

Danuta Rutkowska<sup>\*,\*\*</sup>

## **Sieci rozmyto-neuronowe do klasyfikacji: architektury typu RBF i MLP – system NEFCLASS**

### **1. Wprowadzenie**

Systemy rozmyto-neuronowe powstały jako połączenie systemów rozmytych i sieci neuronowych. Szczególny przypadek stanowią architektury sieciowe, nazywane sieciami rozmyto-neuronowymi, w których system rozmyty jest realizowany w postaci architektury sieci neuronowej. Umożliwiają one wykorzystanie zarówno regułowej reprezentacji wiedzy systemu rozmytego, jak też zdolności uczenia – typowej dla sieci neuronowych.

W systemach rozmytych wnioskowanie odbywa się na podstawie bazy rozmytych reguł typu JEŻELI-TO. Taka regułowa reprezentacja wiedzy jest czytelna, co oznacza że pozwala na łatwe wyjaśnienie działania systemu poprzez uaktywnianie odpowiednich reguł. W sieciach neuronowych wiedza jest rozproszona w wagach sieci, a przez to nieczytelna dla użytkownika. Uczenie sieci neuronowych polega na znalezieniu optymalnego zestawu wag. Sieciowa, wielowarstwowa architektura sieci rozmyto-neuronowych odwzorowuje regułową bazę wiedzy i jednocześnie pozwala na wykorzystanie algorytmów uczenia. W tym przypadku uczenie stosuje się w celu optymalnego doboru parametrów, charakteryzujących zbiory rozmyte występujące w regułach. Parametry te, w odróżnieniu od wag sieci neuronowych mają czytelną interpretację, np. środków i/lub szerokości funkcji przynależności zbiorów rozmytych. Szczegółowe informacje na temat sieci neuronowych oraz systemów rozmytych i rozmyto-neuronowych można znaleźć, np. w [4–7].

Sieci neuronowe MLP (*multi-layer perceptron*) i RBF (*radial basis functions*) są najbardziej znane i najczęściej stosowane. W sieciach MLP dokonuje się uczenia wag za pomocą algorytmu wstecznej propagacji błędów [12]. Sieci RBF zdecydowanie różnią się od MLP. Ich architektura jest bliższa sieciowej implementacji systemu rozmytego. Dlatego sieci rozmyto-neuronowe, rozważane w literaturze [4–7], nazywamy sieciami typu RBF. Sieci te, w zastosowaniu do klasyfikacji, porównuje się w tym artykule z systemem rozmyto-neuronowym NEFCLASS [4], który został zbudowany zgodnie z ideą rozmytego wielowarstwowego perceptronu, czyli MLP.

---

\* Wyższa Szkoła Humanistyczno-Ekonomiczna, Łódź

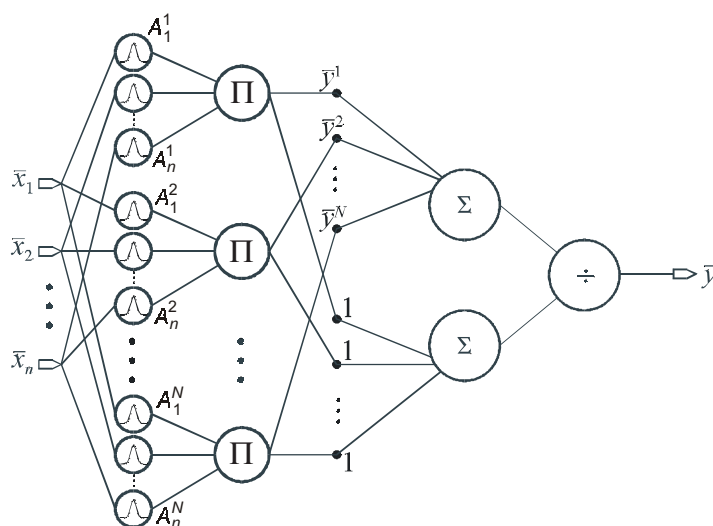
\*\* Politechnika Częstochowska, Częstochowa

## 2. Architektury sieci rozmyto-neuronowych typu RBF

Przedmiotem rozważań są teraz sieci rozmyto-neuronowe, które w wersji podstawowej zaproponowano w [13], a następnie rozwijano w [6–10]. Stanowią one siećową reprezentację systemu rozmytego. Takie sieci, w najprostszej wersji, są bardzo podobne do sieci RBF, a przy pewnych założeniach nawet wykazano ich równoważność [3]. Dlatego mówimy, że są to sieci rozmyto-neuronowe typu RBF.

