

Sieci neuronowe i systemy rozmyte w automatyce

Michał Kochan 197678

17.01.2017

Spis treści

1	Wstęp	1
2	Podstawy modelowania sieci neuronowych	2
2.1	Budowa sieci neuronowych	2
2.2	Struktura sieci neuronowej	3
2.3	Sieci jednokierunkowe jednowarstwowe	3
2.4	Sieci jednokierunkowe wielowarstwowe	4
2.5	Sieci rekurencyjne	4
2.6	Sieci komórkowe	4
3	Uczenie sieci neronowych	5
3.1	Reguła Delta	5
3.2	Reguła Hebba	6
4	Algorytmy uczące w sieciach neuronowych	7
4.1	Algorytm wstecznej propagacji błędu	7
4.2	Algorytm QUICKPROP	8
4.3	Algorytm RPROP	8
5	Podstawy modelowania opartego na logice rozmytej	8
5.1	Reguły sterowania rozmytego	8
5.2	Proces wnioskowania	9
5.2.1	Baza reguł	9
5.2.2	Blok rozmywania	10
5.2.3	Blok wnioskowania (inferencji)	10
5.2.4	Blok wyostrzania	10
5.3	Strojenie sterownika	12
6	Modelowanie rozmyte	12
6.1	Modelowanie za pomocą eksperta	12
6.2	Modele samonastrajające się na bazie danych pomiarowych	13
6.3	Model Mamdaniego	13
6.4	Model Takagi-Sugeno	13
7	Podejście neuronowo-rozmyte	14
7.1	Architektura	14

8	Przykłady użycia	14
8.1	Obszary zastosowań	14
8.2	Zastosowania praktyczne	14

Streszczenie

W pracy zaprezentowano podstawowe zagadnienia z zakresu techniki sieci neuronowych oraz systemów rozmytych. Opisano budowę oraz pracę sieci neuronowej ze szczególnym uwzględnieniem procesu jej uczenia oraz budowę i pracę systemu rozmytego. Podano podstawowe wymogi niezbędne dla prawidłowego funkcjonowania sztucznych sieci neuronowych i systemów rozmytych oraz zaprezentowano ich praktyczne zastosowania.

1 Wstęp

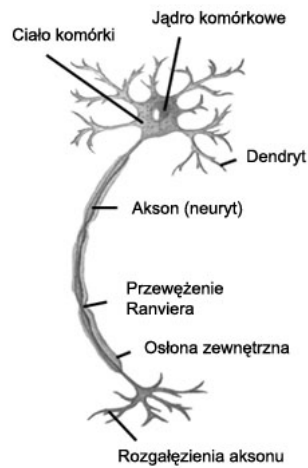
Sztuczne sieci neuronowe stanowią alternatywną metodę symulacji zdolną do odwzorowywania nadzwyczaj złożonych funkcji a nawet do modelowania słabo zdefiniowanych procesów fizycznych. Podstawową ich zaletą jest zdolność do zamodelowania skomplikowanych, nieliniowych zagadnień na drodze prezentacji przykładowych danych wejściowych, ewentualnie zestawów danych wejściowych i wyjściowych. Posiadają one zdolność do obsługiwania dużych i skomplikowanych systemów z wieloma wzajemnie powiązаныmi wielkościami. Cechę tę zawdzięczają umiejętności „uczenia się” zależności istniejących między parametrami wejściowymi i wyjściowymi. Prezentowane sieci dane mogą zawierać sygnały o charakterze zakłóceń a nawet sygnały nadmiarowe. Sieci „potrafią” ignorować dane nadmiarowe oraz te, których wpływ na badane zjawisko jest pomijalnie mały, „koncentrując” się na wielkościach wejściowych o decydującym znaczeniu dla opisu modelowanego procesu. Powyższe zadecydowało o szerokim ich wykorzystaniu w wielu gałęziach nauki i techniki. Dotyczy to w szczególności zagadnień optymalizacji, nadzoru, robotyki, prognozowania nawet obróbki sygnałów, modelowania i sterowania obiektów dynamicznych, predykcji obciążeń systemu elektroenergetycznego, rozpoznawania wzorców, kompresji danych. Szczególnie użyteczne są w obszarze zagadnień związanych z modelowaniem. Modele oparte o technikę obliczeń wykorzystujących sztuczne sieci neuronowe mogą być traktowane jako jedne z metod predykcji oraz analizy inżynierskiej.

Celem wprowadzenia pojęcia i teorii zbiorów rozmytych była potrzeba matematycznego opisanie tych zjawisk i pojęć, które mają charakter wieloznaczny oraz nieprecyzyjny. W teorii tej możemy mówić o częściowej przynależności punktu do rozważanego zbioru. Zamiast zdaniem przyjmującymi wartości prawda lub fałsz posługujemy się zmiennymi lingwistycznymi, które przyjmują jako wartości nieprecyzyjne pojęcia języka mówionego. Tak jak coś co jest ciepłe nie jest ani gorące ani zimne. Dzięki temu możliwe jest opisywanie takich cech obiektów jak: bardzo, trochę, średnio, mało, niewiele. Na systemy rozmyte składają się te techniki i metody, które służą do obrazowania informacji nieprecyzyjnych, nieokreślonych bądź niekonkretnych. Pozwalają one opisywać zjawiska o charakterze wieloznacznym, których nie jest w stanie ująć teoria klasyczna i logika dwuwartościowa. Charakteryzują się tym, że wiedza jest przetwarzana w postaci symbolicznej i zapisywana w postaci rozmytych reguł. Systemy rozmyte znajdują zastosowanie tam, gdzie nie posiadamy wystarczającej wiedzy o modelu matematycznym rządzącym danym zjawiskiem oraz tam gdzie odtworzenie tegoż modelu staje się nieopłacalne lub nawet niemożliwe. Tak więc możemy je spotkać w bazach danych, sterowaniu oraz dziedzinach zajmujących się przetwarzaniem języka naturalnego.

