Idea rozmytego perceptronu jako ogólnego modelu dla systemów neuronowo - rozmytych

Piotr Wardyński,
Wydział Informatyki i Zarządania, Politechnika Wrocławska,
piotr.wardynski@vezyr.pl
Data wygłoszenia seminarium: 12 maj 2009 r.
Data oddania pracy pisemnej: 27 maj 2009 r.

Abstract. Opracowanie przedstawia ideę rozmytego perceptronu zaproponowaną przez Detlefa Nauck oraz Rudolfa Kruse z Uniwersytetu Technicznego w Braundschweig. W pierwszej części sprecyzowane zostało pojęcie systemów neuronowo – rozmytych, w kolejnych zaś przedstawiony został model rozmytego perceptronu, jako trójwarstwowa sieć neuronowa ze zbiorami rozmytymi do modelowania wag, który może posłużyć do zaprezentowania ogólnej idei systemów neuronowo – rozmytych oraz ułatwić porównanie różnych wypracowanych do tej pory rozwiązań. W niniejszym opracowaniu zaprezentowano podstawowe informacje na temat tego perceptronu, a także przedstawiono jedną z jego implementacji.

Keywords: System neuronowo – rozmyty, logika rozmyta, sieć neuronowa

1 Wstęp

Podejście, polegające na połączeniu sieci neuronowych z systemami rozmytymi stało się bardzo popularne w połowie lat 90-tych XX wieku. W okresie tym powstała duża ilość podejść, różniących się od siebie zastosowaną architekturą, funkcjami aktywacji, propagacji czy algorytmów uczących. Nie sprzyjało to oczywiście porównaniu poszczególnych rozwiązań ze sobą. Dlatego na Uniwersytecie Technicznym w Braunschweig Panowie Detlef Nauck oraz Rudolf Kruse opracowali i przedstawili rozmyty perceptron (ang. *Fuzzy perceptron*). Model ten został zaproponowany jako baza dla różnych podejść rozmytych architektur w sieciach neuronowych, ułatwiający ich porównanie. Dodatkowo jest na tyle elastyczny, że możliwe jest jego łatwe rozszerzenie i dostosowanie do konkretnych wymagań, uzyskując system neuronowo – rozmyty spełniający określone założenia. Twórcy tego podejścia jasno określają, że ich celem nie było rozmycie całego perceptronu, lecz zaprezentowanie sposobności na umieszczenie pewnej wiedzy wewnątrz sieci neuronowej w celu skrócenia czasu potrzebnego do jej wyuczenia. Umiejscowienie takiej wiedzy wewnątrz sieci pozwala także na interpretację wyników jej uczenia w formie reguł lingwistycznych.
W punkcie drugim niniejszego opracowania zostanie sprecyzowane pojęcie systemów neuronowo – rozmytych. W punktach 3 i 4 omówiony zostanie rozmyty perceptron oraz ogólny algorytm uczący oparty o rozmyty błąd (ang. *Fuzzy error*).

2 System neuronowo – rozmyty

Obecnie systemy neuronowo – rozmyte są stosowane w wielu różnych sytuacjach, takich jak sterowania, analiza danych czy wsparcie dla systemów decyzyjnych. Nowoczesne systemy neuronowo rozmyte są zazwyczaj reprezentowane jako wielowarstwowa sieć neuronowa, jednak istnieją także rozważania na temat zastosowania innych architektur sieci, np. samoorganizujących map. W systemach neuronowo – rozmytych wagi połączeń czy funkcje aktywacji różnią się od tych, które znane są z klasycznych sieci neuronowych. Ponieważ powstało wiele różnych podejść do tego problemu, pan Nauck w swojej pracy[1] proponuje zdefiniowanie pojęcia *neuronowo – rozmyty* jako system, który:

- System neuronowo – rozmyty jest systemem rozmytym, który uczony jest przez pewien algorytm, zgodnie z teorią o sieciach neuronowych. Procedura ucząca opiera się na informacjach lokalnych i dokonuje tylko i wyłącznie lokalnych modyfikacji w systemie rozmytym. Tak więc proces uczenia nie opiera się na bazie wiedzy, a na danych dostarczonych do systemu.
- System neuronowo – rozmyty może być zobrazowany jako specjalna 3 – warstwowa sieć neuronowa, w której zastosowano t-normy i t-konormy zamiast typowej funkcji aktywacji. Pierwsza warstwa prezentuje dane wejściowe, środkowa (ukryta) odpowiada za reguły rozmyte, ostatnia (trzecia) warstwa jest warstwą wyjściową. Zbiory rozmyte kodowane są jako (rozmyte) wagi połączeń.
- System neuronowo – rozmyty może być zawsze (przed, w czasie oraz po nauce) interpretowany jako system z regułami rozmytymi. Możliwe jest stworzenie systemu od podstaw, nie mając na początku żadnych reguł, jak również możliwe jest zapewnienie reguł rozmytych na starcie oraz ich redukcja w czasie nauki.
- Proces uczenia systemów neuronowo-rozmytych angażuje semantyczne właściwości zasadniczych systemów rozmytych. To skutkuje ograniczeniem możliwych modyfikacji parametrów systemu.
- System neuronowo – rozmyty aproksymuje n – wymiarową (nieznaną) funkcję, której próbka podana jest jako dane do nauki. System neuronowo – rozmyty nie powinien być jednak postrzegany jako pewien rodzaj rozmytego systemu eksperckiego.

Podejścia, w których sieci neuronowe zostały wykorzystane tylko do zapewnienia wejścia dla systemu rozmytego lub do zmiany wyjścia takiego systemu, autorzy wolą nazywać *kombinacją sieci neuronowych i systemów rozmytych* lub *współistniejącymi sieciami neuronowymi i systemami rozmytymi*. Określenie *system neuronowo – rozmyty* zostało więc użyte dla podkreślenia, że reguły rozmyte reprezentujące wiedzę zostały umieszczone wewnątrz sieci neuronowej, co przekłada się na skrócenie czasu potrzebnego do wyuczenia takiej sieci. Zapewnia to również możliwość interpretacji takiej sieci w formie zbioru reguł lingwistycznych typu Jeżeli – to (ang. *If – then rules*).

3 Rozmyty perceptron

Rozmyty perceptron posiada architekturę typową dla wielowarstwowego perceptronu znanego z klasycznego podejścia do sieci neuronowych. Główna różnica polega na modelowaniu wag połączeń jako zbiorów rozmytych w systemach neuronowo – rozmytych. Podejście takie wymagało również pewnych modyfikacji w funkcjach aktywacji, wyjścia oraz propagacji. Założeniem tego modelu było zapewnienie możliwości jego interpretacji w formie reguł lingwistycznych oraz wykorzystania wcześniejszej wiedzy umieszczonej w sieci neuronowej w celu skrócenia czasu potrzebnego do jej wyuczenia. Autorzy tego podejścia zaproponowali taki ogólny model systemu neuronowo – rozmytego jako trójwarstwową sieć neuronową. Z matematycznego punktu widzenia, rozmyty perceptron został opisany w Definicji 1 [2].

Definicja 1 *Trójwarstwowy rozmyty perceptron jest trójwarstwową siecią neuronową zdefiniowaną jako szóstka (U, W, NET, A, O, ex).*

