możliwych do obróbki jak też i szybkości działania.

W celu realizacji założeń niniejszej pracy zaprojektowano i napisano w języku Borland Delphi® 5.0 następujące aplikacje:

- a) symulator sieci neuronowych **Nets2003**
- b) programy do digitalizacji danych graficznych: **Phase_diagr** i **Graph_read**
- c) aplikacje do wstępnego przetwarzania danych: **Preproc**, **Eddan3**, **Erase_0**
- d) programy do manipulacji plikami tekstowymi w celu przygotowania zbiorów uczących i testowych: **Split_file**, **Learn_conv**
- e) programy do obróbki wyników pracy sieci neuronowych: **Classif_compute**, **Gen_compute**
- f) inne: **Launch_file**

2.1. SYMULATOR SIECI NEURONOWYCH

Nets2003 jest systemem do symulacji, testowania i analizy sztucznych sieci neuronowych. Jest rozwinięciem programu **Sieci95** opracowanego po raz pierwszy w 1995 r. Aktualnie liczy ok. 14 000 linii kodu i jest stale udoskonalany. Posiada możliwości pracy w trybie wsadowym tj. treningu i analizy zestawów sieci bez ingerencji użytkownika. Na podstawie wcześniej zdefiniowanych plików konfiguracyjnych jest w stanie podjąć pracę nawet po przerwie w zasilaniu komputera. Po skończonej pracy może wyłączyć komputer. W zależności od wielkości wektora wejściowego może pracować na zbiorach danych liczących ok. 1-5 mln rekordów. Zaimplementowano również system monitorowania kluczowych punktów pętli programu z wyprowadzaniem komunikatów diagnostycznych do pliku zewnętrznego (Error_log.txt) w celu późniejszej analizy przyczyn nieprawidłowej pracy. Program posiada graficzny, intuicyjny interfejs użytkownika pozwalający na szybkie zdefiniowanie struktury sieci neuronowej i jej parametrów uczenia.

Nets2003 wyposażony został w możliwości symulacji następujących typów sieci neuronowych:

- a) perceptronów wielowarstwowych (MLP)
- b) sieci z kontrapropagacją (CP)
- c) sieci z warstwami kontekstu typu Jordana-Elmana
- d) sieci rozmytych (*neuro-fuzzy*)

Podstawowy algorytm uczenia to propagacja wsteczna z modyfikacjami:

- a) technika momentu
- b) zanikające współczynniki uczenia
- c) algorytm *delta-bar-delta*

Dla sieci typu MLP zaimplementowano metodę analizy wrażliwościowej [94].

W zakresie sieci rozmytych, **Nets2003** oferuje różne metody defuzyfikacji jak: COA, MOM, M-SLIDE, a także metody inicjalizacji reguł:

- a) bezpośrednio na danych uczących
- b) metoda góraska
- c) metoda *lookup table* [49, 79, 129]
- d) połączenia tych metod

Zaimplementowano również kryterium Fishera redukcji ilości reguł rozmytych. Do uczenia sieci z kontrapropagacją (CP) symulator posiada także opcję gazu neuronowego.

Nets2003 może symulować sieci neuronowe z maksymalną liczbą 10 warstw ukrytych i 2500 komórek w każdej warstwie. Wartości te mogą być w razie potrzeby modyfikowane. Wszystkie komórki w przylegających warstwach są zawsze połączone na zasadzie „każdy z każdym”.

Zaimplementowano następujące funkcje aktywacji:

- a) sigmoida (funkcja logistyczna)
- b) $a*x$
- c) a/x
- d) $\ln(|x|)$
- e) $\sin(x)$

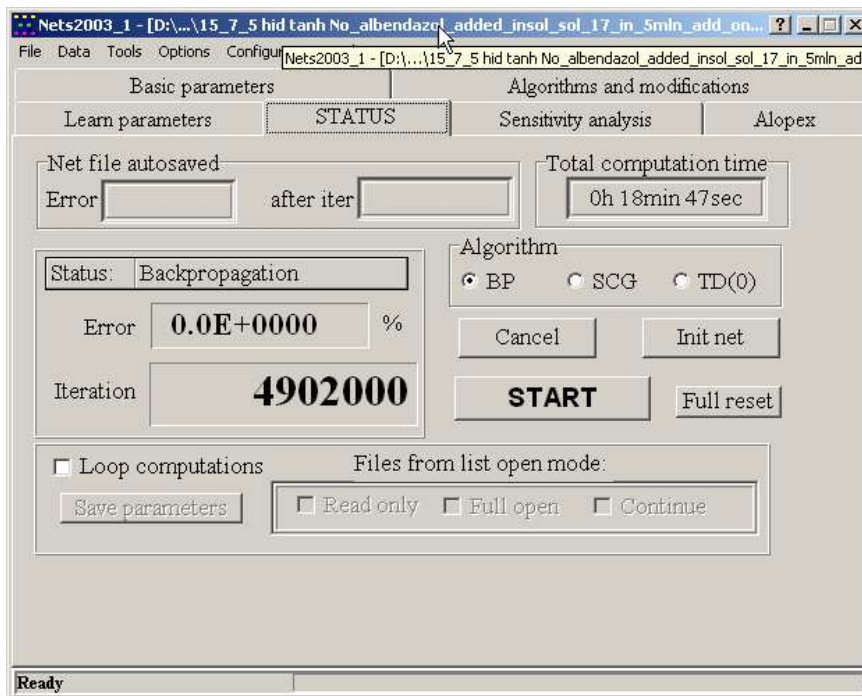
- f) $\cos(x)$
- g) tangens hiperboliczny $\tanh(x)$
- h) funkcja fsr wg Bilskiego [12]

Zasadniczo wszystkie komórki posiadają jednakowe funkcje aktywacji chociaż każda warstwa może mieć zdefiniowane indywidualne parametry dla swoich komórek (funkcja aktywacji, współczynniki uczenia). Jako funkcje kosztu zaimplementowano:

- a) klasyczną funkcję kwadratową
- b) funkcję logarytmiczną
- c) funkcję Hampela

Weryfikacji systemu **Nets2003** dokonano z użyciem klasycznych problemów testowych jak np. problem XOR, czy nauka dodawania. Wyniki jego pracy nie odbiegały znacząco od otrzymanych przez profesjonalne systemy SNNSv4.2 [55] czy PDP++ [54].

Główne okno programu przedstawiono na rycinie 8, zaś dodatkowe opcje zobrazowano na rycinie 30 ze str. 160.



Rycina 8. Widok głównego okna programu *Nets2003*.

2.2. OPROGRAMOWANIE NARZĘDZIOWE

SPECJALISTYCZNE OPROGRAMOWANIE NAPISANE NA POTRZEBY NINIEJSZEJ PRACY

- a) **Gen_compute** (Rycina 34 str. 162) został zaprojektowany do obróbki wyników testowania sieci neuronowych w programie **Nets2003** w celu stworzenia zbiorów uczących dla sieci drugiego rzędu (komitet ekspertów). Pozwala również na obliczenie średniej i odchyłeń standardowych z wyników testowania wielu architektur sieci neuronowych. Jego kod źródłowy liczący ok. 300 linii został wykorzystany w programach **Classif_compute** i **PKB 2.0**
- b) **Classif_compute** (Rycina 31 str. 161) to program obliczający błędy klasyfikacji. Wyposażony jest w algorytm automatycznego doboru progu decyzyjnego zarówno dla klasyfikacji binarnej jak też i wielokrotnej. Pozwala na ocenę błędu generalizacji całej serii sieci zgodnie z metodą k-krotnego wzajemnego sprawdzania. Wyniki pracy zapisywane są w pliku tekstowym. Kod źródłowy liczy ok. 1500 linii
- c) **Preproc** (Rycina 40 str. 165) służy do szeroko rozumianego wstępnego przetwarzania danych. Kod źródłowy zawiera ok. 2000 linii. Program został wyposażony w następujące funkcje:
- odbijanie względem punktu (0,0)
 - binaryzacja danych z automatycznym określeniem wielkości wektora binarnego na podstawie zbioru danych; możliwe jest zadanie dowolnych etykiet w wektorze binarnym (np. -1 i 1)
 - normalizacja za pomocą normy euklidesowej
 - logarytmowanie danych
 - dodawanie kolumn na początku danych
 - mnożenie konkretnej kolumny o stałą
 - usuwanie jednakowych rekordów danych

- balansowanie reprezentacji poszczególnych klas poprzez losowe usuwanie rekordów klas nadmiarowych lub multiplikację rekordów klas mniejszościowych
- sprawdzanie poprawności danych i usuwanie rekordów zawierających inne niż alfanumeryczne znaki
- usuwanie nadmiarowych rekordów danych, czyli takich których wszystkie elementy są identyczne z innym rekordem danych

d) **Eddan3** (Rycina 32 str. 161) to drugi program do wstępnego przetwarzania danych. Kod źródłowy liczy ok. 800 linii. Skonstruowany został w postaci prostego arkusza i udostępnia następujące funkcje:

- multiplikacja rekordów danych z nakładaniem szumu o zadanej amplitudzie i znaku. Amplituda szumu może być zadawana jako stała wartość lub procent danej wartości
- podział na pary zbiorów testowy-uczący względem kolumn zbioru bazowego z równoczesnym przestawianiem wierszy lub bez
- podział na pary zbiorów testowy-uczący względem wierszy zbioru bazowego

e) **Split_file** (Rycina 41 str. 166) realizuje także podział na pary zbiorów testowy-uczący względem wierszy zbioru bazowego. Dzięki inaczej skonstruowanemu algorytmowi operacja ta jest wykonywana wielokrotnie szybciej niż w programie **Eddan3**. Kod źródłowy liczy 300 linii

f) **Learn_conv** (Rycina 36 str. 163) również tworzy pary zbiorów testowy-uczący, ale w przeciwieństwie do obu poprzednich programów tworzy pliki uczące i testowe z serii plików tekstowych, z których każdy opisuje jedną formulację. Obowiązują zasady k -krotnego wzajemnego sprawdzania, gdzie k może przyjmować dowolne wartości naturalne. Program pozwala na losowy lub sekwencyjny dobór plików i generuje

na końcu pracy raport o tym, w którym zbiorze testowym znajdują się dane z pierwotnych plików danych. Zawiera 300 linii kodu

- g) **Erase_0** (Rycina 33 str. 162) pozwala na usuwanie kolumn danych, których wartość maksymalna i minimalna są równe sobie – takie kolumny danych przeszkadzają w procedurze skalowania używanej w **Nets2003**. Kod źródłowy składa się z ok. 600 linii

- h) **Graph_read** (Rycina 35 str. 163) służy do digitalizacji wykresów przebiegu krzywych stężenie-czas publikowanych w piśmiennictwie fachowym. **Graph_read** jest skanerem bitmap obsługiwany przez użytkownika za pomocą myszki, gdzie po zadaniu wartości rzeczywistych skali na osi wykresu kliknięcie w obrębie okna programu jest przetwarzane na współrzędne macierzystego wykresu i zapisywane do pliku tekstowego. Jego kod źródłowy zawiera ok. 500 linii

- i) **Phase_diagr** (Rycina 38 str. 164) zaprojektowano i napisano jako skaner bitmap specyficznym przeznaczone do digitalizacji trójkątów fazowych opisujących skład mikroemulsji. Odpowiednio przygotowana bitmapa skanowana jest pod kątem kolorów pikseli, które porównywane są z uprzednio zadanymi wzorcami. Współrzędne prostokątne bitmapy przeliczane są na współrzędne trójkątne diagramu Gibbsa i następnie zapisywane w pliku tekstowym. Program posiada możliwości kompresowania danych – na potrzeby tej funkcji opracowano algorytm kompresji własnej konstrukcji **Intelliscan** pozwalający ograniczać rozmiar plików wyjściowych za pomocą arbitralnie ustalanego współczynnika kompresji. **Phase_diagr** służy również do wizualizacji wyników predykcji układów fazowych przez sieci neuronowe. Aktualnie kod źródłowy zawiera ok. 1500 linii

- j) **Launch_file** (Rycina 37 str. 164) uruchamia inne programy z półminutowym opóźnieniem, co pozwala użytkownikowi na decyzję co do wstrzymania automatycznego uruchomienia danego programu. Umieszczony jest zazwyczaj w menu „Autostart” systemu Windows® i uruchamia symulator **Nets2003** z parametrem `"/a`” co pozwala na automatyczne podjęcie uczenia zadanych wcześniej architektur sieci neuronowych. Wykorzystywane jest to w wypadku awarii zasilania lub systemu operacyjnego, w wyniku czego należy ponownie uruchomić komputer, co jest jedyną czynnością wymaganą do przywrócenia procesu uczenia w miejscu ostatniego jego zapisu. Jest to mały program liczący ok. 170 linii kodu

INNE OPROGRAMOWANIE

W badaniach używano również komercyjnego arkusza kalkulacyjnego Microsoft Excel® 2000 do statystycznego opracowania wyników oraz niektórych przekształceń danych. Wstępnej obróbki skanowanych trójkątów fazowych dokonano za pomocą pakietu graficznego Corel® 9.0.

2.3. SPRZĘT KOMPUTEROWY

Wszystkie obliczenia zostały wykonane na 4 dwuprocessorowych stacjach roboczych Pentium III 1 GHz pracujących pod kontrolą środowiska Windows® 2000 Professional w trybie 24h/d przez 7 dni w tygodniu. Stacje robocze zarządzane są poprzez sieć lokalną dzięki oprogramowaniu VNC (*Virtual Network Computing*) firmy AT&T (licencja GPL). Stacje robocze posiadały software’owe i hardware’owe zabezpieczenia temperaturowe, w tym własnej konstrukcji program **Exit_NT_auto** wyłączający komputer w razie przegrzania.

3. MODELOWANIE NEURONOWE

Modelowanie neuronowe dotyczyło następujących problemów:

- a) zależność profilu rozpuszczania substancji leczniczej od jakościowego i ilościowego składu stałego rozproszenia
- b) zależność wystąpienia układu mikroemulsyjnego od ilościowego i jakościowego składu formulacji
- c) aproksymacja funkcji stężenie-czas w badaniach *in vitro* i *in vivo*

Modele opisujące właściwości formulacji farmaceutycznych zostały zaprojektowane tak, aby mogły służyć jako podstawa dla systemów doradczych wspomagających proces otrzymywania formulacji. Działanie takich systemów polega na przeszukiwaniu parametrów formulacji w zadanym zakresie i przedstawieniu decyzji uzasadniającej przeprowadzenie eksperymentów laboratoryjnych z konkretnym składem jakościowym i ilościowym przy wykorzystaniu danej technologii produkcji.

System do aproksymacji funkcji stężenie-czas zaprojektowano w celu stworzenia narzędzia pozwalającego na obliczanie dla danej formulacji parametrów zarówno testu rozpuszczania jak też i farmakokinetycznych, bez zakładania modelu matematycznego. Dodatkowo, system neuronowy posłużył do wstępnego przetwarzania danych dla modeli rozpuszczania substancji leczniczych ze stałych rozproszeń. Położono nacisk na otrzymanie uniwersalnego i łatwego w obsłudze narzędzia.

3.1. STAŁE ROZPROSZENIA

Stałe rozproszenia są to układy dwu- lub trójskładnikowe zawierające substancję leczniczą trudno rozpuszczalną w wodzie, nietoksyczny nośnik łatwo rozpuszczalny i w stosowanych ilościach pozbawiony własnego

działania farmakologicznego. Sporządzać je można jedną z następujących metod: stapiania, odparowania, suszenia rozpyłowego, hot-spin mixing (HSM) oraz ekstruzji topliwej [65]. Termin stałe rozproszenia przypisywany jest różnym układom fizykochemicznym takim jak: mieszaniny eutektyczne, stałe roztwory, szkliste roztwory, związki kompleksowe lub mieszaniny w/w form. Dane z piśmiennictwa wskazują na pełne uzasadnienie stosowania w/w układów w technologii postaci leku, ze względu na możliwość znacznej poprawy nie tylko dostępności farmaceutycznej substancji leczniczej, ale również dostępności biologicznej [26, 39, 50, 93, 122].

3.1.1. ZAKRES BADAŃ

- a) Modelowanie profilu rozpuszczania substancji leczniczej ze stałych rozproszeń i mieszanin fizycznych zawierających nośniki o różnej budowie chemicznej

Jako modelową substancję leczniczą przyjęto ketoprofen. Z uwagi na możliwość wpływu na poprawę rozpuszczalności substancji leczniczej uwzględniono również modelowanie profilu rozpuszczania ketoprofenu z mieszanin fizycznych .

Baza danych obejmowała wyniki prac doświadczalnych dotyczących stałych rozproszeń prowadzonych w Katedrze Technologii Postaci Leku i Biofarmacji CMUJ w Krakowie [66]. Przedmiotem badań były stałe rozproszenia zawierające trzy rodzaje nośników o różnych masach cząsteczkowych (Tabela 1).

Tabela 1. Nośniki stałych rozproszeń.

Nośnik	Oznaczenie
Makrogol 4000	PEG 4000
Makrogol 6000	PEG 6000
Makrogol 9000	PEG 9000
Poliwinylopirolidon K15	PVP K15
Poliwinylopirolidon K30	PVP K30
Poliwinylopirolidon K90	PVP K90
Hydroksypropylometyloceluloza	HPMC

- b) Modelowanie profilu rozpuszczania substancji leczniczych o różnej rozpuszczalności ze stałych rozproszeń z nośnikami o różnej budowie chemicznej

Zbiór danych utworzono na podstawie piśmiennictwa. Ocenie poddano stałe rozproszenia zawierające substancje lecznicze trudno lub praktycznie nierozpuszczalne w wodzie. W bazie danych uwzględniono również substancję łatwo rozpuszczalną w wodzie tj, chlorowoderek oksprenololu (Tabela 2).

Tabela 2. Układy stałych rozproszeń będących przedmiotem badań.

Stale rozproszenie		Piśmien- nictwo
Substancja lecznicza	Nośnik	
Albendazol	Poliwinylopirolidon (PVP K-12)	[121]
Chlorowodorek oksprenololu	Etyloceluloza (EC-7, EC-10, EC-22, EC-100), hydroksypropyloceluloza (HPC-SL, HPC-L, HPC-M, HPC-H)	[96, 133]
Ciprofloksacyna	Makrogol (PEG-6000)	[41]
Diazepam	Fosfatydylocholina (PC)	[43]
Dikumarol	Poliwinylopirolidon (PVP K-15), β -cyklodekstryna	[111]
Fenytoina	Makrogol (PEG-4000)	[132]
Flurbiprofen	Hydroksypropyloceluloza (HPC-SSL, HPC-SL, HPC-L, HPC-M, HPC-H)	[134]
	Politlenek etylenu (PEO) , hydroksypropyloceluloza (HPC-M)	[97]
Ibuprofen	Makrogol (PEG-10000), Talk	[68]
Indometacyna	Nikotynamid	[37]
Karbamazepina	Gelucire 44/14, Gelucire 50/13	[99]
Kwas mefenamowy	Makrogol (PEG-3350), Tween 20	[95]
Mebendazol	Makrogol (PEG-4000, PEG-6000)	[24]
Nitrazepam	Fosfatydylocholina (PC)	[43]
Oksazepam	Makrogol (PEG-6000)	[63]
Oksodypina	Makrogol (PEG-6000)	[128]

Tabela 2 c.d.

Stałe rozproszenie		Piśmiennictwo
Substancja lecznicza	Nośnik	
Papaweryna	Makrogol (PEG-6000, PEG-20000)	[35]
Piroksykam	Poliwinylopirolidon (PVP K-30)	[118]
	Poliwinylopirolidon (PVP K-17, PVP K-90)	[119]
Sulfatiazol	Mocznik	[25]

c) Modelowanie profilu rozpuszczania trudno rozpuszczalnych substancji leczniczych z układów zawierających jeden rodzaj nośnika

W zbiorze danych z piśmiennictwa uwzględniono stałe rozproszenia i mieszaniny fizyczne zawierające Makrogol jako nośnik (Tabela 3).

Tabela 3. Układy stałych rozproszeń z Makrogolem będących przedmiotem badań.

Stałe rozproszenie		Piśmiennictwo
Substancja lecznicza	Nośnik	
Ciprofloksacyna	Makrogol (PEG-6000)	[41]
Fenytoina	Makrogol (PEG-4000)	[132]
Ibuprofen	Makrogol (PEG-10000)	[68]
Mebendazol	Makrogol (PEG-4000, PEG-6000)	[24]
Oksazepam	Makrogol (PEG-6000)	[63]
Oksodypina	Makrogol (PEG-6000)	[128]
Papaweryna	Makrogol (PEG-6000, PEG-20000)	[35]

3.1.2. ZDEFINIOWANIE FUNKCJI SYSTEMU NEURONOWEGO

Zadaniem systemu neuronowego było przewidywanie szybkości rozpuszczania substancji leczniczych ze stałych rozproszeń i mieszanin fizycznych w oparciu o zgromadzone dane dotyczące składu jakościowego, ilościowego i technologii sporządzania formulacji.

3.1.3. PRZYGOTOWANIE DANYCH STANOWIĄCYCH BAZĘ WIEDZY

Do utworzenia zbioru danych z piśmiennictwa posłużono się skonstruowanym specjalnie w tym celu programem **Graph_read**, który po zeskanowaniu wykresów i ich obróbce graficznej digitalizował obraz do plików tekstowych możliwych do przedstawienia sieci neuronowej. Ostatecznie bazowy zbiór danych zawierał ok. 3000 rekordów.

Liczba wejść sieci wynosiła minimalnie siedem, a maksymalnie szesnaście. Zależało to od analizowanego zagadnienia i dostępnych dla niego danych:

a) W przypadku modelowania profilu rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych zawierających nośniki o różnej budowie chemicznej baza danych zawierała 100 rekordów danych będących wynikiem testu rozpuszczania substancji leczniczej z 25 formulacji. Pełna charakterystyka opracowywanych formulacji stałych rozproszeń i mieszanin fizycznych wymagała użycia wektora wejściowego z 7 zmiennymi wejściowymi (Tabela 4).

Tabela 4. Wektor wejściowy dla modelu neuronowego rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych.

Nr wejścia	Kodowana informacja
1	Rodzaj formulacji: stałe rozproszenie lub mieszanina fizyczna
2	Stałe rozproszenie sporządzone z mieszaniny fizycznej czy nie
3	Metoda sporządzania: stapiania, odparowania
4	Warunki chłodzenia: w temperaturze pokojowej lub na łaźni lodowej
5	Ilość substancji leczniczej w 1g formulacji
6	Indeks konektywności (CI) nośnika
7	Czas rozpuszczania w minutach

b) Dla modelowania profilu rozpuszczania substancji leczniczych o różnej rozpuszczalności ze stałych rozproszeń z nośnikami o różnej budowie chemicznej użyto całej dostępnej bazy danych (3000 rekordów). Liczba wejść sieci neuronowej wynosiła 16 (Tabela 5).

c) W przypadku modelowania profilu rozpuszczania trudno rozpuszczalnych substancji leczniczych z układów zawierających Makrogol wyselekcjonowano ze zbioru bazowego 64 formulacje, które złożyły się na zbiór danych liczący ok. 800 rekordów. Z uwagi na brak w wyselekcjonowanych formulacjach przykładów trójskładnikowych stałych rozproszeń wektor wejściowy został zmniejszony do 12 elementów (Tabela 6).

Tabela 5. 16-elementowy wektor wejściowy neuronowego modelu profilu rozpuszczania substancji leczniczych ze stałych rozproszeń.

Nr wejścia	Kodowana informacja
1	Masa cząsteczkowa substancji leczniczej
2	Indeks konektywności (CI) substancji leczniczej
3	Rodzaj formułacji: stałe rozproszenie lub mieszanina fizyczna
4	Stałe rozproszenie sporządzone z mieszaniny fizycznej czy nie
5	Metoda sporządzania: stapiania, odparowania, suszenia rozpyłowego
6	Warunki chłodzenia: w temperaturze pokojowej lub na łaźni lodowej
7	Ilość substancji leczniczej w 1g formułacji
8	Zawartość nośnika nr 1 w 1g formułacji
9	Zawartość nośnika nr 2 w 1g formułacji
10	Indeks konektywności (CI) nośnika nr 1
11	Indeks konektywności (CI) nośnika nr 2
12	Masa cząsteczkowa nośnika nr 1
13	Masa cząsteczkowa nośnika nr 2
14	Ilość obrotów łopatki / koszyczka
15	pH medium w teście rozpuszczania (<i>dissolution test</i>)
16	Czas rozpuszczania w minutach

Tabela 6. Wektor wejściowy neuronowego modelu profilu rozpuszczania substancji leczniczych ze stałych rozprożeń z Makrogolem.

Nr wejścia	Kodowana informacja
1	Masa cząsteczkowa substancji leczniczej
2	Indeks konektywności (CI) substancji leczniczej
3	Rodzaj formułacji: stałe rozproszenie lub mieszanina fizyczna
4	Stałe rozproszenie sporządzone z mieszaniny fizycznej czy nie
5	Metoda sporządzania: stapiania, odparowania, suszenia rozpyłowego
6	Ilość substancji leczniczej w 1g formułacji
7	Zawartość nośnika nr 1 w 1g formułacji
8	Indeks konektywności (CI) nośnika nr 1
9	Masa cząsteczkowa nośnika nr 1
10	Ilość obrotów łopatki / koszyczka
11	pH medium w teście rozpuszczania (<i>dissolution test</i>)
12	Czas rozpuszczania w minutach

Komórka wyjściowa była zawsze tylko jedna i wyrażała ilość rozpuszczonej substancji leczniczej z układów proszkowych w warunkach *in vitro* po podanym na wejściu sieci neuronowej czasie.

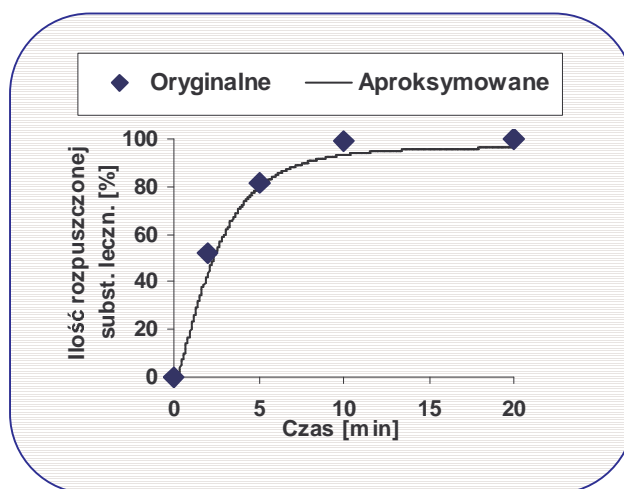
Do kodowania substancji leczniczych i nośników użyto mas cząsteczkowych i indeksu konektywności (*connectivity index* - CI). Parametr ten jest liczbą charakteryzującą strukturę chemiczną cząsteczki danej substancji. CI obliczano na podstawie znanej budowy cząsteczkowej substancji chemicznych za pomocą algorytmu zaproponowanego przez Kier i wsp. [69-73]. W przypadku polimerów CI obliczano dla uśrednionej wielkości cząsteczki obliczonej na podstawie średniej masy cząsteczkowej polimeru. W przypadku kopolimerów uwzględniono dodatkowo dane dotyczące średniej zawartości poszczególnych monomerów oraz rodzaju podstawników [40]. Na

podstawie danych z piśmiennictwa określono także rozpuszczalność substancji leczniczych zaliczając je do dwóch klas: rozpuszczalne i nierozpuszczalne w wodzie [21].

Zbiory danych poddawano liniowemu skalowaniu w zakresach (0,2;0,8) i (-0,8;0,8). Do zbiorów uczących dodawano również rekordy z nałożonym szumem o rozkładzie jednostajnym i amplitudzie $\pm 5\%$ w czterokrotnej ilości w stosunku do wyjściowej liczby rekordów.

Bazowy zbiór danych podzielono na 10 par zbiorów testowy-uczący zgodnie z zasadami metody k -krotnego wzajemnego sprawdzania. Stosowano również podział zgodny z metodyką *leave-one-out*, gdzie wartość k równała się ilości formułacji, jaka była dostępna przy opracowywaniu danego modelu.

Przeprowadzono również eksperymenty z neuronowym wstępnym przetwarzaniem danych (*preprocessing*), które polegało na aproksymacji funkcji stężenie-czas przez system **PKB 2.0** (Rycina 9).



Rycina 9. Przykład aproksymacji przebiegu krzywej rozpuszczania ketoprofenu ze stałego rozproszenia.

Aproksymowane profile rozpuszczania substancji leczniczej próbkowano co 3 sekundy. Tak uzyskane zbiory danych służyły jako baza wiedzy dla właściwych modeli neuronowych przewidujących szybkość rozpuszczania substancji leczniczej ze stałych rozproszeń i mieszanin fizycznych.

3.1.4. BUDOWA MASZYNY WNIOSKUJĄCEJ

Poszukiwania najlepszej architektury sieci neuronowej prowadzone były z użyciem metodyki 10-krotnego wzajemnego sprawdzania. Technika *leave-one-out* stosowana była do dodatkowej weryfikacji optymalnego modelu neuronowego. Poszukiwania prowadzone były w grupie modeli uczonych z nauczycielem. Testowano liniowe i nieliniowe sieci typu MLP oraz oparte na logice rozmytej. Jako funkcje aktywacji testowano funkcje liniowe, logarytmiczne, sigmoidę (funkcję logistyczną), tangens hiperboliczny i funkcję *fsr*. Wszystkie komórki w sieci posiadały zawsze tę samą funkcję aktywacji. Komórki w przylegających warstwach były połączone na zasadzie „każdy z każdym”. Testowano również różne funkcje kosztu: kwadratową, logarytmiczną i Hampela. Liczba warstw ukrytych była zmienna i wahała się od 0 do 5. Algorytm uczący stanowiła zawsze propagacja wsteczna z techniką momentu oraz algorytmem *delta-bar-delta*. Zastosowano również technikę *jog-of-weights* zapobiegającą utykaniu sieci w minimum lokalnym. Technika ta polega na sprawdzaniu szybkości uczenia sieci neuronowej. Jeśli szybkość uczenia spada, tzn. sieć nie poprawiła swego wyniku przez zadaną ilość iteracji (kryterium „cierpliwości” systemu), to do wszystkich wag dodawany jest losowy szum o niewielkiej amplitudzie i rozkładzie gaussowskim. Chwilowe zaburzenie struktury sieci pozwala czasem na ucieczkę z minimum lokalnego i osiągnięcie lepszego rozwiązania w następnych krokach uczenia. Kryterium „cierpliwości” wahało się od 50 000 do 100 000 iteracji. Maksymalna ilość iteracji wahała się od 100 000 do 5 000 000. Wszystkie sieci posiadały jednostkową wartość epoki. Miarą błędu używaną do weryfikacji modeli neuronowych było RMSE (*root mean squared error*) obliczane wg równania 25. Błąd w czasie uczenia (*on-line error*) obliczany na zbiorze uczącym, a dokładniej na jego niezasumionej części, był kryterium poprawności nauczania danej sieci neuronowej. Kryterium selekcji optymalnej architektury był błąd generalizacji obliczany na zbiorach testowych nie biorących udziału w procesie uczenia sieci, zgodnie z metodą 10-krotnego wzajemnego sprawdzania.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (s_i - s_i^{\text{wzorzec}})^2}{n}} \quad (25)$$

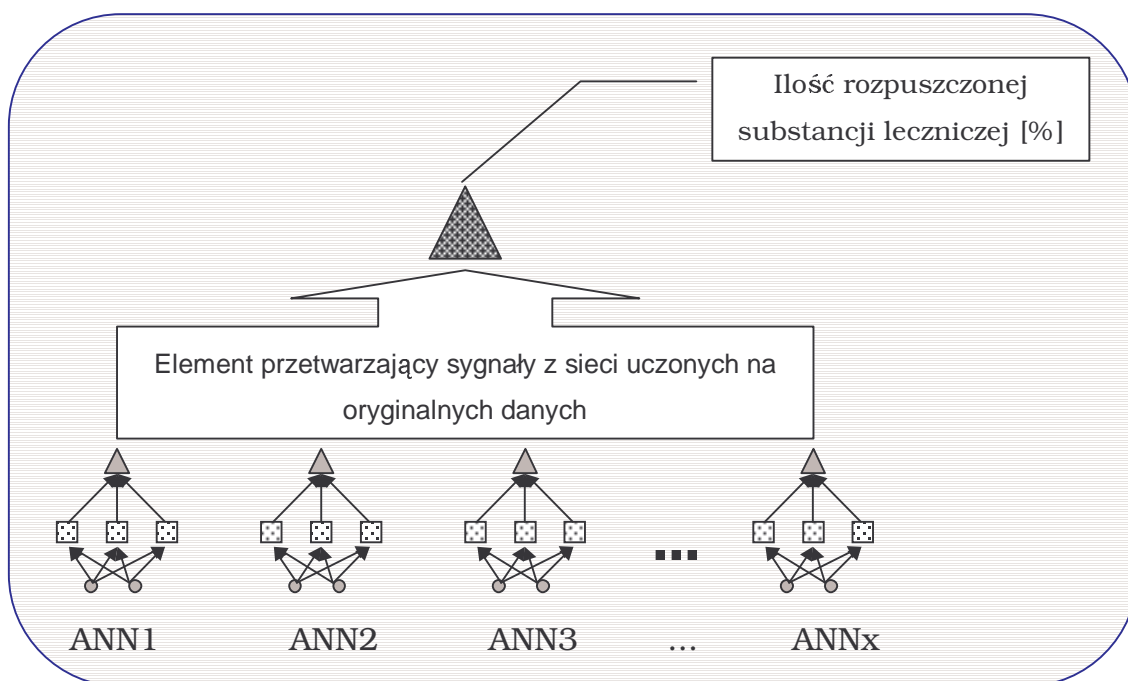
gdzie:

s_i – aktualne wyjście sieci neuronowej

s_i^{wzorzec} – zadane wyjście sieci neuronowej

n – liczba rekordów danych

Optymalne architektury poddawano analizie wrażliwościowej w celu ewentualnej redukcji wektora wejściowego. Zredukowane modele uczone ponownie, aby ocenić czy redukcja nie była nadmierna i czy nie spowodowała utraty precyzji modelu neuronowego.