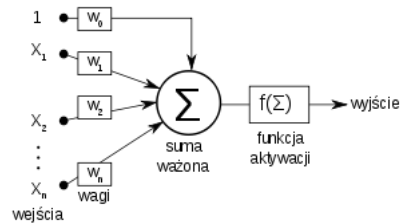
2 Podstawy modelowania sieci neuronowych

2.1 Budowa sieci neuronowych

Sztuczny neuron skonstruowany został na wzór neuronu naturalnego. Wejścia to odpowiedniki dendrytów, wagi to cyfrowe odpowiedniki modyfikacji dokonywanych na sygnałach przez synapsy. Blok sumujący to odpowiednik jądra, blok aktywacji to wzgórek aksonu, a wyjście - to akson. Proces przetwarzania sygnału w sztucznym neuronie w sposób ogólny można przedstawić następująco: Wejścia dostarczają sygnał, który następnie jest mnożony przez współczynniki wag, następnie w bloku sumowania następuje sumowanie pomnożonych sygnałów. Wynikiem tego otrzymujemy sygnał zwany potencjałem membranowym. Następnie sygnał przetworzony zostaje w bloku aktywacji, który w zależności od potrzeb może być opisany różnymi funkcjami - zwanymi funkcjami aktywacji. Wartość funkcji aktywacji jest sygnałem wyjściowym neuronu i propagowana jest do neuronów warstwy następnej.



Rysunek 1: Budowa prawdziwego neuronu.



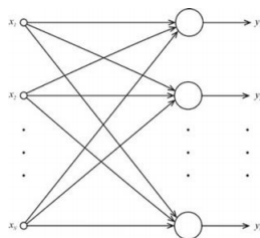
Rysunek 2: Budowa sztucznego neuronu.

2.2 Struktura sieci neuronowej

Sztuczne sieci neuronowe naśladują proces uczenia mózgu człowieka. Sieć stanowi grupa wzajemnie połączonych neuronów. Sieć neuronowa gromadzi i przechowuje informacje w postaci „siły” wzajemnych połączeń neuronów, zwanych wagami. Z uwagi na brak praktycznej możliwości prawidłowego ustawienia wag, konieczne jest wykorzystanie zdolności tzw. „uczenia się sieci”, celem ustawienia wag. Proces taki odbywa się w fazie nauki. Jedną z metod realizacji fazy nauki jest tzw. uczenie nadzorowane, w czasie którego każda z danych wejściowych oraz odpowiadający jej wzorzec są wielokrotnie prezentowane sieci. Różnica pomiędzy wartością wzorca i odpowiadającego mu, wygenerowanego przez sieć, sygnału wyjściowego, stanowi miarę stopnia modyfikacji wag. Najbardziej popularnym i skutecznym algorytmem uczenia wielowarstwowej sieci neuronowej jest metoda wstecznej propagacji błędów (ang. BackPropagation - BP). Modyfikacjami algorytmu BP są: momentowa metoda wstecznej propagacji błędów, metody ze zmiennymi współczynnikami uczenia i momentu oraz metody uwzględniające alternatywną postać funkcji błędu. Aby uzyskać dokładny model dla całej przestrzeni zmian wartości parametrów wejściowych sieć musi posiadać umiejętność uogólniania. W tym celu w procesie uczenia należy uwzględnić dostatecznie dużą liczbę danych uczących, pokrywających swymi wartościami cały zakres zmienności. Sieć neuronowa wyuczona na wąskim zakresie danych wejściowych może nie mieć zdolności do generowania prawidłowych odpowiedzi. Z drugiej jednak strony, sieć wyuczona na zbyt dużym zakresie może utracić zdolności do koncentracji na rzeczywistych zależnościach a stąd dokładność modelu również może ulec pogorszeniu. Poprawę zdolności i dokładności przewidywania sieci neuronowej można uzyskać stosując odpowiednią liczbę neuronów ukrytych oraz poprzez dobór właściwej funkcji aktywacji. Podczas modelowania przy użyciu sieci neuronowych funkcję aktywacji z reguły wybiera twórca sieci a wagi są dopasowywane w fazie nauki sieci.

2.3 Sieci jednokierunkowe jednowarstwowe

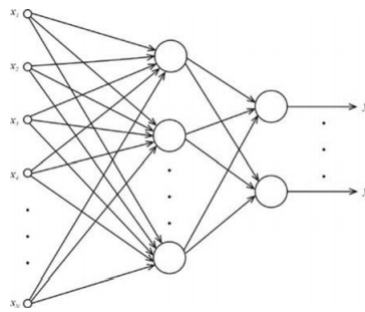
W sieciach tego typu neurony ułożone są w jednej warstwie, która jest zasilana z węzłów wejściowych. Przepływ sygnału w tego typu sieciach przebiega zawsze w ściśle określonym kierunku: od warstwy wejściowej do warstwy wyjściowej. Na węzłach wchodzących nie znajdują się warstwy neuronów, gdyż nie zachodzi w nich żaden proces obliczeniowy. Dobór wag następuje tu w procesie uczenia sieci, czyli dopasowania sygnałów wyjściowych y_i do wartości oczekiwanej d_i .



Rysunek 3: Sieć jednokierunkowa jednowarstwowa.