- *U jest niepustym zbiorem jednostek, które wchodzą w skład poszczególnych warstw systemu.* U_1 *reprezentuje warstwę wejściową,* U_2 *– warstwę ukrytą,* U_3 *– warstwę wyjściową.*
- *W określa strukturę sieci neuronowej jako zbiór połączeń pomiędzy jej elementami.*
- *A definiuje funkcję aktywacji* A_u *dla każdej jednostki* $u \in U$ *do obliczenia aktywacji* a_u.
- *O definiuje dla każdego* $u \in U$ *funkcję wyjścia* O_u *do obliczenia wartości sygnału wyjściowego danej jednostki.*
- *NET definiuje dla każdego* $u \in U$ *funkcję propagacji* NET_u.
- *ex definiuje dla każdej jednostki* $u \in U_1$ *jej zewnętrzne wejście* $ex(u) = ex_u$. *Pozwala to na propagację sygnału wejściowego na elementy warstwy wejściowej. Dla wszystkich pozostałych jednostek (pozostałych warstw) ex nie jest definiowana.*

Rozmyty perceptron może być postrzegany jako trzywarstwowy perceptron, który został rozmyty w pewnych obszarach. Z definicji tylko wagi, funkcja propagacji oraz funkcja aktywacji elementów warstwy wyjściowej (i tylko elementów warstwy wyjściowej) są modelowane jako zbiory rozmyte. Rozmyty perceptron, podobnie jak jego klasyczny odpowiednik, używany jest do aproksymacji funkcji. Największą zaletą takiego podejścia jest fakt, że jego struktura może być interpretowana w formie reguł lingwistycznych, ponieważ rozmyte wagi mogą skojarzone z pewnymi lingwistycznymi określeniami. Z tego powodu rozmyty perceptron został zdefiniowany jako trójwarstwowa sieć. Możliwe jest jednak skierowanie wyjść warstwy trzeciej (wyjściowej) na dodatkową, kolejną warstwę ukrytą (z kolejnymi regułami), po której występuje kolejna warstwa wyjściowa. W ten sposób rozmyty perceptron może być rozbudowywany o kolejne warstwy ukryte, które mogą zapewniać różną funkcjonalność.

4 Algorytm uczący

Algorytm uczący w systemach neuronowo – rozmytych nie jest tak prostolinijny jak w typowych sieciach neuronowych. Wynika to z faktu zastosowania *t-norm* oraz *t-konorm* jako funkcji aktywacji dla poszczególnych elementów systemu, które nie są różniczkowalne. Nie można więc w tym podejściu wykorzystać znanej metody gradientowej. Wymagało to więc opracowania specjalnego algorytmu uczącego opartego o rozmyty błąd (ang. *Fuzzy error*). Metoda obliczenie błędu opiera się na różnicy pomiędzy oczekiwanym sygnałem wyjściowym a aktualnym sygnałem wyjściowym (jest to więc metoda nauczania z nadzorem) i została przedstawiona w Definicji 2 [2].

Definicja 2 *Przyjmijmy trójwarstwowy rozmyty perceptron, który posiada n wejść i m wyjść. Niech* $t_u^{(p)}$ *będzie oczekiwanym sygnałem wyjściowym pojedynczego elementu warstwy wyjściowej dla wektora wejściowego* $i^{(p)}$, $o_u^{(p)}$ *będzie aktualną wartością wyjścia oraz* $range_u$ *będzie różnicą pomiędzy maksymalną i minimalną wartością wyjścia dla pojedynczej jednostki warstwy wyjściowej u. Rozmyty błąd może zostać obliczony wg wzoru:*

$$E_u^{(p)} = 1 - exp\left(\beta \left(\frac{t_u^{(p)} - o_u^{(p)}}{range_u}\right)^2\right) \tag{1}$$

gdzie $\beta \in R$ *określa poziom skalowania.*

Poziom skalowania β pozwala na dostosowanie czułości błędu rozmytego. Dzięki dobraniu odpowiedniej wartości tego parametru, zależnej od konkretnego problemu, możemy zwiększyć lub zmniejszyć tolerancję odchylenia otrzymanego sygnału wejściowego od wartości oczekiwanej. Jest to użyteczne np. w problemach klasyfikacji, gdzie nie ma potrzeby otrzymania dokładnie takiej samej wartości wyjściowej, aby zakwalifikować obiekt do danej grupy.
Posiadając jasno określony wzór na obliczenie błędu wyjścia systemu neuronowo – rozmytego, możemy zdefiniować algorytm propagacji tego błędu. Ponieważ przyjęty model posiada architekturę wielowarstwową, niezbędne było opracowanie specjalnej wersji algorytmu propagacji wstecznej błędu rozmytego, który przedstawia Definicja 3 [2].