Rycina 10. Schemat komitetu ekspertów. ANN – sztuczna sieć neuronowa.

Zaprojektowano również systemy drugiego rzędu, tzw. komitety ekspertów, które bazowały na wyjściach sieci uczonej na danych oryginalnych (Rycina 10).

Analiza wyjść zestawu sieci stanowiącego komitet ekspertów prowadzona była w dwojaki sposób:

- a) uśrednienie wyjść wszystkich sieci
- b) podanie wyjść sieci neuronowych uczonych na oryginalnych danych na wejścia sieci drugiego rzędu, która po wytrenowaniu przewiduje procent rozpuszczonej substancji leczniczej

3.1.5. BUDOWA INTERFEJSU UŻYTKOWNIKA

Zaprojektowano i napisano w języku Delphi® system **SD_expert** (Rycina 43 str. 167) jako interfejs pomiędzy użytkownikiem a systemem neuronowym. System ten wykorzystuje część kodu **Nets2003** i jest przeznaczony do odczytywania formatu plików sieci neuronowych zapisywanego przez **Nets2003**. Aktualnie liczy 12 000 linii kodu. Jako podstawy systemu użyto najlepsze modele neuronowe wyselekcjonowane w czasie badań. Istotą działania **SD_expert** jest testowanie modeli neuronowych poprzez zadawanie pytań systemom neuronowym. Program wyposażono w funkcje wizualizacji pracy w postaci wykresów liniowych, a także zapisywania wyników do plików tekstowych. **SD_expert** posiada wbudowaną funkcję przeszukiwania przestrzeni możliwych rozwiązań w celu znalezienia optymalnej formułacji. Kryteria optymalności formułacji definiuje użytkownik – mogą one dotyczyć szybkości uwalniania substancji leczniczej z formułacji, składu lub technologii przygotowania danej formułacji stałego rozproszenia. Poszukiwanie optymalnej formułacji odbywa się w sposób iteracyjny i całkowicie automatyczny.

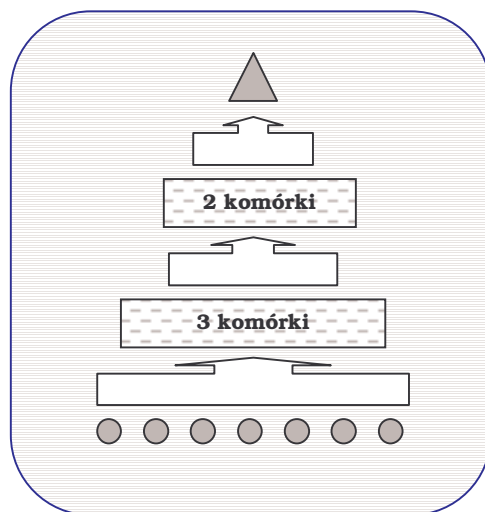
Przeprowadzono eksperyment numeryczny, który polegał na użyciu systemu **SD_expert** do symulowanego poszukiwania optymalnej formułacji stałego rozproszenia i mieszaniny fizycznej. Wykorzystano w tym celu komitet ekspertów uczony z użyciem jednego ze zbiorów danych wchodzącego w skład 10 par zbiorów uczący-testowy zastosowanych w metodzie 10-krotnego wzajemnego sprawdzania. Kryterium poszukiwań były dane dotyczące formułacji stałych rozproszeń oraz mieszanin fizycznych papaweryny, które pochodziły ze zbioru testowego. Oznacza to, że system

SD_expert musiał generalizować wyniki poza swoją bazę wiedzy. Zadaniem systemu było znalezienie optymalnego składu ilościowego nowych formułacji względem zadanych profili rozpuszczania i składów jakościowych układów z papaweryną.

3.1.6. WYNIKI BADAŃ

MODELOWANIE PROFILU ROZPUSZCZANIA KETOPROFENU ZE STAŁYCH ROZPROSZEŃ I MIESZANIN FIZYCZNYCH Z NOŚNIKAMI O RÓŻNEJ BUDOWIE CHEMICZNEJ

W toku eksperymentów numerycznych wyselekcjonowano optymalną architekturę sieci neuronowej typu MLP 3_2 hid tanh (Rycina 11). Jako funkcję aktywacji sieć ta posiadała tangens hiperboliczny, a jako funkcję kosztu funkcję Hampela. Sieć uczono do 300 000 iteracji.



Rycina 11. Optymalna architektura sieci neuronowej typu MLP (3_2 hid tanh).

Najlepsze wyniki modelowania za pomocą sieci rozmytych otrzymano za pomocą algorytmu *lookup table*, który automatycznie dobiera liczbę reguł i ich parametry. Z uwagi na to nie jest możliwe określenie jednolitej architektury optymalnej sieci rozmytej, gdyż dla innej pary zbiorów testowych algorytm *lookup table* stworzył inne architektury – liczba reguł rozmytych wahała się od 40 do 50.

Wyniki zebrane w tabeli 7 wskazują, że sieci rozmyte można traktować jako równoważne optymalnej architekturze MLP. Świadczy o tym nieistotna statystycznie różnica błędów RMSE otrzymanych w wyniku testowania obydwu rodzajów sieci neuronowych w procedurze 25-krotnego wzajemnego sprawdzania (Tabela 7).

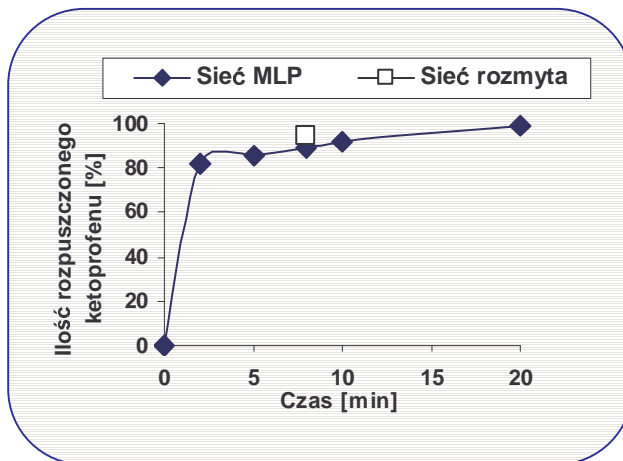
Tabela 7. Błędy RMSE najlepszych sieci neuronowych modelu profilu rozpuszczania ketoprofenu z badanych układów.

Sieć neuronowa	RMSE
3_2 hid tanh	11,8
Sieć rozmyta	11,6

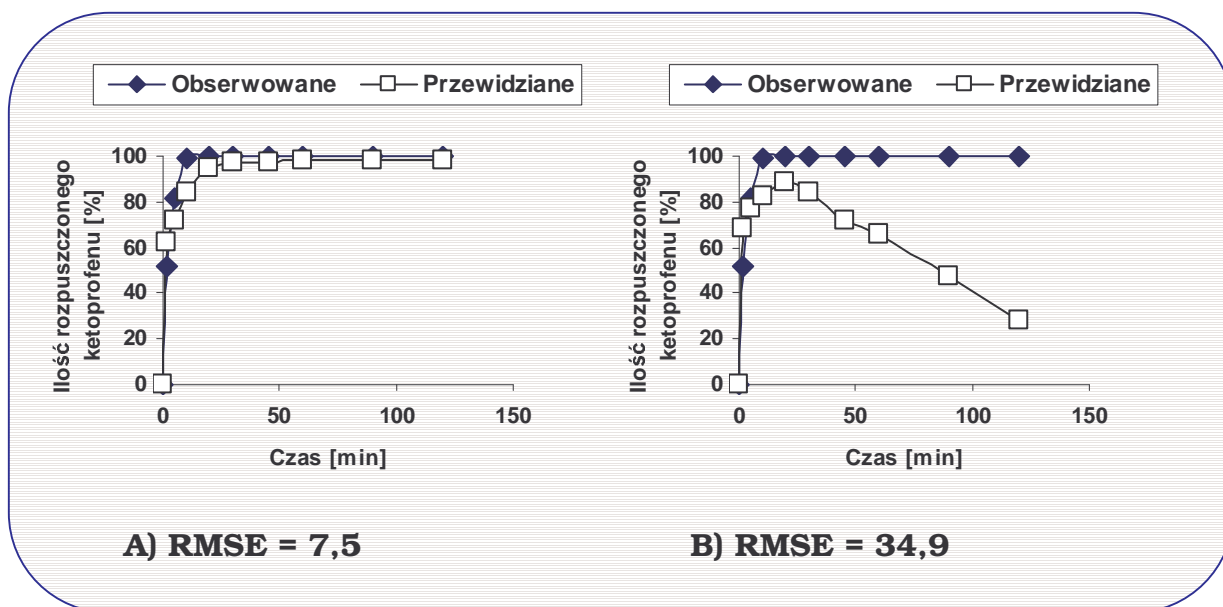
Obydwa modele zostały wytrenowane na danych zawierających punkty czasowe testu rozpuszczania od 0 do 20 minut. Przeprowadzono eksperyment polegający na testowaniu optymalnych architektur z wykorzystaniem zbioru danych opisującego profil rozpuszczania ketoprofenu z nieznaną siecią formułacji w zakresie punktów czasowych od 0 do 120 minut. Jako formułację testową wybrano stałe rozproszenie ketoprofen + PVP K30 (1+9). Porównanie obserwowanych profili rozpuszczania ketoprofenu z przewidzianymi przez oba typy sieci neuronowych przedstawiono na rycinie 12.

Zdolność predykcyjna to jedna z najistotniejszych własności modeli neuronowych umożliwiająca ich wykorzystanie do wspomagania procesu formułacji. Jednak oprócz bezpośredniego przewidywania zachowania się nieznaną formułacji można również analizować sam model neuronowy pod kątem reguł opisujących badany problem. Modele rozmyte nadają się bezpośrednio do takiej analizy z uwagi na swoją strukturę, którą można bezpośrednio interpretować jako zestaw reguł logicznych. Dzięki analizie wybranej sieci rozmytej znaleziono regułę opisującą szybkie rozpuszczanie ketoprofenu ze stałego rozproszenia w Makroglolu 9000 w stosunku 0,3+9,7.

Formulacja ta nie była obecna w zbiorze uczącym, wobec czego analiza reguł logicznych doprowadziła w tym przypadku do ekstrapolacji.



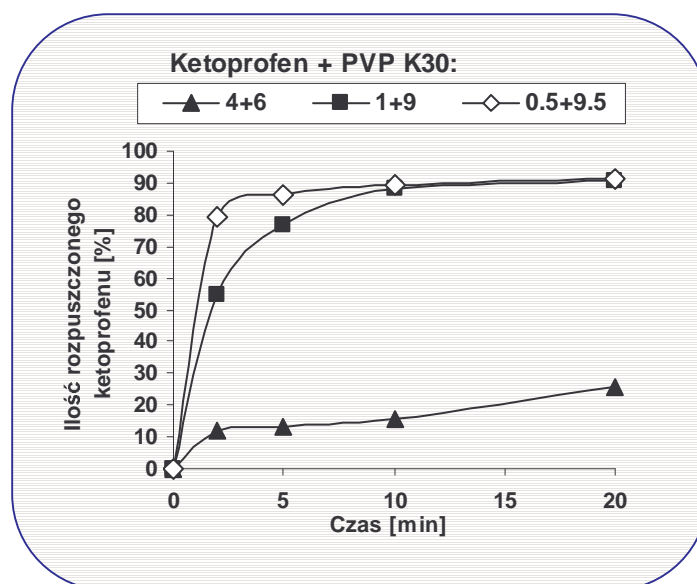
Rycina 12. Porównanie zdolności ekstrapolacyjnych modeli sieci MLP i rozmytych na przykładzie predykcji szybkości rozpuszczania ketoprofenu ze stałego rozproszenia ketoprofen + PVP K30 (1+9).



Rycina 13. Weryfikacja reguły logicznej utworzonej automatycznie przez system oparty na logice rozmytej za pomocą sieci MLP.

Weryfikacja poprawności tej reguły polegała na podaniu danych opisujących ww. stałe rozproszenie sieci typu MLP, która przewidziała podobny stopień rozpuszczenia ketoprofenu w zadanym czasie 8 min. Wyniki powyższych eksperymentów przedstawiono na rycinie 13.

Mimo, że sieci MLP posiadają charakter „czarnej skrzynki”, możliwe jest również znalezienie pewnych reguł jakie modele te tworzą do opisu badanego problemu. Ekstrakcji reguł logicznych można dokonać poprzez specyficzną konstrukcję zbioru testowego. Wykonano eksperyment z podaniem wytrenowanej sieci MLP o optymalnej architekturze numerycznie stworzonych danych opisujących trzy stałe rozproszenia ketoprofenu z PVP o zmieniających stosunkach ilościowych substancji leczniczej do nośnika: 4+6, 1+9 i 0,5+9,5. Profile rozpuszczania ketoprofenu wygenerowane przez sieć 3_2 hid tanh dla poszczególnych formułacji testowych przedstawiono na rycinie 14. Model neuronowy przewidział wzrost szybkości rozpuszczania ketoprofenu wraz ze zmniejszeniem jego zawartości w stałym rozproszeniu.



Rycina 14. Profile rozpuszczania ketoprofenu z symulowanych numerycznie formułacji stałych rozproszeń z PVP K30.

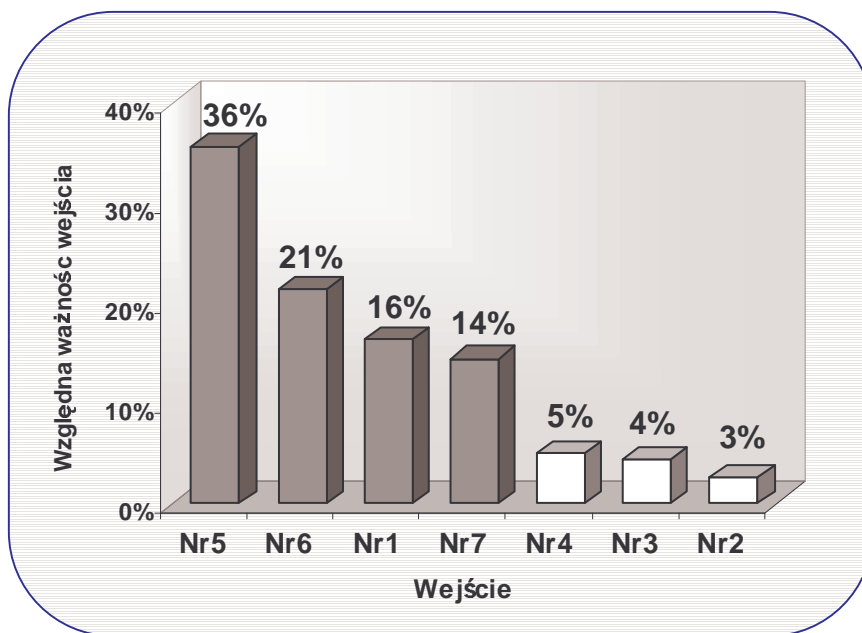
Innym sposobem analizy modelu neuronowego jest analiza wrażliwościowa, która pozwala określić zmienne kluczowe dla danego problemu.

Wykonano analizę wrażliwościową optymalnej architektury MLP. Wyniki przedstawione na rycinie 15 pozwalają na ograniczenie liczby wejść sieci neuronowej. Procedura usuwania wejść polega na znalezieniu największego gradientu względnej ważności wejść. Analizując względne ważności wejść nr 7 i nr 4 można dojść do wniosku, że jest to miejsce największego spadku względnej ważności wejść. Wszystkie wejścia posiadające względną ważność mniejszą od wejścia nr 4 zostały usunięte z wektora wejściowego sieci neuronowej, co doprowadziło do redukcji liczby wejść sieci neuronowej z pierwotnych siedmiu do czterech.

Wejścia zredukowanego modelu wydają się być zmiennymi kluczowymi zależności profilu rozpuszczania ketoprofenu od składu i rodzaju formułacji (Tabela 8). Najistotniejsze okazały się: rodzaj formułacji, zawartość w nich substancji leczniczej, rodzaj nośnika oraz punkty czasowe. Czteroelementowy wektor wejściowy stał się podstawą do wytrenowania modeli neuronowych o takiej samej architekturze i parametrach uczenia jak poprzednio otrzymane optymalne modele neuronowe. Jediną różnicę stanowi liczba wejść.

Tabela 8. Wektor wejściowy zredukowany po analizie wrażliwościowej.

Nr wejścia	Kodowana informacja
1	Rodzaj formułacji: stałe rozproszenie lub mieszanina fizyczna
2	Ilość substancji leczniczej w 1g formułacji
3	Indeks konektywności (CI) nośnika
4	Czas w minutach



Rycina 15. Wyniki analizy wrażliwościowej optymalnej architektury MLP.

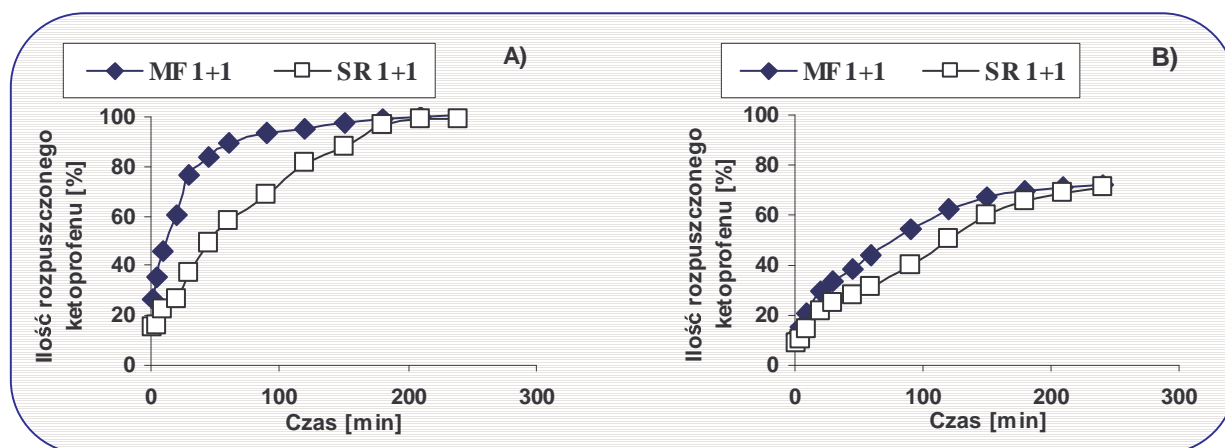
Błędy generalizacji przedstawione w tabeli 9 mogą świadczyć o tym, że sieć typu MLP wykazuje znacznie mniejszy błąd niż przed redukcją wektora wejściowego. Tak więc paradoksalnie: mniejsza ilość informacji na wejściu sieci pozwala otrzymać bardziej precyzyjny model. Takie relacje błędów pomiędzy modelami o różnej ilości wejść mogą być związane z zasadą „przekleństwa wymiarowości” (*curse of dimensionality*) [10, 57]. Sieci rozmyte tym razem okazały się być gorsze niż MLP chociaż różnica błędów nie jest istotna statystycznie.

Tabela 9. Błędy RMSE najlepszych sieci neuronowych po redukcji wymiaru wektora wejściowego.

Sieć neuronowa	RMSE
Optymalna sieć MLP	8,6
Sieć rozmyta	9,2

W celu wykazania skuteczności tak otrzymanego modelu neuronowego przetestowano zredukowaną sieć MLP na zbiorze danych zawierającym zupełnie nieznaną nośnik jakim była hydroksypropylometyloceluloza (HPMC). Prace doświadczalne prowadzone nad otrzymaniem stałego rozproszenia ketoprofenu z HPMC wykazały, że nośnik ten nie jest odpowiedni do formułowania stałych rozproszeń z ketoprofenem. Wyrażało się to m.in. wolniejszym rozpuszczaniem ketoprofenu ze stałego rozproszenia w porównaniu do odpowiedniej mieszaniny fizycznej. Sieć neuronowa prawidłowo przewidziała obserwowaną zależność pomiędzy rodzajem formułacji a szybkością rozpuszczania ketoprofenu (Rycina 16).

Przeprowadzono również uczenie sieci neuronowych na zbiorach przetwarzanych z udziałem systemu **PKB 2.0**. Eksperyment ten miał na celu potwierdzenie założenia o wpływie neuronowego wstępnego przetwarzania (*preprocessing*) danych na jakość modelu rozpuszczania ketoprofenu. Uczenie przeprowadzono z użyciem optymalnej architektury sieci MLP. Wyniki przedstawiono w tabeli 10. Analiza wariancji wskazuje na istotną statystycznie różnicę średnich błędów RMSE dla $\alpha=0,1$.

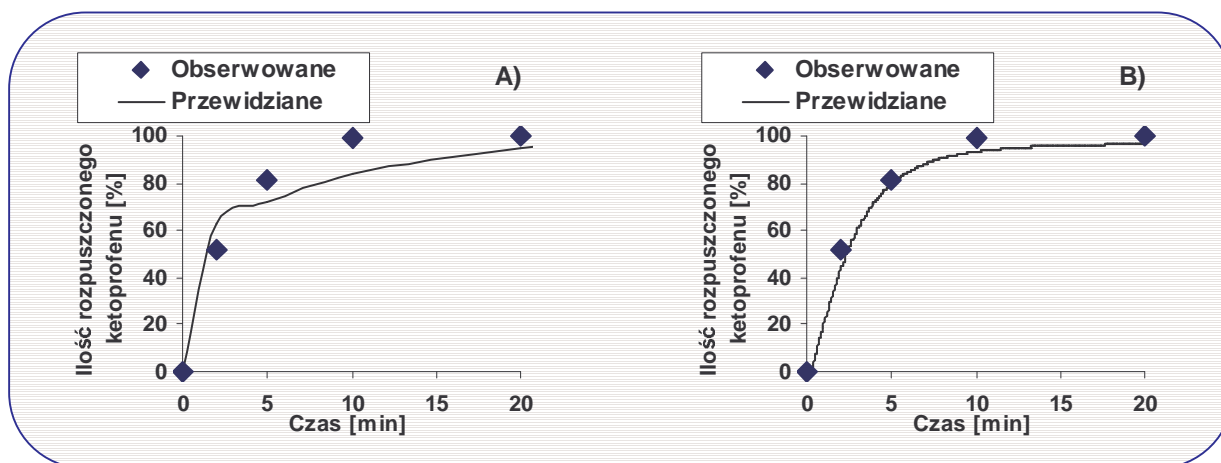


Rycina 16. Porównanie profilu rozpuszczania ketoprofenu z mieszaniny fizycznej (MF) i stałego rozproszenia (SR). (A) dane doświadczalne, (B) symulowane przez sieć neuronową.

Tabela 10. Wyniki testowania architektury 3_2 hid tanh uczonej na oryginalnych zbiorach danych i z wykorzystaniem neuronowego wstępnego przetwarzania.

Zbiór uczący	Zbiór testowy	
	Z wstępnym przetwarzaniem danych	Bez wstępnego przetwarzania danych
Z wstępnym przetwarzaniem danych	8,1	8,7
Bez wstępnego przetwarzania danych	12,8	8,6

Poprawę zdolności interpolacyjnych sieci MLP po nauczaniu na danych z wstępnym przetwarzaniem zilustrowano na rycinie 17. Obserwowana gładkość profilu rozpuszczania ketoprofenu pomiędzy punktami czasowymi dostępnymi w danych oryginalnych potwierdza pozytywny wpływ wstępnego przetwarzania danych na jakość modelu neuronowego (Rycina 17B).



Rycina 17. Porównanie predykcji profilu rozpuszczania ketoprofenu ze stałego rozproszenia otrzymanej przez optymalną sieć MLP uczoną na danych oryginalnych (A) i z neuronowym wstępnym przetwarzaniem (B).

MODEL ROZPUSZCZANIA SUBSTANCJI LECZNICZYCH O RÓŻNEJ ROZPUSZCZALNOŚCI ZE STAŁYCH ROZPROSZEŃ I MIESZANIN FIZYCZNYCH Z NOŚNIKAMI O ZRÓŻNICOWANEJ BUDOWIE CHEMICZNEJ

Pierwotnym założeniem badań było przedstawienie sieci wszystkich dostępnych danych w celu stworzenia przez nią modelu profilu rozpuszczania dla dowolnej substancji leczniczej. Użyto tutaj 16-elementowego wektora wejściowego (Tabela 5). Nie udało się jednak otrzymać reprezentatywnego modelu neuronowego, gdyż minimalny błąd RMSE jaki się udało uzyskać wynosił ponad 19. Wynik ten należy przypisać prawdopodobnie obecności w bazie danych zarówno stałych rozproszeń z substancjami leczniczymi trudno- jak też i dobrze rozpuszczalnymi w wodzie.

W celu poprawy jakości modelu neuronowego dodano jeszcze jedną zmienną do wektora wejściowego klasyfikującą substancje lecznicze jako dobrze lub trudno rozpuszczalne (zmienna binarna) w wyniku czego ostatecznie otrzymano 17-elementowy wektor wejściowy (Tabela 11).

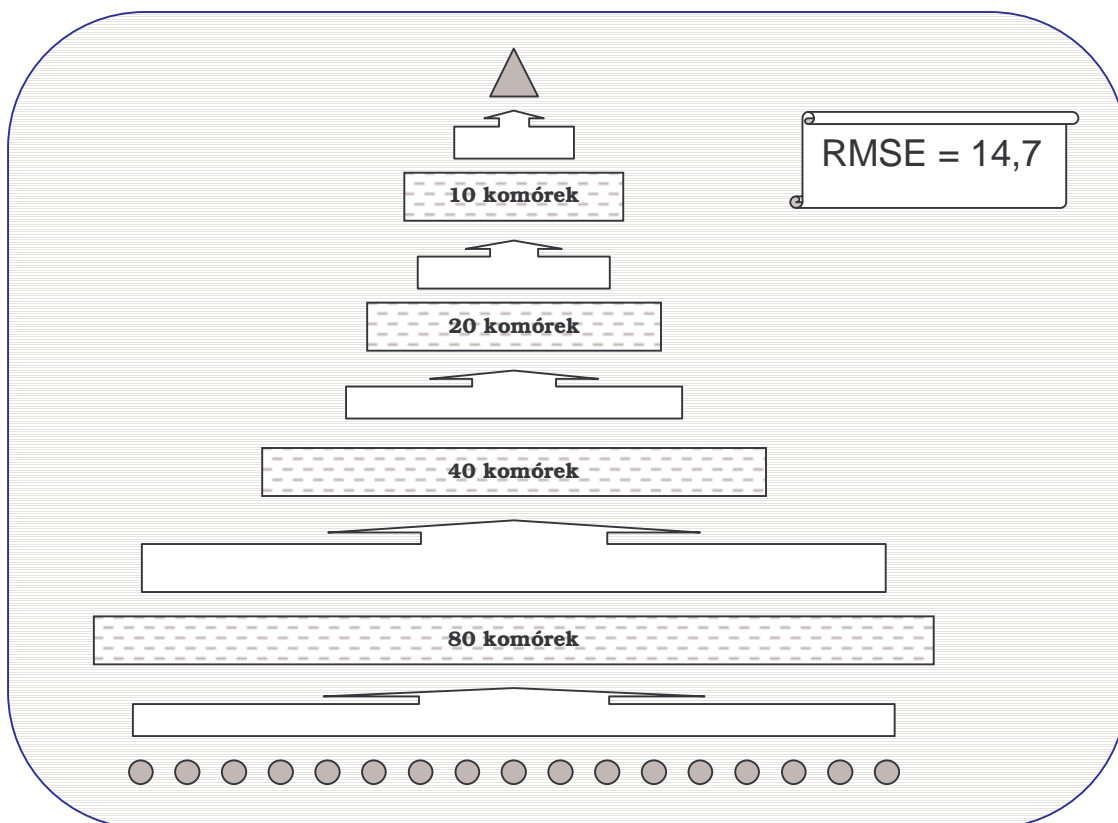
Należy wspomnieć, iż na żadnym etapie eksperymentów nie udało się zredukować wymiaru wektorów wejściowych poprzez analizę wrażliwościową, jak to miało miejsce w przypadku modelu rozpuszczania ketoprofenu. Można wnioskować, iż zmienne sterujące, które były prezentowane modelom neuronowym, są niezbędne do opisu zagadnienia.

Dodanie zmiennej klasyfikującej rozpuszczalność substancji leczniczej

Otrzymano uniwersalny model rozpuszczania substancji leczniczych ze stałych rozproszeń o najlepszych jak dotąd własnościach predykcyjnych. Poszukiwania najlepszej architektury sieci neuronowej zgodnie z metodą 10-krotnego wzajemnego sprawdzania pozwoliły wyselekcjonować sieć o skomplikowanej architekturze 80_40_20_10 hid tanh_5mln_scaled+- uczoną do 5 000 000 iteracji. Architekturę optymalnej sieci i jej błąd generalizacji przedstawiono na rycinie 18.

Tabela 11. 17-elementowy wektor wejściowy neuronowych modeli rozpuszczania substancji leczniczych ze stałych rozproszeń.

Nr wejścia	Kodowana informacja
1	Masa cząsteczkowa substancji leczniczej
2	Indeks konektywności (CI) substancji leczniczej
3	Rozpuszczalność substancji leczniczej w wodzie (rozpuszczalna/nierozpuszczalna)
4	Rodzaj formułacji: stałe rozproszenie lub mieszanina fizyczna
5	Stałe rozproszenie sporządzone z mieszaniny fizycznej czy nie
6	Metoda sporządzania: stapiania, odparowania, suszenia rozpyłowego
7	Warunki chłodzenia: w temperaturze pokojowej lub na łaźni lodowej
8	Ilość substancji leczniczej w 1g formułacji
9	Zawartość nośnika nr 1 w 1g formułacji
10	Zawartość nośnika nr 2 w 1g formułacji
11	CI nośnika nr 1
12	CI nośnika nr 2
13	Masa cząsteczkowa nośnika nr 1
14	Masa cząsteczkowa nośnika nr 2
15	Ilość obrotów łopatki / koszyczka
16	pH medium w teście rozpuszczania (<i>dissolution test</i>)
17	Czas rozpuszczania w minutach



Rycina 18. Optymalna architektura sieci uniwersalnego modelu rozpuszczania różnych substancji leczniczych (80_40_20_10 hid tanh_5mln_scaled+-).