2.4 Sieci jednokierunkowe wielowarstwowe

Cechą wyróżniającą te sieci od pozostałych jest występowanie oprócz warstwy wejściowej i wyjściowej, co najmniej jednej warstwy ukrytych neuronów. Sygnały wejściowe są podawane na pierwszą warstwę ukrytą neuronów, a te z kolei stanowią sygnały źródłowe dla kolejnej warstwy. Na wartość sygnału wejściowego mają wpływ wagi obu warstw, podczas gdy sygnały wytwarzane w warstwie ukrytej nie zależą od wag warstwy wyjściowej. Uczenie sieci wielowarstwowej odbywa się zwykle z nauczycielem. Uczenie odbywa się w celu określenia wag na wszystkich warstwach sieci, tak, aby przy zadanej wartości $x(i)$ uzyskać na wyjściu wartość $y(i)$, która będzie odpowiadała z dużą dokładnością wartości żądanej $d(i)$.



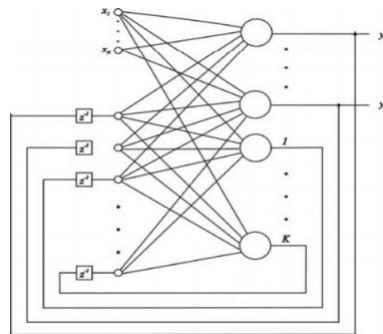
Rysunek 4: Sieć jednokierunkowa wielowarstwowa.

2.5 Sieci rekurencyjne

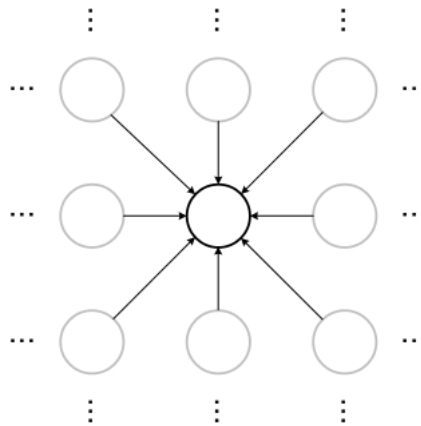
Możemy tu wyróżnić sieci rekurencyjne jednowarstwowe mające jedną warstwę neuronów wyjściowych oraz sieci rekurencyjne wielowarstwowe, które posiadają dodatkową warstwę ukrytą. Od sieci jednokierunkowych różni je występowanie sprzężenia zwrotnego między warstwami wyjściowymi i wejściowymi. Sprzężenie zwrotne polega tu na przekazywaniu sygnałów z warstwy wyjściowej bądź ukrytej do warstwy wejściowej. Zmiana stanu jakiegokolwiek neuronu na skutek sprzężenia zwrotnego zostaje przeniesiona na całą sieć, wywołując przy tym stan przejściowy, kończący się określonym stanem, który został uprzednio ustalony, ogólnie innym, niż wcześniejszy.

2.6 Sieci komórkowe

W tych sieciach sprzężenia wzajemne między elementami przetwarzającymi dotyczy jedynie najbliższego sąsiedztwa. Połączenia te są w ogólności nieliniowe i opisane poprzez układ równań różniczkowych. Podstawową trudność w stosowaniu tego typu sieci stanowi opracowanie skutecznej, efektywnej i uniwersalnej metody projektowania. Typowym przykładem sieci komórkowej może być sieć typu mapa Kohonena.



Rysunek 5: Sieć rekurencyjna wielowarstwowa.



Rysunek 6: Sieć komórkowa.

3 Uczenie sieci neronowych

Zostaną omówione dwie podstawowe reguły uczenia: reguła Delta leżąca u podstaw większości algorytmów uczenia z nauczycielem oraz reguła Hebba stanowiąca przykład uczenia bez nauczyciela.

3.1 Reguła Delta

Reguła delta jest regułą uczenia z nauczycielem. Polega ona na tym, że każdy neuron po otrzymaniu na swoich wejściach określone sygnały (z wejść sieci albo od innych neuronów, stanowiących wcześniejsze piętra przetwarzania informacji) wyznacza swój sygnał wyjściowy wykorzystując posiadaną wiedzę w postaci wcześniej ustalonych wartości współczynników wzmocnienia (wag) wszystkich wejść oraz (ewentualnie) progu. Wartość sygnału wyjściowego, wyznaczonego przez neuron na danym kroku procesu uczenia porównywana jest z odpowiedzią wzorcową podaną przez nauczyciela w ciągu uczącym. Jeśli występuje rozbieżność - neuron wyznacza różnicę pomiędzy swoim sygnałem wyjściowym a tą wartością sygnału, która była by - według nauczyciela prawidłowa. Sygnał

błędu (delta) wykorzystywany jest przez neuron do korygowania swoich współczynników wagowych (i ewentualnie progę), stosując następujące reguły:

- wagi zmieniane są tym silniej, im większy błąd został wykryty;
- wagi związane z tymi wejściami, na których występowały duże wartości sygnałów wejściowych, zmieniają się bardziej, niż wagi wejść, na których sygnał wejściowy był niewielki.

Znając błąd popełniony przez neuron oraz jego sygnały wejściowe możemy łatwo przewidzieć, jak będą się zmieniać jego wagi.

Sieć stosując opisane metody w praktyce sama przerywa proces uczenia gdy jest już dobrze wytrenowana, gdyż małe błędy powodują jedynie minimalne korekty wag. Jest to logiczne, podobnie jak zasada uzależniania wielkości korekty od wielkości wejściowego sygnału przekazywanego przez rozważaną wagę, te wejścia, na których występowały większe sygnały miały większy wpływ na wynik działania neuronu, który okazał się błędny, trzeba je więc silniej "temperować".