### 2.1. Podstawowa architektura sieciowa

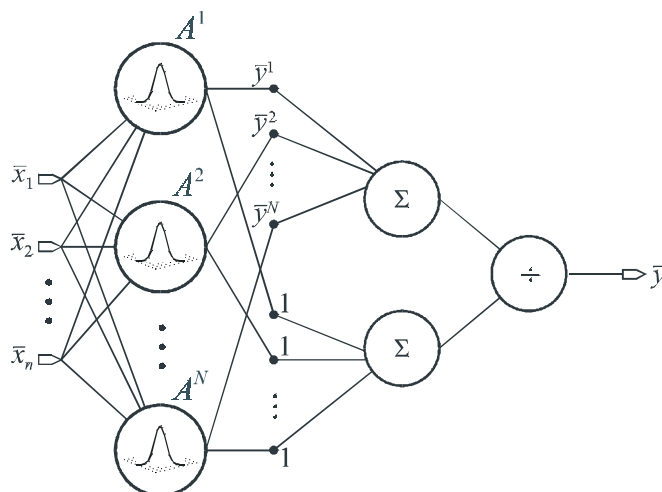
Najbardziej popularną architekturę sieciową systemu rozmyto-neuronowego pokazuje rysunek 1. Sieć ta reprezentuje najczęściej stosowany system rozmyty, nazywany systemem Mamdaniego [10], w którym dokonuje się rozmywania typu *singleton* oraz wyostrażania zgodnie z funkcją realizowaną przez elementy sumowania i dzielenia [6, 7].



Rys. 1. Architektura sieci typu RBF z operacją iloczynu

Tę samą architekturę sieciową można przedstawić w postaci jak na rysunku 2. Zamiast elementów realizujących jednowymiarowe funkcje przynależności zbiorów rozmytych oraz elementów iloczynu, potrzebnych do utworzenia iloczynu kartezjańskiego tych zbiorów rozmytych, pokazano elementy (neurony), które realizują wielowymiarowe funkcje przynależności tych iloczynów kartezjańskich.

Architektura sieci z rysunku 1 jest szczególnym przypadkiem architektury przedstawionej na rysunku 2, gdy za pomocą operacji iloczynu tworzy się iloczyn kartezjański zbiorów rozmytych. Zamiast iloczynu, można zastosować operację minimum, jak w klasycznych systemach rozmytych Mamdaniego; wtedy elementy iloczynu na rysunku 1 należy zastąpić neuronami realizującymi operację minimum.



Rys. 2. Architektura sieciowa typu RBF

Sieć pokazana na rysunku 2 stanowi odwzorowanie następującej bazy reguł

$$R^k : \mathbf{IF} \mathbf{x} \text{ is } A^k \mathbf{ THEN } y \text{ is } B^k \quad (1)$$

gdzie:

$$k = 1, \dots, N,$$

$$\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbf{X} \subset \mathbf{R}^n,$$

$$y \in \mathbf{X} \subset \mathbf{R},$$

$$A^k = A_1^k \times \dots \times A_n^k,$$

$\mathbf{B}^k$  – zbiory rozmyte o funkcjach przynależności  $\mu_{A^k}(\mathbf{x}), \mu_{B^k}(y)$ , które najczęściej przyjmuje się jako funkcje gaussowskie;  
 $\mathbf{x}$  i  $y$  – zmienne lingwistyczne.

Wzór (1) przedstawia ogólną postać bazy reguł, którą można wyrazić jako

$$R^k : \mathbf{IF} x_1 \text{ is } A_1^k \mathbf{ AND} \dots \mathbf{ AND} x_n \text{ is } A_n^k \mathbf{ THEN } y \text{ is } B^k \quad (2)$$

Bazę reguł daną wzorem (2) odwzorowuje sieć rozmyto-neuronowa z rysunku 1.

Architektura sieci rozmyto-neuronowej z rysunków 1 i 2 stanowi tzw. znormalizowaną sieć RBF. W tym przypadku wagi  $\bar{y}^k$  oznaczają środki funkcji przynależności zbiorów rozmytych  $\mathbf{B}^k$ , dla  $k = 1, \dots, N$ . Szczegóły na temat tego typu sieci można znaleźć w [10].