W celu poprawy wyniku uzyskanego przez optymalną architekturę sieci neuronowej użyto 10 najlepszych architektur do stworzenia komitetu ekspertów:

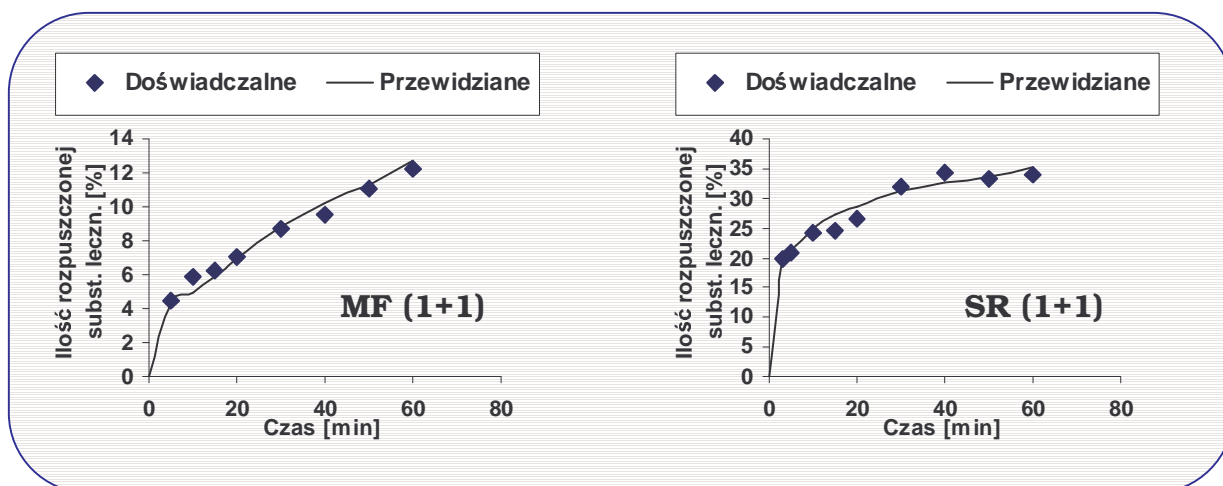
1. 15_7_5 hid tanh_5mln
2. 20_16_8_4 hid tanh_5mln
3. 40_10_5 hid tanh_5mln
4. 60_20_8 hid tanh_10mln
5. 60_20_8 hid tanh_5mln_scaled+- (dane skalowane od -0,8 do 0,8)
6. 80_20_10 hid tanh_10mln_scaled+-
7. 80_20_10 hid tanh_5mln
8. 80_40_20_10 hid tanh_10mln
9. 80_40_20_10 hid tanh_5mln_scaled+-
10. 80_40_20_10 hid fsr_1mln_scaled+-

W wyniku użycia komitetu ekspertów po uśrednieniu wyników sieci uczonych na danych oryginalnych uzyskano wartości RMSE = 13,4 czyli o 1,3 jednostki mniejsze niż najlepszej, pojedynczej sieci neuronowej. Użycie innej sieci neuronowej w roli elementu przetwarzającego wyjścia sieci uczonych na danych oryginalnych nie wpłynęło na poprawę tego wyniku. Opisany powyżej komitet ekspertów stanowi podstawę działania systemu **SD_expert** służącego do wspomagania formulacji stałych rozproszeń.

Symulowano użycie systemu **SD_expert** do poszukiwania optymalnych formulacji stałych rozproszeń (SR) i mieszanin fizycznych (MF). Jako przykłady wybrano układy papaweryny (Pap) z Makrogolem 6000 i Makrogolem 20000 (0,5+0,95 oraz 1+1). Przykłady formulacji przedstawione zostały na rycinie 19. Przewidziane przez **SD_expert** składy ilościowe były zbliżone do danych doświadczalnych w 6 przypadkach na 8 testowanych formulacji (Tabela 12).

Tabela 12. Porównanie doświadczalnych składów ilościowych wybranych formulacji z przewidzianymi przez system *SD_expert*.

Formulacja	Skład ilościowy (subst. leczn. + nośnik)	
	Doświadczalne	Znalezione
MF Pap+PEG 6000	1+1	1+1 lub 1+1,5
SR Pap+PEG 6000	1+1	1+2,3 lub 1+9
MF Pap+PEG 6000	0,5+9,5	2+8 lub 7,5+2,5
SR Pap+PEG 6000	0,5+9,5	0,5+9,5 lub 1,5+8,5
MF Pap+PEG 20000	1+1	1+1 lub 1+0,9
SR Pap+PEG 20000	1+1	1+1,2 lub 1,9+1
MF Pap+PEG 20000	0,5+9,5	1+9 lub 4+6
SR Pap+PEG 20000	0,5+9,5	0,5+9,5 lub 8,3+1,7



Rycina 19. Wyniki przewidywania profilu rozpuszczania papaweryny z układów z Makrogolem 6000 przez system *SD_expert*. MF – mieszanina fizyczna. SR – stałe rozproszenie.

Przeprowadzono również eksperyment z uczeniem kilku najlepszych architektur metodą *leave-one-out*. Wyniki przedstawiono w tabeli 13.

Tabela 13. Wyniki uczenia kilku architektur metodą *leave-one-out*.

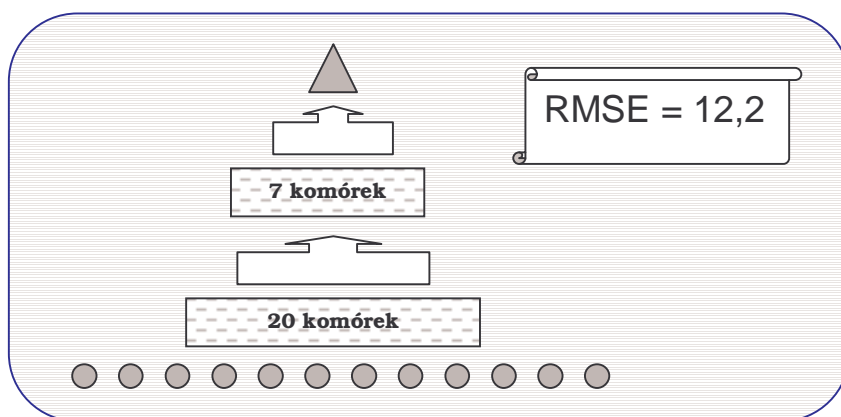
Architektura	RMSE
20_16_8_4 hid tanh_5mln	10,8
40_10_5 hid tanh_5mln	12,3
60_20_8 hid tanh_5mln	11,5
80_20_10 hid tanh_5mln	10,7
80_40_20_10 hid fsr_1mln_scaled+-	11,9

Uczenie metodą *leave-one-out* pozwoliło na ocenę zdolności modeli neuronowych do generalizacji składu ilościowego stałego rozproszenia. Wynika to z obecności w każdym zbiorze uczącym informacji o wszystkich substancjach leczniczych, gdyż podczas tworzenia par zbiorów uczący-testowy za każdym razem wyłączana była tylko jedna formuła. W celu oceny zdolności do generalizacji składu jakościowego dokonano podziału

zbioru bazowego na 18 par zbiorów uczący-testowy. Każdy zbiór testowy zawierał dane dotyczące formułacji z jedną tylko substancją leczniczą, a odpowiadający mu zbiór uczący został całkowicie pozbawiony informacji o danej substancji leczniczej. Tak przygotowane zbiory danych posłużyły do wytrenowania zespołów sieci składających się na opisany powyżej komitet ekspertów. Przeprowadzono 18-krotne wzajemne sprawdzanie. W wyniku testowania wytrenowanych komitetów ekspertów otrzymano wartość błędu $RMSE = 35,9$. Dodatkowy eksperyment zmierzający do oceny zdolności generalizacyjnej systemu neuronowego polegał na dodaniu do zbioru uczącego jednego rekordu ze zbioru testowego. Wyniki generalizacji tak nauczonych sieci neuronowych są lepsze ($RMSE = 27,1$), chociaż spadek błędu nie jest istotny statystycznie.

MODEL ROZPUSZCZANIA TRUDNO ROZPUSZCZALNYCH SUBSTANCJI LECZNICZYCH Z UKŁADÓW ZAWIERAJĄCYCH MAKROGOL

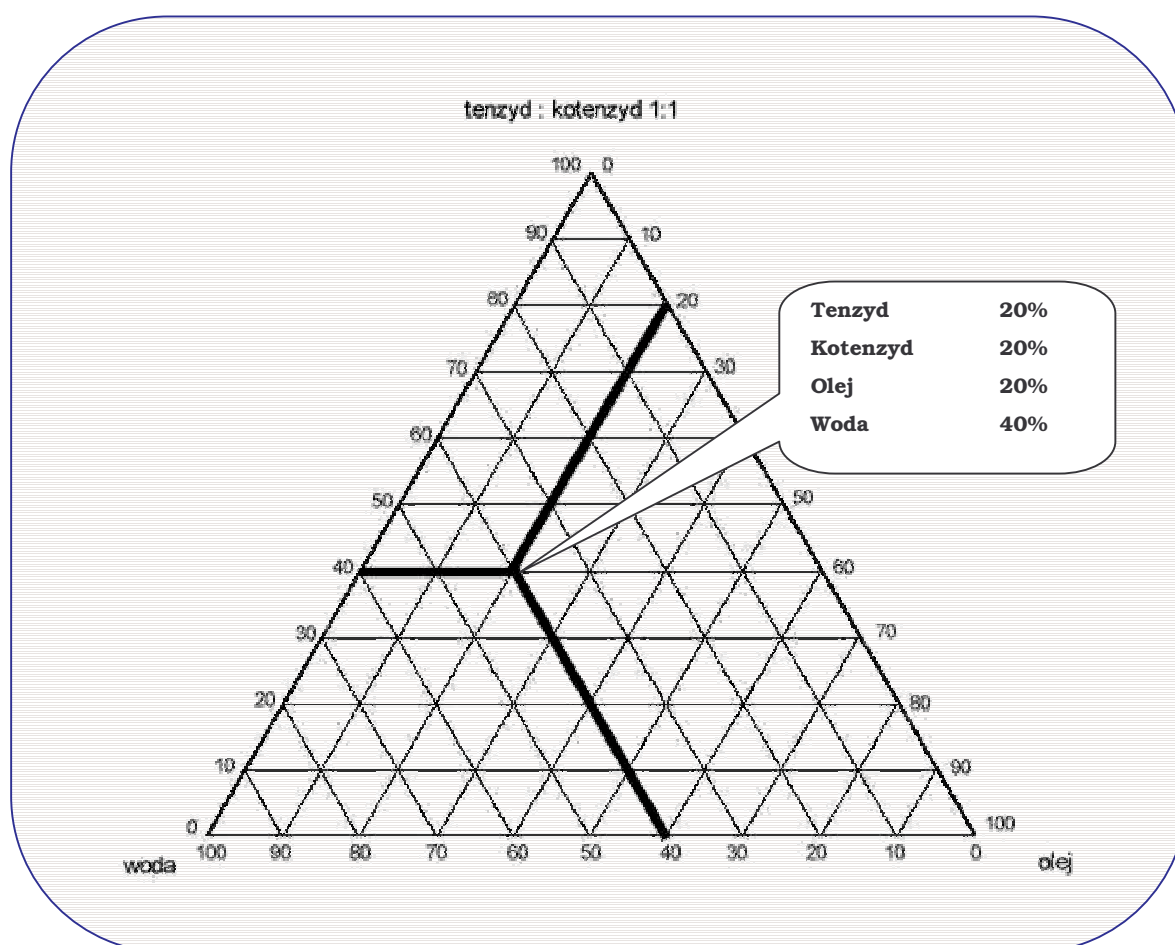
Równoległe z modelami całościowymi, opisanymi powyżej, zaprojektowano model profilu rozpuszczania słabo lub praktycznie nierozpuszczalnych w wodzie substancji leczniczych ze stałych rozproszeń formułowanych z nośnikami należącymi do grupy polioksyetylenoglikoli. Na rycinie 20 przedstawiono optymalną architekturę 20_7 hid tanh, której błąd $RMSE$ otrzymany w procedurze 10-krotnego wzajemnego sprawdzania jest mniejszy niż w przypadku komitetu ekspertów uniwersalnego modelu profilu rozpuszczania substancji leczniczych rozpuszczalnych i nierozpuszczalnych w wodzie ze stałych rozproszeń i mieszanin fizycznych ($RMSE = 13,4$). Tak więc model bardziej wyspecjalizowany jest też bardziej precyzyjny.



Rycina 20. Optymalna architektura sieci neuronowej dla modelu rozpuszczania różnych substancji leczniczych ze stałych rozproszeń zawierających Makrogol.

3.2. MIKROEMULSJE

Mikroemulsje to izotropowe, termodynamicznie trwałe układy fazy wodnej, olejowej oraz od jednego do trzech tenzydów [23, 32, 74, 90, 105]. Powstają spontanicznie po zmieszaniu składników w odpowiednich proporcjach. Są to przezroczyste lub lekko opalizujące ciecze, albo żele, które znajdują coraz większe zastosowanie jako postaci leku [11, 64]. Zainteresowanie mikroemulsjami jako postaciami leku wiąże się z ich trwałością oraz zdolnością do solubilizacji i przedłużania, bądź kontroli uwalniania substancji leczniczej. Obszar mikroemulsji przedstawiany jest zwykle na tzw. diagramach Gibbsa, które są wykresami w trójkątnym układzie współrzędnych: olej, woda, tenzydy.



Rycina 21. Przykładowy diagram fazowy.

Przykładowo na rycinie 21 przedstawiono diagram fazowy z zaznaczonym punktem o składzie mikroemulsji: 20% tenzydu, 20% kotenzydu, 20% oleju i 40% wody.

3.2.1. ZAKRES BADAŃ

Zbiór danych stanowiły wyniki prac doświadczalnych prowadzonych w Katedrze Technologii Postaci Leku i Biofarmacji CMUJ w Krakowie oraz dane z piśmiennictwa. Wyselekcjonowano 118 formułacji, które opisane zostały jako układy izotropowe (Tabela 14) i stały się podstawą do konstrukcji bazy wiedzy systemu neuronowego. Z uwagi na to, że pojęcie mikroemulsji jest różnie definiowane przez autorów, przyjęto na potrzeby niniejszej pracy pojęcie mikroemulsji równoznaczne z roztworem micelarnym nie będącym ciekłym kryształem.

3.2.2. ZDEFINIOWANIE FUNKCJI SYSTEMU NEURONOWEGO

Zadaniem sieci neuronowych było przewidywanie wystąpienia układu izotropowego dla danego składu jakościowego i ilościowego substancji pomocniczych.

Tabela 14, Zestawienie składów jakościowych mikroemulsji w bazie wiedzy.

Faza olejowa	Tenzyd	Kotenzyd	Písmien-nictwo
Kwas olejowy, oleinian etylu, olej sojowy, Miglyol® 812, kwas oktanowy, oktanian etylu	Ovothin® 200	n-butanol, n-propanol	[2]
Metakrylan metylu	Laurylosiarczan sodowy	2-hydroksyetylo-akrylan metylu	[127]
Mirystynian izopropylu	Tween® 60	n-butanol	[75]
Mirystynian izopropylu	Epikuron® 200 (lecytyna sojowa), Ovothin® 200 (lecytyna z jaj kurzych)	Etanol	[110]
Mirystynian izopropylu	Tween® 40	Sorbitol	[9]
Mirystynian izopropylu	Epikuron® 200, Ovothin® 200	n-propanol, izopropanol, n-butanol, sec-butanol, izobutanol, tert-butanol, n-pentanol	[1]
Myritol® 318 Mirystynian izopropylu	Epikuron® 200, Epikuron® H100, „butanoilo-lecytyna” (m-PC4), „heksanoilo-lecytyna” (m-PC6), „oktanoilo-lecytyna” (m-PC8), „dekanoilo- lecytyna” (m-PC10), „heksadekanoilo- lecytyna” (m-PC16)	Etanol	[123]

Tabela 14 c. d.

Faza olejowa	Tenzyd	Kotenzyd	Písmien -nictwo
Mirystynian izopropylu	Epikuron® 200	n-butanol, sec-butanol, tert-butanol, n-pentanol, kwas n-heksanowy, kwas n-pentanowy, sec-butyloamina, tert-butyloamina, 2-aminopentan, 3-aminopentan, 1,2-butanodiol, 1,2-pentanodiol, 1,2-heksanodiol, eter monobutylowy glikolu dietylenowego, eter monopentylowy glikolu dietylenowego, eter monoheksylowy glikolu dietylenowego, n-heksanol, 2-pentanol, 3-pentanol	[3]
Olej sojowy, Captex® 355	Arlacel® 80, Arlacel® 83, Arlacel® 186, Myvero® 18-99, Tween® 80, Capmul® MCM	-	[29]
Parafina ciekła	Brj® 96	Gliceryna, glikol etylenowy, glikol propylenowy	[92]
Captex® 300	monooleinian heksaglicerolu (MO 500) monooleinian dekaglicerolu (MO 750), sesquiolenian dekaglicerolu (SO 750), monolaurynian tetraglicerolu (ML 310)	Etanol, n-butanol, n-propanol	[56]

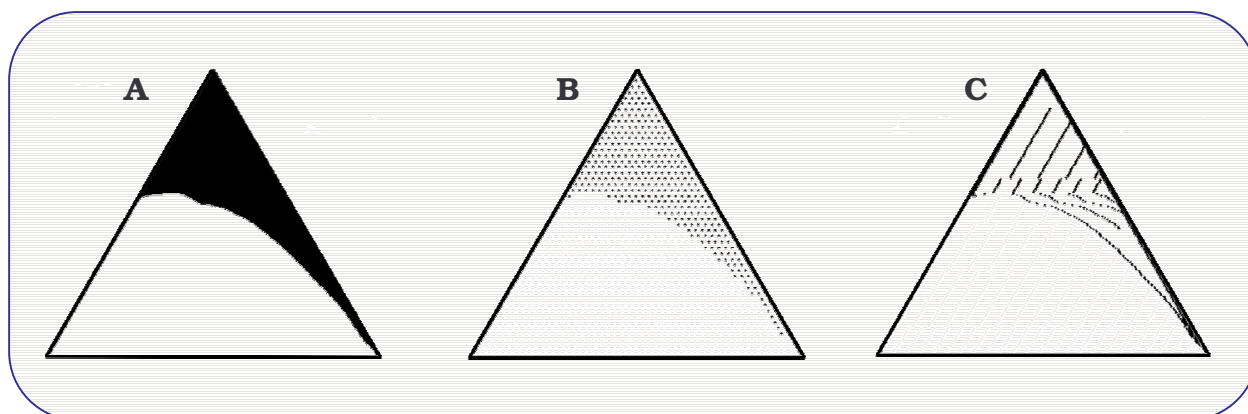
3.2.3. PRZYGOTOWANIE DANYCH STANOWIĄCYCH BAZĘ WIEDZY

Z uwagi na to, że układy fazowe prezentowane są w postaci graficznej jako trójkąty Gibbsa, do digitalizacji trójkątów fazowych użyto zaprojektowanego specjalnie w tym celu programu **Phase_diagr**.

Zastosowano dwa systemy digitalizacji trójkątów fazowych:

- a) skanowanie równomierne z krokiem co 2% zawartości danego składnika
- b) skanowanie z kompresją za pomocą napisanego specjalnie w tym celu algorytmu **Intelliscan** zaimplementowanego w programie **Phase_diagr**

Na rycinie 22 zobrazowano różnice w algorytmach skanowania trójkątów fazowych.



Rycina 22. Digitalizacja trójkątów fazowych. A - oryginalna bitmapa; B - skanowanie równomierne co 2%; C - Intelliscan. Kolor czarny oznacza obszar mikroemulsji.

Skanowanie równomierne pozwoliło na stworzenie bazy danych, której całkowity rozmiar wynosił około 130 000 rekordów danych. Algorytm **Intelliscan** zaprojektowano tak, aby przy największym nawet współczynniku kompresji wykrywał granice przejść fazowych i umieszczał je w plikach wynikowych. Z tego względu, mimo wysokiej, ustalonej arbitralnie wartości

współczynnika kompresji (50), rozmiar bazy danych wyniósł ok. 65 000 rekordów.

Tabela 15. Wektor wejściowy modeli neuronowych składu mikroemulsji.

Nr wejścia	Kodowana informacja
1	Indeks konektywności (CI) kotenzydu
2	Indeks konektywności (CI) fazy olejowej
3	Indeks konektywności (CI) tenzydu głównego
4	Indeks konektywności (CI) tenzydu pomocniczego
5	HLB mieszaniny tenzydów
6	Gęstość fazy olejowej
7	Siła jonowa fazy wodnej
8	Ilość fazy olejowej [%]
9	Ilość kotenzydu [%]
10	Ilość tenzydu głównego [%]
11	Ilość tenzydu pomocniczego [%]
12	Ilość fazy wodnej [%]

Dane w postaci plików tekstowych obrobiono następnie za pomocą arkusza kalkulacyjnego MS Excel® 2000 tak, aby przekształcić zestaw trzech współrzędnych trójkątnych do pełnego opisu 4- lub 5-cioskładnikowego układu fazowego. Pełny opis wektora wejściowego sieci neuronowych składający się z 12 zmiennych przedstawiono w tabeli 15. Skład jakościowy kodowano w postaci indeksu konektywności (CI) obliczanego na podstawie znanej budowy chemicznej. W przypadku substancji pomocniczych pochodzenia naturalnego lub półsyntetycznych, będących niejednokrotnie mieszaninami kilku składników określano średnią masę cząsteczkową substancji. Wartości HLB tenzydów i gęstości faz olejowych zostały znalezione w piśmiennictwie. Siła jonowa fazy wodnej była obliczana na

podstawie znanego składu ilościowego i jakościowego fazy wodnej. Wyjście sieci stanowiła jedna komórka, gdzie oczekiwana była binarna decyzja (TAK/NIE). Decyzja „TAK” oznaczała obecność, zaś decyzja „NIE” brak obecności mikroemulsji.

Testowano różne sposoby wstępnego przetwarzania danych (*data preprocessing*):

- a) skalowanie liniowe w zakresach (0,2;0,8)
- b) normalizacja wektorów wejściowych
- c) binaryzacja
- d) wzbogacanie za pomocą reguł zdroworozsądkowych

PROCEDURA SKALOWANIA LINIOWEGO

Skalowanie liniowe jest standardową procedurą dostosowywania wektorów wejściowych i wyjściowych do dziedziny funkcji aktywacji. Procedura jest realizowana przez system **Nets2003**. Równanie 26 przedstawia sposób wyznaczania przeskalowanej wartości współrzędnej wektora wejściowego lub wyjściowego.

$$x_i^{sc} = R_i^{\min} + (x_i - x_i^{\min}) \cdot \frac{(R_i^{\max} - R_i^{\min})}{(x_i^{\max} - x_i^{\min})} \quad (26)$$

gdzie:

x_i^{sc} - skalowana wartość i-tej składowej wektora

x_i - oryginalna wartość i-tej składowej wektora

x_i^{\min} i x_i^{\max} - minimalna i maksymalna wartość i-tej składowej wektora znaleziona dla wszystkich rekordów zbioru danych

R_i^{\min} i R_i^{\max} - ustalona arbitralnie minimalna i maksymalna wartość zakresu skalowania dla i-tej składowej wektora

PROCEDURA NORMALIZACJI

Normalizacja dotyczyła tylko wektorów wejściowych sieci neuronowych i przeprowadzona została za pomocą równania 27 [94] z użyciem programu **Preproc**.

$$x_i^n = \frac{x_i}{\sqrt{\sum_{i=1}^N x_i^2}} \quad (27)$$

gdzie:

x_i^n - znormalizowana i-ta składowa wektora wejściowego

x_i - i-ta składowa wektora wejściowego

N – rozmiar wektora wejściowego

PROCEDURA BINARYZACJI

Binaryzacja danych polegała na przekształceniu reprezentacji wektorów wejściowych i wyjściowych o składowych ciągłych w wektory o składowych binarnych. Dokonano tego za pomocą przekształcenia liczb rzeczywistych zaokrąglonych do najbliższych im wartości całkowitych w ich dwójkowe reprezentacje. Procedura ta spowodowała zwiększenie ilości wejść z 12 do 75. Binaryzacja przeprowadzona została za pomocą programu **Preproc**.

PROCEDURA WZBOGACANIA ZBIORU UCZĄCEGO O REGUŁY ZDROWOROZSĄDKOWE

W celu sprawdzenia możliwości implementacji wiedzy eksperta do systemu przewidującego skład mikroemulsji rozbudowano zbiory uczące wszystkich 10 par zbiorów uczący-testowy otrzymanych za pomocą systemu skanowania **Intelliscan**. Polegało to na przekształcaniu zbioru uczącego poprzez zerowanie kolejno: zawartości tenzydu, fazy olejowej i wodnej, a także ich opisów jakościowych. Kolumny wyjściowe zostały również wyzerowane. Oznaczało to decyzję „NIE”, czyli brak mikroemulsji. Tak przekształcone zbiory zostały następnie dołączone do oryginalnego zbioru

uczącego. Opisana wyżej procedura przekształcania zbioru uczącego miała za zadanie dostarczyć sieci informacji o niemożności stworzenia mikroemulsji bez obecności któregośkolwiek ze składników: tenzydu, fazy olejowej i wodnej. Wytrenowano również sieci na bazowym zbiorze danych poddanych wzbogaceniu o reguły zdroworozsądkowe i określano następnie błąd na zbiorze testowym PRAKT.

3.2.4. BUDOWA MASZYNY WNIOSKUJĄCEJ

Optymalny model neuronowy znaleziony został w toku szeroko zakrojonych eksperymentów numerycznych prowadzonych zgodnie z metodą 10-krotnego wzajemnego sprawdzania. Testowano sieci uczone z nauczycielem, typu MLP, jak też i oparte na logice rozmytej. W przypadku tych ostatnich ustalono metodę defuzyfikacji jako M-SLIDE oraz metodę inicjalizacji reguł na podstawie losowo wybranych rekordów danych. Sieci MLP posiadały od 0 do 5 warstw ukrytych. Algorytmem uczącym była zawsze propagacja wsteczna z techniką momentu i algorytmem *delta-bar-delta*. Jako funkcje aktywacji testowano sigmoidę (funkcję logistyczną), tangens hiperboliczny i fsr [12]. Wszystkie komórki w danej sieci posiadały zawsze tę samą funkcję aktywacji. Komórki w przylegających warstwach były połączone na zasadzie „każdy z każdym”. Funkcja kosztu była zawsze funkcją kwadratową. Zastosowano technikę *jog-of-weights* z kryterium cierpliwości ustalonym na 300 000 iteracji. Ponieważ na wyjściu sieci neuronowej jest zmienna binarna przyjęto jako miarę błędów błąd klasyfikacji.

Stosowano trzy miary sprawności modeli neuronowych klasyfikacji:

- a) błąd całkowity (EG) obliczany jako procent błędnie sklasyfikowanych rekordów niezależnie od decyzji sieci na TAK lub na NIE, obliczany względem całości zbioru testowego (równanie 28)
- b) ilość decyzji fałszywie pozytywnych (FD⁺) tj. wtedy, gdy sieć przewiduje istnienie mikroemulsji, której w rzeczywistości nie

zaobserwowano – obliczana jako procent całości zbioru testowego (równanie 29)

- c) ilość poprawnie sklasyfikowanych mikroemulsji (GD^+) obliczana jako procent ilości wszystkich zaobserwowanych mikroemulsji w zbiorze testowym (równanie 30)

$$EG = \frac{E_A}{n} \cdot 100\% \quad (28)$$

gdzie:

EG – błąd całkowity [%]

E_A – ilość wszystkich błędnie sklasyfikowanych rekordów

n – całkowita ilość rekordów testowych

$$FD^+ = \frac{E_+}{n} \cdot 100\% \quad (29)$$

gdzie:

FD^+ – ilość decyzji fałszywie pozytywnych [%]

E_+ – ilość błędnie sklasyfikowanych rekordów z decyzją o wystąpieniu mikroemulsji w przypadku jej fizycznego braku

n – całkowita ilość rekordów testowych

$$GD^+ = \frac{G_+}{n_+} \cdot 100\% \quad (30)$$

gdzie:

GD^+ – ilość decyzji poprawnych [%]

G_+ – ilość poprawnie sklasyfikowanych rekordów z decyzją o wystąpieniu mikroemulsji

n_+ – ilość przypadków stwierdzenia obecności mikroemulsji

Kryterium selekcji optymalnej architektury był błąd generalizacji obliczany zgodnie z metodą 10-krotnego wzajemnego sprawdzania. Mimo, iż w założeniu odpowiedź sieci neuronowej miała być binarna, to użycie sieci neuronowych z ciągłymi funkcjami aktywacji spowodowało otrzymanie wartości wyjściowych należących do zbioru liczb rzeczywistych. Powstała konieczność użycia progu decyzyjnego, który separowałby pobudzenia komórki wyjściowej interpretowane jako odpowiedź „TAK” od odpowiedzi „NIE”. Dobór progu decyzyjnego przeprowadzono dwojako:

- a) arbitralnie ustalona wartość – mediana przedziału wartości pobudzenia komórki wyjściowej
- b) indywidualnie dla danej architektury z użyciem kryterium błędu całkowitego

O ile pierwszy sposób z arbitralnym, jednakowym progiem pobudzenia jest standardową metodą, o tyle kontrowersje może budzić drugi sposób. Indywidualny dobór progu decyzyjnego dla każdej architektury nie pozwala obliczyć błędu generalizacji, gdyż wykorzystuje się tu informację o wartości oczekiwanej ze zbiorów testowych. Metoda ta wynikała z obserwacji własnych, wg których różne architektury sieci neuronowych charakteryzowały się różnymi optymalnymi progami decyzyjnymi, które pozwalały na osiągnięcie znacznie większej sprawności modelu niż w stosunku do założonego sztywnego progu w postaci mediany przedziału pobudzenia komórki wyjściowej. Założono, iż w pewnych przypadkach próg decyzyjny może być charakterystyczny dla architektury sieci neuronowej. Założenie to zostało zweryfikowane na osobnym zbiorze walidacyjnym, który nie brał udziału ani w uczeniu sieci neuronowej ani w procesie doboru optymalnego progu decyzyjnego. Optymalizacja progu decyzyjnego pozwoliła na oszacowanie maksymalnej sprawności modelu neuronowego. Próg wyrażany był w procentach wielkości przedziału odpowiedzi komórki wyjściowej (równanie 31).

$$\text{Thr}(\%) = \left| \frac{A - x_{\min}}{x_{\max} - x_{\min}} \right| \cdot 100\% \quad (31)$$

gdzie :

Thr(%) – próg pobudzenia [%]

x_{\min} – wartość minimalna przedziału obserwowanych pobudzeń komórki wyjściowej

x_{\max} – wartość minimalna przedziału obserwowanych pobudzeń komórki wyjściowej

A – aktualna wartość liczbowa progu

3.2.5. BUDOWA INTERFEJSU UŻYTKOWNIKA

Zaprojektowano i napisano w języku Borland Delphi® 5.0 system **ME_expert** (Rycina 42 str. 166) liczący 14 000 linii kodu. Funkcjonalnie system ten odpowiada systemowi **SD_expert**. Obok wspólnych elementów jak odczytywanie formatu plików z systemu **Nets2003**, iteracyjne poszukiwanie optymalnej formulacji, czy też obsługa komitetu ekspertów, system **ME_expert** pozwala na binaryzację zbiorów danych w czasie pracy oraz graficzną prezentację wyników pracy w postaci diagramów fazowych. Optymalizacja formulacji może być prowadzona zarówno ze względu na skład jakościowy jak i ilościowy.

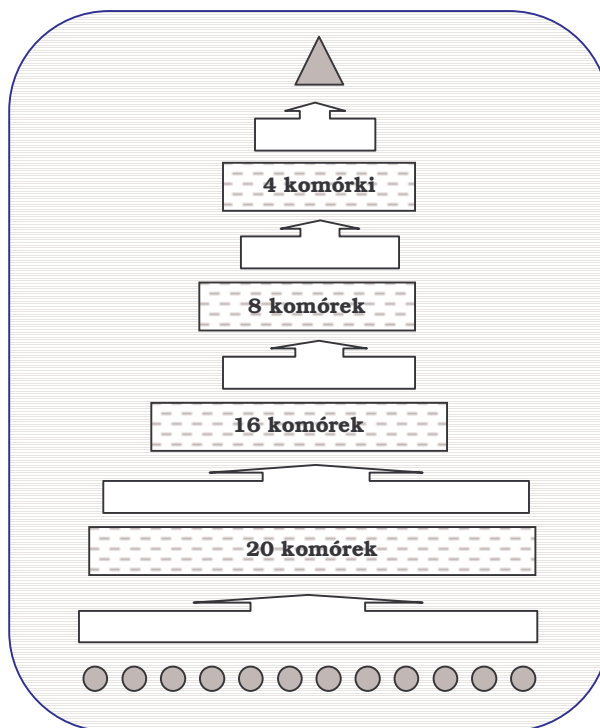
3.2.6. WYNIKI BADAŃ

DANE Z INTELLISCAN

Skalowanie liniowe w zakresie (0,2;0,8)

Budowa modelu neuronowego składu mikroemulsji została rozpoczęta od danych otrzymanych za pomocą algorytmu **Intelliscan** z uwagi na mniejszy koszt obliczeniowy związany z mniejszym rozmiarem zbioru

uczącego. Wyselekcjonowano optymalną architekturę sieci neuronowej typu MLP 20_16_8_4 hid tanh uczona do 5 000 000 iteracji (Rycina 23).



Rycina 23. Optymalna architektura sieci neuronowej dla zbiorów danych po skanowaniu algorytmem *Intelliscan*.

Błąd generalizacji optymalnej architektury określony za pomocą procedury 10-krotnego wzajemnego sprawdzania przedstawiono w tabeli 16.

Tabela 16. Wyniki 10-krotnego wzajemnego sprawdzania dla architektury 20_16_8_4 hid tanh.

EG	FD ⁺	GD ⁺	Thr(%)
17,4%	4,7%	62,3%	50%
17,3%	4,7%	62,3%	35%*

* optymalny próg

Powyższa sieć posłużyła jako narzędzie do projektowania formułacji mikroemulsji, które następnie zostały wykonane w laboratorium. Wytrenowano sieć 20_16_8_4 hid tanh na całym dostępnym zbiorze danych,

a następnie przygotowano 55 formułacji, które zostały wskazane przez sieć 20_16_8_4 hid tanh jako izotropowe układy fazowe. Formułacje te złożyły się na zbiór PRAKT, który liczył ok. 4200 rekordów danych (Tabela 17).