W praktycznych realizacjach opisanego wyżej algorytmu dochodzi kilka dalszych godnych uwagi elementów. W pierwszej kolejności twórca sieci musi zdecydować, jak silne powinny być zmiany wag powodowane przez określone wartości sygnałów wejściowych i określoną wielkość błędu. Ten współczynnik proporcjonalności h , zwany learning rate, może być wybierany dowolnie, jednak każda konkretna decyzja ma określone konsekwencje. Wybranie współczynnika za małego prowadzi do bardzo powolnego procesu uczenia (wagi są poprawiane w każdym kroku bardzo słabo, żeby więc osiągnęły pożądane wartości trzeba wykonać bardzo dużo takich kroków). Z kolei wybór za dużego współczynnika uczenia powoduje bardzo gwałtowne zmiany parametrów sieci, które w krańcowym przypadku prowadzić mogą nawet do niestabilności procesu uczenia. Konieczny jest więc kompromisowy wybór współczynnika uczenia, uwzględniający zarówno korzyści związane z szybką pracą, jak i względy bezpieczeństwa, wskazujące na konieczność uzyskania stabilnej pracy procesu uczenia.

3.2 Reguła Hebba

Jest to jedna z najpopularniejszych metod samouczenia sieci neuronowych. Polega ona na tym, że sieci pokazuje się kolejne przykłady sygnałów wejściowych, nie podając żadnych informacji o tym, co z tymi sygnałami należy zrobić. Sieć obserwuje otoczenie i odbiera różne sygnały, nikt nie określa jednak, jakie znaczenie mają pokazujące się obiekty i jakie są pomiędzy nimi zależności. Sieć na podstawie obserwacji występujących sygnałów stopniowo sama odkrywa, jakie jest ich znaczenie i również sama ustala zachodzące między sygnałami zależności.

Po podaniu do sieci neuronowej każdego kolejnego zestawu sygnałów wejściowych tworzy się w tej sieci pewien rozkład sygnałów wyjściowych - niektóre neurony sieci są pobudzone bardzo silnie, inne słabiej, a jeszcze inne mają sygnały wyjściowe wręcz ujemne. Interpretacja tych zachowań może być taka, że niektóre neurony „rozpoznają” podawane sygnały jako „własne” (czyli takie, które są skłonne akceptować), inne traktują je „obojętnie”, zaś jeszcze u innych neuronów wzbudzają one wręcz „awersję”. Po ustaleniu się sygnałów wyjściowych wszystkich neuronów w całej sieci - wszystkie wagi wszystkich neuronów

są zmieniane, przy czym wielkość odpowiedniej zmiany wyznaczana jest na podstawie iloczynu sygnału wejściowego, wchodzącego na dane wejście (to którego wagę zmieniamy) i sygnału wyjściowego produkowanego przez neuron, w którym modyfikujemy wagi.

Dokładniejsza analiza procesu samouczenia metodą Hebba pozwala stwierdzić, że w wyniku konsekwentnego stosowania opisanego algorytmu początkowe, najczęściej przypadkowe „preferencje” neuronów ulegają systematycznemu wzmocnieniu i dokładnej polaryzacji. Jeśli jakiś neuron miał „wrodzoną skłonność” do akceptowania sygnałów pewnego rodzaju - to w miarę kolejnych pokazów nauczy się te sygnały rozpoznawać coraz dokładniej i coraz bardziej precyzyjnie. Po dłuższym czasie takiego samouczenia w sieci powstaną zatem wzorce poszczególnych typów występujących na wejściu sieci sygnałów. W wyniku tego procesu sygnały podobne do siebie będą w miarę postępu uczenia coraz skuteczniej grupowane i rozpoznawane przez pewne neurony, zaś inne typy sygnałów staną się „obiektem zainteresowania” innych neuronów. W wyniku tego procesu samouczenia sieć nauczy się, ile klas podobnych do siebie sygnałów pojawia się na jej wejściach oraz sama przyporządkuje tym klasom sygnałów neurony, które nauczą się je rozróżniać, rozpoznawać i sygnalizować.

Proces samouczenia ma niestety wady. W porównaniu z procesem uczenia z nauczycielem samouczenie jest zwykle znacznie powolniejsze. Co więcej bez nauczyciela nie można z góry określić, który neuron wyspecjalizuje się w rozpoznawania której klasy sygnałów. Stanowi to pewną trudność przy wykorzystywaniu i interpretacji wyników pracy sieci. Co więcej - nie można określić, czy sieć uczona w ten sposób nauczy się wszystkich prezentowanych jej wzorców. Dlatego sieć przeznaczona do samouczenia musi być większa niż sieć wykonująca to samo zadanie, ale trenowana w sposób klasyczny, z udziałem nauczyciela.

4 Algorytmy uczące w sieciach neuronowych

Celem zastosowania algorytmów uczących jest adaptacyjny dobór wag, tak aby umożliwić sieci działanie w warunkach określonych w postaci wymagań niezbędnych do odwzorowania informacji wejściowych na dane wyjściowe.

4.1 Algorytm wstecznej propagacji błędu

Nazwa algorytmu pochodzi od sposobu obliczania błędu w warstwach sieci neuronowej. Począwszy od warstwy wyjściowej, na podstawie różnicy pomiędzy sygnałem otrzymanym i wartością pożądaną wagi połączeń neuronów są korygowane. W kolejnych warstwach błąd obliczany jest na podstawie funkcji błędu neuronów warstwy poprzedzającej. Algorytm wstecznej propagacji błędu należy do grupy gradientowych metod optymalizacji, czyli zakłada, że gradient funkcji wskazuje kierunek jej najszybszego wzrostu, a zmieniając znak na przeciwny wskaże kierunek najszybszego spadku funkcji. W ten sposób można minimalizować funkcję poprzez modyfikację współczynników jej wag, aż do osiągnięcia szybkiego spadku funkcji celu. W każdym cyklu uczącym algorytm wstecznej propagacji błędu wyróżnia następujące etapy:

-Analiza sieci neuronowej.