Definicja 3 *Załóżmy trójwarstwowy rozmyty perceptron oraz zadanie testowe L. Ogólny algorytm propagacji wstecznej błędu rozmytego można zdefiniować wg schematu:*

1. *Wybierz dowolny wzorzec uczący* $p \in L$, *a następnie rozpropaguj wektor wejściowy* $i^{(p)}$.
2. *Oblicz:*

$$\delta_u^{(p)} = \begin{cases} \operatorname{sgn}(t_u^{(p)} - o_u^{(p)}) \cdot E_u^{(p)} & for\ u \in U_3 \\ \sum_{v \in U_3} a_u^{(p)} \cdot \delta_v^{(p)} & for\ u \in U_2 \end{cases} \tag{2}$$

3. *Oblicz:* $\Delta_p W(u,v) = f(\delta_v^{(p)}, a_u^{(p)}, \mathrm{net}_v^{(p)}), \ (u \in U_i, \ v \in U_j, \ i,j \in M, j = i+1)$

Powtarzaj powyższe kroki dla wszystkich $p \in L$, dopóki ogólny błąd

$$E = \sum_{p \in \mathcal{L}} \sum_{u \in U_3} E_u^{(p)} \tag{4}$$

będzie większy, niż tolerancja po zakończeniu epoki uczącej.

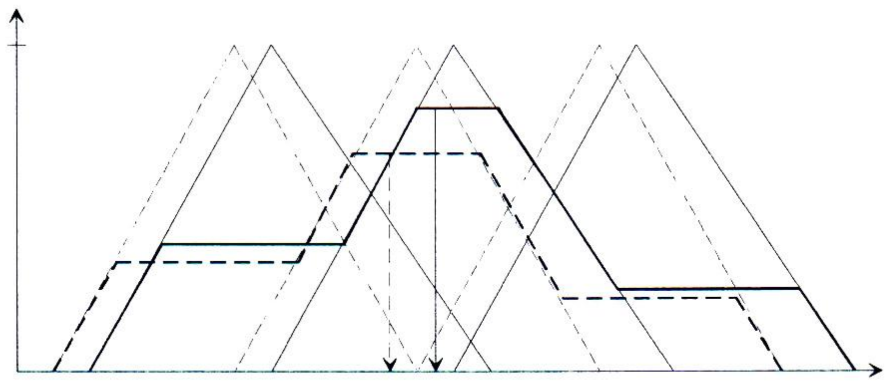
Jak można zaobserwować, w pierwszej kolejności obliczana jest wartość błędu dla jednostek wchodzących w skład warstwy wyjściowej. Dzięki zastosowaniu funkcji *sgn* wartość błędu jest dodatkowo modelowana, przyjmując wartość dodatnią, ujemną lub zerową, gdy wartość otrzymana jest równa wartości oczekiwanej. Tak obliczona wartość δ jest propagowana na elementy warstwy ukrytej. Dla pojedynczego elementu w warstwie ukrytej wartości z poszczególnych elementów warstwy wyjściowej, z którymi ten element posiada połączenie, są mnożone przez wartość funkcji aktywacji, a następnie sumowane ze sobą. W ten sposób obliczana jest wartość sygnału δ dla elementu warstwy ukrytej. Wartość ta jest wykorzystywana w funkcji modyfikującej wartości wag połączeń elementów poszczególnych warstw. Zależność ta powinna być wyspecyfikowana zależnie od aktualnie wykorzystywanego rozmytego perceptronu. Zazwyczaj zbiory rozmyte są zdefiniowane przez sparametryzowaną funkcje przynależności, a wagi połączeń modelowane są właśnie przez zbiory rozmyte. Wszelkie modyfikacje wag powinny więc polegać na modyfikacji parametrów funkcji przynależności opisujących zbiór rozmyty, będący wagą danego połączenia.