Tabela 17. Zestawienie składów jakościowych 55 formułacji zaproponowanych przez sieć 20_16_8_4 hid tanh.

Faza olejowa	Tenzyd	Kotenzyd
Mirystynian izopropylu	Brij® 96, laurylosiarczan sodowy	Glikol propylenowy
Mirystynian izopropylu	Brij® 58	Glikol propylenowy
Mirystynian izopropylu	Tween® 60	-
Mirystynian izopropylu	Tween® 60, laurylosiarczan sodowy	-
Mirystynian izopropylu	Tween® 60	Glikol propylenowy
Mirystynian izopropylu	Tween® 60	Makrogol (PEG 4000)
Parafina ciekła	Brij® 96	-
Parafina ciekła	Brij® 96	Makrogol (PEG 4000)
Parafina ciekła	Brij® 58	-
Parafina ciekła	Brij® 58	Makrogol (PEG 4000)
Parafina ciekła	Brij® 58	Glikol propylenowy
Parafina ciekła	Tween® 60	Glikol propylenowy
Parafina ciekła	Arlacel® 20	Glikol propylenowy
Parafina ciekła	Arlacel® 20, laurylosiarczan sodowy	Glikol propylenowy
Parafina ciekła	Tween® 20	-
Parafina ciekła	Tween® 20	Glikol propylenowy
Parafina ciekła	Tween® 20	Makrogol (PEG 4000)

Spośród zaproponowanych przez sieć neuronową 55 układów fazowych 15 okazało się być układami izotropowymi. Należy zaznaczyć, iż w zbiorze PRAKT znajdowały się formułacje z zupełnie nieznanymi sieciami neuronowym substancjami pomocniczymi jak Brij® 58 czy Arlacel® 20. Oznacza to, że sieci musiały generalizować zarówno skład ilościowy jak i jakościowy mikroemulsji. Wyniki testowania sieci sieci 20_16_8_4 hid tanh na zbiorze PRAKT przedstawiono w tabeli 18.

Tabela 18. Wyniki testowania na zbiorze PRAKT dla architektury 20_16_8_4 hid tanh.

EG	FD ⁺	GD ⁺	Thr(%)
60,4%	59,2%	73,7%	50%
6,7%	0,3%	3,2%	90%

Ustalenie progu 90% wynikało z niemożności znalezienia optymalnej jego wartości dla założonego kryterium błędu całkowitego EG. W przeciwieństwie do 10-krotnego wzajemnego sprawdzania, gdzie próg optymalny był mniejszy od maksymalnego, w tym przypadku próg maksymalny pozwalał na uzyskanie najmniejszej wartości EG, wobec czego zrezygnowano z procedury optymalizacji.

Równocześnie z eksperymentami laboratoryjnymi prowadzono eksperymenty numeryczne zmierzające do poprawy jakości modelu neuronowego. Zbudowano komitet ekspertów, w którym jako elementu przetwarzającego wyjścia sieci uczonych na danych oryginalnych użyto sieci neuronowej drugiego rzędu.

W skład komitetu ekspertów wchodziły następujące architektury sieci neuronowych:

1. 15_7_4_4 hid tanh
2. 15_7_4_4 hid fsr
3. 15_7_4 hid tanh
4. 15_7_4 hid fsr
5. 20_16_8_4hid tanh
6. 20_16_8_4hid fsr
7. 60_20_8hid tanh
8. 60_20_8hid fsr
9. 80_20_10_4hid tanh
10. 80_20_10_4hid fsr

Tabela 19. Wyniki testowania na zbiorze PRAKT dla sieci drugiego rzędu 7_5_3 hid fsr.

EG	FD ⁺	GD ⁺	Thr(%)
29,8%	26,8%	35,3%	50%
7,0%	3,3%	20,0%	90%

Analiza wyników testowania komitetu ekspertów na zbiorze PRAKT przedstawionych w tabeli 19 pozwala stwierdzić, że ustalenie progu decyzyjnego sieci neuronowej na 90% pozwoliło na prawie 3-krotne zmniejszenie błędu całkowitego EG. Ilość poprawnie przewidzianych układów izotropowych uległa zmniejszeniu do 20%. Istotna jest tutaj niska wartość FD⁺, która wyniosła 3,3%.

Przy użyciu komitetu ekspertów przeprowadzono poszukiwania układu izotropowego zawierającego duże ilości fazy wodnej. Zadaniem systemu **ME_expert** użytego do tego zadania było znalezienie optymalnego składu jakościowego tenzydów, kotenzydu i fazy olejowej dla zadanego składu ilościowego. Dzięki iteracyjnej pracy systemu **ME_expert** znaleziono 8 formułacji potencjalnych mikroemulsji (Tabela 20). Ilościowy skład wszystkich formułacji był taki sam i został przedstawiony na diagramie fazowym (Rycina 24).

Tabela 20. Formulacje wyselekcjonowane przez system *ME_expert* jako potencjalne mikroemulsje. Stosunek kotenzydu do tenzydu 1:4.

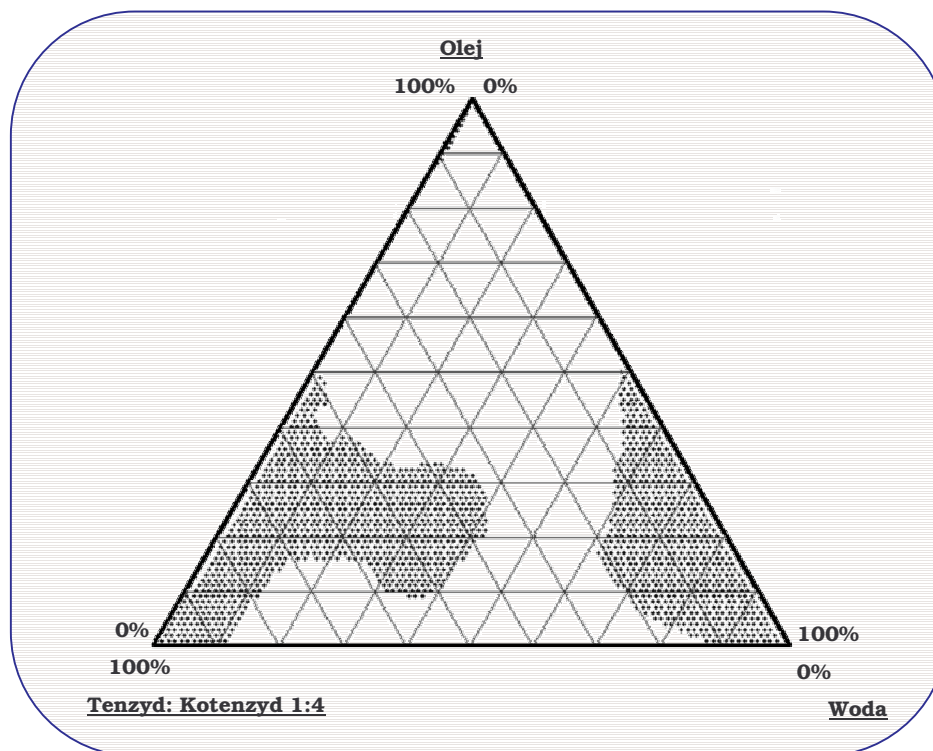
Faza olejowa	Tenzyd	Kotenzyd
Triacetyna	Pluronic® F68	Arlacel® 20
Triacetyna	Pluronic® F68	Makrogol (PEG 400)
Mirystynian izopropylu	Pluronic® F68	Arlacel® 20
Mirystynian izopropylu	Pluronic® F68	Makrogol (PEG 400)
Triacetyna	Solutol® HS 15	Makrogol (PEG 400)
Triacetyna	Solutol® HS 15	Arlacel® 20
Mirystynian izopropylu	Solutol® HS 15	Makrogol (PEG 400)
Mirystynian izopropylu	Solutol® HS 15	Arlacel® 20

Weryfikacja doświadczalna potwierdziła obecność mikroemulsji dla 4 układów. Wyniki prac doświadczalnych stanowiły zbiór PRAKT1. Trafność przewidywań systemu *ME_expert* na zbiorze PRAKT1 przedstawiono w tabeli 21. Na uwagę zasługuje fakt, iż Pluronic® F68 i Solutol® HS 15 są tenzydami zupełnie nieznanymi systemowi neuronowemu tak samo jak użycie Makrogolu 400 w charakterze kotenzydu.

Tabela 21. Błędy systemu *ME_expert* w przewidywaniu zbioru PRAKT1.

EG	FD+	GD+	Thr(%)
23,1%	1,6%	76,8%	50%

Próg 50% (Tabela 21) ustalony został arbitralnie w celu selekcji szerszego zakresu formulacji.



Rycina 24. Skład ilościowy formułacji zaproponowany przez system *ME_expert*.

Normalizacja

W celu wykazania wpływu sposobu wstępnego przetwarzania danych wytrenowano architekturę 20_16_8_4 hid tanh na danych poddanych normalizacji. Pary zbiorów uczący-testowy odpowiadały uprzednio użytym zbiorom danych. Wyniki 10-krotnego wzajemnego sprawdzania wytrenowanej sieci neuronowej przedstawiono w tabeli 22.

Tabela 22. Wyniki 10-krotnego wzajemnego sprawdzania dla architektury 20_16_8_4 hid tanh uczonej na znormalizowanych danych.

EG	FD ⁺	GD ⁺	Thr(%)
18,0%	7,6%	69,2%	50%
17,9%	6,5%	66,2%	55%*

* optymalny próg

Sieć 20_16_8_4 hid tanh wytrenowano na całym zbiorze danych i przetestowano na zbiorze PRAKT poddanemu również normalizacji (Tabela 23).

Tabela 23. Wyniki testowania na zbiorze PRAKT architektury 20_16_8_4 hid tanh uczonej na danych znormalizowanych.

EG	FD ⁺	GD ⁺	Thr(%)
9,1%	4,5%	0,0%	50%
4,5%	0,0%	0,0%	90%

Binaryzacja

W związku z poczynionymi założeniami dotyczącymi wstępnego przetwarzania danych wytrenowano architekturę 20_16_8_4 hid tanh na danych binarnych. Pary zbiorów testowy-uczący odpowiadały uprzednio użytym zbiorom danych. Przeprowadzono również 10-krotne wzajemne sprawdzanie i obliczono błędy generalizacji (Tabela 24).

Tabela 24. Wyniki 10-krotnego wzajemnego sprawdzania dla architektury 20_16_8_4 hid tanh uczonej na danych binarnych.

EG	FD ⁺	GD ⁺	Thr(%)
17,6%	6,3%	66,5%	50%
17,6%	5,9%	65,5%	59%*

* optymalny próg

Dane z piśmiennictwa wskazują na dobre zdolności klasyfikacyjne sieci rozmytych pracujących z użyciem binarnych danych uczących [42]. Podjęto więc poszukiwania optymalnej architektury sieci rozmytej. Poszukiwania architektury odbywały się metodą prób i błędów z uwagi na to, że algorytm automatycznej konstrukcji architektury sieci rozmytych tworzyły zbyt duże sieci neuronowe. Jako optymalną znaleziono sieć z 60 regułami rozmytymi: 60 hid fuzzy_M-SLIDE. Analiza wyników 10-krotnego wzajemnego

sprawdzania sieci 60 hid fuzzy_M-SLIDE przedstawionych w tabeli 25 pozwala stwierdzić, że użycie sieci rozmytych jest korzystne w przypadku danych binarnych. Świadczą o tym wartości błędu całkowitego EG mniejsze o ponad 1% w stosunku do otrzymanych przez sieć 20_16_8_4 hid tanh.

Tabela 25. Wyniki 10-krotnego wzajemnego sprawdzania dla architektury 60 hid fuzzy_M-SLIDE.

EG	FD ⁺	GD ⁺	Thr(%)
16,4%	5,2%	67,0%	50%
16,4%	5,9%	69,1%	48%*

* optymalny próg

Kontynuując pracę nad binarną reprezentacją danych zbudowano komitet ekspertów w oparciu o 6 najlepszych architektur sieci neuronowych:

- a) 15_7_5 hid tanh
- b) 20_16_8_4 hid fsr
- c) 80_40_20_10 hid fsr
- d) 60 hid fuzzy_M-SLIDE
- e) 80 hid fuzzy_M-SLIDE
- f) 100 hid fuzzy_M-SLIDE

Jako elementu przetwarzającego wyjścia sieci uczonych na danych oryginalnych użyto sieci neuronowej drugiego rzędu. Wybrano sieć o architekturze 7_5_3 hid fsr. Wyniki testowania komitetu ekspertów na zbiorze PRAKT przedstawiono w tabeli 26

Tabela 26. Wyniki testowania sieci drugiego rzędu 7_5_3 hid fsr na zbiorze PRAKT. Dane binarne.

EG	FD ⁺	GD ⁺	Thr(%)
7,6%	4,1%	22,6%	50%
4,4%	0,3%	9,5%	90%

Interesujące wyniki uzyskano również w przypadku sieci 15_7_5 hid tanh (Tabela 27). Cechą wyróżniającą tę architekturę jest brak konieczności ustalania progu decyzyjnego komórki wyjściowej wynikający z praktycznie binarnych odpowiedzi komórki wyjściowej – odchylenia od wartości 1 lub 0 były nieistotne. Wyniki testowania tej architektury na zbiorze PRAKT przedstawione w tabeli 27 świadczą o dobrej zdolności do generalizacji.

Tabela 27. Wyniki testowania sieci drugiego rzędu 15_7_5 hid tanh na zbiorze PRAKT. Dane binarne.

EG	FD ⁺	GD ⁺	Thr(%)
9,3%	6,3%	34,2%	Brak

Podobny efekt stwierdzono w przypadku architektury 30_20_10 hid tanh, jednakże błąd całkowity (EG) był większy o ok. 2%.

W celu stwierdzenia wpływu binaryzacji na jakość modeli neuronowych dobrano tak próg decyzyjny dla sieci drugiego rzędu, aby błąd całkowity (EG) był taki sam jak dla sieci drugiego rzędu opracowującej wyjścia sieci trenowanych na zbiorze danych skalowanym od 0,2 do 0,8 dla progu 90% (Tabela 28). Dowodem na wpływ procesu binaryzacji byłyby różne wartości GD⁺ oraz błędów FD⁺.

Tabela 28. Wyniki testowania na zbiorze PRAKT dla sieci drugiego rzędu 7_5_3 hid fsr.

Dane	EG	FD ⁺	GD ⁺	Thr(%)
SCALE	7,0%	3,3%	20,0%	90%
BINAR	7,0%	3,4%	21,6	65%

SCALE – sieć trenowana na zbiorze danych *Intelliscan* skalowanym w zakresie od 0,2 do 0,8

BINAR - sieć trenowana na zbiorze danych *Intelliscan* poddanym binaryzacji

Analogiczną procedurę przeprowadzono dla sieci 15_7_5 hid tanh, z tym że tutaj dobierano próg pobudzenia sieci 15_7_5 hid tanh uczonej na danych skalowanych od 0,2 do 0,8 z uwagi na niewrażliwość na wartość progu sieci

uczonej dla danych binarnych. Wyniki eksperymentu numerycznego przedstawiono w tabeli 29.

Tabela 29. Wyniki testowania na zbiorze PRAKT dla sieci drugiego rzędu 15_7_5 hid tanh.

Dane	EG	FD+	GD+	Thr(%)
SCALE	9,3%	5,7%	21,0%	99%
BINAR	9,3%	6,3%	34,2%	Brak

SCALE – sieć trenowana na zbiorze danych *Intelliscan* skalowanym w zakresie od 0,2 do 0,8

BINAR - sieć trenowana na zbiorze danych *Intelliscan* poddanym binaryzacji

Przeprowadzono dodatkowe porównanie optymalnych architektur sieci drugiego rzędu komitetów ekspertów opracowanych dla danych skalowanych w zakresie 0,2 do 0,8 i binarnych. Użyto analogicznej procedury doboru progu decyzyjnego. Błędy architektury 7_5_3 hid fsr (skalowanie od 0,2 do 0,8) i 15_7_5 hid tanh (dane binarne) przedstawiono w tabeli 30.

Tabela 30. Wyniki testowania na zbiorze PRAKT dla najlepszych sieci drugiego rzędu.

Dane	EG	FD+	GD+	Thr(%)
SCALE	9,3%	5,7%	20,5%	86%
BINAR	9,3%	6,3%	34,2%	Brak

SCALE – sieć 7_5_3 hid fsr trenowana na zbiorze danych *Intelliscan* skalowanym w zakresie od 0,2 do 0,8

BINAR - sieć 15_7_5 hid tanh trenowana na zbiorze danych *Intelliscan* poddanym binaryzacji

Reguły zdroworozsądkowe

Przetestowano sieć 20_16_8_4 hid tanh nauczoną na bazowym zbiorze danych z dodatkiem reguł zdroworozsądkowych. Testowanie przeprowadzono na zbiorach zawierających tylko rekordy opisujące reguły zdroworozsądkowe. Było to równoznaczne z zadaniem sieci pytania, czy może istnieć mikroemulsja bez jednego z jej składników, tj. fazy wodnej, olejowej, czy też tenzydu. Wyniki testowania przedstawiono w tabeli 31. Z uwagi na

jednakowe odpowiedzi dla wszystkich rekordów testowych przedstawiono uśrednione wyniki testowania. Brak progów oznacza, że nie był on potrzebny z uwagi na jednolity charakter wyjścia, które w tym przypadku przyjmowało wartości bliskie 0.

Tabela 31. Testowanie sieci 20_16_8_4 hid tanh uczonej na danych wzbogaconych o reguły zdroworozsądkowe.

EG	FD ⁺	GD ⁺	Thr(%)
0,0%	0,0%	0,0%	Brak

W celu wykazania wpływu dodatku reguł zdroworozsądkowych użyto tej samej procedury testowania dla sieci 20_16_8_4 hid tanh uczonej na zbiorze bazowym bez dodatku reguł zdroworozsądkowych. Wyniki przedstawione w tabeli 32 charakteryzują się takim samym rozkładem odpowiedzi.

Tabela 32. Testowanie sieci 20_16_8_4 hid tanh uczonej na zbiorze bazowym bez dodatku reguł zdroworozsądkowych.

EG	FD ⁺	GD ⁺	Thr(%)
0,0%	0,0%	0,0%	Brak

Sieć 20_16_8_4 hid tanh uczona na danych wzbogaconych o reguły zdroworozsądkowe została przetestowana również na zbiorze PRAKT. Wyniki testowania dla dwóch progów decyzyjnych przedstawiono w tabeli 33. Brak pozytywnych odpowiedzi może być spowodowany niejednorodną dystrybucją klas w zbiorze uczącym, w którym tylko ok. 8,5% rekordów opisywało istniejącą mikroemulsję. W celu wykluczenia tego efektu przeprowadzono równoważenie ilości klas za pomocą procedury „**balance+**” programu **Preproc**. Procedura ta spowodowała dodanie do zbioru uczącego nadmiarowej ilości rekordów mniej licznie reprezentowanej klasy w taki

sposób, aby sumarycznie liczba rekordów dla obu klas była równa. Jakość informacji nie zmieniła się. Zmianie uległa tylko częstotliwość spotykania się sieci z każdą klasą w czasie uczenia. Wyniki testowania sieci 20_16_8_4 hid tanh nauczonej na tak przekształconym zbiorze danych przedstawiono w tabeli 34.

Tabela 33. Wyniki testowania na zbiorze PRAKT sieci 20_16_8_4 hid tanh uczonej na zbiorze bazowym z dodatkiem reguł zdroworozsądkowych.

EG	FD ⁺	GD ⁺	Thr(%)
4,6%	0,0%	0,0%	50%
4,6%	0,0%	0,0%	90%

Tabela 34. Wyniki testowania na zbiorze PRAKT sieci 20_16_8_4 hid tanh uczonej na zbiorze bazowym z dodatkiem reguł zdroworozsądkowych i równoważeniem klas.

EG	FD ⁺	GD ⁺	Thr(%)
53,0%	51,7%	71,0%	50%
49,0%	47,6%	70,0%	90%

DANE ZE SKANOWANIA RÓWNOMIERNEGO CO 2%

W celu wykazania wpływu sposobu skanowania bitmap wytrenowano sieć 20_16_8_4 hid tanh na zbiorze uczącym pochodzącym ze skanowania równomiernego co 2% (**scan_2_2**).

Wyniki generalizacji po 10-krotnym wzajemnym sprawdzaniu przedstawione w tabeli 35 świadczą o dobrych właściwościach generalizacyjnych tak przygotowanego modelu. Próg decyzyjny 39% został znaleziony jako optymalny.

Tabela 35. Wyniki generalizacji sieci 20_16_8_4 hid tanh scan_2_2 po 10-krotnym wzajemnym sprawdzaniu.

EG	FD+	GD+	Thr(%)
10,5%	4,8%	74,9%	50%
10,4%	4,4%	77,2%	39%*

* optymalny próg

Sieć 20_16_8_4 hid tanh uczoną na zbiorze bazowym **scan_2_2** przetestowano również na zbiorze PRAKT. Wyniki testowania zamieszczono w tabeli 36.

Tabela 36. Wyniki testowania na zbiorze PRAKT sieci 20_16_8_4 hid tanh scan_2_2 uczonej na zbiorze bazowym.

EG	FD+	GD+	Thr(%)
24,8%	22,2%	42,1%	50%
23,4%	20,7%	40,5%	90%

Charakter zbiorów testowych uległ zmianie w stosunku do danych uzyskanych ze skanowania algorytmem **Intelliscan**, zarówno z uwagi na losowy dobór formułacji do każdej pary zbiorów testowy-uczący, jak też i z powodu różnej wielkości zbiorów testowych. Z tego powodu dokonano porównania obu systemów skanowania na podstawie metody *leave-one-out* z zamianą zbiorów testowych. Procedura polegała na wytrenowaniu architektury 20_16_8_4 hid tanh zarówno z użyciem danych **scan_2_2** jak też i **Intelliscan**, a następnie przetestowaniu obydwu zestawów sieci na zbiorach testowych **scan_2_2** i **Intelliscan**. W wyniku opisanego wyżej postępowania otrzymano 4 zestawy wyników przedstawionych w tabelach 37 do 40.

Tabela 37. Wyniki generalizacji sieci 20_16_8_4 hid tanh scan_2_2 w metodzie leave-one-out. Test na zbiorach scan_2_2.

EG	FD ⁺	GD ⁺	Thr(%)
9,6%	3,6%	62,%	50%*

* optymalny próg

Tabela 38. Wyniki generalizacji sieci 20_16_8_4 hid tanh Intelliscan w metodzie leave-one-out. Test na zbiorach *Intelliscan*.

EG	FD ⁺	GD ⁺	Thr(%)
17,1%	6,4%	56,0%	50%
16,9%	5,7%	54,2%	53%*

* optymalny próg

Tabela 39. Wyniki generalizacji sieci 20_16_8_4 hid tanh Intelliscan w metodzie leave-one-out. Test na zbiorach scan_2_2.

EG	FD ⁺	GD ⁺	Thr(%)
10,6%	4,4%	61,8%	50%*

* optymalny próg

Tabela 40. Wyniki generalizacji sieci 20_16_8_4 hid tanh_scan_2_2 w metodzie leave-one-out. Test na zbiorach *Intelliscan*.

EG	FD ⁺	GD ⁺	Thr(%)
15,6%	6,1%	58,7%	50%
15,4%	5,5%	57,2%	56%*

* optymalny próg

Porównano średnie błędy całkowite (EG) sieci testowanych na zbiorach pochodzących z tych samych systemów skanowania. Żadne z odpowiadających sobie średnich błędów w teście t-Studenta nie różniły się od siebie istotnie statystycznie.

W wyniku binaryzacji zbiorów danych **scan_2_2** zbudowano komitet ekspertów analogicznie jak dla danych z **Intelliscan**. Nie otrzymano jednak lepszych wyników. Żadna z sieci drugiego rzędu nie wykazywała też braku wrażliwości na wartość progu decyzyjnego. Wyniki generalizacji zbioru PRAKT przez optymalną sieć 5 hid tanh scan_2_2 zestawiono w tabeli 41.

Tabela 41. Wyniki testowania na zbiorze PRAKT sieci drugiego rzędu 5 hid tanh scan_2_2 uczonej na zbiorze bazowym. Dane binarne.

EG	FD ⁺	GD ⁺	Thr(%)
6,4%	2,8%	18,9%	50%
6,4%	2,7%	18,4%	90%

3.3. APROKSYMACJA FUNKCJI STĘŻENIE-CZAS W BADANIACH IN VITRO I IN VIVO

3.3.1. ZAKRES BADAŃ

Przeprowadzono eksperymenty z aproksymacją funkcji jednej zmiennej stężenie-czas dla badań *in vitro* i badań *in vivo*.

W zakresie badań *in vitro* wykorzystano wyniki badań szybkości rozpuszczania ketoprofenu z dwuskładnikowych stałych rozproszeń. W zakresie badań *in vivo* użyto danych dotyczących dożołądkowego podania szczurom mupirocyny zamkniętej w liposomach

3.3.2. ZDEFINIOWANIE FUNKCJI SYSTEMU NEURONOWEGO

Celem badań było stworzenie uniwersalnego systemu bazującego na sieciach neuronowych do oceny dowolnej krzywej przedstawiającej zarówno zmiany szybkości uwalniania substancji leczniczej z postaci leku *in vitro* jak też i zmiany stężeń substancji leczniczej *in vivo*.

3.3.3. PRZYGOTOWANIE DANYCH STANOWIĄCYCH BAZĘ WIEDZY

Dane pochodziły zarówno z badań doświadczalnych przeprowadzonych w Katedrze Technologii Postaci Leku i Biofarmacji CMUJ, jak i symulacji numerycznych. Symulacje numeryczne prowadzono w celu zweryfikowania sprawności systemu neuronowego i porównania aproksymacji neuronowej z klasyczną metodą statystyczną. Symulowane krzywe były próbkowane z zadaniem krokiem na osi czasu i następnie poddawane zaszumianiu. Porównywano takie parametry jak: stężenie maksymalne (C_{max}), czas osiągnięcia stężenia maksymalnego (t_{max}), pole pod krzywą stężenie-czas (AUC) obliczone wg modelu statystycznego i neuronowego, które powstały na podstawie zaszumionych danych. Wartości ww. parametrów porównywano

również z otrzymanymi dla oryginalnych danych. Wektor wejściowy był zawsze jednoelementowy i wyrażał czas w minutach, po którym sieć miała za zadanie przewidzieć stężenie substancji leczniczej oznaczone w danym medium. Sieć posiadała zawsze tylko jedno wyjście. Dane poddawano wstępnemu przetwarzaniu, które polegało na odbijaniu współrzędnych stężenie-czas względem punktu (0,0), co nadawało sigmoidalny kształt danym uczącym zgodny z charakterem funkcji aktywacji użytych w modelach neuronowych. Dane poddawano również liniowemu skalowaniu w zakresie od -0,8 do 0,8 i zaszumianiu z amplitudą $\pm 5\%$. Do zbioru oryginalnych danych dodawano czterokrotną ilość rekordów zaszumionych.

3.3.4. BUDOWA MASZYNY WNIOSKUJĄCEJ

Przeprowadzono serię eksperymentów numerycznych w celu znalezienia optymalnej architektury sieci neuronowych. Liczba warstw ukrytych była zmienna i wahała się od 0 do 6. Testowano różne funkcje aktywacji: liniową, sigmoidę (funkcję logistyczną), tangens hiperboliczny, fsr, sinus i cosinus. Kolejne warstwy ukryte posiadały niekiedy różne funkcje aktywacji. Algorytm uczący stanowiła zawsze propagacja wsteczna z techniką momentu oraz algorytmem *delta-bar-delta*. Zastosowano również technikę *jog-of-weights* z kryterium cierpliwości ustalonym na 5000 iteracji. Kryterium błędu był błąd RMSE obliczany wg równania 25 lub absolutny błąd całkowity (MAE) obliczany wg równania 32.

$$MAE = \sum_{i=1}^n \frac{|s_i - s_i^{wzorzec}|}{n} \quad (32)$$

gdzie:

s_i – aktualne wyjście sieci neuronowej

$s_i^{wzorzec}$ – zadane wyjście sieci neuronowej

n – liczba rekordów danych

Podstawą do selekcji optymalnej architektury był błąd nauczania sieci na różnych zbiorach reprezentujących zarówno profile uwalniania substancji leczniczej *in vitro* jak i stężenie w płynach ustrojowych. Zadaniem sieci neuronowej było jak najwierniejsze reprezentowanie wyników różnych przebiegów stężeń. Oceniano również wizualnie gładkość aproksymacji.

Po etapie selekcji optymalnej architektury rozpoczęto testowanie na zbiorach danych symulowanych i pochodzących z badań doświadczalnych. Symulacje prowadzono w celu weryfikacji prawidłowości aproksymacji przez sieci neuronowe. Dla przebiegów krzywych stężenie-czas *in vitro* użyto do symulacji funkcji logarytmicznej, której parametry dobierano a priori. Funkcje logarytmiczne następnie próbkowano na osi czasu z krokiem o nierównej długości. Miało to na celu utrudnić zadanie aproksymacji jak też i zbliżyć charakter danych symulowanych do rzeczywistych wyników obserwowanych w czasie prac eksperymentalnych. Otrzymany zestaw wartości funkcji zaszumiano z amplitudą $\pm 10\%$ i poddawano regresji za pomocą arkusza kalkulacyjnego MS Excel® 2000. Równocześnie zbiór zaszumionych danych podawano na wejście opisanemu poniżej systemowi **PKB 2.0**, który używał uprzednio wyselekcjonowanych sieci neuronowych. Rezultaty pracy systemu próbkowano co 0.5 jednostki czasu i dobierano parametry funkcji logarytmicznej za pomocą tego samego narzędzia MS Excel® 2000. Równanie 33 przedstawia funkcję logarytmiczną z parametrami A i B . Porównanie parametrów funkcji logarytmicznej (A i B) dobranych na danych zaszumionych i na danych poddanych neuronowej aproksymacji z oryginalnymi, założonymi a priori, posłużyło do oceny przydatności neuronowej aproksymacji w analizie przebiegów krzywych stężenie-czas *in vitro*. Za miarę błędu posłużył błąd MAE (równanie 32).

$$y = A \ln(x) + B \quad (33)$$

gdzie:

A i B – parametry funkcji logarytmicznej

Analogicznie przygotowywano symulowane dane dla przebiegów czasowych *in vivo*, z tym że tutaj wykorzystano równanie dwuwykładnicze dla modelu jednokompartamentowego i podania pozanaczyniowego [30] – równanie 34.

$$y = A \exp(Bx) + C \exp(Dx) \quad (34)$$

gdzie:

A,B,C,D – stałe równania dwuwykładniczego

Zarówno dla modelu matematycznego jak też i neuronowego obliczano parametry takie jak: stężenie maksymalne (C_{\max}), czas osiągnięcia stężenia maksymalnego (t_{\max}) i pole pod krzywą stężenie-czas w zakresie od 0 do określonego czasu ($AUC^{0 \rightarrow t}$), czyli bez pola resztkowego. Porównanie wartości tych parametrów pomiędzy modelem neuronowym a matematycznym posłużyło do określenia jakości modelu neuronowego. Wartości t_{\max} i C_{\max} obliczano dla modelu matematycznego wyliczając pochodną z równania 34 i przyrównując ją do wartości 0. W przypadku modelu neuronowego system **PKB 2.0** przeszukuje wszystkie dostępne punkty testowania znajdując największą wartość C_{\max} i odczytując dla niej t_{\max} . Wartości $AUC^{0 \rightarrow t}$ dla modelu matematycznego obliczano całkując równanie 34 w granicach od 0 do t. Dla modelu neuronowego użyto metody trapezów.

W drugim etapie badań użyto danych z badań wpływu postaci leku na dostępność biologiczną mupirocyny. Użyto systemu **PKB 2.0** do aproksymacji przebiegu stężeń mupirocyny w osoczu szczura po podaniu dożołądkowym zawiesiny liposomów z mupirocyną.