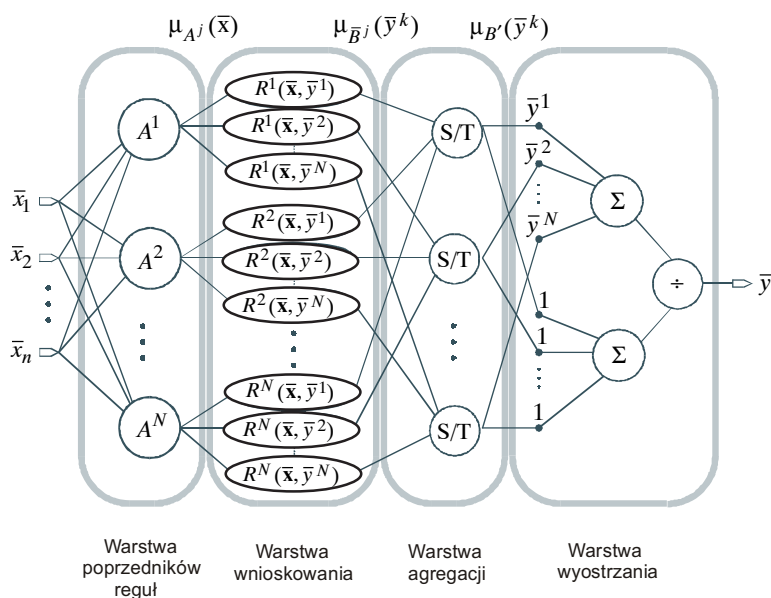
## 2.2. Uogólniona architektura sieciowa

Sieci rozmyto-neuronowe przedstawione na rysunkach 1 i 2 reprezentują najbardziej popularny system Mamdaniego, stosowany głównie w sterownikach rozmytych. W systemie tym wnioskowanie opiera się na założeniu, że reguły wyrażone wzorem (1) lub (2) są traktowane jako tzw. implikacje inżynierskie [5], a nie prawdziwe implikacje w sensie logicznym.

W literaturze, oprócz systemów Mamdaniego, rozważa się też rozmyte systemy logiczne, czyli traktujące reguły jak prawdziwe implikacje. Znanych jest wiele przykładów prawdziwych implikacji, zatem rozważa się różne systemy rozmyte typu logicznego, w których te implikacje wykorzystywane są w procesie wnioskowania [1, 4, 5, 8, 10].

Na rysunku 3 przedstawiono uogólnioną architekturę sieci typu RBF, uwzględniającą zarówno sieci Mamdaniego, jak i logiczne. Istotną różnicą między tą siecią a pokazanymi na rysunkach 1 i 2 stanowią dodatkowe warstwy, oznaczone jako warstwa wnioskowania i warstwa agregacji. Zauważmy, że po usunięciu tych warstw z rysunku 3 otrzymujemy sieć zamieszczoną na rysunku 2. Warto podkreślić, że warstwa poprzedników reguł zawiera elementy (neurony) realizujące wielowymiarowe funkcje przynależności zbiorów rozmytych  $A^k = A_1^k \times \dots \times A_n^k$ , dla  $k = 1, \dots, N$ , przy czym najczęściej przyjmujemy gaussowską postać tych funkcji. Warstwa wyostrzenia na rysunku 3 jest identyczna jak na rysunkach 1 i 2.

Różne szczególne przypadki sieci rozmyto-neuronowych, o architekturze z rysunku 3, w tym też z rysunków 1 i 2, rozważano w [8, 10]. Wśród nich są reprezentacje systemów logicznych, dla konkretnych postaci rozmytych implikacji.