5 Rozmyty perceptron typu Min – Max

Przykładem zastosowania perceptronu rozmytego, który zobrazuje sposób opisu oraz modyfikacji wag połączeń, jest rozmyty perceptron typu Min – Max[2]. Jest to trójwarstwowy rozmyty perceptron, w którym użyto funkcji *min* oraz *max* jako t – norm i t – konorm. Wagi połączeń przychodzących do elementów poszczególnych warstw zostały rozpisane jako $\mu_{i,j} = W(u_i,u_j)$, gdzie $u_i \in U_1$, $u_j \in U_2$ oraz $v_{j,k} = W(u_j,u_k)$, gdzie $u_j \in U_2$, $u_k \in U_3$. Każda z wag została opisana przez funkcję trójkątną, która może być zdeterminowana przez podanie trzech parametrów $l_{i,j}$, $c_{i,j}$, $r_{i,j}$, gdzie jeżeli $\mu_{i,j}(c_{i,j}) = 1$, to $\mu_{i,j}(l_{i,j}) = \mu_{i,j}(r_{i,j}) = 0$ oraz $l_{i,j} \le c_{i,j} \le r_{i,j}$. Zbiór $[l_{i,j}, r_{i,j}]$ możemy nazwać nośnikiem (ang. *support*) zbioru rozmytego. Nośnik jest zbiorem elementów, dla których wartość funkcji jest większa od 0. Przy funkcji trójkątnej, wykres swoim kształtem przypomina trójkąty – w takim przypadku $l_{i,j}$ byłoby równe współrzędnej x lewego wierzchołka przy podstawie trójkąta, natomiast $r_{i,j}$ – prawego wierzchołka przy podstawie trójkąta. Wartość $c_{i,j}$ określana jest jako rdzeń (ang. *core*) – jest to współrzędna x punktu, w którym wartość funkcji jest maksymalna (np. równa 1). W takim przypadku funkcja przynależności, która opisuje zbiór rozmyty będący wagą połączenia przyjmuje postać:

$$\mu_{i,j}(x) = \begin{cases} \dfrac{x - l_{i,j}}{c_{i,j} - l_{i,j}} & \text{dla} \quad x \in [l_{i,j}, c_{i,j}), \\ \dfrac{r_{i,j} - x}{r_{i,j} - c_{i,j}} & \text{dla} \quad x \in [c_{i,j}, r_{i,j}], \\ 0 & \text{w pozostałych przypadkach} \end{cases} \tag{5}$$

W ten sam sposób opisane są wagi $v_{j,k}$.
Jeżeli wartość wyjścia pojedynczego elementu warstwy wyjściowej musi zostać zwiększona, aby odpowiadać wartości oczekiwanej tego elementu, nośnik zbioru rozmytego powinien zostać przesunięty do większych wartości. Oznacza to zwiększenie wartości parametrów *l*, *c* oraz *r*. Jeżeli stopień przynależności sygnału wejściowego do funkcji musi zostać zwiększony, należy rozszerzyć nośnik zbioru rozmytego. Oznacza to np. zwiększenie wartości parametru *r* oraz zmniejszenie wartości parametru *l*. Taką operację można zaobserwować na wykresie funkcji jako rozciągnięcie jej wzdłuż osi x. Rysunek 1 przedstawia zmianę w wykresie funkcji przynależności, która opisuje zbiór rozmyty (czyli wagę połączenia) za pomocą wcześniej opisanej strategii zmiany parametrów *l*, *c* oraz *r*.