3.3.5. BUDOWA INTERFEJSU UŻYTKOWNIKA

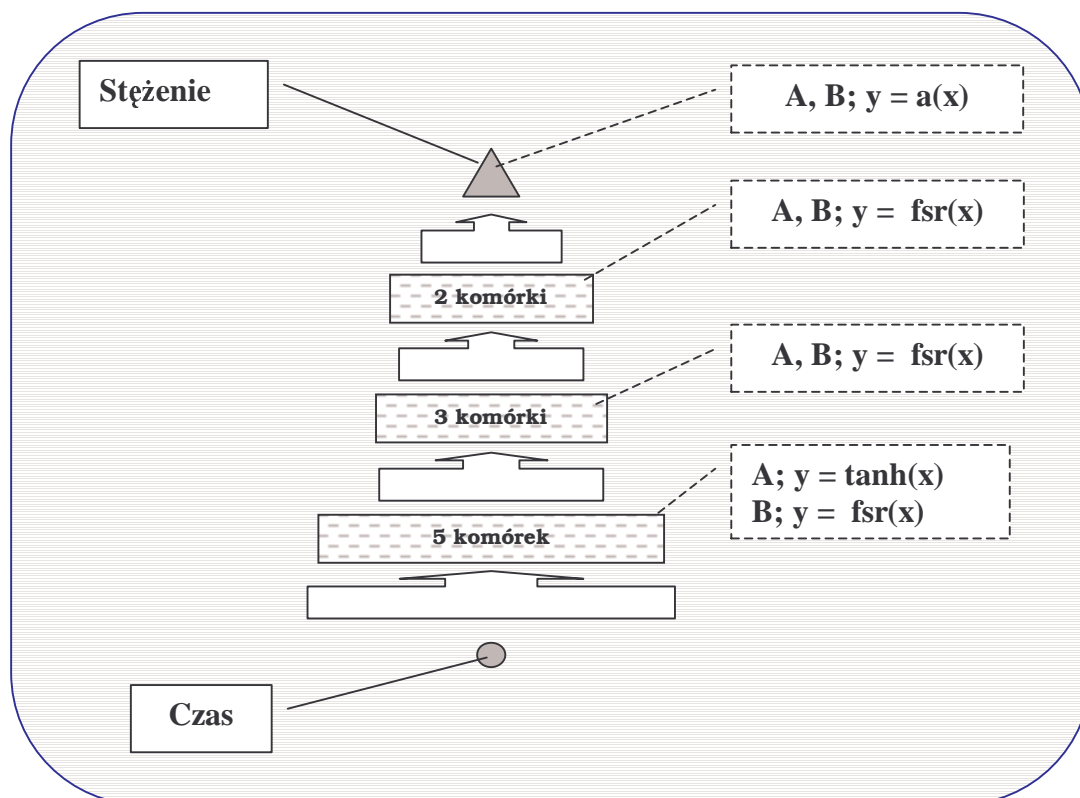
Do pracy z neuronowymi systemami aproksymującymi zaprojektowano i napisano w języku Borland Delphi® 5.0 system **PKB**, którego aktualna wersja **PKB 2.0** liczy ok. 10 000 linii kodu (Rycina 39 str. 165). Jest to system bazujący aktualnie na dwóch wyselekcjonowanych sieciach neuronowych. Sposób pracy różni się od poprzednio przedstawionych

systemów **SD_expert** i **ME_expert**. Korzystając z doświadczeń z komitetem ekspertów zaprojektowano sposób pracy **PKB 2.0** jako wielokrotne uczenie sieci neuronowych i uśrednianie wyników ich testowania na zbiorze uczącym, który jest równocześnie zbiorem danych poddawanych aproksymacji. Ilość powtórzeń uczenia danej architektury może być dowolnie regulowana – standardowo obie architektury uczone są po 10 razy. Wyciągnięcie średniej i obliczenie 99% przedziału ufności pozwala na otrzymanie ostatecznego modelu aproksymującego i określenie powtarzalności uczenia sieci neuronowych. Dokładność próbkowania modelu jest określana przez użytkownika za pomocą osobnego pliku testowego. Wstępne przetwarzanie danych (*preprocessing*) oraz proces uczenia są zautomatyzowane i nie wymagają ustawiania jakichkolwiek parametrów.

3.3.6. WYNIKI BADAŃ

SELEKCJA OPTYMALNYCH ARCHITEKTUR SIECI NEURONOWYCH

W toku eksperymentów numerycznych wyselekcjonowano dwie architektury sieci neuronowych. Sieci A i B (Rycina 25) są czterowarstwowe i różnią się od siebie tylko obecnością w pierwszej warstwie ukrytej sieci A tangensa hiperbolicznego jako funkcji aktywacji. Obie sieci posiadają funkcję liniową jako funkcję aktywacji komórki wyjściowej, która reprezentuje wartość stężenia odpowiadającego podanemu na jedyne wejście sieci punktowi czasowemu.



Rycina 25. Optymalne architektury sieci neuronowych użytych do aproksymacji funkcji stężenie-czas. Sieć A – pierwsza warstwa ukryta z tangensem hiperbolicznym jako funkcją aktywacji. Sieć B – wszystkie warstwy ukryte posiadają funkcję fsr.

MODELOWANIE PRZEBIEGÓW KRZYWYCH STĘŻENIE-CZAS *IN VITRO*

W pierwszym etapie badań symulowano przebieg krzywych stężenie-czas za pomocą funkcji logarytmicznej. Przeprowadzono 10 symulacji krzywych dla różnych parametrów A i B funkcji logarytmicznej. Obliczono błędy dla parametrów A i B otrzymanych zarówno w procedurze regresji, jak też i modelowania neuronowego. Wyniki przedstawiono w tabeli 42.

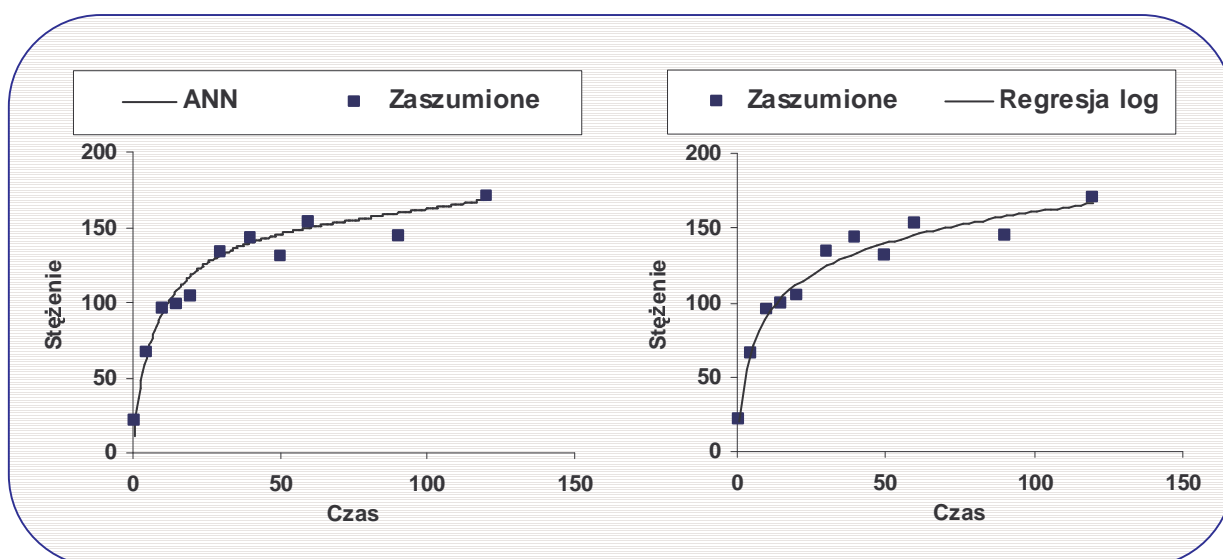
Tabela 42. Średnie błędy absolutne (MAE) przewidywania parametrów równania logarytmicznego A i B.

Parametr	REG NOISED	REG ANN
A	0,16	0,28
B	0,24	0,87

REG NOISED - regresja na danych zaszumionych.

REG ANN - regresja na danych po aproksymacji neuronowej.

Chociaż błędy dla aproksymacji neuronowej są większe, to jednak ich różnica w stosunku do błędów regresji nie jest istotna statystycznie. Przykład aproksymacji neuronowej zaszumionych danych przedstawiono na rycinie 26. Użycie systemu **PKB 2.0** pozwoliło na wzbogacenie zbioru danych dla ketoprofenu dzięki próbkowaniu modelu neuronowego z dokładnością większą niż w oryginalnych danych. Przykład aproksymacji funkcji stężenie-czas dla badań nad stałymi rozproszzeniami przedstawiono na rycinie 9 ze str. 61.



Rycina 26. Porównanie aproksymacji neuronowej i regresji logarytmicznej dla danych obrazujących symulowany przebieg stężeń in vitro. ANN – aproksymacja za pomocą modelu neuronowego. Regresja log – regresja logarytmiczna.

MODELOWANIE PRZEBIEGÓW KRZYWYCH STĘŻENIE-CZAS *IN VIVO*

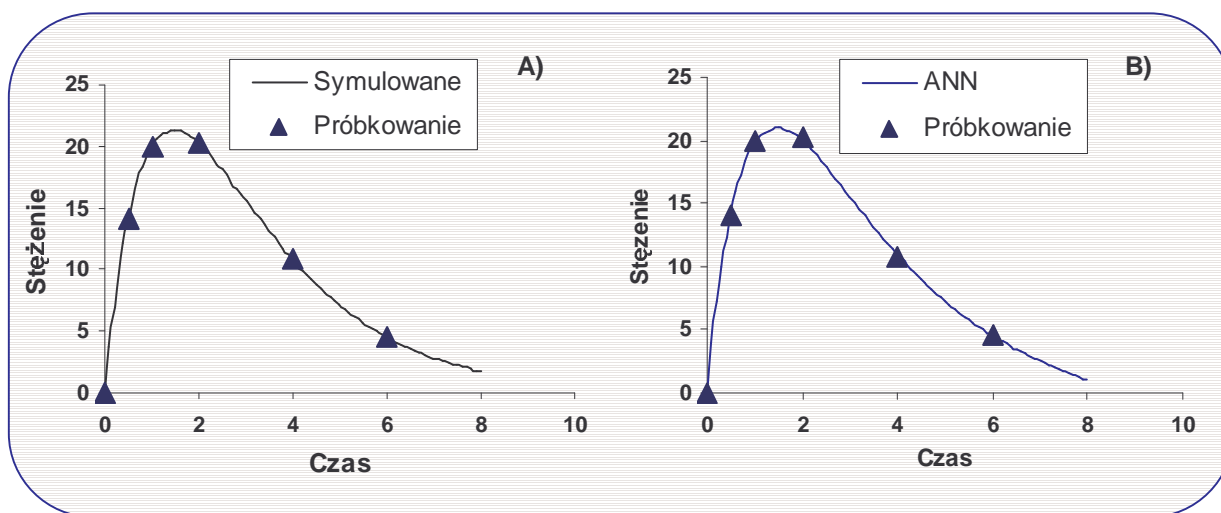
Przebadano 10 symulowanych modeli farmakokinetycznych. Obliczono średnie parametry i porównano ze sobą obliczone na podstawie symulowanych krzywych z wyznaczonymi dla modelu neuronowego. Wyniki porównania przedstawiono w tabeli 43.

Tabela 43. Porównanie średnich parametrów C_{\max} , t_{\max} i $AUC^{0 \rightarrow t}$ dla symulowanych danych i modelu neuronowego.

Parametr	Symulacja	Model neuronowy
C_{\max}	9,04	9,23
t_{\max}	1,14	1,18
$AUC^{0 \rightarrow t}$	34,17	34,03

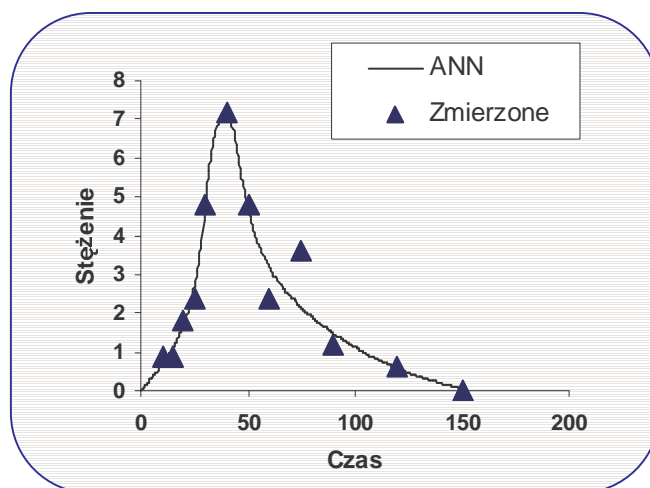
Jak widać różnice są bardzo małe, a ponadto nieistotne statystycznie.

Przykład aproksymacji jednego z symulowanych przebiegów krzywych stężenie-czas przedstawiony został na rycinie 27.



Rycina 27. Porównanie symulowanego (A) przebiegu krzywej stężenie-czas *in vivo* z aproksymowanym przez sieć neuronową (B). ANN – aproksymacja neuronowa.

Następnym etapem pracy była próba aproksymacji przebiegów zmian stężeń w żywym organizmie zmierzonych u szczurów po podaniu dożołądkowym mupirocyny w postaci liposomów. Wyniki te zostały wybrane z powodu trudności w modelowaniu klasycznymi modelami kompartmentowymi. Przykład aproksymacji przez sieć neuronową jednego z tych przebiegów przedstawiono na rycinie 28.



Rycina 28. Aproksymacja przebiegu zmian stężeń mupirocyny w osoczu szczura po podaniu dożołądkowym w postaci liposomów. ANN – aproksymacja neuronowa.

IV. OMÓWIENIE I DYSKUSJA WYNIKÓW

1. STAŁE ROZPROSZENIA

Model profilu rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych posłużył jako ilustracja możliwości odkrywania przez sieć neuronową reguł zdroworozsądkowych tj. ogólnie znanych zależności dotyczących danego zagadnienia. Jedną z doświadczalnie ustalonych reguł jest odwrotnie proporcjonalna zależność szybkości rozpuszczania substancji leczniczej od jej zawartości w stałym rozproszeniu. Testowanie optymalnej architektury sieci 3_2 hid tanh (Rycina 11) na odpowiednio przygotowanym zbiorze danych potwierdziło zdolność modelu neuronowego do identyfikacji tej reguły mimo, iż numerycznie stworzone formułacje stałych rozproszeń nie były znane w czasie uczenia (Rycina 14). Najszybsze rozpuszczanie ketoprofenu wykazała formułacja ketoprofen PVP K30 (0,5+9,5). Zwraca też uwagę nieliniowy przebieg symulowanych krzywych, co potwierdza zdolność sieci neuronowej do identyfikacji zależności nieliniowych.

Z uwagi na to, że sieci uczono w zakresie punktów czasowych od 0 do 20 min. istotne było ustalenie odpowiedzi systemu neuronowego w reakcji na dane obejmujące punkty czasowe powyżej 20 min. Stabilne zachowanie sieci neuronowej (Rycina 12A), wyrażone asymptotycznym dążeniem wykresu do wartości 100% ilości rozpuszczonej substancji leczniczej, pozwala na określenie modelu neuronowego jako wykazującego „zdrowy rozsądek”. Określenie to wiąże się z zachowaniem empirycznego modelu, który jest w stanie generalizować reguły obserwowane w rzeczywistości, w oparciu o dane eksperymentalne, bez wykorzystania wiedzy *a priori* na temat badanego zagadnienia. Wydaje się na pierwszy rzut oka oczywiste, że żadna z wartości na osi Y nie powinna przekraczać 100% ilości rozpuszczonej substancji leczniczej. Jednak sieć neuronowa nie posiada żadnej wiedzy na temat fizycznej interpretacji stosowanych jednostek (ilości w procentach). Tak więc regułę o nieprzekraczaniu 100% ilości rozpuszczonej substancji leczniczej sieć neuronowa musiała stworzyć empirycznie, na podstawie danych.

Dodatkowym i zamierzonym utrudnieniem podczas testowania omawianej sieci 3_2 hid tanh było to, iż w całym zbiorze bazowym obecne były tylko dwie formułacje stałego rozproszenia ketoprofenu z PVP K30, różniące się składem ilościowym: ketoprofen + PVP K30 (1+1) i ketoprofen + PVP K30 (1+9). Aby prawidłowo przewidzieć zachowanie się formułacji 1+9, system neuronowy musiał bazować na informacji o formułacji 1+1 oraz o formułacjach 1+9 sporządzonych z innymi nośnikami. Prawidłowe przewidywanie zachowania się formułacji 1+9 jest przykładem dobrych zdolności systemów neuronowych do generalizacji nieznanymi formułacji. Można stwierdzić, że analiza reguł tworzonych przez modele neuronowe jest w stanie dostarczyć dodatkowego narzędzia do walidacji. Jeżeli reguły systemu neuronowego są zgodne z doświadczeniem empirycznym zebranych na dany temat, to może to stanowić pośredni dowód na poprawność modelu neuronowego.

Ze względu na to, że błąd RMSE architektury 3_2 hid tanh był wyższy od 10% maksymalnej wartości zmiennej wyjściowej podjęto próby poprawy zdolności predykcyjnych modelu neuronowego. Zastosowanie sieci rozmytych nie przyniosło oczekiwanych rezultatów (Tabela 7), gdyż w tym przypadku nie posiadają one zadowalających zdolności do ekstrapolacji (Rycina 12B). Zaobserwowano spadek przewidywanych ilości rozpuszczonego ketoprofenu ze stałego rozproszenia po czasie 20 minut. W przeciwieństwie do sieci rozmytych, sieć MLP (*Multi Layer Perceptron*) prawidłowo przewidziała powolne dążenie ilości rozpuszczonej substancji leczniczej do wartości 100%.

Mimo to użycie sieci rozmytych okazało się cenne, dzięki bezpośredniej analizie i interpretacji reguł logicznych znalezionych przez system rozmyty. Stanowi to podstawową przewagę nad sieciami MLP. Założenia prac badawczych dotyczyły poprawy szybkości rozpuszczania ketoprofenu ze stałych rozprożeń. W związku z tym otrzymany zestaw reguł logicznych systemu rozmytego przeszukano pod kątem powyższych założeń. Pozwoliło to znaleźć nową, hipotetyczną formułację charakteryzującą się dużą szybkością rozpuszczania ketoprofenu z zaproponowanego układu (Rycina 13).

Wobec trudności związanych z poprawą jakości modelu neuronowego przydatną okazała się tzw. analiza wrażliwościowa jako narzędzie redukcji

wektora wejściowego. Jest to procedura matematyczna określająca względny wpływ danego wejścia na otrzymaną wartość wyjściową. Zgodnie z zasadą „przekleństwa wymiarowości” (*curse of dimensionality*) [10, 57], im większy jest rozmiar wektora wejściowego, tym trudniej jest go systemowi neuronowemu opracować. Mimo, iż sieci neuronowe radzą sobie z analizą wielowymiarowych danych znacznie lepiej niż metody statystyczne, to jednak możliwość redukcji liczby wymiarów powinna być zawsze rozważana jako próba poprawy jakości modelu neuronowego. Usunięcie mało znaczących zmiennych poprawia zwykle zdolności generalizacyjne sieci neuronowych, ponieważ przyczynia się do zmniejszenia stopnia komplikacji sieci (liczby wag) w stosunku do wielkości zbioru uczącego [94]. W konsekwencji, mniej skomplikowana sieć uczy się tylko podstawowych zależności, a nie szczegółów opisujących dany problem, czyli jest w stanie tworzyć bardziej ogólne reguły opisujące dane zagadnienie. Powyższe rozważania wiążą się z udowodnioną zależnością zdolności do uogólniania sieci neuronowej od tzw. kryterium Vapnika-Chervonenkisa (*VCdim*), które jest miarą wielkości sieci neuronowej [94]. Zależność ta jest odwrotnie proporcjonalna. Bezpośrednie obliczenie wartości *VCdim* dla sieci wielowarstwowej jest trudne. Wzór na oszacowanie dolnej i górnej granicy *VCdim* wiąże jego wartość m.in. z ilością wejść sieci neuronowej. Z uwagi na to, że jest to zależność wprost proporcjonalna, redukcja liczby wejść przyczynia się do zwiększenia zdolności generalizacyjnych sieci neuronowej.

Pierwotny model profilu rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych posiadał 7 wejść (Tabela 4). Wykonanie analizy wrażliwościowej doprowadziło do redukcji liczby wejść do 4 (Rycina 15, Tabela 8). Wyselekcjonowane zmienne (Tabela 8) można określić mianem zmiennych kluczowych. Testowanie zredukowanego modelu neuronowego wykazało przydatność analizy wrażliwościowej. Wykazano, iż błąd generalizacji był mniejszy niż dla modelu z 7 wejściami, a różnica błędów była istotna statystycznie (Tabela 9). Mimo znacznej redukcji liczby wejść możliwe było efektywne uczenie, gdyż pozostałe wejścia pozwalały na zróżnicowanie formulacji przedstawianych sieci neuronowej. Należy zaznaczyć, że o ile redukcja wartości *VCdim* poprzez usunięcie części wejść

jest korzystna z punktu widzenia zdolności generalizacji sieci neuronowej, to jednak nadmierna redukcja może spowodować usunięcie istotnych informacji niezbędnych do otrzymania prawidłowego modelu zagadnienia. W celu potwierdzenia zasadności redukcji wejść przeprowadza się zwykle weryfikację zredukowanego modelu z użyciem zbioru danych nie przedstawianego sieci podczas uczenia [94].

Weryfikacja zredukowanej sieci neuronowej została przeprowadzona z użyciem zbioru danych, zawierającego informacje na temat szybkości rozpuszczania ketoprofenu ze stałych rozprożeń i mieszanin fizycznych zawierających jako nośnik hydroksypropylometylocelulozę (HPMC). Nośnik ten stanowił nową strukturą chemiczną dla sieci neuronowej. Wyniki prac doświadczalnych w tym zakresie wykazały, że szybkość rozpuszczania ketoprofenu ze stałego rozproszenia ketoprofen + HPMC (1+1) była mniejsza niż z mieszaniny fizycznej o takim samym składzie. Sieć neuronowa, mimo że nie przewidziała dokładnie przebiegu profilu rozpuszczania ketoprofenu, to jednak była w stanie w zadowalający sposób przewidzieć lepsze wyniki rozpuszczania ketoprofenu z mieszaniny fizycznej niż z odpowiadającego jej składem stałego rozproszenia (Rycina 16).

Użycie neuronowego wstępnego przetwarzania danych miało służyć dalszej poprawie jakości neuronowego modelu profilu rozpuszczania ketoprofenu. Wstępne przetwarzanie danych jest rozumiane jako wszelkie transformacje danych pozwalające sieciom na łatwiejsze uczenie. Dotyczy to m.in. dostosowywania zakresów danych do dziedziny funkcji aktywacji, normalizacji rozkładów statystycznych, kompresji itd. Tak długo, jak nie zostaną zatracone jakiegokolwiek istotne właściwości danych, dopuszczalne są dowolne procedury wstępnego przetwarzania [81, 91]. W związku z tym w niniejszych badaniach testowano możliwości użycia sieci neuronowych jako narzędzi przetwarzających dane dla innych modeli neuronowych. Celem aproksymacji przebiegu funkcji stężenie-czas przez system **PKB 2.0** (Rycina 9) była poprawa jakości neuronowego modelu rozpuszczania ketoprofenu ze stałych rozprożeń. Dla zredukowanego do 4 wejść modelu 3_2 hid tanh wykazano, że poprawa nie dotyczyła bezpośrednio właściwości predykcyjnych, a raczej zdolności interpolacyjnych sieci neuronowej (Rycina

17). Świadczą o tym statystycznie istotnie różnice wyników testowania modeli neuronowych na zbiorach danych z wstępnym przetwarzaniem (Tabela 10). Otrzymany błąd testowania sieci uczonych na zbiorach oryginalnych różnił się o ok. 3 jednostki od sieci uczonych na danych z wstępnym przetwarzaniem. Neuronowe wstępne przetwarzanie danych wzbogaciło właściwy model profilu rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych dzięki dostarczeniu informacji niedostępnej w danych oryginalnych.

Wyniki powyższych badań skłoniły do rozbudowy bazy wiedzy. Uzyskano zbiór danych liczący 3000 rekordów, które opisują 269 formułacji stałych rozproszeń i mieszanin fizycznych. Pozwoliło to na stworzenie uniwersalnego modelu zależności profilu rozpuszczania substancji leczniczych z układów z nośnikami o zróżnicowanej budowie chemicznej. Z uwagi na wymagania związane z opisem składu formułacji liczbę wejść zwiększono do 16. Niezbędne okazało się wprowadzenie m.in. zmiennych charakteryzujących substancję leczniczą i warunki prowadzenia badań szybkości rozpuszczania (Tabela 5). Próba modelowania tak przygotowanych danych skończyła się niepowodzeniem. System neuronowy nie potrafił rozróżnić substancji leczniczej charakteryzujących się dobrą lub słabą rozpuszczalnością na podstawie tylko indeksu konektywności (CI) i masy cząsteczkowej. W większości przypadków stosowanie stałych rozproszeń ma na celu zwiększenie rozpuszczalności substancji leczniczej trudno rozpuszczalnej w wodzie. Znane są jednak przykłady zastosowania metody sporządzania stałych rozproszeń w celu spowolnienia szybkości rozpuszczania substancji leczniczych dobrze rozpuszczalnych w wodzie [65]. Tak więc obecność w jednym zbiorze uczącym dwóch rodzajów stałych rozproszeń może nastęrczać problemy w kompleksowej analizie szybkości rozpuszczania substancji leczniczej. W związku z tym celowa była rozbudowa wektora wejściowego o zmienną klasyfikującą rozpuszczalność substancji leczniczej.

Poprzez dodanie zmiennej kodującej rozpuszczalność substancji leczniczej spowodowano zwiększenie liczby wejść modelu neuronowego do 17 (Tabela 11). Tak sformatowane dane pozwalają na zachowanie

uniwersalności modelu neuronowego. Poszukiwanie optymalnej architektury sieci neuronowej doprowadziło do wyselekcjonowania pięciowarstwowej sieci 80_40_20_10 hid tanh_5mln_scale+- (Rycina 18). Zwraca uwagę jej stopień komplikacji skorelowany z rozmiarem bazy wiedzy liczącej ok. 3000 rekordów. Jest to jedna z największych sieci neuronowych użytych w niniejszej pracy. Dla zobrazowania jej stopnia komplikacji wystarczy przytoczyć całkowitą liczbę wag, która wynosi 5570. Mimo, iż zanotowano znaczny spadek błędu generalizacji w stosunku do sieci opartych na 16-elementowym wektorze wejściowym (z RMSE = 19 na RMSE = 14,7), do poprawy zdolności predykcyjnej modelu neuronowego opartego na 17-elementowym wektorze wejściowym użyto komitetu ekspertów. Komitet ekspertów złożony z 10 sieci neuronowych okazał się bardziej skutecznym narzędziem przewidywania szybkości rozpuszczania substancji leczniczych ze stałych rozproszeń i mieszanin fizycznych. Użyto średniej arytmetycznej jako elementu przetwarzającego komitetu ekspertów, gdyż zastosowanie innej sieci neuronowej w tym charakterze nie przyniosło spodziewanej poprawy jakości modelu. Wartość błędu generalizacji RMSE wyniosła 13,4. Uzyskano więc dodatkowy spadek błędu o 1,3 jednostki.

Otrzymano znacznie lepsze wyniki modelu neuronowego opartego na wektorze wejściowym składającym się z 17 elementów i zawierającym zmienną klasyfikującą rozpuszczalność substancji leczniczej (Tabela 11), niż dla sieci opartej na 16-elementowym wektorze wejściowym bez klasyfikacji rozpuszczalności substancji leczniczej w wodzie (Tabela 5). Stwierdzone błędy wskazują na to, że CI i masa cząsteczkowa substancji leczniczej nie są wystarczająco reprezentatywnymi zmiennymi dla określenia jej rozpuszczalności.

Weryfikacja zdolności generalizacyjnych otrzymanych modeli neuronowych przeprowadzona została w celu oceny możliwości przewidywania nieznanymi składów ilościowych, jak też i jakościowych stałych rozproszeń. Wyniki testowania niektórych architektur sieci neuronowych za pomocą procedury *leave-one-out* pozwalają stwierdzić, że modele neuronowe są w stanie dobrze przewidywać profil rozpuszczania substancji leczniczych z nieznanymi składów ilościowych stałych rozproszeń

(Tabela 13). Zastosowanie specyficznego podziału zbioru bazowego ze względu na substancje lecznicze ujawniło gorsze zdolności generalizacyjne w zakresie składów jakościowych formulacji. Dotyczy to przypadku, kiedy nie jest znana systemowi neuronowemu budowa chemiczna substancji leczniczej, której profil rozpuszczania ze stałego rozproszenia należy przewidzieć.

Weryfikacja systemu **SD_expert** przeprowadzona na wybranych przykładach pozwoliła na stwierdzenie przydatności **SD_expert** do optymalizacji ilościowego składu formulacji względem zadanego profilu rozpuszczania substancji leczniczej (Rycina 19). Praca systemu polegała na równoczesnej optymalizacji wartości dwóch wejść określających zawartość substancji leczniczej i nośnika. Nie zawsze wartość jednego parametru stanowiła dopełnienie wartości drugiego do jedności. Warunek taki zawsze powinien być spełniony dla dwuskładnikowych stałych rozproszeń. Stąd też w kolumnie „Znalezione” (Tabela 12) jako odpowiedzi systemu podano dwie propozycje systemu przyjmując raz przewidzianą zawartość substancji leczniczej, a drugi raz zawartość nośnika za podstawę do obliczenia składu formulacji. Znalezienie właściwych składów ilościowych w 6 na 8 testowanych przypadków wskazuje na dobre własności generalizacyjne systemu **SD_expert** w zakresie wspomaganie doboru składu ilościowego w nieznannej formulacji. Należy w tym miejscu przypomnieć, że sieci neuronowe użyte w tym eksperymencie nie miały w swojej bazie wiedzy formulacji przedstawianych im do znalezienia. Tym samym potwierdzona została funkcja doradcza systemu **SD_expert**.

Eksperyment z dodawaniem jednego rekordu zbioru testowego do zbioru uczącego pozwolił na ukazanie możliwości zastosowania systemu neuronowego do *screeningu in silico* [80] w celu znalezienia optymalnej postaci leku. Niewielki dodatek informacji uczącej spowodował spadek błędu testowania o ok. 6 jednostek. Prowadzi to do wniosku, że aby wykorzystać w pełni możliwości systemu neuronowego należy wykonać limitowaną serię badań doświadczalnych, których wyniki posłużą następnie do douczenia sieci neuronowych. Tak douczony system neuronowy jest bardziej precyzyjny i może znacznie efektywniej wspomagać poszukiwania nowej formulacji.

Neuronowy model profilu rozpuszczania dowolnej substancji leczniczej ze stałych rozproszeń z wybranymi nośnikami posłużył jako kolejny przykład zależności stopnia komplikacji optymalnej architektury od wielkości zbioru uczącego. W oparciu o wyselekcjonowane dane opisujące jedną grupę nośników, tj. polioksyetylenoglikole, zbudowano specjalistyczny model neuronowy przewidujący profil rozpuszczania różnych substancji leczniczych ze stałych rozproszeń. Bazowy zbiór danych liczył ok. 800 rekordów. Spośród 20 wytrenowanych architektur sieci neuronowych wybrano optymalną architekturę 20_7 hid tanh (Rycina 20). Architektura ta okazała się znacznie mniejsza niż sieć 80_40_20_10 hid tanh_5mln_scale+- będąca optymalną dla opisanego powyżej uniwersalnego modelu profilu rozpuszczania substancji leczniczych ze stałych rozproszeń z nośnikami o różnej budowie chemicznej (Rycina 18).

2. MIKROEMULSJE

Modele neuronowe opracowane dla układów mikroemulsyjnych są, podobnie jak dla stałych rozproszeń, modelami predykcyjnymi. Z punktu widzenia metodyki, budowa neuronowego modelu zależności powstawania mikroemulsji od rodzaju i ilości poszczególnych składników opierała się na takich samych zasadach jak dla stałych rozproszeń. Jednak pomimo użycia w procesie poszukiwania najlepszego modelu niejednokrotnie identycznych, jak dla stałych rozproszeń, architektur sieci neuronowych, optymalne architektury dla układów mikroemulsyjnych znacznie się różniły.

Skonstruowano opis jakościowego i ilościowego składu mikroemulsji z użyciem wektora wejściowego liczącego 12 zmiennych (Tabela 15). Rozmiar wektora wejściowego okazał się wystarczający do zróżnicowania 118 formułacji mikroemulsji znajdujących się w bazie wiedzy. Dane z piśmiennictwa jako baza wiedzy systemu służyły trenowaniu i selekcji sieci neuronowych.