Rys. 3. Uogólniona architektura sieci typu RBF

Szczególny przypadek sieci rozmyto-neuronowej, przedstawionej na rysunku 2, otrzymujemy z architektury sieci z rysunku 3 po bezpośrednim połączeniu wyjść warstwy poprzedników reguł z odpowiednimi wejściami warstwy wyostrzania. Oznacza to, że

$$\tau'_k = \tau_k \quad (3)$$

gdzie

$$\tau_k = \mu_{A^k}(\bar{x}) \quad (4)$$

natomiast

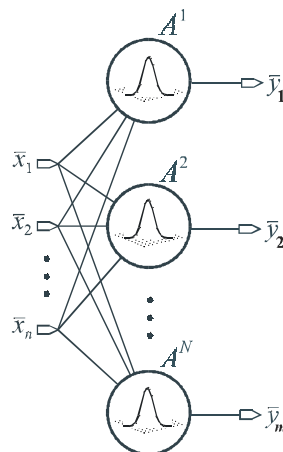
$$\tau'_k = \mu_{B^k}(\bar{y}^k) \quad (5)$$

Wartość  $\tau_k$ , dla  $k = 1, \dots, N$ , dana wzorem (4), jest stopniem aktywacji reguły  $R^k$ ; określa w jakim stopniu wektor wejściowy  $\bar{x} = [\bar{x}_1, \dots, \bar{x}_n]^T$  pasuje do poprzednika reguły, czyli części IF we wzorze (2).

Wartość  $\tau'_k$ , dla  $k = 1, \dots, N$ , jak podaje wzór (5) jest wartością funkcji przynależności zbioru rozmytego  $B^k$ , otrzymanego w wyniku wnioskowania systemu rozmytego na podstawie bazy reguł (2), w punkcie stanowiącym środek funkcji przynależności zbioru rozmytego  $B^k$ , występującego w następniku reguły  $R^k$ , danej wzorem (2). Jeżeli wnioskowanie na podstawie pojedynczej reguły  $R^k$  daje rezultat w postaci zbioru rozmytego  $\bar{B}^k$ , to zbiór rozmyty  $B^k$  jest tworzony na podstawie zbiorów rozmytych  $\bar{B}^k$ , dla  $k = 1, \dots, N$ , jako efekt agregacji reguł. Szczegóły można znaleźć w literaturze [6, 10].

### 2.3. Sieci rozmyto-neuronowe typu RBF do klasyfikacji

Sieć rozmyto-neuronowa typu RBF, w zastosowaniu do klasyfikacji, może mieć prostszą postać niż przedstawiona na rysunkach 1, 2 i 3. Taką sieć ilustruje rysunek 4. Będziemy ją nazywać rozmyto-neuronowym (R-N) klasyfikatorem typu RBF.



Rys. 4. Sieć typu RBF do klasyfikacji

Sieć przedstawiona na rysunku 4, w porównaniu z siecią rozmyto-neuronową z rysunku 2, nie posiada warstwy wyostrzania. Wyjścia sieci pokazanej na rysunku 4 odpowiadają klasom, w rozważanym problemie klasyfikacji. Taka sieć odzwierciedla następującą bazę reguł

$$R^k : \mathbf{IF} \mathbf{x} \text{ is } A^k \mathbf{ THEN } \mathbf{x} \in C_l \quad (6)$$

gdzie  $C_l$ , dla  $l = 1, \dots, m$ , oznacza daną klasę. Poszczególne wyjścia sieci z rysunku 4, czyli  $\bar{y}_l$ , są skojarzone z odpowiednimi klasami. Liczba reguł może być równa lub większa od liczby klas, co oznacza że więcej niż jedno wyjście może być skojarzone z tą samą klasą. Maksymalna wartość na określonym wyjściu wskazuje na decyzję odnośnie do wyboru klasy w zadaniu klasyfikacji.

Warto podkreślić, że sieć z rysunku 2 także może być stosowana do tych samych problemów klasyfikacji. W tym przypadku, wartość wyjściowa  $\bar{y}$ , otrzymana w wyniku uśrednienia wartości  $\tau_k$ , dla  $k = 1, \dots, N$ , decyduje o wyborze klasy. Jednakże sieć z rysunku 4 jest prostsza, zawiera mniej elementów (neuronów).

Zwróćmy teraz uwagę na zastosowanie do klasyfikacji sieci rozmyto-neuronowej z rysunku 3. Po usunięciu warstwy wyostrzania otrzymujemy uogólnioną architekturę sieci typu RBF do klasyfikacji, w wersji analogicznej jak sieć na rysunku 4. W tym przypadku na wyjściach otrzymujemy wartości  $\tau'_k$ , dla  $k = 1, \dots, N$ , a każde z tych wyjść, podobnie jak w sieci z rysunku 4, jest skojarzone z odpowiednią klasą.