Rysunek 1: Dostosowanie funkcji przynależności opisującej zbiór rozmyty. Linią przerywaną oznaczono stan początkowy, linią ciągłą – stan końcowy, po zmianie parametrów. ([2], str. 5)

Wprowadzony przykład perceptronu rozmytego miał na celu nie tylko zademonstrowanie, w jaki sposób mogą być opisane wagi połączeń w systemie neuronowo – rozmytym, ale także by zobrazować sposób modyfikacji wag w procesie uczenia. Jak wcześniej zostało wspomniane, nie można podać jednego, uniwersalnego wzoru na modyfikację wag połączeń, gdyż wagi te nie są opisane przez liczby, lecz zbiory rozmyte. Dlatego funkcja modyfikacji wag musi być dobierana do konkretnego, aktualnie wykorzystywanego perceptronu. W przypadku perceptronu

Min – Max algorytm propagacji wstecznej błędu rozmytego może przyjąć postać, jak w Definicji 4.

Definicja 4 *Mając dany trójwarstwowy rozmyty perceptron, w którym użyto funkcji min oraz max jako t – norm i t – konorm, algorytm wstecznej propagacji błędu może zostać zdefiniowany w następujący sposób:*

Ustaw ogólny błąd E = 0, a następnie dla każdego wzorca uczącego $p \in L$ powtarzaj następujące kroki:

1. *Wybierz dowolny wektor uczący $p \in L$, który nie został jeszcze wybrany w bieżącej epoce uczenia i rozpropaguj wektor wejściowy $i^{(p)}$.*
2. *Oblicz δ_u dla każdego elementu warstwy ukrytej oraz wyjściowej oraz zaktualizuj ogólny błąd E, zgodnie z Definicją 3.*
3. *Oblicz zmianę parametrów wag rozmytych. Dla elementów warstwy wyjściowej, w której występują współczynniki uczenia σ_l, σ_c, $\sigma_r \in R$:*

$$\begin{aligned} \Delta_p l_{j,k} &= \sigma_l \cdot \delta_{u_k} \cdot (c_{j,k} - l_{j,k}), \\ \Delta_p c_{j,k} &= \sigma_c \cdot \delta_{u_k} \cdot (r_{j,k} - l_{j,k}), \\ \Delta_p r_{j,k} &= \sigma_r \cdot \delta_{u_k} \cdot (r_{j,k} - c_{j,k}), \end{aligned} \tag{6}$$

   oraz dla elementów warstwy ukrytej, w której występują współczynniku uczenia η_l, η_c, $\eta_r \in R$:

$$\begin{aligned} \Delta_p l_{i,j} &= -\eta_l \cdot \delta_{u_j} \cdot (c_{j,k} - l_{i,j}), \\ \Delta_p c_{i,j} &= \eta_c \cdot \delta_{u_j} \cdot (a_{u_i} - c_{i,j}), \\ \Delta_p r_{i,j} &= \eta_r \cdot \delta_{u_j} \cdot (r_{i,j} - c_{j,k}). \end{aligned} \tag{7}$$

Algorytm kończy swoje działanie, kiedy ogólny błąd E jest stosunkowo mały i akceptowalny. W przeciwnym wypadku, powtarzamy wszystkie kroki dla kolejnej epoki.

Wykorzystanie współczynników uczenia pozwala na kontrolowanie wielkości zmian poszczególnych parametrów modelujących wagi połączeń. Może być to przydatne, kiedy chcemy np. tylko w nieznacznym stopniu zmieniać pozycję zbioru rozmytego i spróbować wyuczyć sieć głównie przez rozszerzanie lub zwężanie nośnika zbioru rozmytego. Należy także pamiętać, aby po każdej modyfikacji parametrów sprawdzać, czy dokonane zmiany nie powodują sprzeczności warunku $l \leq c \leq r$. Obliczone wg powyższych wzorów zmiany poszczególnych parametrów są dodawane do ich aktualnej wartości.