- Uczenie sieci poprzez odwrócenie kierunku przepływu sygnałów.
 - Uczenie sieci poprzez adaptację wag.
- Cały proces jest powtarzany dla wszystkich danych uczących, aż do momentu osiągnięcia przez sieć wartości błędu określającej dokładność procesu uczenia

4.2 Algorytm QUICKPROP

Został opracowany przez Falhmana w 1988 roku. QUICKPROP11 jest algorytmem iteracyjnym opartym na wyznaczaniu minimum funkcji błędu sieci neuronowej, która opiera się na metodzie Newtona połączonej z wiedzą heurystyczną. Na zmianę wagi pomiędzy neuronami mają wpływ składowe gradientu, aktualna wartość wagi oraz czynnik momentu związany z wartością poprzednią. Zastosowanie tego algorytmu w sieci neuronowej znacząco wpływa na przyspieszenie procesu uczenia. Zabezpiecza także przed utknięciem procesu uczenia w płytkim minimum lokalnym, gdzie ze względu na niską wartość pochodnej funkcji postęp uczenia jest niewielki.

4.3 Algorytm RPROP

RPROP (ang. Resilient Propagation) został opracowany przez Redmilla i Brauna w 1992 roku. Jest to jeden z najlepszych algorytmów trenujących ogólnego przeznaczenia. W większości przypadków jego zastosowanie znacząco usprawnia działanie sieci, co w przypadku klasyfikacji przekłada się na dokładniejsze wyniki rozpoznawania wzorców. Zastosowanie algorytmu RPROP wraz z niektórymi funkcjami aktywacji powoduje pojawienie się problemu płaskiego miejsca (ang. Flat Spot), czyli braku reakcji sieci na jakiegokolwiek wzorce. Dodatkowo RPROP zyskuje przewagę nad algorytmami wstecznej propagacji, gdyż nie wymaga określenia parametrów początkowych.

5 Podstawy modelowania opartego na logice rozmytej

Aby móc sterować pewnym procesem technologicznym lub też pracą urządzeń konieczne jest zbudowanie modelu, na podstawie którego można będzie podejmować decyzje związane ze sterowaniem. Jednakże często znalezienie odpowiedniego modelu jest problemem trudnym, niekiedy wymagającym przyjęcia różnego typu założeń upraszczających. Zastosowanie systemów rozmytych nie wymaga od nas znajomości tych procesów. Konstruujemy po prostu rozmyte reguły postępowania w postaci zdań warunkowych: IF ... THEN ...

5.1 Reguły sterowania rozmytego

Działanie sterownika rozmytego (ang. fuzzy logic controller) opiera się na zasadzie aproksymacji funkcji realizującej rzeczywisty proces. Sterownik otrzymuje wartości opisujące stan urządzenia czy systemu i przetwarza to w ostrą wartość sterującą tym urządzeniem czy systemem. Klasyczny sterownik składa się z czterech części: baza reguł, blok rozmywania, blok wnioskowania, blok wyostrzania. Każda z nich może być realizowana na wiele sposobów.



Rysunek 7: Schemat sterownika rozmytego

5.2 Proces wnioskowania

Wiele rozumowań ludzi nie ma charakteru formalnego i nie opiera się o schematy wnioskowania logiki. Dzieje się tak dlatego, że przesłanki, którymi się posługują nie są do końca pewne i przez to otrzymują wnioski uznawane tylko w pewnym stopniu. Sterowniki rozmyte naśladują ludzkie rozumowanie poprzez bardzo proste radzenie sobie w złożonych sytuacjach.

Jako regułę wnioskowania dla sterowników rozmytych stosuje się rozmytą regułę modus ponens. Reguła ta wygląda następująco:

Przesłanka: x jest A'
 Implikacja: If x jest A THEN y jest B
 Wniosek: y jest B'

Reguła ta pozwala na wywnioskowanie o prawdziwości następnika na podstawie prawdziwości poprzednika przy czym implikacja traktowana jest jako pewna relacja rozmyta. Zauważmy jednak, że A' (B') wcale nie musi być równy zbiorowi A (B). Pozwala to na pewną elastyczność. Jeśli bowiem A' jest trochę podobny do zbioru A , to zbiór B jest zbliżony do zbioru B' . Zilustrujmy to przykładem: Załóżmy, że mamy regułę

Jeśli prędkość samochodu jest duża, to poziom hałasu jest wysoki.

Niech teraz przesłanka mówi: Prędkość samochodu jest średnia. Sterownik powinien na podstawie tego wywnioskować, że: Poziom hałasu jest średnio wysoki.

5.2.1 Baza reguł

Baza reguł stanowi reprezentację wiedzy eksperta o możliwych wartościach zmiennych stanu, o pożądanym stanie urządzenia, itp. Przyjmuje się dla potrzeb sterowania, że przesłanka jak i wniosek są koniunkcjami prostych faktów rozmytych. Na bazę reguł składa się więc zbiór pewnych rozmytych reguł postaci

$$\text{IF } (x_1 \text{ jest } A_1) \text{ AND } \dots \text{ AND } (x_n \text{ jest } A_n) \text{ THEN } (y_1 \text{ jest } B_1) \text{ AND } \dots \text{ AND } (y_m \text{ jest } B_m),$$

gdzie A_i, B_j są zbiorami rozmytymi, x_i są danymi wejściowymi, a y_j są zmiennymi wyjściowymi modelu lingwistycznego.