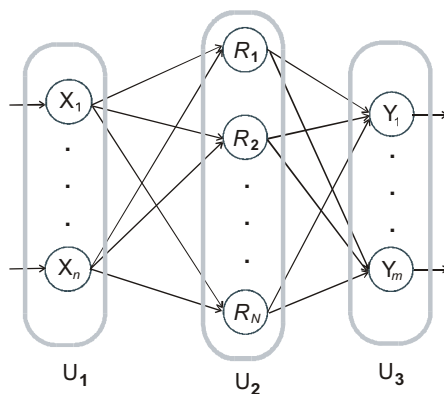
Wyjścia wskazujące na tę samą klasę można oczywiście połączyć w jedno. Wtedy liczba wyjść będzie równa liczbie klas (i mniejsza od liczby reguł). Dotyczy to obu sieci do klasyfikacji (czyli R-N klasyfikatorów typu RBF), zarówno tej uogólnionej, jak i sieci z rysunku 4.

Stosowanie uogólnionego R-N klasyfikatora typu RBF jest szczególnie uzasadnione w przypadkach, gdy klasy w problemie klasyfikacji nie są całkowicie rozłączne, czyli  $C_l$ , dla  $l = 1, \dots, m$ , należy traktować jak zbiory rozmyte. Można wtedy rozważać przypadek OCFC (*overlapping consequent fuzzy sets*), kiedy nie jest spełniony warunek, że wartość funkcji przynależności tych zbiorów rozmytych w punkcie będącym „środkiem” jednego z nich jest równa lub bliska zeru dla pozostałych [10].

### 3. Architektura sieci rozmyto-neuronowych typu MLP

#### 3.1. Rozmyty perceptron

Na rysunku 5 pokazano architekturę sieciową tzw. rozmytego perceptronu [4], który od klasycznego, wielowarstwowego perceptronu (MLP) różni się tym, że wagi połączeń w tej sieci przyjmują wartości rozmyte (liczby rozmyte). Poza tym taka sieć realizuje rozmyte wnioskowanie, typowe dla systemu rozmytego. Wobec tego neurony tej sieci realizują inne funkcje niż klasyczny model neuronu w sieci MLP. W tym sensie przypominają one elementy (neurony) sieci omówionych w rozdziale 2, czyli sieci typu RBF. Dokładną definicję takiego ogólnego, rozmytego perceptronu przedstawiono w [4].



Rys. 5. Perceptron wielowarstwowy (trójwarstwowy)

### 3.2. System NEFCLASS

System NEFCLASS został zaprojektowany do zadań klasyfikacji (nazwa oznacza *Neuro-Fuzzy CLASSifier*). Opracowano go [4] na podstawie architektury rozmytego perceptronu, pokazanego na rysunku 5.

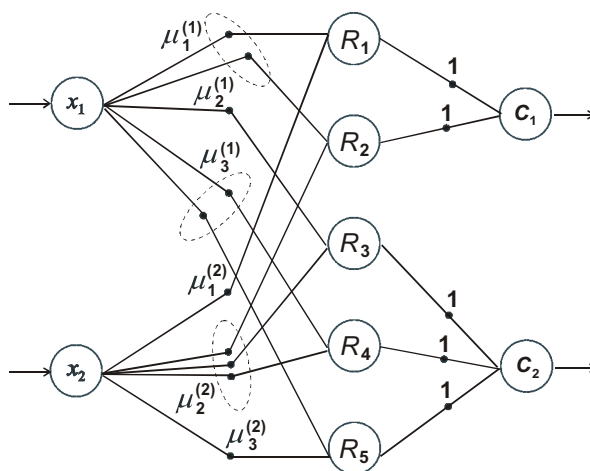
Architekturę sieciową systemu NEFCLASS ilustruje rysunek 6. Pokazany tu, przykładowy system, dokonuje klasyfikacji na podstawie następującej bazy reguł:

$$\begin{aligned}
 \mathbf{R}_1 &: \text{IF } \mathbf{x}_1 \text{ is } \mathbf{A}_1^{(1)} \text{ AND } \mathbf{x}_2 \text{ is } \mathbf{A}_1^{(2)} \text{ THEN } \mathbf{x} \in C_1 \\
 \mathbf{R}_2 &: \text{IF } \mathbf{x}_1 \text{ is } \mathbf{A}_1^{(1)} \text{ AND } \mathbf{x}_2 \text{ is } \mathbf{A}_2^{(2)} \text{ THEN } \mathbf{x} \in C_1 \\
 \mathbf{R}_3 &: \text{IF } \mathbf{x}_1 \text{ is } \mathbf{A}_2^{(1)} \text{ AND } \mathbf{x}_2 \text{ is } \mathbf{A}_2^{(2)} \text{ THEN } \mathbf{x} \in C_2 \\
 \mathbf{R}_4 &: \text{IF } \mathbf{x}_1 \text{ is } \mathbf{A}_3^{(1)} \text{ AND } \mathbf{x}_2 \text{ is } \mathbf{A}_2^{(2)} \text{ THEN } \mathbf{x} \in C_2 \\
 \mathbf{R}_5 &: \text{IF } \mathbf{x}_1 \text{ is } \mathbf{A}_3^{(1)} \text{ AND } \mathbf{x}_2 \text{ is } \mathbf{A}_3^{(2)} \text{ THEN } \mathbf{x} \in C_2
 \end{aligned}
 \tag{7}$$

gdzie:

$$\begin{aligned}
 \mathbf{x} &= [\mathbf{x}_1, \mathbf{x}_2]^T \\
 \mathbf{x}_i, & \text{ dla } i = 1, 2 \text{ – zmienne lingwistyczne,} \\
 \mathbf{A}_j^{(i)}, & \text{ dla } j = 1, 2, 3 \text{ – zbiory rozmyte (liczbami rozmytymi) o funkcjach przynależności } \mu_j^{(i)}, \\
 C_i & \text{ – odpowiednia klasa.}
 \end{aligned}$$

W ogólnym przypadku taki system może posiadać  $n$  zmiennych wejściowych, tzn.  $i = 1, \dots, n$ , a także liczbę reguł równą  $N$ , czyli  $j = 1, \dots, N$ .



Rys. 6. Architektura sieciowa systemu NEFCLASS

W sieci NEFCLASS zbiory rozmyte o funkcjach przynależności  $\mu_j^{(i)}$  są traktowane jak rozmyte wagi. Na rysunku 6 zaznaczono wspólne wagi, co oznacza takie same zbiory rozmyte, występujące w poprzednikach kilku (nie tylko jednej) reguł. Wynika to z faktu, że w systemie NEFCLASS dużą wagę przywiązuje się do „czytelności” reguł, czyli ich zrozumiałej interpretacji semantycznej, np. w znaczeniu „mały”, „średni”, „duży” [4].

Wagi połączeń w ostatniej warstwie są równe 1 i jak widać na rysunku 6, połączono tu wyjścia warstwy poprzedniej, wskazujące na tę samą klasę, o czym wspomniano w odniesieniu do sieci R-N typu RBF do klasyfikacji w podrozdziale 2.3.

Krótkie informacje o systemie NEFCLASS można też znaleźć w [2].

#### 4. Porównanie R-N klasyfikatora typu RBF i systemu NEFCLASS

System NEFCLASS został opracowany na bazie rozmytego perceptronu [4]. Architektura sieciowa tego systemu (rys. 6) jest w istocie siecią rozmytą, gdyż zawiera rozmyte wagi. Jednakże, jak wykazano w [9], sieć ta jest równoważna sieci typu RBF, która nie jest rozmyta w tym sensie, że nie przetwarza rozmytych wartości, mimo że odwzorowuje działanie systemu rozmytego. Taka sama równoważność zachodzi między siecią typu RBF a systemem NEFCON, służącym do sterowania, jak też systemem NEFPROX – do aproksymacji [4]. Porównania sieci rozmyto-neuronowych typu RBF i systemu NEFCON dokonano też w [11]. Warto podkreślić, że pierwsza część tych sieciowych architektur, dotycząca poprzedników reguł (IF...) jest taka sama, natomiast różnica dotyczy przede wszystkim części następników reguł (THEN...).