Taki typ architektury systemu neuronowo – rozmytego może zostać wykorzystany do aproksymacji funkcji, której posiadamy pewien zbiór przykładowych danych. Wyuczony system może być wtedy użyteczny do takich celów, jak klasyfikacja bądź zadania kontroli. Dodatkowo otrzymujemy zbiór rozmytych reguł Jeżeli – To (ang. *fuzzy if-then rules*) do interpretacji przykładowych danych, jeżeli możliwe jest przypisanie reguł językowych (lingwistycznych) do zbiorów rozmytych rozwiniętych podczas procesu uczenia.

6 Podsumowanie

Zaprezentowany trójwarstwowy model rozmytego perceptrony jest bardzo użyteczną kombinacją sieci neuronowych oraz systemów rozmytych. Model taki może zostać użyty jako ogólny model dla różnych podejść w dziedzinie systemów neuronowo – rozmytych. Ułatwi to wytłumaczenie zasady działania takich systemów, modelowania wag w postaci zbiorów rozmytych i modyfikacji parametrów opisujących te wagi. Może być również pomocny w porównaniu różnych podejść, które początkowo różniły się architekturą, ilością warstw czy funkcjami aktywacji. Zaprezentowany model rozmytego perceptronu może być zainicjowany korzystając z wcześniejszej wiedzy podanej w postaci rozmytych reguł Jeżeli – To, co może przyczynić się do skrócenia czasu potrzebnego do wyuczenia się sieci. Możliwa jest również interpretacja takiego perceptronu w postaci reguł rozmytych. Sieć neuronowa przestaje być wtedy czarną skrzynką, do której podajemy dane i otrzymujemy wynik, nie wiedząc, w jaki sposób został uzyskany. Możliwe jest oczywiście traktowanie systemu neuronowo – rozmytego bez interpretacji jego struktury, traktując go jako typową sieć neuronową. Jednak możliwość interpretacji oraz jasność semantyczna takiego systemu zapewnia możliwość sprawdzenia i utrzymania wiarygodności takiego systemu, co jest niewątpliwą zaletą takiego podejścia.

7 Bibliografia

6.1 Prace, do których występuje bezpośrednie nawiązanie w tekście

1. Nauck: Neuro-Fuzzy Systems: Review and Prospects, opracownie z Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, Sep. 8-11, 1997
   http://fuzzy.cs.uni-magdeburg.de/archive/pub/papers/eufit97b.ps.gz
2. Nauck: A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches, opracownie z 2nd German GI-Workshop "Fuzzy-Systeme'94 in Munich, Oct. '94"
   http://fuzzy.cs.uni-magdeburg.de/archive/pub/papers/fuzsys94.ps.gz

6.2 Prace pomocne przy opracowaniu tematu

3. Kruse, Nauck: Learning Methods for Fuzzy Systems, opracowanie z Third German GI-Workshop "Fuzzy-Neuro-Systeme'95", Darmstadt, Germany, November 15 - 17, 1995
   http://fuzzy.cs.uni-magdeburg.de/archive/pub/papers/fuz95c.ps.gz
4. Nauck, Klawonn, Kruse: Combining Neural Networks and Fuzzy Controllers.
5. Nauck: Beyond Neuro-Fuzzy: Perspectives and Directions, opracowanie z Third European Congress on Intelligent Techniques and Soft Computing (EUFIT'95) in Aachen
   http://fuzzy.cs.uni-magdeburg.de/archive/pub/papers/eufit95.ps.gz
6. Nauck, Kruse: Neuro-Fuzzy Systems for Function Approximation, opracowanie z 4. International Workshop Fuzzy-Neuro Systems 1997 in Soest
   http://fuzzy.cs.uni-magdeburg.de/archive/pub/papers/fns97.ps.gz