Od trafnego wyboru zmiennych stanu i zmiennych sterujących pracą urządzenia zależy właściwe funkcjonowanie sterownika.

5.2.2 Blok rozmywania

W tej fazie konkretna wartość liczbową skojarzona z daną zmienną lingwistyczną poddana zostaje operacji rozmywania (ang. fuzzification), w wyniku której odwzorowujemy ją na zbiór rozmyty, ponieważ system sterowania z logiki rozmyta operuje na zbiorach rozmytych. W praktyce, o ile wartość na wejściu nie jest podana wraz z zakłóceniem, stosuje się rozmywanie typu singleton. W wyniku tego zmiennej x skojarzonej ze zmienną lingwistyczną A przypisuje się funkcję przynależności równą 1 dla \bar{x} i 0 wszędzie poza tym.

Jeśli mamy do czynienia z zakłóceniem lub ewentualnie błędem pomiaru możemy zastosować rozmycie gaussowskie

$$\mu(x) = \exp\left(-\frac{(x - \bar{x})^2}{\delta}\right),$$

gdzie $\delta > 0$.

5.2.3 Blok wnioskowania (inferencji)

W bloku wnioskowania ma miejsce uruchomienie każdej z reguł, której przesłanki są spełnione. Ogólnie rzecz biorąc w oparciu o przesłanki znajdujemy odpowiedni zbiór rozmyty będący wnioskiem z przyjętych reguł rozmytych.

Na podstawie bazy reguł

$$\begin{aligned} &\text{IF } (x_1 \text{ jest } A_1^1) \text{ AND } \dots \text{ THEN } (y_1 \text{ jest } B_1), \\ &\text{IF } (x_1 \text{ jest } A_1^2) \text{ AND } \dots \text{ THEN } (y_1 \text{ jest } B_1), \\ &\quad \dots \\ &\text{IF } (x_1 \text{ jest } A_1^n) \text{ AND } \dots \text{ THEN } (y_1 \text{ jest } B_1), \end{aligned}$$

oraz danych wejściowych, tzn. znajomości stopnia spełnienia przesłanki, wyliczane są zbiory rozmyte B'_i .

W przypadku, gdy mamy do czynienia z przesłanką prostą

$$\text{IF } x_i \text{ jest } A_i \text{ THEN } \dots$$

oraz rozmywania typu singleton funkcję charakterystyczną łatwo policzyć w oparciu o znajomość $\mu_{A_i}(x)$.

Podsumowując na wyjściu otrzymujemy zbiór B' (po zastosowaniu metody modus ponens), które zależą od metody rozmywania, sposobu zdefiniowania zbiorów rozmytych.

5.2.4 Blok wyostrzania

Do konkretnego sterowania potrzebna jest konkretna wielkość. Dlatego dane wyjściowe bloku wnioskowania (jeden lub więcej zbiorów rozmytych) odwzorowuje się w jedną wielkość, która będzie wyjściowym sygnałem sterowania. Można to zrealizować na wiele sposobów. Ogólnie dzielimy je na metody center

of area (COA) oraz metody średniej z największych (MOM). Przykładem pierwszej jest środek ciężkości zbioru rozmytego (CoG). Przykładem metody drugiego typu jest metoda "pierwszy z największych" (FM, ang. first of maximum).

Założmy, że blok wnioskowania zwrócił zbiór rozmyty B' . Omówimy teraz możliwe metody defuzyfikacji.

Metoda "środka najlepszego odcinka" (MBS - Middle of Best Sector)

Najpierw szukamy odcinek, dla którego wartości funkcji przynależności są maksymalne. W przypadku gdy jest ich kilka umawiamy się, że wybieramy pierwszy (ostatni, najdłuższy, ...). Gdy takiego odcinka nie ma, to bierzemy pierwszą wartość szczytowa. W przypadku gdy taki odcinek jest, to jako wartość zwracaną bierzemy jego środek.

Metoda "pierwsze z lewej maksimum" (LMM - Left Most Maximum)

Zwaracaną wartością jest x_0 takie, że

$$\mu_{B'}(x_0) = \max_{x \in X}(\mu_{B'}(x)) \text{ oraz } \forall_{x < x_0} \mu_{B'}(x) < \mu_{B'}(x_0).$$

Metoda "pierwsze z prawej maksimum" (RMM - Right Most Maximum)

Zgodnie z nazwą otrzymujemy wartość x_0 taką, że

$$\mu_{B'}(x_0) = \max_{x \in X}(\mu_{B'}(x)) \text{ oraz } \forall_{x > x_0} \mu_{B'}(x) > \mu_{B'}(x_0).$$

Metoda "środka ciężkości" (CoG - Center of Gravity)

Wartość x_0 wyliczamy jako środek ciężkości funkcji przynależności zbioru B' . W przypadku ciągłym i dyskretnym korzystamy odpowiednio ze wzorów:

$$x_0 = \frac{\int x \mu_{B'}(x) dx}{\int \mu_{B'}(x) dx} \quad x_0 = \frac{\sum_{i=1}^N x_i \mu_{B'}(x_i)}{\sum_{i=1}^N \mu_{B'}(x_i)}.$$

Metoda "maksimum funkcji przynależności"

Wartość x_0 obliczamy zgodnie z zależnością

$$\mu_{B'}(x_0) = \sup_{x \in X} \mu_{B'}(x)$$

przy założeniu, że funkcja $\mu_{B'}$ jest unimodalna. Metoda ta nie uwzględnia kształtu funkcji przynależności.