Optymalna sieć neuronowa, znaleziona w toku eksperymentów numerycznych, posiadała pięciowarstwową, skomplikowaną architekturę

20_16_8_4 hid tanh (Rycina 23). Jej błąd całkowity (EG) wyniósł około 17% (Tabela 16) przy optymalnym progu decyzyjnym dla tej architektury. Architektura sieci z czterema warstwami ukrytymi jest rzadko spotykana w piśmiennictwie. Znane jest twierdzenie o tym, iż wystarczy jedna warstwa ukryta do przybliżenia dowolnej funkcji przez sieć neuronową [52]. Jednak z doświadczeń własnych jak i z pojawiających się ostatnio doniesień o użyciu więcej niż jednej warstwy ukrytej w sieciach typu MLP [5, 31] wynika, że bardziej skomplikowane sieci mogą być skuteczniejsze w modelowaniu niż sieci dwuwarstwowe. Tak też i stało się w przypadku modelowania układów mikroemulsyjnych, gdzie pięciowarstwowa sieć neuronowa okazała się być najlepszą spośród testowanych sieci jedno- i wielowarstwowych (testowano sieci z liczbą warstw maksymalnie do 6). Istotą powodzenia wielowarstwowych architektur wydaje się być ich elastyczność wynikająca z bardziej zróżnicowanej topologii oraz poszczególnych przekształceń realizowanych przez kolejne warstwy ukryte. Przekształcenia te można często utożsamiać z kompresją danych, która jeśli jest odpowiednio przeprowadzona, może ułatwić pracę dalszym warstwom i pozwolić na wyłuskanie najistotniejszych informacji. Jest to analogiczny proces jak w przypadku sieci z kontrapropagacją [52, 114], z tym że tutaj kompresja informacji jest bardziej stopniowa. Należy też w tym miejscu zauważyć, że nadmierna ilość warstw może przyczyniać się do destabilizacji sieci neuronowej [77] objawiającej się np. jednakową odpowiedzią na wszystkie wzorce uczące i testowe. Takie zachowanie było niejednokrotnie obserwowane w czasie eksperymentów numerycznych, chociaż dotyczyło zazwyczaj nadmiernie długiego treningu, podczas którego pojawiał się tzw. efekt „zmęczenia” sieci [81].

Sporządzono 55 formułacji mikroemulsji, kierując się przewidywaniami optymalnej architektury 20_16_8_4 hid tanh, które złożyły się na weryfikacyjny zbiór danych (PRAKT). Potwierdzono obecność układów izotropowych w 15 spośród 55 zaproponowanych formułacji. Duża wartość błędu całkowitego osiągnięta podczas testowania zbioru PRAKT (Tabela 18) wynikała z obecności formułacji zawierających składniki, które nie występowały dotychczas w zbiorze uczącym, jak np. Brij® 58. Oznacza to, że

sieć 20_16_8_4 hid tanh jest w stanie stosunkowo poprawnie generalizować skład ilościowy mikroemulsji, natomiast generalizacja składu jakościowego jest trudniejsza. Z uwagi na to, zmodyfikowano model neuronowy poprzez budowę komitetów ekspertów i wstępne przetwarzanie danych.

Komitety ekspertów budowano w oparciu o najlepsze z trenowanych architektur sieci podczas 10-krotnego wzajemnego sprawdzania jako alternatywne rozwiązanie w stosunku do pojedynczej sieci 20_16_8_4 hid tanh. Wykazano znacznie mniejszy błąd całkowity testowania na zbiorze PRAKT niż osiągnięty przez pojedynczą sieć neuronową (Tabela 19). Mimo, iż ilość trafnych odpowiedzi zmniejszała się wraz z błędem całkowitym EG, to jednak była ona wyższa niż w przypadku pojedynczej sieci 20_16_8_4 hid tanh przy wartości progu decyzyjnego 90%. Próg ten został wybrany *a priori* z uwagi na napotkane trudności w jego optymalizacji, jak też i założenia dotyczące funkcji systemu neuronowego. Z punktu widzenia funkcji doradczej systemu korzystne jest, aby jak najmniejsza ilość odpowiedzi fałszywie pozytywnych FD^+ przypadała na jak największą liczbę trafień GD^+ . Dzięki temu decyzja systemu o występowaniu mikroemulsji charakteryzuje się wysokim prawdopodobieństwem trafienia.

Opisany komitet ekspertów stał się podstawą systemu **ME_expert** zaprojektowanego z myślą o poszukiwaniu optymalnego składu mikroemulsji względem zadanych kryteriów. Przeprowadzono eksperymenty z poszukiwaniem mikroemulsji o dużej zawartości fazy wodnej. Iteracyjna praca systemu **ME_expert** polegała na doborze składu jakościowego do założonego składu ilościowego. W wyniku eksperymentów numerycznych zaproponowane zostało 8 formuł potencjalnych mikroemulsji (Tabela 20), które charakteryzowały się jednakowym składem ilościowym (Rycina 24). Weryfikacja doświadczalna przewidywań systemu neuronowego przedstawiona w tabeli 21 w pełni potwierdza skuteczność neuronowego systemu wspomagania poszukiwania składu mikroemulsji. Błąd całkowity EG rzędu 23% przy trafności GD^+ ok. 77% to wynik, który należy uznać za bardzo dobry. Należy również podkreślić, że zaproponowane przez system **ME_expert** formułacje zawierają składniki mikroemulsji nieznane systemowi neuronowemu. Potwierdza to, że dobrze wytrenowany model neuronowy

może ekstrapolować właściwości substancji pomocniczych i generalizować w sposób prawidłowy powstające na ich bazie formułacje. Zwraca również uwagę niski poziom ilości odpowiedzi fałszywie pozytywnych FD^+ .

Masters [81] oraz Marciniak i wsp. [91] zwrócili uwagę, że dobór właściwej metody przetwarzania danych może decydować o poprawie jakości modelu neuronowego. Z tych względów zastosowano modyfikacje wstępnego przetwarzania danych, które dotyczyły zarówno sposobu skanowania danych z graficznych plików obrazujących trójkąty fazowe jak też i późniejszej obróbki danych cyfrowych.

Opracowane dwa różne algorytmy skanowania plików graficznych obrazujących trójkąty fazowe znacznie się od siebie różnią (Rycina 22). Skanowanie równomierne co 2% na osi zawartości składników mikroemulsji (**scan_2_2**) pozwala na otrzymanie jednolitego obrazu trójkąta fazowego niezależnie od wielkości obszaru zajmowanego przez mikroemulsję, zaś algorytm **Intelliscan** wykrywa głównie granice przejść fazowych. Procedura **Intelliscan** programu **Phase_diagr** została zaprojektowana z myślą o znacznej kompresji danych z jednoczesnym zachowaniem dokładności opisu trójkąta fazowego. Algorytm **Intelliscan** wykorzystuje prostą redukcję ilości pikseli z zeskanowanej linii poprzez podzielenie ich ilości przez wartość współczynnika kompresji. Jedynymi odstępstwami od tej reguły są piksele, których sąsiedzi mają inny kolor, co jest równoznaczne z zachowaniem granic jednakowych obszarów. Założono, iż sieci neuronowe będą w stanie nauczyć się zasady, że w środku obszaru ograniczonego pikselami opisującymi obecność mikroemulsji lub jej brak, nie ma żadnych przejść fazowych. Mimo swej prostoty **Intelliscan** pozwolił na efektywną digitalizację danych graficznych i przedstawienie tak przetworzonych danych sieciom neuronowym.

Wyniki 10-krotnego wzajemnego sprawdzania dla sieci 20_16_8_4 hid tanh_scan_2_2 uczonej na zbiorze danych **scan_2_2** są znacznie lepsze niż dla tej samej sieci uczonej na danych uzyskanych z algorytmu **Intelliscan** (Tabela 16 i 35). Błąd całkowity EG wynosił ok. 10%. Podczas testowania na zbiorze PRAKT błąd całkowity EG przy progu decyzyjnym 90% wyniósł ok

23% (Tabela 36). Jest to ponad trzykrotnie większa wartość błędu EG w porównaniu do architektury uczonej na danych **Intelliscan** (Tabela 18 i 36).

Bezpośrednie porównywanie tych dwóch sieci jest jednak niemożliwe z uwagi na inny rozmiar zbiorów testowych i dystrybucję klas wynikającą z odmiennego charakteru algorytmów skanowania. Inny jest też skład zbiorów testowych z powodu losowej procedury wyboru formułacji podczas tworzenia par zbiorów uczący-testowy przez program **Learn_conv**. Biorąc pod uwagę opisane powyżej właściwości zbiorów danych, przeprowadzono trening sieci 20_16_8_4 hid tanh w systemie leave-one-out zarówno na danych pochodzących ze skanowania **Intelliscan** jak też i **scan_2_2**. Miało to na celu udowodnienie wpływu algorytmu skanowania bitmap na jakość modelu neuronowego. Każdy z wytrenowanych modeli przetestowano na danych testowych **Intelliscan** i **scan_2_2**. Trenowanie architektury 20_16_8_4 hid tanh w trybie leave-one-out zapobiegło problemom z dystrybucją danych w zbiorach testowych, ponieważ ilość formułacji w obydwu przypadkach była taka sama (118), a tryb wyboru danych do zbiorów testowych był deterministyczny. Kolejność wyłączenia poszczególnych formułacji ze zbioru bazowego do testowego podczas tworzenia par zbiorów uczący-testowy była taka sama zarówno w przypadku danych **Intelliscan** jak i **scan_2_2**. Dzięki temu, testowanie np. sieci uczonej na zbiorze **Intelliscan** z użyciem zbioru testowego **Intelliscan**, czy odpowiadającego mu **scan_2_2** jest zawsze generalizacją. Umożliwiło to bezpośrednie porównanie średnich wyników generalizacji sieci 20_16_8_4 hid tanh uczonych na danych **Intelliscan** z sieciami uczonymi na danych **scan_2_2** (Tabela 37 do 40). Analiza wyników wskazuje, że mniejszy jest błąd testowania na danych **scan_2_2** niż na danych **Intelliscan**. Stwierdzono, że rodzaj zbiorów uczących nie jest w tym przypadku czynnikiem istotnym. Testowanie na danych **Intelliscan** zależy od danych uczących. Sieci uczone na danych **scan_2_2** uzyskały lepsze wyniki generalizacji danych **Intelliscan**. Różnica błędów całkowitych EG ok. 1,5 % dla optymalnych progów decyzyjnych nie jest istotna statystycznie (Tabela 38 i 40). Wskazuje to na brak wpływu sposobu skanowania danych graficznych na jakość modelu neuronowego. Z uwagi na mniejszy rozmiar zbiorów uczących **Intelliscan**, a tym samym mniejszy nakład obliczeniowy

niezbędny do stworzenia modelu neuronowego. Korzystniejsze wydało się więc stosowanie tak otrzymanych zbiorów uczących do przygotowania modeli neuronowych.

Modyfikacje danych liczbowych polegały na przekształcaniu tych danych w celu ułatwienia uczenia sieci neuronowych. Wszystkie modyfikacje przeprowadzono tylko dla danych z *Intelliscan*, co wynikało z poprzednich rozważań dotyczących braku wpływu algorytmu skanowania bitmap na jakość sieci neuronowych. Jedynym wyjątkiem była procedura binaryzacji, której użyto również w przypadku zbiorów *scan_2_2*, co można z kolei uzasadnić znaczną różnicą ilości wejść w stosunku do reprezentacji ciągłej. Inna ilość wejść może mieć ewentualny wpływ na wrażliwość tak skonstruowanych sieci na sposób skanowania bitmap, co należało zweryfikować.

Skalowanie linowe w zakresie od 0,2 do 0,8 stosowane było rutynowo wszędzie tam, gdzie nie dokonywano innych przekształceń danych, jako podstawowy sposób dostosowywania danych do dziedziny funkcji aktywacji modeli neuronowych.

O ile procedury normalizacji i binaryzacji stanowią matematyczne przekształcenia, o tyle dodawanie reguł zdroworozsądkowych to już świadoma ingerencja użytkownika w dane uczące polegająca na dodaniu informacji o niemożności stworzenia mikroemulsji bez jednego z trzech jej podstawowych składników.

Wyniki badań wykazały, że normalizacja (Tabela 22) nie jest lepszym rozwiązaniem od skalowania liniowego (Tabela 16). System neuronowy oparty na znormalizowanych danych nie posiada zdolności generalizacyjnych, co uwidocznili również testowanie na zbiorze PRAKT (Tabela 23). Przyczyną może tu być m.in. rodzaj zastosowanej normy euklidesowej. Doniesienia na temat neuronowych modeli składu mikroemulsji zdają się potwierdzać małą skuteczność normalizacji danych jako metody wstępnego przetwarzania [106].

Binaryzacje potraktowano w niniejszej pracy jako zmianę reprezentacji wektora wejściowego z ciągłej na binarną poprzez przekodowanie danych z systemu dziesiętnego na dwójkowy. Wprawdzie w wyniku zastosowania

referencyjnej sieci 20_16_8_4 hid tanh nie uzyskano mniejszego błędu (Tabela 24) niż sieć uczona na danych ciągłych (Tabela 16), to jednak już sieć rozmyta 60 hid fuzzy_M-SLIDE wykazywała nieznaczną poprawę wyników badań (Tabela 25).

Zaprojektowano komitet ekspertów dla modeli opartych na danych binarnych. Jako element przetwarzający wybrano sieć neuronową o takiej samej architekturze jak dla danych ciągłych – 7_5_3 hid fsr. Wyniki testowania na zbiorze PRAKT (Tabela 26) nie odbiegają znacznie od wyników uzyskanych dla komitetu ekspertów opartego na danych skalowanych linowo w zakresie od 0,2 do 0,8 (Tabela 19). Widoczne jest to zwłaszcza dla progu decyzyjnego 90%.

W celu wykazania wpływu procesu binaryzacji na jakość modelu neuronowego przeprowadzono eksperyment z doprowadzeniem błędu całkowitego EG komitetu ekspertów uczonego na danych binarnych, do takiego samego poziomu, co błąd komitetu ekspertów uczonego na danych ciągłych. Dokonano tego poprzez zmianę progu decyzyjnego. Obydwa systemy neuronowe wykazały podobną charakterystykę odpowiedzi zarówno FD^+ jak i GD^+ (Tabela 28). Użycie sieci o architekturze 15_7_5 hid tanh w charakterze elementu przetwarzającego komitetu ekspertów dla danych binarnych pozwoliło poprawić wyniki. Testowanie na zbiorze PRAKT wskazuje, że odpowiedzi sieci 15_7_5 hid tanh są praktycznie binarne, tj. nie występują wartości pośrednie pomiędzy 0 a 1 (Tabela 27). Spontaniczna dychotomizacja odpowiedzi systemu neuronowego jest niezwykle cenna z uwagi na brak problemu optymalizacji progu decyzyjnego. Wprawdzie błąd uzyskany dla tej sieci jest wyższy ($EG = 9\%$), ale za to poziom prawidłowych decyzji GD^+ jest znacznie wyższy, bo aż ok. 34%. Stosując opisaną wyżej procedurę doprowadzania błędu całkowitego do jednakowego poziomu porównano architektury 15_7_5 hid tanh będące elementami przetwarzającymi komitetów ekspertów opracowanych dla danych binarnych i ciągłych. Osiągnięto różnicę ok. 13% prawidłowo sklasyfikowanych odpowiedzi GD^+ (Tabela 29) na korzyść systemów opartych na danych binarnych.

Komitet ekspertów dla danych skalowanych liniowo w zakresie od 0,2 do 0,8 składał się z 10 sieci neuronowych uczonych na danych oryginalnych, podczas gdy komitet ekspertów dla danych binarnych zbudowano w oparciu o 6 sieci uczonych na danych oryginalnych. Z tego względu zdecydowano, że obok porównywania odpowiadających sobie architektur sieci drugiego rzędu, należy porównać ze sobą optymalne architektury sieci drugiego rzędu. Zasada eksperymentu polegała na doprowadzeniu jednej z sieci neuronowych do poziomu błędu całkowitego EG drugiej. Kryterium różnicowania były wartości GD^+ i FD^+ . Wyniki zastosowania powyższej procedury do porównania między sobą sieci 15_7_5 hid tanh (dane binarne) z siecią 7_5_3 hid fsr (dane skalowane liniowo w zakresie od 0,2 do 0,8) świadczą o przydatności techniki binaryzacji w konstruowaniu neuronowego modelu mikroemulsji (Tabela 30). Różnica w wartości GD^+ pomiędzy obydwoma sieciami wynosiła 14%. Dodatkowo wykazano, iż dla każdej reprezentacji danych istnieje inna, optymalna architektura sieci neuronowej.

Mimo zastosowania binaryzacji danych pochodzących ze skanowania równomiernego co 2% nie uzyskano lepszych wyników predykcji (Tabela 41). Stanowi to potwierdzenie skuteczności stworzonego algorytmu kompresji **Intelliscan**. Wydaje się on w zupełności wystarczający do digitalizacji danych z trójkątów fazowych bez utraty jakości modelu neuronowego.

Dodawanie reguł zdroworozsądkowych potraktowano jako metodę wstępnego przetwarzania danych mającą za zadanie wzbogacenie wiedzy systemu neuronowego o doświadczenia empiryczne. Jako przykład wzięto pod uwagę znaną zasadę o niezbędności trzech podstawowych składników (tenzyd, faza olejowa i wodna) do powstania mikroemulsji. Odpowiednio zmodyfikowane dane dodane zostały do zbioru uczącego, za pomocą którego wytrenowano wyselekcjonowaną wcześniej optymalną architekturę 20_16_8_4 hid tanh. Wyniki testowania sieci 20_16_8_4 hid tanh na danych obrazujących reguły zdroworozsądkowe wskazują na to, że sieć jest w stanie prawidłowo przewidywać brak mikroemulsji podczas nieobecności któregokolwiek z podstawowych składników (Tabela 31). Na tym samym zbiorze przetestowano sieć 20_16_8_4 hid tanh uczoną bez dodatku reguł zdroworozsądkowych. Wyniki testowania (Tabela 32) nie pozostawiają

wątpliwości, że sieci uczone na oryginalnych danych, same w czasie procesu uczenia stworzyły reguły zdroworozsądkowe dotyczące konieczności użycia poszczególnych składników do powstania mikroemulsji. Uzasadnieniem powyższego stwierdzenia są zerowe wartości błędów EG i FD^+ oraz GD^+ , które wynikają z odpowiedzi sieci reprezentujących brak mikroemulsji dla wszystkich rekordów zbioru testowego, co jest zgodne z założoną postacią reguły zdroworozsądkowej.

Testowanie sieci 20_16_8_4 hid tanh uczonej na danych wzbogaconych o reguły zdroworozsądkowe na zbiorze PRAKT ujawniło trudności z generalizacją tak przygotowanego systemu (Tabela 33). Wskazuje na to zerowa wartość GD^+ , która oznacza, że system nie przewidział w żadnym przypadku obecności mikroemulsji. Obserwowana nadmierna zachowawczość sieci neuronowej może wynikać z niewielkiej liczby rekordów opisujących występowanie mikroemulsji w stosunku do wielkości całego zbioru uczącego. Oznacza to nierówną dystrybucję klas w zbiorze uczącym. Może to spowodować, że mniej licznie reprezentowana klasa będzie w czasie uczenia traktowana przez sieć neuronową jako zakłócenie, a nie ważna przesłanka do stworzenia reprezentatywnego modelu mikroemulsji. Niebezpieczeństwo potraktowania ważnej informacji jako szumu przez sieć neuronową wynika ze sposobu uczenia sieci. Uczenie polega na iteracyjnym przedstawianiu sieci neuronowej poszczególnych rekordów danych i dobieraniu wartości wag po obliczeniu błędu sieci dla danego wzorca uczącego. Prezentacja rekordów danych do uczenia odbywa się w sposób losowy z wyłączeniem [81] tj. każdy rekord raz prezentowany sieci jest wyłączany z uczenia dopóki nie zostanie przedstawiony sieci cały zbiór uczący. Nie ma więc tutaj preferencji dla żadnych rekordów danych, co oznacza, że to ile razy sieć spotka się z daną klasą wartości wyjściowych zależeć będzie tylko od dystrybucji klas w zbiorze uczącym. Od ilości prezentacji danej klasy zależy to, czy sieć zapamięta tę klasę jako istotną, czy też potraktuje jako chwilowe zaburzenie w danych. Pewnym paradoksem jest to, że w takiej sytuacji zdolności generalizacyjne sieci neuronowej, za które odpowiedzialna jest m.in. umiejętność ignorowania rzadko występujących klas wyjść, mogą stać się przyczyną złego uwarunkowania modelu.

W celu wyjaśnienia, czy nierówna dystrybucja klas w zbiorze uczącym jest przyczyną trudności z generalizacją, przeprowadzono dodatkowo eksperyment z równoważeniem ilości rekordów dla każdej z klas odpowiedzi systemu neuronowego.

Wyniki testowania na zbiorze PRAKT sieci 20_16_8_4 hid tanh uczonej na zbiorze wzbogaconym o reguły zdroworozsądkowe i poddanym równoważeniu klas (Tabela 34) odbiegają od poprzednio uzyskanych dla sieci bez równoważenia klas (Tabela 33). Równoważenie klas doprowadziło do nadmiernego „optymizmu” sieci neuronowej popełniającej teraz znacznie więcej błędów FD⁺. Wydaje się, że takie zachowanie może być spowodowane znaczną różnicą w wielkości zbiorów uczących. Zbiór danych wzbogacony o reguły zdroworozsądkowe liczył ok. 250 000 rekordów, zaś ten sam zbiór poddany procesowi równoważenia klas ok. 470 000 rekordów. Architektura 20_16_8_4 hid tanh, chociaż optymalna dla oryginalnych danych (zbiór bazowy ok. 65 000 rekordów), jest niewystarczająca dla większych zbiorów danych. Pewną przesłanką uzasadniającą powyższe twierdzenie jest fakt zakończenia uczenia sieci 20_16_8_4 hid tanh uczonej na danych z równoważeniem klas, już po ok. 6 mln. iteracji, chociaż trenowano ją do 20 mln iteracji. Należy więc stwierdzić, że wprowadzenie dodatkowo sztucznie utworzonych danych nie wpłynęło na poprawę jakości modelu neuronowego. Sztuczne dodawanie rekordów uczących okazało się nie tylko szkodliwe dla modelu neuronowego, ale również niepotrzebne z uwagi na to, iż sieć neuronowa stworzyła samodzielnie regułę o niezbędności podstawowych składników mikroemulsji.

3. APROKSYMACJA FUNKCJI STĘŻENIE-CZAS W BADANIACH *IN VITRO* I *IN VIVO*.

Sieci neuronowe są nazywane uniwersalnymi aproksymatorami z uwagi na swoje własności adaptacyjne będące rezultatem zdolności do uczenia się [52, 77, 91, 94, 109, 114, 136]. Aproksymacja różnego rodzaju funkcji stała się podstawowym zadaniem, w którym sieci neuronowe udowodniły swoją przydatność. Jest to również jeden z głównych sposobów testowania jakości nowych algorytmów uczących.

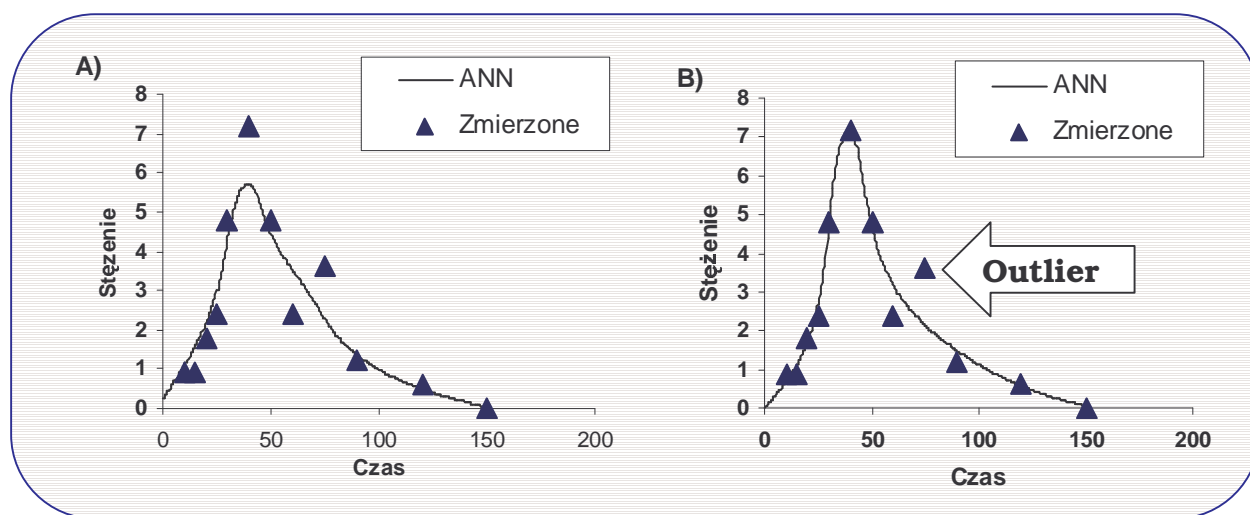
Do aproksymacji funkcji stężenie-czas wykorzystano w większym stopniu zdolności sieci neuronowych do interpolacji niż do ekstrapolacji. Oznacza to, że model neuronowy powinien dokładnie przewidywać przebieg krzywej stężenie-czas w zakresie bazy wiedzy. Nie jest wymagane od modelu neuronowego, aby przewidywał wyniki spoza zakresu bazy wiedzy.

System **PKB 2.0** skonstruowany do przeprowadzenia aproksymacji funkcji stężenie-czas bazował na dwóch sieciach neuronowych (Rycina 25), które pracowały w charakterze komitetu ekspertów. Założono, że ta sama architektura uczona wielokrotnie od losowo dobranych startowych wartości wag znajdzie inne minimum lokalne funkcji celu, bliskie minimum globalnemu. Taki typ pracy sieci znany jest pod nazwą multistartu [77, 94]. Dodatkowo, obie sieci A i B różniły się kryteriami zatrzymania uczenia: sieć A uczona była maksymalnie do 500 000 iteracji i minimalnego błędu RMSE = 0,02, podczas gdy sieć B uczona była maksymalnie do 1mln iteracji i błędu RMSE = 0,01. Tak więc model neuronowy aproksymowanej funkcji powstawał dwuetapowo: w pierwszej kolejności dobierany był ogólny charakter modelu przez sieć A, a w drugim etapie sieć B dopasowywała szczegółową postać modelu neuronowego.

Przykład aproksymacji przebiegu stężeń mupirocyny w osoczu szczura (Rycina 29) ilustruje opisany powyżej tok pracy systemu neuronowego. Na rycinie 29A przedstawiono wstępny etap uczenia sieci neuronowej. Widoczne jest słabe dopasowanie modelu do danych w przeciwieństwie do dokładnej aproksymacji przedstawionej na rycinie 29B, która obrazuje końcowy etap

uczenia i uśredniania modeli neuronowych. Zwraca uwagę także zdolność sieci neuronowych do odrzucania wyników obarczonych dużymi błędami (*outliers*).

Wykorzystanie danych symulowanych pozwoliło na weryfikację dokładności aproksymacji zarówno przebiegu krzywej stężenie-czas *in vitro* jak i *in vivo*. Porównanie parametrów obliczonych bezpośrednio z równań symulowanych przebiegów krzywych stężenie-czas z otrzymanymi przez sieci neuronowe pozwala stwierdzić, że system neuronowy jest w stanie dorównać regresji logarytmicznej (Rycina 26, Tabela 42), jak również prawidłowo przewidywać proste modele kompartmentowe (Rycina 27, Tabela 43). Przykład regresji logarytmicznej wskazuje, że jeśli jest znana postać matematyczna modelu opisującego dane zjawisko, to metody statystyczne dają dokładniejsze wyniki niż sieć neuronowa, chociaż różnica pomiędzy obydwojema modelami nie jest istotna statystycznie. Należy również zwrócić uwagę na niewielką ilość danych na podstawie których sieci musiały stworzyć modele przebiegu zmian stężeń, tj. 11 punktów czasowych w przypadku badań *in vitro* i 6 w przypadku badań *in vivo*.



Rycina 29. Dwa etapy aproksymacji funkcji stężenie-czas przez system *PKB 2.0*. A - etap początkowy. B - etap końcowy. Przebieg stężeń mupirocyny w osoczu szczura po podaniu dożołądkowym mupirocyny w postaci liposomów.

Modele neuronowe można próbkować z dowolną dokładnością co znajduje zastosowanie przede wszystkim w całkowaniu numerycznym. System **PKB 2.0** jest wyposażony w moduł obliczania pola powierzchni pod krzywą stężenie-czas (AUC). Wykorzystano do tego celu metodę trapezów należącą do tzw. kwadratur Newtona-Cotesa. Dokładność metody trapezów zależy odwrotnie proporcjonalnie od długości przedziału próbkowania całkowanej funkcji [38]. Istotne jest więc próbkowanie aproksymowanej funkcji małymi przedziałami o jednakowej długości. Stanowi to wymóg prawidłowego stosowania metody trapezów. Właściwości takie są oferowane przez system **PKB 2.0**.

Modelowanie danych doświadczalnych również okazało się możliwe do wykonania za pomocą systemu **PKB 2.0**. Użycie tego systemu pozwoliło dopasować krzywą stężenie-czas do bardziej skomplikowanego przebiegu zmian stężeń mupirocyny w osoczu szczura po podaniu dożołądkowym (Rycina 28). Nie powiodło się modelowanie powyższych danych z użyciem prostych modeli kompartmentowych. W tej sytuacji używając klasycznych narzędzi należałoby znaleźć odpowiedni, niestandardowy model farmakokinetyczny i dobrać jego parametry. Jest to zadanie trudne, z uwagi na to, że przestrzeń możliwych rozwiązań jest nieograniczona. W celu znalezienia odpowiedniego modelu byłoby konieczne testowanie nieskończenie wielu postaci funkcji stężenie-czas. Zastosowane sieci neuronowe znalazły samodzielnie optymalny model zmian stężeń mupirocyny w osoczu szczura. Warto więc podkreślić raz jeszcze automatyczny proces znajdowania postaci modelu, który wobec żmudnego poszukiwania odpowiedniej funkcji metodami klasycznymi staje się podstawową przewagą sieci neuronowych użytych w zadaniach aproksymacyjnych.

W zakresie prac dotyczących biofarmaceutycznych aspektów formułowania postaci leku dotychczas wykorzystywano tylko predykcyjne właściwości modeli neuronowych [22, 33, 116]. W tym kontekście cenne wydają się możliwości aproksymacyjne udostępniane przez system **PKB 2.0** dla badań *in vitro* i *in vivo*, tym bardziej wobec faktu, że ilość prac dotyczących tego zagadnienia nie jest znaczna. Być może jest to

spowodowane koniecznością posiadania dużych baz danych, co w przypadku danych pochodzących z badań *in vivo* nierzadko jest trudne do osiągnięcia.

PODSUMOWANIE

Zaletą wykorzystania sieci neuronowych jest fakt, iż stopień komplikacji hipotetycznej funkcji opisującej dane zagadnienie jest mniej istotny niż w przypadku modelowania statystycznego. Sieci neuronowe są w stanie identyfikować wielowymiarowe zależności w sposób automatyczny i w oparciu o dane doświadczalne je modelować i generalizować. Równie ważna jest zdolność sieci neuronowych do tworzenia modeli nieliniowych, które są trudne do otrzymania metodami klasycznymi [36, 52, 77, 94, 114].

Na przykładzie aproksymacji funkcji logarytmicznej stwierdzono, że jeżeli znany jest model statystyczny zagadnienia, to nie jest możliwe uzyskanie lepszego modelu neuronowego niż wyspecjalizowany model statystyczny. Tak więc zastosowanie sieci neuronowych powinno się ograniczać do problemów, które nie mogą być opisane przez modele statystyczne, lub gdy zakładane modele nie są wystarczająco precyzyjne.

Modelowanie neuronowe może też być rozpatrywane jako wstępny etap opracowywania teorii opisującej dane zagadnienie. W tym kontekście atrakcyjną wydaje się być możliwość ekstrakcji reguł z sieci rozmytych lub MLP, co może dostarczyć łatwej do interpretacji wiedzy nabytej przez system w czasie uczenia.

Z uwagi na pozytywne wyniki badań nad modelowaniem neuronowym można sądzić, że opracowane neuronowe systemy ekspertowe **ME_expert**, **SD_expert** znajdą zastosowanie w projektowaniu formulacji mikroemulsji i stałych rozproszeń na zasadzie *screeningu in silico* [80]. Świadczą o tym m.in. zakończone sukcesem poszukiwania nowych składów mikroemulsji przeprowadzone za pomocą systemu **ME_expert**. Wykazana skuteczność systemu **PKB 2.0** w zadaniach aproksymacyjnych pozwala sądzić, że system ten znajdzie zastosowanie w opracowywaniu różnorodnych przebiegów krzywych stężenie-czas w badaniach *in vitro* i *in vivo*. Dodatkowym polem zastosowań systemu **PKB 2.0** stanie się też prawdopodobnie wstępne

przetwarzanie danych dla modeli neuronowych rozpuszczania substancji leczniczych ze stałych rozproszeń.

Zakres zastosowań sieci neuronowych jest praktycznie nieograniczony, co potwierdzają doniesienia z piśmiennictwa [91, 114]. Wydaje się, że modele empiryczne, jakimi są sieci neuronowe, stają się coraz powszechniej akceptowanym narzędziem modelowania w sytuacjach, gdy nie ma statystycznej alternatywy [36, 78].

Powyższe rozważania oraz przedstawione wyniki badań wskazują, że opracowane oprogramowanie i metodyka użycia sieci neuronowych w technologii postaci leku i biofarmacji pozwolą w przyszłości konstruować modele dla innych postaci leku. Dzięki temu zostaną uzyskane cenne narzędzia wspomagające proces optymalizacji postaci leku. Przygotowanie modeli neuronowych opisujących inne postaci leku jest możliwe m.in. dzięki opracowanej bazie specjalistycznych programów narzędziowych służących do przekształcania danych na potrzeby sieci neuronowych. Wraz ze skonstruowanym symulatorem **Nets2003** stanowią one standardowe narzędzia do modelowania neuronowego.