W odniesieniu do części poprzedników reguł równoważność sieci typu RBF i systemu NEFCLASS polega na realizacji funkcji przynależności zbiorów rozmytych jako funkcji radialnych w sieci typu RBF i rozmytych wag w sieci NEFCLASS. W istocie obie sieci



odwzorowują ten sam sposób rozmytego wnioskowania, ale pierwsza z tych sieci nie jest rozmyta, podczas gdy druga tak, gdyż posiada wagi o wartościach rozmytych (liczby rozmyte).

Wykazaną równoważność sieci typu RBF i systemu NEFCLASS można rozszerzyć także do uogólnionej wersji sieci rozmyto-neuronowej, czyli tej z warstwą wnioskowania i agregacji pokazaną na rysunku 3. Można bowiem rozważać system NEFCLASS z wnioskowaniem na podstawie prawdziwych implikacji. Uwagi odnośnie rozmytości pozostają takie same.

## 5. Uwagi końcowe i wnioski

Można zaproponować ogólną architekturę sieciową, która reprezentuje system rozmyty i może być interpretowana zarówno jako rozmyty perceptron, jak też sieć typu RBF. Będzie ona złożona z dwóch części:

- 1) dotyczącej poprzedników reguł,
- 2) dla następników reguł.

Jej szczególnymi przypadkami mogą być sieci typu RBF, także architektury uogólnione, jak też system NEFCLASS dla różnych postaci reguł (również wykorzystujących prawdziwe implikacje).

Warto podkreślić, że każdą z tych części można realizować za pomocą klasycznych sieci neuronowych, podobnie jak w [11].

Ciekawym rozwiązaniem problemu klasyfikacji może być zastosowanie R-N klasyfikatora typu RBF, gdzie część reguł ma postać (6), które odzwierciedla architektura sieci z rysunku 4, podczas gdy niektóre reguły wykorzystują prawdziwe implikacje i OCFC, o czym mówiono w podrozdziale 2.3. Takie podejście pozwala na lepszą klasyfikację, z rozróżnianiem przypadków, których zaklasyfikowanie do danej klasy nie budzi wątpliwości od trudnych do klasyfikacji z uwagi na częściową przynależność do wielu klas.

## Literatura

- [1] Czogała E., Łęski J.: *Fuzzy and Neuro-Fuzzy Intelligent Systems*. Heidelberg, Springer 2000
- [2] Gorzałczany M.B.: *Computational Intelligence Systems and Applications: Neuro-Fuzzy and Neural Synergisms*. Heidelberg, Springer 2002
- [3] Jang J.-S.R., Sun C.-T.: *Functional equivalence between radial basis function networks and fuzzy inference systems*. IEEE Transactions on Neural Networks, t. 4, nr 1, 1993, 156–159
- [4] Nauck D., Klawonn F., Kruse R.: *Foundations of Neuro-Fuzzy Systems*. New York, Wiley 1997
- [5] Mendel J.M.: *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Englewood Cliffs, Prentice-Hall 2001
- [6] Rutkowska D., Piliński M., Rutkowski L.: *Sieci neuronowe, algorytmy genetyczne, systemy rozmyte*. Warszawa, PWN 1997
- [7] Rutkowska D.: *Inteligentne systemy obliczeniowe: Algorytmy genetyczne i sieci neuronowe w systemach rozmytych*. Warszawa, Akademicka Oficyna Wydawnicza PLJ 1997

- 
- [8] Rutkowska D., Nowicki R.: *Implication-based neuro-fuzzy architectures*. International Journal of Applied Mathematics and Computer Science, t. 10, nr 4, 2000, 675–701
  - [9] Rutkowska D.: *Type 2 Fuzzy Neural Networks: An Interpretation Based on Fuzzy Inteference Neural Networks with Fuzzy Parameters*. Proceedings of the 2002 IEEE World Congress on Computational Intelligence, 2002, 1180–1185
  - [10] Rutkowska D.: *Neuro-Fuzzy Architectures and Hybrid Learning*. Heidelberg, Springer, 2002
  - [11] Rutkowska D.: *Sieci rozmyto-neuronowe do sterowania: architektury typu RBF I MLP – system NEFCON*. XV Krajowa Konferencja Automatyki, t. III, 2005, 31–36
  - [12] Tadeusiewicz R.: *Sieci neuronowe*. Warszawa, Akademicka Oficyna Wydawnicza RM 1993
  - [13] Wang L.-X.: *Adaptive Fuzzy Systems and Control*. Englewood Cliffs, Prentice-Hall 1994