Jeśli blok wnioskowania zwrócił więcej niż jeden zbiór rozmyty możemy zastosować którąś z poniższych metod.

Metoda "środka zbioru rozmytego" (center average defuzzification)

Niech wielkością wyjściową bloku wnioskowania będą zbiory B_i , dla $i = 1, 2, \dots, N$. Zwracaną wartość w tej metodzie odczytujemy ze wzoru

$$x_0 = \frac{\sum_{i=1}^N \mu_{B'_i}(\bar{x}_i) \bar{x}_i}{\sum_{i=1}^N \mu_{B'_i}(\bar{x}_i)},$$

gdzie \bar{x}_i jest punktem, w którym μ_{B_i} przyjmuje maksimum. Punkt ten nazywamy środkiem (ang. center) zbioru rozmytego B_i .

Metoda center of sums defuzzification

Wartość wynikowa x_0 jest liczona zgodnie ze wzorem

$$x_0 = \frac{\int_X x \sum_{i=1}^N \mu_{B'_i}(x) dx}{\int_X \sum_{i=1}^N \mu_{B'_i}(x) dx}.$$

5.3 Strojenie sterownika

Strojenie polega na poprawianiu znaczeń terminów zmiennych lingwistycznych, tzn. zmianie ich parametrów, obszaru odniesienia lub odpowiednim przeskalowaniu.

Nie ma ogólnej metody optymalnego strojenia parametrów. Często wykorzystuje się wtedy różne metody numeryczne (np. metodę największego spadku) a nawet algorytmy genetyczne.

6 Modelowanie rozmyte

Celem modelowania jest uzyskanie coraz większej dokładności oraz upraszczanie struktur. Zastosowanie zbiorów rozmytych umożliwia stworzenie modelu systemu, reprezentującego istotne cechy przy pomocy aparatu teorii zbiorów rozmytych. Istnieje wiele metod tworzenia modeli. Do najprostszych należy stworzenie zestawu odpowiednich zmiennych lingwistycznych oraz zbioru reguł postaci IF ... THEN ... stanowiących jakościowy opis systemu, najbardziej bliski językowi naturalnemu.

6.1 Modelowanie za pomocą eksperta

Pierwszym rodzajem modelowania rozmytego zastosowanego w praktyce było tworzenie modeli rozmytych na bazie wiedzy eksperta. Oparte jest ono o wiedzę i doświadczenie eksperta doskonale znającego system i wszystkie jego zachowania. Model werbalny tworzony na bazie wiedzy eksperta pozwala stworzyć tylko model Mamdaniego, jednakże może to być pierwszy etap.

Modele stworzone przez różnych ekspertów mogą się różnić, ale niekiedy daje się skonstruować precyzyjne modele, np. w przypadku systemów mechanicznych

i elektrycznych. Sytuacja taka jest rzadka w przypadku systemów termicznych czy chemicznych; najgorzej jest w systemach socjologicznych i ekonomicznych.

Głównym problemem tutaj jest to, że ekspert nie zna mechanizmów wnioskowania zachodzących w jego umyśle. Dlatego jego wiedza charakteryzuje się bardziej "intuicyjnością" niż ścisłością określenia. Mając jednak wstępnie zdefiniowane reguły i zbiory rozmyte, można taki system poddać procesowi dostrajania przez co dopercyzować uzyskaną od eksperta wiedzę. Strojenie można zrealizować metodą algorytmów genetycznych lub przekształcając sterownik w rozmytą sieć neuronową.

6.2 Modele samonastrajające się na bazie danych pomiarowych

W tym modelu mamy do czynienia ze stałą bazą reguł i zbiorów rozmytych (nie podlega ona zmianom). Strojenie polega na takim dobraniu funkcji przynależności by zminimalizować błąd modelu względem modelowanego systemu. Stosuje się wtedy przeważnie przekształcenie systemu rozmytego w sieć neuronową, algorytmy genetyczne poszukujące optymalnych parametrów, metody heurystyczne i oparte na klasteryzacji.

Metody klasteryzacji polegają na automatycznym wykrywaniu skupisk (klastarów) próbek pomiarowych, które tworzą się w charakterystycznych dla systemu punktach, oraz wyciągania na tej podstawie wniosków.

6.3 Model Mamdaniego

Modele Mamdaniego stosują najbardziej naturalne z punktu widzenia logiki rozmytej podejście ponieważ opierają się na bazie reguł i stosowaniu operatorów lingwistycznych. W przypadku modelu Mamdaniego funkcją przynależności rozmytej implikacji $A \Rightarrow B$, która jest równoważna pewnej relacji rozmytej $R \subset X \times Y$, wyznaczamy następująco

$$\mu_{A \Rightarrow B}(x, y) = T(\mu_A(x), \mu_B(y)),$$

gdzie T jest dowolną t-normą, np. MIN lub iloczyn (patrz tabelka 2).

6.4 Model Takagi-Sugeno

Wadą modeli takich jak powyżej jest to, że nie zawierają jawnej postaci wiedzy obiektywnej, pomimo iż ta wiedza jest czasami dostępna. Dlatego Sugeno i współpracownicy zaproponowali bazę reguł specjalnego formatu, w której następni są typu funkcyjnego. Najczęściej mamy tu do czynienia z funkcjami liniowymi lub sklejeniem funkcji liniowych.