Z drugiej strony bezpośrednia ingerencja w kod źródłowy pozwoli na ewentualne dostosowywanie tych narzędzi do specyfiki konkretnego problemu. Przykładem może tu być program **Phase_diagr** zaprojektowany specjalnie z myślą o skanowaniu trójkątów fazowych w celu digitalizacji danych graficznych. Edycja kodu źródłowego opracowanego symulatora sieci neuronowych **Nets2003** pozwala na jego ulepszanie i wykorzystanie najnowszych trendów w uczeniu sieci neuronowych.

Skonstruowane neuronowe systemy ekspertowe **ME_expert**, **SD_expert** i **PKB 2.0** są tzw. systemami o otwartej architekturze. Oznacza to, że po drobnych modyfikacjach interfejsu oraz wyposażeniu ich w odpowiednio przygotowane sieci neuronowe, można modelować za ich pomocą zupełnie inne niż dotychczas zależności. Tak więc elastyczność opracowanych neuronowych systemów ekspertowych wynika z automatycznego procesu budowy maszyny wnioskującej polegającego na uczeniu sieci neuronowych z użyciem odpowiednich danych. Proces ten wymaga minimalnej interwencji użytkownika w porównaniu do żmudnego i

czasochłonnego kodowania wiedzy w klasycznych systemach ekspertowych [78, 89].

V. WNIOSKI

1. Opracowane oprogramowanie pozwala na skuteczną symulację modeli neuronowych niezbędnych do analizy danych w technologii postaci leku, a także biofarmacji. Systemy ekspertowe **ME_expert** i **SD_expert** oraz **PKB 2.0** spełniają prawidłowo zadania, do których zostały zaprojektowane.
2. Neuronowe modele predykcyjne mogą być wykorzystane do ukierunkowania prac doświadczalnych, których celem jest znalezienie optymalnej formulacji danej postaci leku. W tym kontekście modele neuronowe można traktować jako maszyny wnioskujące neuronowych systemów ekspertowych. Wspomaganie procesu formulacji postaci leku odbywa się na drodze screeningu *in silico*.
3. Możliwość stosowania jednolitej metodyki modelowania neuronowego oraz jednolitego oprogramowania zarówno w przypadku modelowania składu mikroemulsji, jak i zależności profilu rozpuszczania substancji leczniczej od składu stałych rozproszeń, wskazuje na uniwersalny charakter sieci neuronowych.
4. Sieci neuronowe pozwalają na budowę uniwersalnego modelu zależności profilu rozpuszczania substancji leczniczej od składu stałych rozproszeń.
5. Modele neuronowe, opracowane dla układów mikroemulsyjnych mogą generalizować skład nowych formulacji. Model neuronowy 15_7_5 hid tanh należy uznać jako optymalny.

6. Generalizacja przez modele neuronowe ilościowego składu nowych formułacji farmaceutycznych jest bardziej precyzyjna od generalizacji nowego składu jakościowego.
7. Aproksymacyjne modele neuronowe wykorzystane do obliczania parametrów krzywych stężenie-czas w badaniach *in vitro* i *in vivo* charakteryzują się porównywalną dokładnością z metodami statystycznymi. Uniwersalny charakter modeli neuronowych pozwala na aproksymację dowolnych przebiegów krzywych przez jeden system neuronowy.
8. Sieci neuronowe zdolne są do selekcji zmiennych kluczowych za pomocą analizy wrażliwościowej. Właściwa selekcja zmiennych kluczowych może poprawić jakość modelu neuronowego.
9. Dodawanie reguł zdroworozsądkowych do danych uczących w celu poprawy jakości modelu neuronowego nie jest potrzebne jeżeli model już takie reguły sam znalazł na podstawie danych empirycznych.
10. Zespoły sieci neuronowych w postaci komitetu ekspertów są z reguły sprawniejsze niż pojedyncze sieci neuronowe.
11. Sieci neuronowe oparte na logice rozmytej mogą stanowić komplementarne narzędzie modelowania wraz z sieciami typu MLP.

VI. STRESZCZENIE

Niniejsza praca dotyczyła modelowania neuronowego w technologii postaci leku i biofarmacji.

W celu oceny zdolności sieci neuronowych do generalizacji nowych formuacji farmaceutycznych przygotowano neuronowe modele profilu rozpuszczania substancji leczniczych ze stałych rozproszeń oraz składu mikroemulsji. Opracowanie neuronowych modeli aproksymujących krzywe stężenie-czas w badaniach *in vitro* i *in vivo* służyło weryfikacji założenia o uniwersalności modelowania neuronowego w zakresie aproksymacji funkcji jednej zmiennej.

Szeroki zakres podejmowanych zagadnień wymagał użycia różnorodnego oprogramowania takiego jak: symulator sieci neuronowych, skanery bitmap, arkusze kalkulacyjne, czy też narzędzia obróbki plików tekstowych. W celu zapewnienia maksymalnej elastyczności i wydajności oraz dostosowania do specyficznych wymogów prowadzenia eksperymentów numerycznych, podjęto prace nad zaprojektowaniem i wykonaniem zestawu niezbędnych programów komputerowych. Realizacja powyższego zadania obejmowała projektowanie algorytmów, struktur danych, formatów plików, graficznych interfejsów użytkownika, a także implementację metod numerycznych. Jako język programowania wybrano Borland Delphi® 5.0.

Opracowano neuronowe modele predykcyjne przewidujące skład mikroemulsji i profil rozpuszczania substancji leczniczej ze stałych rozproszeń.

Wykazano, że zastosowanie odpowiednio przygotowanych systemów neuronowych, przede wszystkim w formie zespołów sieci neuronowych, stanowi interesujące rozwiązanie przy opracowywaniu postaci leków. Zaprojektowano też specjalne programy **ME_expert** i **SD_expert** służące do prowadzenia screeningu *in silico* dla odpowiednich postaci leku: mikroemulsji i stałych rozproszeń. Obydwa systemy oparte na sieciach neuronowych pozwoliły na generalizację nowych formuacji. Skuteczność przewidywań nowych składów mikroemulsji za pomocą modelu neuronowego

wynosiła 77%. Tym samym została potwierdzona doradcza funkcja neuronowych systemów ekspertowych.

Opracowany na potrzeby modeli składu mikroemulsji algorytm **Intelliscan** pozwolił na uzyskanie znacznej kompresji danych bez utraty jakości kodowanej informacji. Dzięki temu możliwe było szybsze uczenie sieci neuronowych z uwagi na mniejszy koszt obliczeniowy opracowania zbiorów danych z mniejszą ilością rekordów.

W modelowaniu składu mikroemulsji wykazano, że sieci neuronowe z czterema warstwami ukrytymi stanowią optymalne architektury niezbędne do rozwiązania postawionego problemu.

Oprócz własności predykcyjnych, wykorzystano z powodzeniem także zdolności sieci neuronowych do aproksymacji funkcji w analizie danych w technologii postaci leku, jak również w biofarmacji. Skonstruowano system **PKB 2.0**, który używając dwóch sieci neuronowych pozwala na aproksymację praktycznie dowolnego przebiegu krzywych stężenie-czas w badaniach *in vitro* i *in vivo* z zadaną dokładnością. Wykazano, iż aproksymacja neuronowa, jakkolwiek nieco mniej precyzyjna, może być utożsamiana z metodą regresji. Przedstawiono też przykład użycia systemu **PKB 2.0** do aproksymacji przebiegu krzywej stężenie-czas *in vivo*, dla którego nie udało się znaleźć prostego modelu farmakokinetycznego. System **PKB 2.0** znalazł automatycznie odpowiedni model przebiegu krzywej stężenie-czas i pozwolił na obliczenie odpowiednich parametrów farmakokinetycznych. Dodatkowo użyto z powodzeniem systemu **PKB 2.0** do wstępnego przetwarzania danych na potrzeby modeli profilu rozpuszczania substancji leczniczych ze stałych rozprożeń. Wykazano pozytywny wpływ neuronowego wstępnego przetwarzania danych na zdolności interpolacyjne modeli neuronowych profilu rozpuszczania substancji leczniczych ze stałych rozprożeń.

Wyniki badań potwierdzają słuszność postawionych założeń. Potwierdzono możliwość skutecznego zastosowania sieci neuronowych w technologii postaci leku, a także w biofarmacji. Wykazano uniwersalny charakter sieci neuronowych używając jednolitej metodyki, a także

oprogramowania do konstruowania modeli neuronowych różnych postaci leku.

VII. SYMBOLE

Spis symboli użytych w pracy:

- ART** – rodzaj sieci uczonej bez nadzoru (*Adaptive Resonance Theory*)
- AUC** – pole pod krzywą stężenie-czas
- BAM** - typ sieci rekurencyjnej (*Bidirectional Associative Memory*)
- C_{max}** – stężenie maksymalne
- CP** – sieć z kontrapropagacją
- d** - błąd sieci neuronowej
- E** - funkcja błędu
- EG** – całkowity błąd klasyfikacji
- FD⁺** – ilość fałszywych decyzji o wystąpieniu mikroemulsji
- f_{sr}** – logarytmiczna funkcja aktywacji wg Bilskiego
- GD⁺** – ilość prawidłowych decyzji o wystąpieniu mikroemulsji
- i, j** – indeksy komórek w sieciach neuronowych
- k** – krotność podziału zbioru bazowego w metodzie wzajemnego sprawdzania
- MAE** – miara błędu (*mean absolute error*)
- MLP** – preceptron wielowarstwowy (*Multi Layer Perceptron*)
- M-SLIDE** – metoda defuzyfikacji stosowana w sieciach rozmytych
- RMSE** – miara błędu (*root mean squared error*)
- S** – wektor stanów układu i równocześnie wyjść sieci neuronowych
- t** – krok uczenia sieci neuronowej
- t_{max}** – czas osiągnięcia stężenia maksymalnego
- W** – wektor wag
- w_{ij}** – pojedyncza waga
- X** – wektor parametrów układu i równocześnie wejść sieci neuronowych
- x_i** – pojedyncze wejście sieci neuronowej
- α** i **η** – współczynniki uczenia
- Δw_{ij}** – zmiana wagi

VIII. PIŚMIENNICTWO

1. Aboofazeli R., Lawrence M.J. Investigations into the formation and characterization of phospholipid microemulsions.I. Pseudo-ternary phase diagrams of systems containing water-lecithin-alcohol-isopropyl mirystate. *Int. J. Pharm.* 93,1993, 161-175.
2. Aboofazeli R., Lawrence M.J. Investigations into the formation and characterization of phospholipid microemulsion; *Int. J. Pharm.* 125, 1995, 107-116.
3. Aboofazeli R., Lawrence M.J. Investigations into the formation and characterization of phospholipid microemulsions. III. *Int. J. Pharm.* 111, 1994, 63-73.
4. Abu-Mostafa Y.S. Uczące się maszyny. *Świat Nauki*, 46, 6, 1995.
5. Agatonovic-Kustrin S., Alany R.G. Role of genetic algorithms and artificial neural networks in predicting the phase behavior of colloidal delivery systems. *Pharm Res.* 2001, 18, 7,1049-55.
6. Agatonovic-Kustrin S., Alany R.G., Rades T., Tucker I.G., Use of Artificial Neural Networks To Predict Quaternary Phase Systems From Limited Experimental Data, *Proc. 2nd World Meeting APGI/APV*, Paris 25-28 May 1998.
7. Agatonovic-Kustrin S., Beresford R., Razzak M. Determination of enantiomeric composition of ibuprofen in solid state mixtures of the two by by DRIFT spectroscopy. *Analytica Chimica Acta*, 417, 2000, 31-39.
8. Antikainen O. Modelling of Granule Growth in a Fluidized Bed Granulator with Fuzzy Logic. *World Meeting ADRITELF/AVP/APGI*, Florence, 8/11 April 2002. p. 215.
9. Attwood D. A study on factors influencing the droplet size in nonionic oil-in-water microemulsions. *Int. J. Pharm.* 88, 1992, 417-422.

10. Bengio, S., Bengio, Y. Taking on the Curse of Dimensionality in Joint Distributions Using Neural Networks. *IEEE T. Neur. Networ.* 2000, 11, 550-557.
11. Bhargawa H.N., Narurkar A., Lieb L.M. Using microemulsions for drug delivery - *Pharm. Techn.* 3, 46, 1987
12. Bilski J. The backpropagation learning with logarithmic transfer function. *Proceedings of the Fifth Conference Neural Networks and Soft Computing, Zakopane 2000*, 71-77.
13. Błasiak A., Mendyk A., Jachowicz R. Projektowanie zbioru uczącego neuronowego modelu szybkości rozpuszczania substancji leczniczych ze stałych rozproszeń i mieszanin fizycznych. XVIII Naukowy Zjazd PTFarm „Farmacja w XXI wieku”, Poznań 20-22.09.2001, Streszczenia, S17, str. 849.
14. Bourquin J., Shmidli H., van Hoogevest P., Leuenberger H. Advantages of Artificial Neural Networks (ANNs) as alternative modeling technique for data sets showing non-linear relationship using data from a galenical study on a solid dosage form. *Eur.J.Pharm.Sci.*, 7, 1998, 5-16.
15. Bourquin J., Shmidli H., van Hoogevest P., Leuenberger H. Comparison of artificial neural networks (ANN) with classical modeling techniques using different experimental designs and data from a galenical study on a solid dosage form, *Eur.J.Pharm.Sci.*, 6, 1998, 287–300.
16. Bourquin J., Shmidli H., van Hoogevest P., Leuenberger H. Pitfalls of artificial neural networks (ANN) modeling technique for data sets containing outlier measurements using a study on mixture properties of a direct compressed dosage form. *Eur.J.Pharm.Sci.*, 7, 1998, 17-28.
17. Brier M. E., Aronoff G. R. Application of artificial neural networks to clinical pharmacology. *Int. Jour. Clin. Pharm. Ther.* 34, 510, 1996.
18. Brier M. E., Smith B. P.J. Statistical Approach to Neural Network Model Building for Gentamycin Peak Predictions., *Pharm.Sci.*, 85, 65, 1996.

19. Brier M. E., Żurada J. M. Neural Network Predicted Peak and Trough Gentamicin Concentrations. *Pharm.Res.*, 12, 406, 1995.
20. Buciński A., Nasal A., Kaliszan R. Pharmacological Classification of Drugs Based on Neural Network Processing of Molecular Modeling Data. *Combinatorial Chemistry & High Throughput Screening*, 3, 2000, 525-533.
21. Budavari S. – editor. *The Merck Index*. Merck&Co. 11th ed. Rahway, N.J., USA 1989.
22. Chen Y., McCall T.W., Baichwal A.R., Meyer M.C. The application of an artificial neural network and pharmacokinetic simulations in the design of controlled-release dosage form., *J Contr Release* 59, 1999, 33-41
23. Chevalier Y., Zemb T. The structure of micelles and microemulsions - *Rep. Prog. Phys.* 53, 270, 1990.
24. Chiba Y., Kohri N., Iseki K., Miyazaki K. Improvement of dissolution and bioavailability for mebendazol, an agent for human echinococcosis, by preparing solid dispersions with polyethylene glycol. *Chem.Pharm.Bull.* 39, 8, 1991, 2158-2160.
25. Chiou W.L., Niazi S. Phase diagram and dissolution – rate studies on sulfathiazol – urea solid dispersions. *J.Pharm.Sci.* 60, 9, 1971.
26. Chiou W.L., Riegelman S. Pharmaceutical application of solid dispersion systems. *J. Pharm. Sci.*, 60, 1971, 1281-1302.
27. Chow H-H., Tolle K.M., Roe D.J., Elsberry V., Chen H. Application of Neural Networks to Population Pharmacokinetic Data Analysis. *J.Pharm.Sci.* 86, 7, 1997, 840-845.
28. Cierniak R., Cpałka K. The analysis of fetal heart rate using neural network and neuro-fuzzy approaches. *Proceedings of the Fourth Conference Neural Network and Their Applications, Zakopane 1999*, 502-508.
29. Constantinides P.P., Scalart J-P. Formulation and physical characterization of water-in-oil microemulsions containing long-versus medium-chain glycerides. *Int. J. Pharm.* 158, 1997 57-68.
30. Danek A. *Chemia fizyczna*. PZWL, Warszawa, 1982.

31. Degim T., Hadgraft J., Ilbasimis S., Ozkan Y. Prediction of Skin Penetration Using Artificial Neural Network (ANN) Modeling. *J. Pharm. Sci.* 92, 3, 2003, 656-664.
32. Dekker M., Attwood D. Colloidal Drug Delivery Systems. Edited by Jörg Kreuter. Inc. New York, 1994, 31-71.
33. Dowell J., Hussain A., Devane J., Young D., Artificial Neural Networks Applied to the In Vitro - In Vivo Correlation of an Extended-Release Formulation: Initial Trials and Experience. *J. Pharm. Sci.* 88, 1, 1999, 154-160.
34. Ebube N., Owusu – Ababio G., Adeyeye Ch.M. Preformulation studies and characterization of the physiochemicak properties of amorphous polymers using artificial neural networks. *Int.J.Pharm.* 196, 2000, 27-35.
35. El-Gindy N.A., Karara A.H., Abd El-Khalek M.M. Enhanced dissolution rate of papaverine via the solid dispersion technique. *Sci.Pharm.* 44, 4, 1976, 283-288.
36. Erb, R. The Backpropagation Neural Network - A Bayesian Classifier. *Clin. Pharmacokinet.* 1995, 29, 69-79.
37. Eshra A.G., Naggar V.F., Boraie N.A. A study of indometacin – nicotinamide solid dispersions. *Pharm.Ind.*, 48, 12, 1986
38. Fortuna Z., Macukow B., Wąsowski J. *Metody numeryczne.* WNT, Warszawa 1993.
39. Fawaz F., Bonini F., Guyot M., Bildet J., Maury M., Lagueny A.M. Bioavailability of norfloxacin from PEG 6000 solid dispersion and cyclodextrin inclusion complexes in rabbits. *Int. J. Pharm.* 132, 1996, 271-275.
40. Fiedler H.P. *Lexikon der Hilfsstoffe.* Editio Cantor Verlag Aulendorf 1996.
41. Francés C., Veiga M.D., Español O.M., Cadórniga R. Preparation, characterization and dissoluton of ciprofloxacin/PEG 6000 binary systems. *Int.J.Pharm.*, 77, 1991, 193-198.

42. Fu A., Wong M.H., Sze S.C., Wong W.C., Wong W.L., Yu W.K. Finding Fuzzy Sets for the Mining of Fuzzy Association Rules for Numerical Attributes. <http://citeseer.nj.nec.com/>.
43. Fuji M., Hasegawa J., Kitajima H., Matsumoto M. The solid dispersions of bezodiazepins with phosphatidylcholine. The effect of substituents of bezodiazepins on the formulation of solid dispersions. *Chem.Pharm.Bull.* 39, 11, 1991, 3013-3017
44. Gašperlin M., Tušar L., Tušar M., Kristl J., Šmid-Korbar J., Lipophilic semisolid emulsion systems: viscoelastic behavior and prediction of physical stability by neural network modeling. *Int. J. Pharm.* 168, 1998, 243-254.
45. Gašperlin, M., Tušar, L., Tušar, M., Šmid-Korbar, J., Zupan, J., Kristl, J. Viscosity prediction of lipophilic semisolid emulsion systems by neural network modeling. *Int. J. Pharm.* 2000, 196, 37-50.
46. Gobburu V.S., Chen E.P., Artificial Neural Networks As a Novel Approach to Integrated Pharmacokinetic - Pharmacodynamic Analysis. *J. Pharm. Sci.* 85, 5, 1996.
47. Gobburu V.S., Shelver W.H. Quantitative Structure-Pharmacokinetic Relationship (QSPR) of Beta Blockers Derived Using Neural Networks. *J. Pharm. Sci.* 84, 7, 1996, 862-865.
48. Goh W.Y., Lim C.P., Peh K.K., Predicting Drug Dissolution Profiles With an Ensemble of Boosted Neural Networks: A Time Series Approach. *IEEE T. Neur. Networ.* 11, 2, 2003, 459-463.
49. Govindaraju, V., Ianakiev, K. Potential Improvement of Classifier Accuracy by Using Fuzzy Measures. *IEEE T. Fuzzy Syst.*, 8, 2000, 679-690.
50. Grave A., Mielck J.B. Chemical stability of octreotide lyophilized from buffered solutions and in buffered solid dispersion with polycarbophil. 4-th World Meeting ADRITELF/APGI/APV, Florence, April 2002, 435-436.

51. Harrald P.G., Kamstra M. Evolving Artificial Neural Networks to Combine Financial Forecasts. *IEEE T. Evol. Comput.* 1, 1, 1997, 40-52.
52. Hertz J., Krogh A., Palmer R.G. Wstęp do teorii obliczeń neuronowych. wyd. II. WNT Warszawa 1995.
53. Hinton G.E. Jak sieci neuropodobne uczą się na podstawie doświadczeń. *Świat Nauki*, 11, 1992.
54. <http://www.cnbc.cmu.edu/Resources/PDP++//PDP++.html>
55. <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
56. Ho H-O., Hsiao C-C., Sheu M-T., Preparation of Microemulsion Using Polyglycerol Fatty Acid Esters as Surfactant for the Delivery of Protein Drugs. *J. Pharm. Sci.* 85, 2, 1996.
57. Hong X., Harris Ch.J. Generalized Neurofuzzy Network Modeling Algorithms Using Bézier-Bernstein Polynomial Functions and Additive Decomposition. *IEEE T. Neur. Networ.*, 11, 4, 2000, 839-902.
58. Hussain A.S., Johnson R.D., Vachcarajani N.N., Ritschel W.A. Feasibility of Developing a Neural Network for Prediction of Human Pharmacokinetic Parameters from Animal Data. *Pharm. Res.*, 10, 3, 1993, 466-469.
59. Hussain A.S., Shivanand P., Johnson R.D. Application of neural computing in pharmaceutical product development: computer aided formulation design. *Drug Dev. Ind. Pharm.* 1994, 20, 1739-1752.
60. Hussain A.S., Yu X., Johnson R.D. Application of Neural Computing in Pharmaceutical Product Development. *Pharm. Res.*, 8, 1991, 1248-1252.
61. Huuskonen J., Salo M., Taskinen J. Neural Network Modeling for Estimation of the Aqueous Solubility of Structurally Related Drugs. *J. Pharm. Sci.*, 86, 4, 1997, 450-454.
62. Inghelbrecht S., Remon J-P., de Aguiar P.F., Walczak B., Massart D. L., Van De Velde F., De Baets P., Vermeersch H., De Backer P. Instrumentation of a roll compactor and the evaluation of the

- parameter settings by neural networks. *Int. J. Pharm.* 148, 1, 1997, 103-115.
63. Jachowicz R. Dostępność farmaceutyczna oksazepamu ze stałego rozproszenia. *Farmacja Polska* 50, 3, 1994.
64. Jachowicz R. Możliwości technologii postaci leku w rozwiązywaniu skutecznej terapii lekami peptydowymi. *Farmacja Polska* 53, 1997, 584-588.
65. Jachowicz R. Stałe rozproszenia jako układy o kontrolowanym uwalnianiu substancji leczniczych. *Farm.Pol.* 55, 3, 1999.
66. Jachowicz R., Nürnberg E., Pieszczek B., Kluczykowska B., Maciejewska A. Solid dispersion of ketoprofen in pellets. *Int. J. Pharm.* 206, 2000, 13-21.
67. Kandimalla K. K., Kanikkannan N., Singh M. Optimization of a vehicle mixture for the transdermal delivery of melatonin using artificial neural networks and response surface method. *J.Contr.Release.*, 61, 1999, 71-82.
68. Khan G.M., Jiabi Z. Preparation, characterization, and dissolution studies of ibuprofen solid dispersions using polyethylene glycol (PEG), talc, and PEG-talc as dispersion carriers. *Drug.Dev.Ind.Pharm.* 24, 5, 1998, 455-462
69. Kier L.B., Hall L.H. Molecular Connectivity VII: specific treatment of heteroatoms. *J. Pharm. Sci.* 65, 12, 1976, 1806-9.
70. Kier L.B., Hall L.H., Murray W.J. Molecular Connectivity II: Relationship to Water Solubility and Boiling Point; *J. Pharm. Sci.* 12, 65, 1976.
71. Kier L.B., Hall L.H., Murray W.J. Molecular Connectivity III: Relationship to Partition Coefficients. *J. Pharm. Sci.* 64, 12, 1975, 1978-81.
72. Kier L.B., Hall L. H., Murray W. J. Molecular Connectivity V: Connectivity Series Concept Applied to Density; *J. Pharm. Sci.* 8, 65, 1976.

73. Kier L.B., Hall L. H., Murray W. J., Randic M. Molecular Connectivity I: Relationship to Nonspecific Local Anesthesia; *J. Pharm. Sci.* 12, 64, 1975, 1971-1974.
74. Kluczykowska B. Mikroemulsje – nowe postacie leku. *Farmacja Polska* 19, 52, 1996.
75. Kluczykowska B., Dziekan A., Jachowicz R., The effect of water phase components on the formation of o/w microemulsion. Proceedings of 3rd World Meeting on Biopharmaceutics Pharmaceutical Technology, Berlin 2000, 573-574.
76. Kohonen T., Self-organization and associative memory. Springer-Verlag Berlin, Heidelberg, 1984.
77. Korbicz J., Obuchowski A., Uciński D. Sztuczne sieci neuronowe - podstawy i zastosowania. Akademska Oficyna Wydawnicza PLJ Warszawa 1994.
78. Lisboa P.J.G. A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Networks* 15, 2002, 11-39.
79. Liu, Z., Yan, F. Fuzzy Neural Network in Case-Based Diagnostic System. *IEEE T. Fuzzy Syst.*, 5, 1997, 209-222.
80. Lohmann R., Voss H. In Silico Screening, Predictable Properties in Drug Development. *New Drugs.* 4, 2002, 46-48.
81. Masters T. Sieci neuronowe w praktyce. Programowanie w języku C++. Wydawnictwa Naukowo-Techniczne Warszawa 1996.
82. McCulloch J.L., Pitts W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 1943
83. Mendyk A., Jachowicz R. A simple approach to neural optimisation of pharmaceutical dosage form. Fifth conference "Neural networks and soft computing", Zakopane, 6-10.06.2000.
84. Mendyk A., Jachowicz R. Perspektywy zastosowania sztucznych sieci neuronowych w technologii postaci leku i biofarmacji. XVIII Naukowy Zjazd PTFarm „Farmacja w XXI wieku”, Poznań 20-22.09.2001, Streszczenia, S17, str. 836.

85. Mendyk A., Jachowicz R. Próby przewidywania za pomocą sieci neuronowych profilu rozpuszczania ketoprofenu ze stałych rozproszeń i mieszanin fizycznych”. XVII Naukowy Zjazd Polskiego Towarzystwa Farmaceutycznego „Farmacja w perspektywie XXI w.” Kraków 10-12.09.1998.
86. Mendyk A., Jachowicz R. Sztuczne sieci neuronowe – nowe narzędzie badawcze w farmacji. *Farm. Pol.* 54, 19, 1998.
87. Mendyk A., Jachowicz R., Błasiak A., Neural Expert System for Pharmaceutical Formulation Development – Focus on Solid Dispersions. Proceedings of the Sixth International Conference on Neural Networks and Soft Computing, Zakopane, Poland, June 11-15, 2002 eds. Leszek Rutkowski, Janusz Kacprzyk . Heidelberg, New York : Physica-Verlag, 2003. (Advances in Soft Computing), 480-485.
88. Mendyk A., Jachowicz R., Chłosta S., Dziekan A., “Neural Expert System for Optimization of Microemulsions Quantitative and Qualitative Composition.” 4th World Meeting ADRITELF/AVP/APGI, Florence, 8/11 April 2002, 1519-1520.
89. Mulawka J. W. Systemy ekspertowe. Wydawnictwa Naukowo Techniczne, Warszawa 1996.
90. Müller R.H., Hildebrand G.E. Z oryg. niem. Tłum. Jachowicz R., Kubis A.A., Klawe J. Technologia nowoczesnych postaci leku - PZWL Warszawa, 1998
91. Nałęcz M. - red. Biocybernetyka i inżynieria biomedyczna 2000. Tom 6. Sieci neuronowe. Duch. W., Korbicz J., Rutkowski T., Tadeusiewicz R., - red. tomu, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2000.
92. Neelima J.K., Allen L.V. Studies on microemulsions using Brij96 as surfactant and glycerin, ethylene glycol and propylene glycol as cosurfactants. *Int. J. Pharm.* 57, 1989, 87-93.
93. Okonogi S., Oguchi T., Yonemochi E., Puttipatkhachorn S., Yamamoto K. Improved dissolution of ofloxacin via solid dispersion. *Int. J. Pharm.* 156, 1997, 175-180.

94. Osowski S. Sieci neuronowe. Oficyna Wydawnicza Politechniki Warszawskiej Warszawa 1996.
95. Owusu-Ababio G., Ebube N.K., Reams R., Habib M. Comparative dissolution studies for mefenamic acid – polyethylene glycol solid dispersions systems and tablets. *Pharm.Dev.Technol.*, 3, 3, 1993, 405-412
96. Ozeki T., Yuasa H., Kanaya Y., Oishi K. Application of the solid dispersion method to the controlled release of medicine. V. Suppression mechanism of the medicine release rate in the three – component solid dispersion system. *Chem.Pharm.Bull.* 42, 2, 1994, 337-343
97. Ozeki T., Yuasa H., Kanaya Y. Application of the solid dispersion method to the controlled release of medicine. IX. Difference in the release of flurbiprofen from solid dispersions with poly(ethylene oxide) and hydroxypropylcellulose and interaction between medicine and polymers. *Int.J.Pharm.* 155, 1997, 209-217.
98. Peh K.K., Lim C.P., Quek S.S., Khoh K.H. Use of Artificial Neural Networks to Predict Drug Dissolution Profiles and Evaluation of Network Performance Using Similarity Factor. *Pharm. Res.* 17, 11, 2000, 1384-1388.
99. Perissutti B., Rubessa F., Princivalle F. Solid dispersions of carbamazepine with Gelucire 44/14 and 50/13. *S.T.P. Pharma Sciences*, 10, 6, 2000, 479-484.
100. Piegat A. Modelowanie i sterowanie rozmyte. Akademicka Oficyna Wydawnicza EXIT Warszawa 1999.
101. Polak S., Brandys J., Mendyk A. Sztuczne sieci neuronowe jako narzędzie analizy danych w epidemiologii i ekonomice zdrowia. *Polski Przegląd Kardiologiczny*, 4, 2002, 79-83.
102. Polak S., Mendyk A., Brandys J. Artificial Neural Network as an Internet-based Tool for Initial Screening of Gestational Diabetes Mellitus. International Conference on E-health in Common Europe. Kraków 05-06.06.2003.

103. Polak S., Mendyk A., Brandys J., Using Artificial Neural Network as a Tool for Epidemiological Data Analysis. Proceedings of the Sixth International Conference on Neural Networks and Soft Computing, Zakopane, Poland, June 11-15, 2002 eds. Leszek Rutkowski, Janusz Kacprzyk. Heidelberg, New York : Physica-Verlag, 2003. (Advances in Soft Computing), 486-491.
104. Polański J., Self-organizing neural networks for pharmacophore mapping. *Adv. Drug Delivery Rev.* 55, 9, 2003, 1149-1162.
105. Prince L.M. Microemulsions, theory and practice - Academic Press, New York, 1977.
106. Richardson C.J., Mbanefo A., Aboofazeli R., Lawrence M.J., Barlow D.J. Prediction of Phase Behavior in Microemulsion Systems Using Artificial Neural Networks. *J Colloid Interface Sci.* 1997, 2, 187, 296-303.
107. Rocksloh K., Rapp F.R., Abed Abu S., Müller W., Reher M., Gauglitz G., Schmidt P.C., Optimization of Crushing Strength and Disintegration Time of a High-Dose Plant Extract Tablet by Neural Networks., *Drug Dev Ind Pharm* 25, 9, 1999, 1015-1025.
108. Rosenblatt F. Principles of Neurodynamics. New York: Spartan 1962.
109. Rutkowska D., Piliński M., Rutkowski L., Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. PWN, Warszawa-Łódź 1997.
110. Saint Ruth H. Attwood D., Ktistis G., Taylor C.J., Phase studies and particle size analysis of oil-in-water phospholipid microemulsions. *Int. J. Pharm.* 116, 1995, 253-261.
111. Seikawa H., Fukuda N., Takada M., Ohtani K., Arita T., Nakano M. Dissolution behavior and gastrointestinal absorption of dicumarol from solid dispersion systems of dicumarol – polyvinylpyrrolidone and dicumarol – β -cyclodextrin. *Chem.Pharm.Bull.* 31, 4, 1983, 1350-1356.
112. Stolorz P., Lapedes A., Xia Y. Predicting protein secondary structure using neural net and statistical methods. *J.Mol.Biol.* 225, 2, 1992.

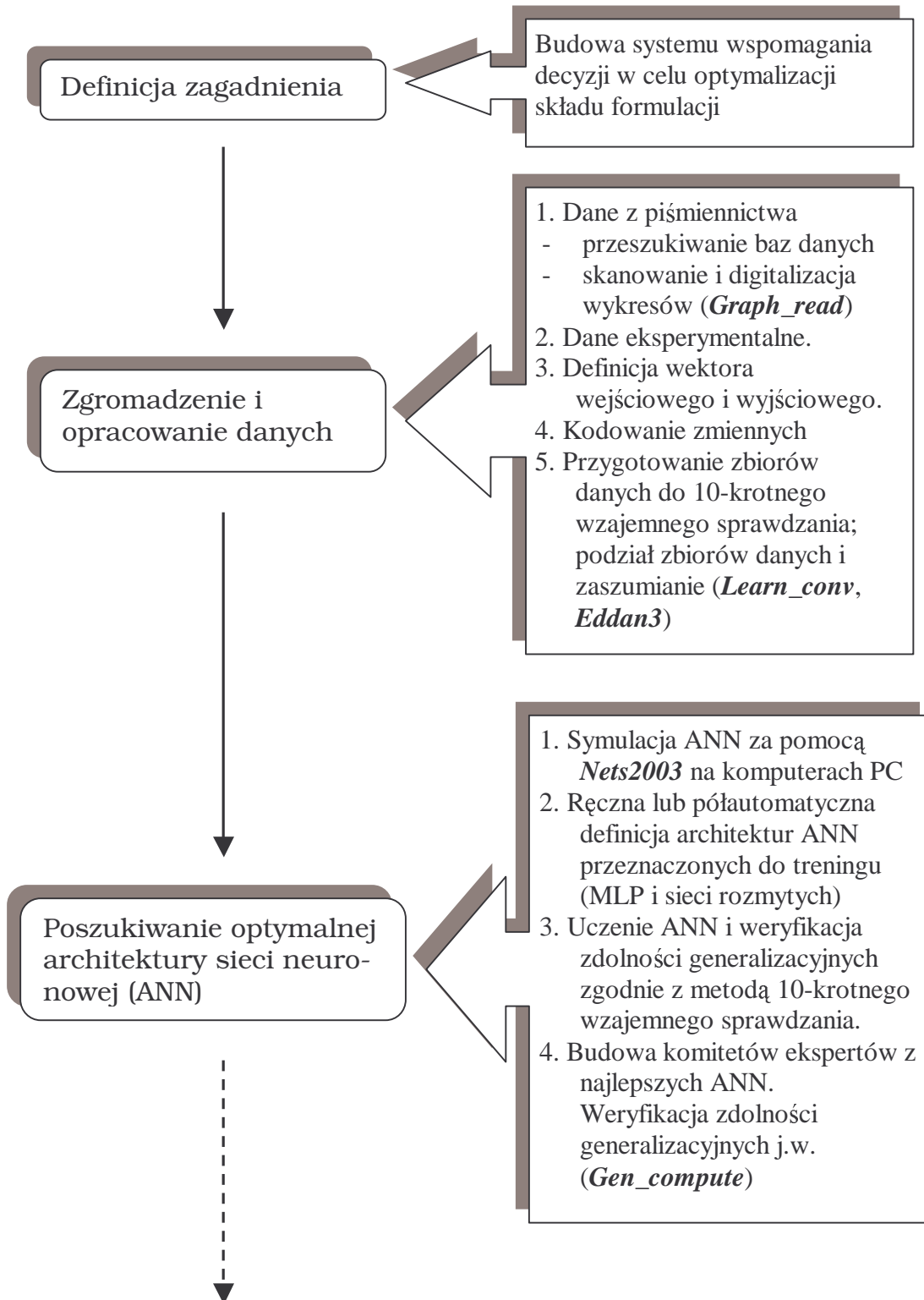
113. Sun Y., Peng Y., Chen Y., Shukla A.J. Application of artificial neural networks in the design of controlled release drug delivery systems. *Adv. Drug Delivery Rev.* 55, 9, 2003, 1201-1215.
114. Tadeusiewicz R. *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM Warszawa 1993.
115. Takahara J., Takayama K., Nagai T. Multi – objective optimization technique based on an artificial neural network in sustained release formulations. *J.Control.Release* 49, 1997, 11-20.
116. Takayama K., Fujikawa M., Nagai T. Artificial Neural Network as a Novel Method to Optimize Pharmaceutical Formulation. *Pharm Res* 16, 1, 1999, 1-6.
117. Takayama K., Fujikawa M., Obata Y., Morishita M. Neural network based optimization of drug formulations. *Adv. Drug Delivery Rev.* 55, 9, 2003, 1217-1231.
118. Tantishaiyakul V., Kaewnopparat N., Ingkatawornwong S. Properties of solid dispersions of piroxicam in polyvinylpyrrolidone K-30. *Int.J.Pharm.* 143, 1996, 59-66
119. Tantishaiyakul V., Kaewnopparat N., Ingkatawornwong S. Properties of solid dispersions of piroxicam in polyvinylpyrrolidone. *Int.J.Pharm.*, 181, 1999, 143-151.
120. Taskinen J., Yliruusi J. Prediction of physicochemical properties based on neural network modelling. *Adv. Drug Delivery Rev.* 55, 9, 2003, 1163-1183.
121. Torrado S., Torrado S., Torrado J.J., Cadórniga R. Preparation, dissolution and characterization of albendazole solid dispersions. *Int.J.Pharm.* 140, 1996, 247-250.
122. Trapani G., Franco M., Latrofa A., Pantaleo M.R., Provenzano M.R., Sanna G., Maciocco E., Liso G. Physicochemical characterization and in vivo properties of Zolpidem in solid dispersions with polyethylene glycol 4000 and 6000. *Int. J. Pharm.* 184, 1999, 121-130.

123. Trotta M., Gallarate M., Pattarino F., Carlotti M.E. Investigation of the phase behaviour of systems containing lecithin and 2-acyl lysolecithin derivatives. *Int. J. Pharm.* 190, 1999, 83-89.
124. Türkoğlu M., Özarslan R., Sakr A. Artificial Neural Network Analysis of a Direct Compression Tableting Study, *Eur.J.Pharm.Biopharm.*, 41, 1995, 315-322.
125. Veng-Pedersen P. Modi N.B. Application of Neural Networks to Pharmacodynamics. *J. Pharm. Sci.* 82, 9, 1993.
126. Veng-Pedersen P. Modi N.B. Neural Networks in Pharmacodynamic Modeling. Is Current Modeling Practice of Complex Kinetic Systems at a Dead End ? *J. Pharm. Biopharm.* 20, 4, 1992.
127. Venkatesan A. Activated Microporous Materials through Polymerization of Microemulsion Precursors. A Dissertation Proposal Submitted to the Doctoral Dissertation Committee. The Department of Chemical Engineering University of Akron, 1998.
128. Vera N., Veiga M.D., Cadoringa R. Solid dispersions of oxodipine/PEG 6000. Characterization and dissolution study. *S.T.P. Pharma Sciences*, 1, 2, 1991, 125-129.
129. Wang, L., Wei, C. Approximation Accuracy of Some Neuro-Fuzzy Approaches. *IEEE T. Fuzzy Syst.*, 8, 2000, 470-478.
130. Wu P-C., Obata Y., Fujikawa M., Li C.J., Higashiyama K., Takayama K. Simultaneous Optimization Based on Artificial Neural Networks in Ketoprofen Hydrogel Formula Containing O-Ethyl-3-butylcyclohexanol as Percutaneous Absorption Enhancer. *J. Pharm. Sci.* 90, 8, 2001, 1004-1014.
131. Yager R. R., Filev D. P., Podstawy modelowania i sterowania rozmytego. WNT Warszawa 1995.
132. Yaoku S., Umehara K., Sonobe T, Nagai T., Sugihara M., Fukuyama Y. Particle size dependency of dissolution rate and human bioavailability of phenytoin in powders and phenytoin – polyethylene glycol solid dispersions. *Chem.Pharm.Bull.* 32, 10, 1984, 4130-4136.

133. Yuasa H., Ozeki T., Kanaya Y., Oishi K. Application of the solid dispersion method to the controlled release of medicine. IV. Precise control of the release rate of a water soluble medicine by using the solid dispersion method applying the difference in the molecular weight of a polymer. *Chem.Pharm.Bull.* 41,5, 1993, 933-936.
134. Yuasa H., Ozeki T., Takahashi H., Kanaya Y., Ueno M. Application of the solid dispersion method to the controlled release of medicine. VI. Release mechanism of slightly water soluble medicine and interaction between flurbiprofen and hydroksypropyl cellulose in solid dispersion. *Chem.Pharm.Bull.* 42, 2, 1994, 354-358.
135. Zadeh L.A. Fuzzy Sets. *Information and Control*, 8, 1965, 338-353.
136. Żurada J., Barski M., Jędruch W. *Sztuczne sieci neuronowe. Podstawy teorii i zastosowania.* PWN Warszawa 1996.

IX. OPROGRAMOWANIE

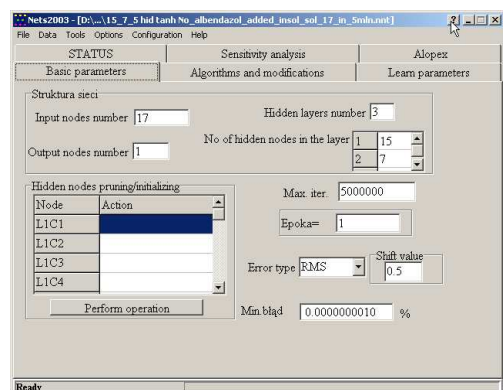
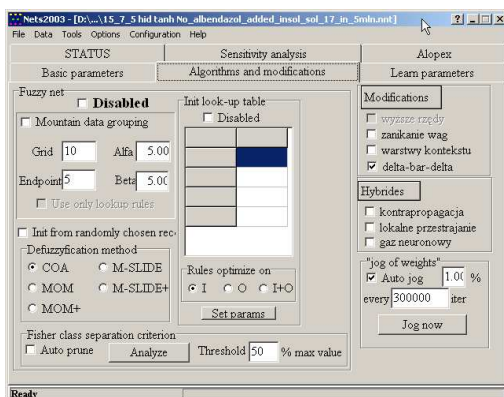
1. ALGORYTM BUDOWY SYSTEMU *SD_EXPERT*



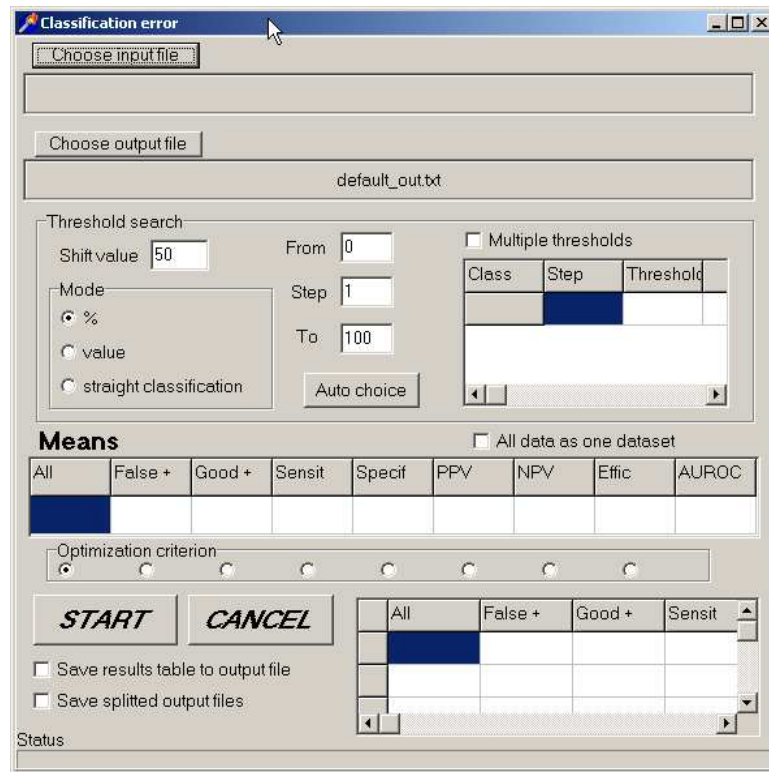
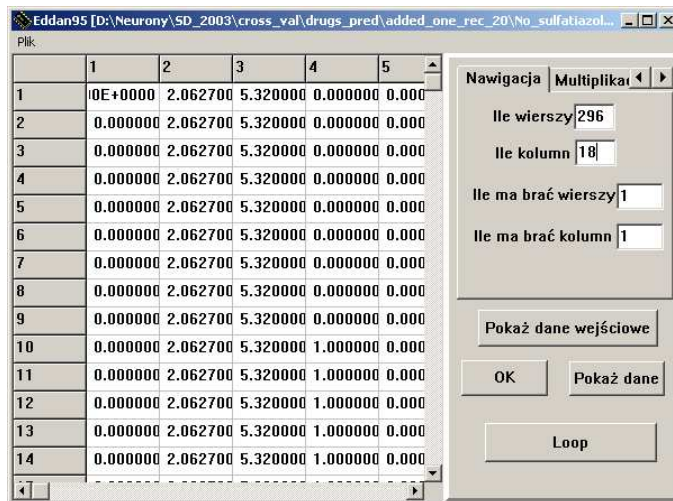
Opracowanie ostatecznego kształtu systemu **SD_expert**

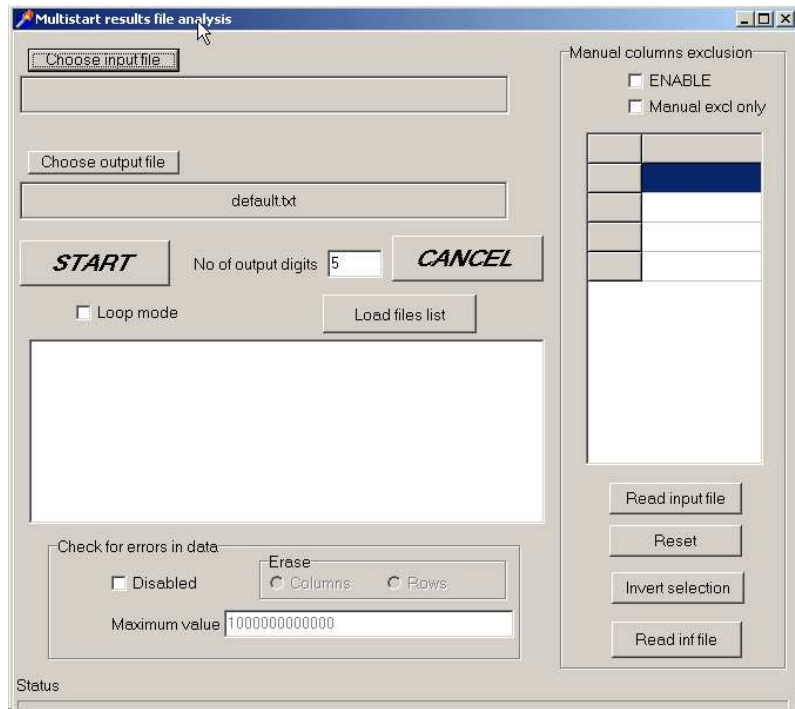
1. Wykorzystanie części kodu **Nets2003**
2. Użycie komitetów ekspertów ANN w charakterze maszyny wnioskującej.
3. Automatyzacja pracy przy poszukiwaniu optymalnej formułacji. Iteracyjne przeszukiwanie przestrzeni rozwiązań
4. Wykres profilu rozpuszczania substancji leczniczej jako kryterium optymalności formułacji
5. Możliwość wielowymiarowej optymalizacji zarówno dla składu jakościowego jak i ilościowego
6. Weryfikacja systemu na wybranych przykładach

2. ZRZUTY EKRAKOWE NAPISANYCH PROGRAMÓW

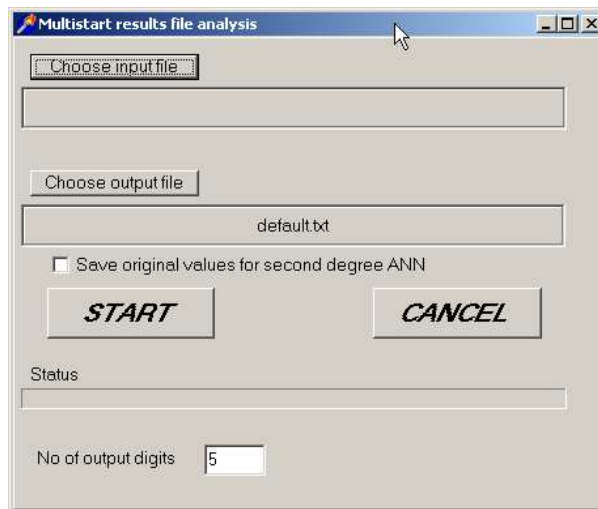


Rycina 30. Opcje dostępne w programie *Nets2003*

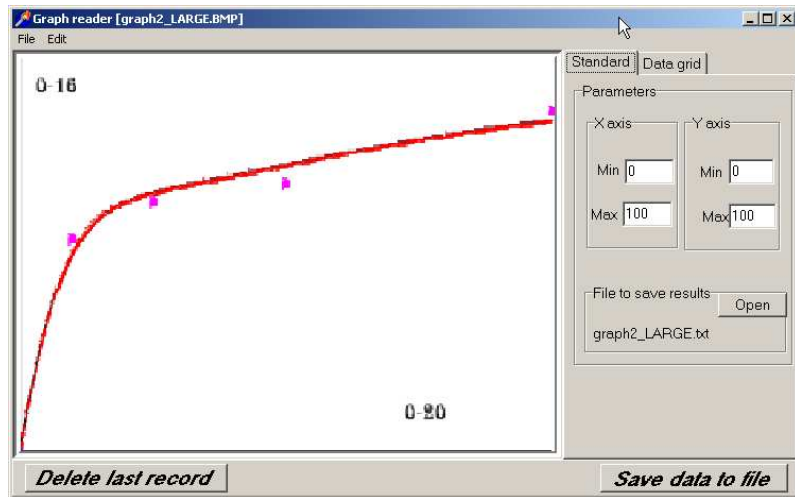
Rycina 31. Program *Classif_compute*.Rycina 32. Główne okno programu *Eddan3*



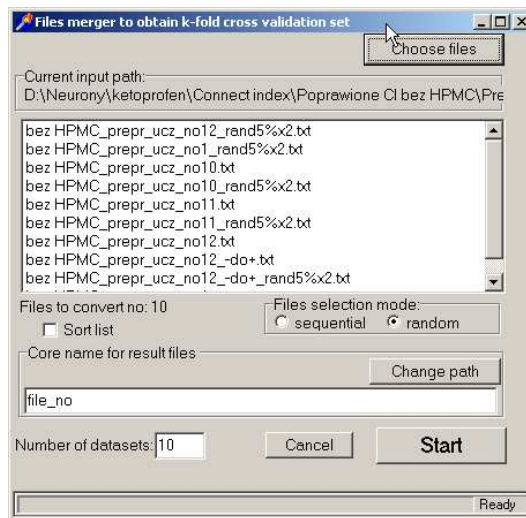
Rycina 33. Program *Erase_0*.



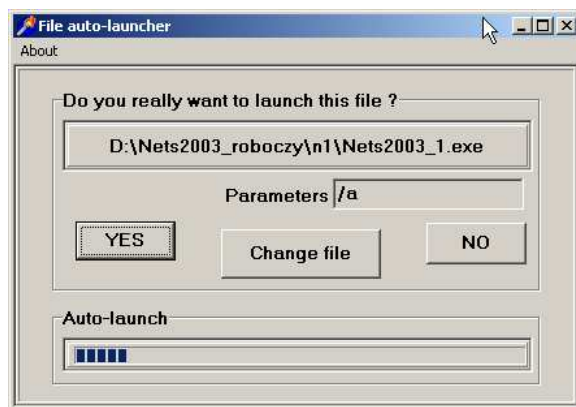
Rycina 34. Program *Gen_compute*.



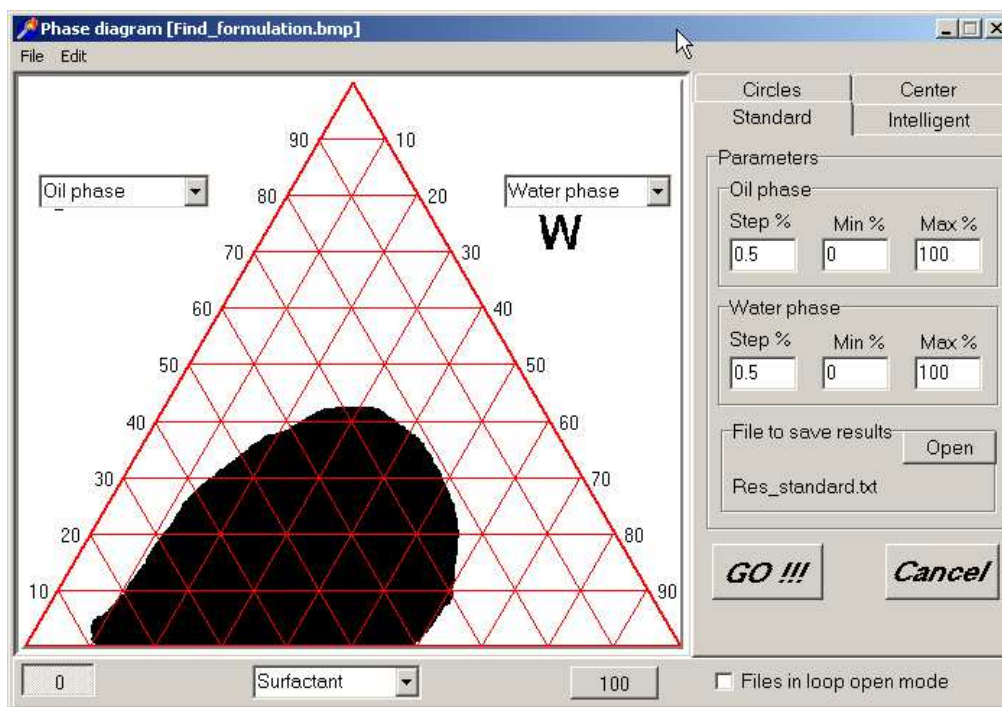
Rycina 35. Program *Graph_read*.



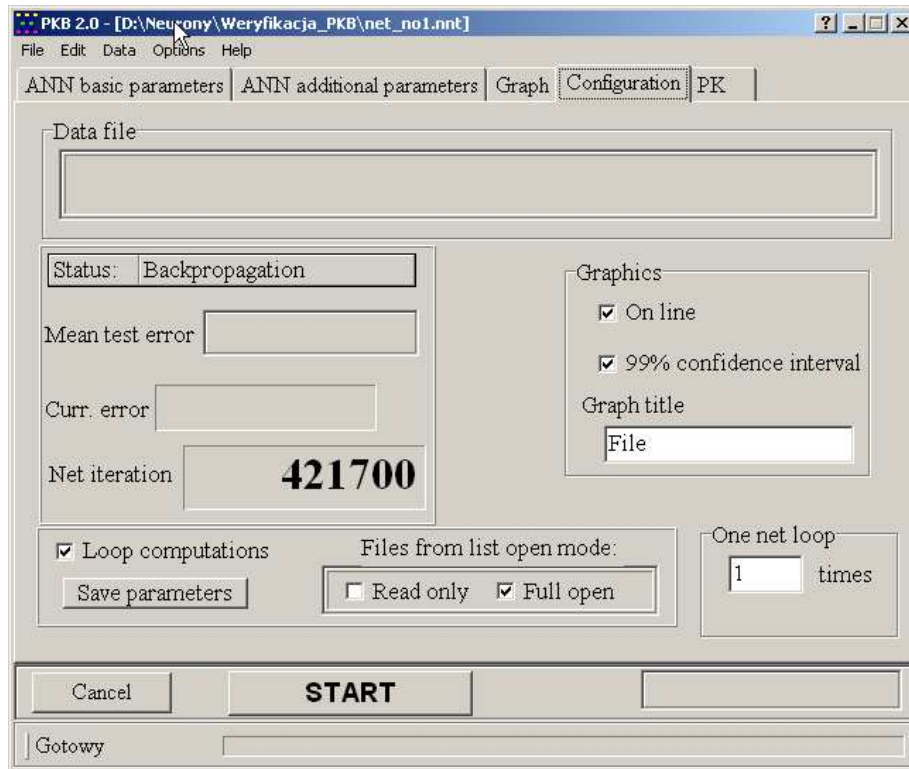
Rycina 36. Program *Learn_conv*.



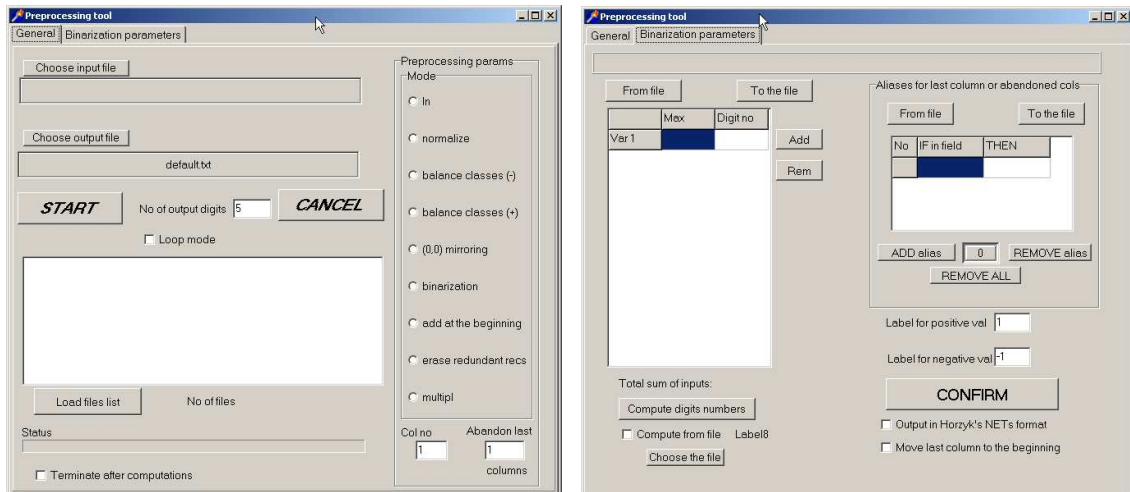
Rycina 37. Program *Launch_file*.



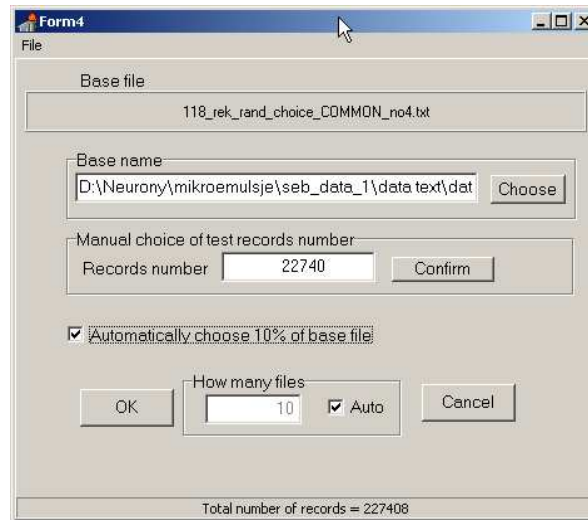
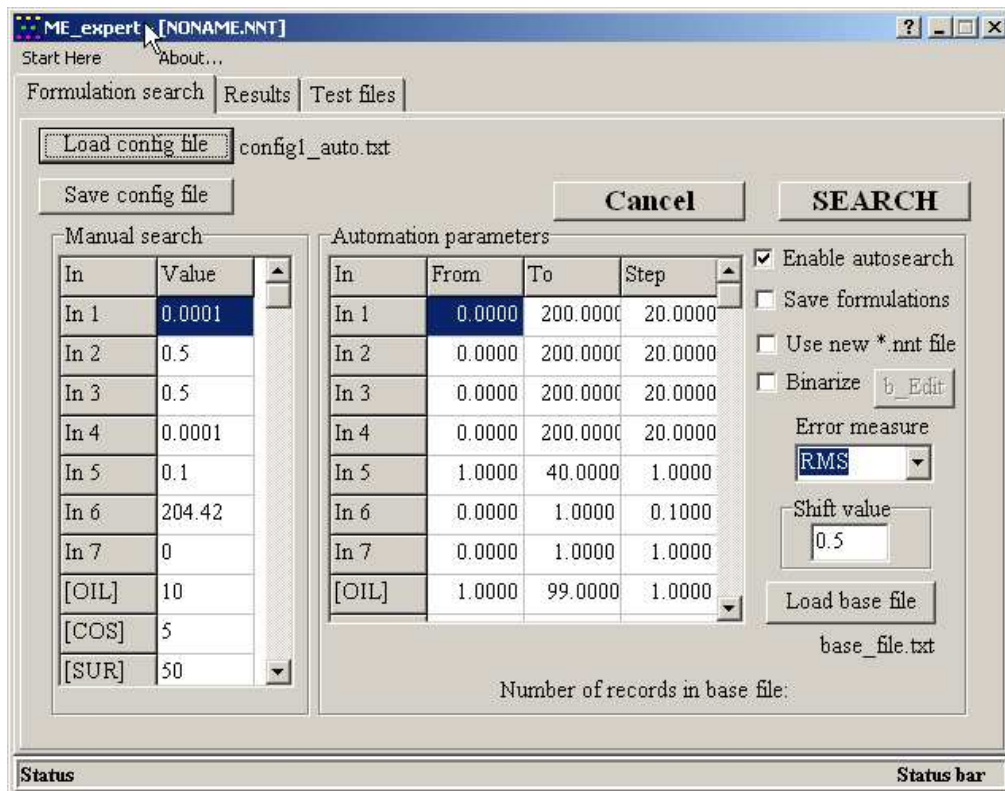
Rycina 38. Program *Phase_diagr* z załadowaną bitmapą obrazującą trójkąt fazowy.

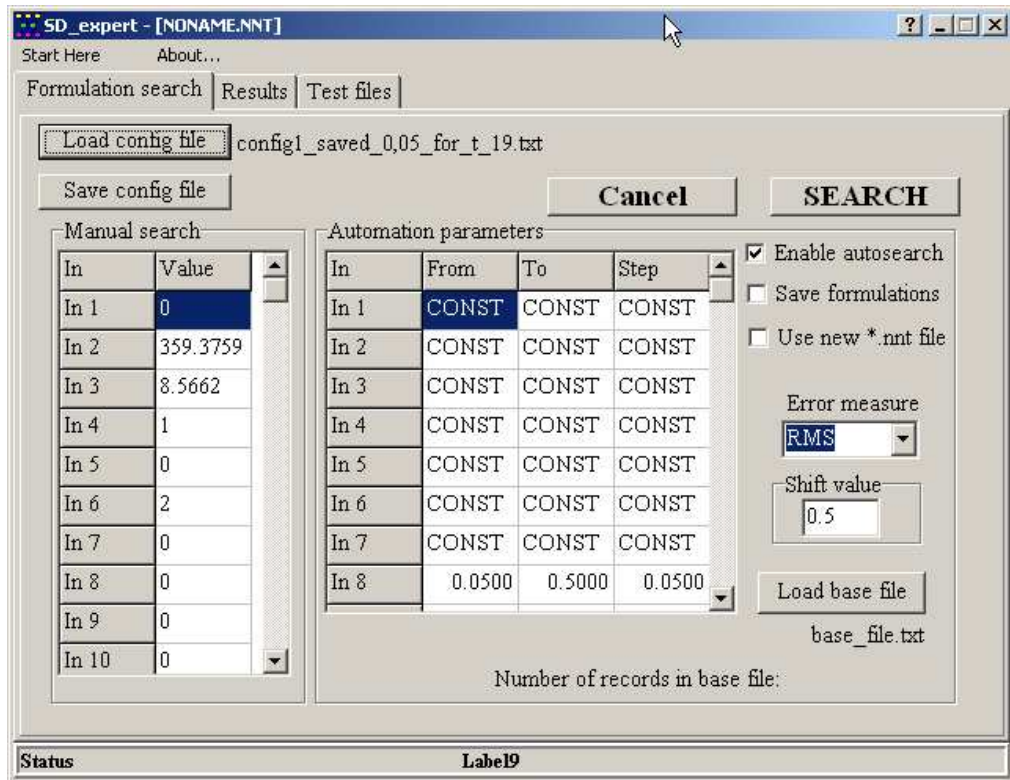


Rycina 39. Główne okno systemu PKB2.0.



Rycina 40. Okna programu Preproc.

Rycina 41. Program *Split_file*.Rycina 42. System *ME_expert*.

Rycina 43. System *SD_expert*.