7 Podejście neuronowo-rozmyte

Systemy oparte na logice rozmytej, jak już wiemy, znajdują zastosowanie szczególnie tam, gdzie trudno jest opisać model opisujący dane urządzenie. Nietety przy rosnącej złożoności modelowanych procesów rosną też problemy z układaniem rozmytych reguł i funkcji przynależności. Zaprowadziło to do powstania nowego podejścia - systemów neuronowo-rozmytych. Systemy rozmyte posiadają zdolność do radzenia sobie z nieprecyzyjnymi danymi, a sieci neuronowe mają zdolność uczenia się i tworzenia nowych reguł, które uzupełniłyby brakującą wiedzę eksperta, który projektował bazę reguł. Systemy te łącząc cechy zarówno sieci neuronowych jak i systemów rozmytych znalazły się więc jednocześnie w głównym kregu zainteresowań. Urządzenia zbudowane w oparciu o architekturę systemów neuronowo-rozmytych znalazły zastosowanie w szczególności w procesach sterowania.

7.1 Architektura

System neuro-rozmyty składa się z tych samych komponentów co sterownik rozmyty, z tym że na każdym etapie obliczenia wykonywane są przez uczące się sieci neuronowe. W bloku rozmywania każdy neuron reprezentuje funkcję przynależności poprzedzającą go reguły rozmytej.

Najczęściej spotykaną architekturą jest architektura składająca się z pięciu bloków wymyślona przez C.-T. Lina.

8 Przykłady użycia

8.1 Obszary zastosowań

Sieci neuronowe i systemy rozmyte znalazły szerokie zastosowanie w wielu dziedzinach. Przykłady zastosowania:

- Bazy danych,
- Rozpoznawanie obrazów i kształtów,
- Zastosowania w systemach medycznych,
- Przetwarzanie obrazów,
- Uczenie maszynowe,
- Rozmyte zarządzanie pakietami w sieci,
- Ekonomia,

8.2 Zastosowania praktyczne

Systemy neuronowo-rozmyte znalazły szerokie zastosowanie w praktyce, których przykłady zamieszczono poniżej:

- NASA wykorzystuje systemy neuronowo-rozmyte do sterowania ramieniem manipulatora działającego w ładowni promów kosmicznych, co pozwala na utrzymywanie manipulowanych obiektów w niezmiennym położeniu w warunkach nieważkości,
- Uczeni z New York University Medical Center zastosowali systemy neuronowo-

rozmyte jako alternatywę dla złożonych i czasochłonnych tensorowych obliczeń parametrów ruchu robota, co umożliwiło przyspieszenie działania systemu sterowania a przez to pracę robota w czasie rzeczywistym,

- Firma General Dynamics opracowała dla US Navy system oparty na systemach neuronowo- rozmytych klasyfikujących i rozpoznających sygnały sonarowe, pozwalający na identyfikację jednostki pływającej a nawet obiektów nadwodnych (np. helikopter unoszący się nad powierzchnią oceanu),
- Szpital Anderson Memorial Hospital w południowej Karolinie wykorzystał systemy neuronowo- rozmyte dla celów optymalizacji leczenia, co pozwoliło na uzyskanie znacznych oszczędności, a przede wszystkim na uratowanie życia kilkudziesięciu pacjentów,
- Producent raket Genera Devices Space Systems Division użył systemy neuronowo- rozmytych do sterowania pracą 150 zaworów doprowadzających paliwo i tlen do silników rakiety Atlas, co pozwoliło na zastąpienie dotychczas stosowanego kosztownego i zawodnego, złożonego systemu automatyki opartego na setkach sensorów,
- Firma Eaton Corporation wykorzystwała systemy neuronowo- rozmyte w układzie sterowania wspomagającym pracę kierowcy dużej ciężarówki (pięć osi, osiemnaście kół) przy wykonywaniu niektórych szczególnie trudnych manewrów (np. cofaniez naczepą),
- Amerykańskie siły powietrzne US, Air Force używają systemów neuronowo- rozmytych do rozwoju symulatorów lotu,
- Koncern Ford Motor Company przygotował nowy system diagnostyczny dla silników,
- Linie TWA stosują systemy neuronowo- rozmyte do lokalizacji bomb w swoim terminalu na lotnisku JFK w Nowym Yorku,
- W energetyce, np. w elektrowni BC Hydro w Vancouver systemy neuronowo- rozmyte znalazły zastosowanie do prognozowania zapotrzebowania na moc elektryczną,
- Firma Halliburton wykorzystuje systemy neuronowo- rozmyte do identyfikacji typu skał napotykanych podczas prowadzenia odwiertów przy poszukiwaniu złóż ropy i gazu.

Literatura

- [1] „Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji”, Andrzej Łachwa, Wydawnictwo EXIT, 2001.
- [2] „Metody i techniki sztucznej inteligencji”, Leszek Rutkowski, Wydawnictwo Naukowe PWN, 2005.
- [3] „Modelowanie rozmyte”, Grzegorz Głowaty, praca magisterska napisana po kierunku A. Łachwy, 2003.
- [4] „An Introduction to Fuzzy and Neurofuzzy Systems”, M. Brown, 1996.
- [5] „Wybrane problemy zastosowania zbiorów rozmytych”, Magdalena E. Kostrzeńska, Warszawa 2006.
- [6] „Sieci neuronowe”, Nałęcz M. , Akademicka Oficyna Wydawnicza EXIT , Warszawa,2000.
- [7] „Sieci neuronowe i neurokomputery”,Rutkowski L.,Wydawnictwo Politechniki Częstochowskiej, Częstochowa,1996
- [8] „ Sieci neuronowe, algorytmy genetyczne i systemy rozmyte”, Rutkowska D., Piliński M., Rutkowski L., PWN, Warszawa,1997
- [9] „ Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami”, Tadeusiewicz R., Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1998