

## 6. Inteligentne systemy decyzyjne w technice fonicznej

Wybrane zagadnienia związane z tematem niniejszego rozdziału zostaną przedstawione w pewnym uproszczeniu, przede wszystkim na przykładzie ich zastosowań w dziedzinie rekonstruowania sygnałów fonicznych. Istnieje wiele zastosowań metod inteligentnych w technice fonicznej. Niektóre z nich, jak np. rozpoznawanie sygnałów (mowy, dźwięków instrumentów muzycznych, fraz muzycznych) oraz sterowanie urządzeniami fonicznymi (charakterystykami kierunkowości mikrofonów, automatycznym doбором prądu podkładu w zapisie magnetycznym i in.) były przedmiotem badań prowadzonych w Katedrze Inżynierii Dźwięku Politechniki Gdańskiej (KID PG). W okresie poprzedzającym napisanie niniejszej pracy, a także w trakcie jej redagowania, przedmiotem badań Katedry była także tematyka rekonstruowania sygnałów fonicznych. Dziedzina ta obejmuje zarówno zastosowania specyficzne, takie jak rekonstruowanie nagrań archiwalnych, jak również zastosowania o szerszym znaczeniu na gruncie współczesnej techniki, których celem jest redukcja szumów i zniekształceń sygnału fonicznego w torach jego rejestracji i transmisji. Badania, o których mowa, podobnie jak wcześniej wymienione eksperymenty, związane z zastosowaniami inteligentnych systemów decyzyjnych, dofinansowywał Komitet Badań Naukowych, w ramach indywidualnych projektów badawczych, którymi kierował autor pracy bądź jego współpracownicy z KID PG.

Rekonstruowanie sygnałów jest zagadnieniem, które ma długą historię rozwoju (patrz roz. 4), jednak wykorzystanie do tego celu algorytmów z dziedziny sztucznej inteligencji oraz inteligentnych technik obliczeniowych (ang. *computational intelligence, soft computing*) doprowadziło do opracowania szeregu oryginalnych metod i uzyskania interesujących wyników. Zanim zastosowania te zostaną przedstawione, w kolejnych paragrafach niniejszego rozdziału zostaną przytoczone podstawy teoretyczne wybranych algorytmów decyzyjnych, takich jak logika rozmyta, sieci neuronowe i metoda zbiorów przybliżonych.

### 6.1 Logika rozmyta

Wnioskowanie oparte na dwuwartościowej logice Arystotelesa oraz klasycznie pojmowanych zbiorach nie jest w stanie rozwiązać wielu

sprzeczności i niejednoznaczności, jakie występują przy przetwarzaniu rzeczywistych danych.

Zaproponowana przez Zadeha w 1965 roku wielowartościowa logika - logika rozmyta (ang. *fuzzy logic*) wraz z opartym na niej systemem wnioskowania okazała się niezwykle przydatna w zastosowaniach inżynierskich i znalazła sobie trwałe miejsce we współczesnych naukach technicznych. Systemy logiki rozmytej charakteryzują się dużą prostotą i elastycznością struktury przy jednoczesnym zachowaniu wysokiej skuteczności. Są one oparte na bazie reguł typu IF-THEN, przy czym tworzenie bazy reguł jest proste i naturalne.

Ze względu na swoją efektywność w zastosowaniach do przetwarzania danych rzeczywistych, wnioskowanie rozmyte wykorzystywane jest w różnego rodzaju systemach eksperckich i decyzyjnych. Logika rozmyta została z powodzeniem zastosowana również w przetwarzaniu sygnałów, w tym do zastosowań opisywanych w niniejszym rozdziale, które opracowano w KID PG.

### 6.1.1. Wstęp do teorii zbiorów rozmytych

#### 6.1.1.1. Zbiory rozmyte

Teoria zbiorów rozmytych jest uogólnieniem klasycznej teorii zbiorów. Klasyczna, Kantorowska teoria zbiorów zakłada, że dowolny element *należy* lub *nie należy* do danego zbioru. Niech  $U = \{x\}$  będzie zbiorem wszystkich elementów ze względu na pewną właściwość, zaś  $\mu(x)$  funkcją określającą przynależność elementu  $x$  do zbioru  $X$ , będącego podzbiorem  $U$ . Wówczas funkcja  $\mu(x):U \Rightarrow [0,1]$  jest zdefiniowana jako:

$$\forall_{x \in U} \mu_x(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases} \quad (6-1)$$

W teorii zbiorów rozmytych element może częściowo należeć do pewnego zbioru, a przynależność tę można wyrazić przy pomocy liczby rzeczywistej z przedziału  $[0,1]$ . Zatem *funkcja przynależności* (ang. *membership function*)  $\mu(x):U \Rightarrow [0,1]$  jest zdefiniowana następująco:

$$\forall_{x \in U} \mu_x(x) = \begin{cases} f(x), & x \in X \\ 0, & x \notin X \end{cases} \quad (6-2)$$

gdzie  $f(x)$  jest funkcją przyjmującą wartości z przedziału  $[0,1]$ . Wartości takiej funkcji nazywane są stopniami przynależności (ang. *membership grades*).

Zbiory z funkcją przynależności określoną na podstawie zależności (6-2) dla zawartych w nich elementów są nazywane *zbiorami rozmytymi* (ang. *fuzzy sets*). Dwa zbiory rozmyte  $X_1, X_2 \subset U$  są równe, gdy spełniona jest równość:

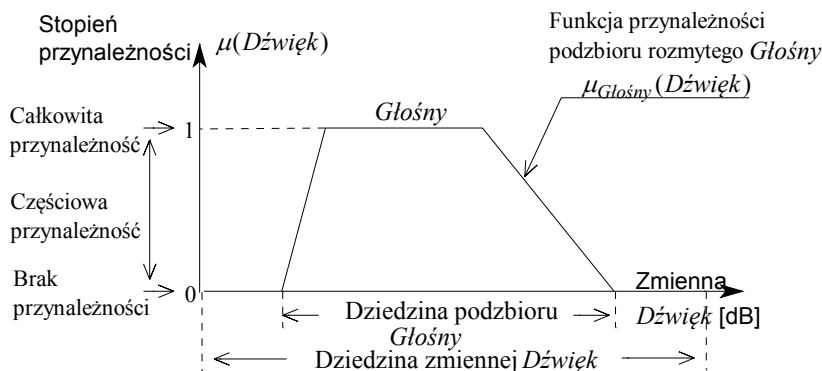
$$\forall_{x \in U} \mu_{X_1}(x) = \mu_{X_2}(x) \tag{6-3}$$

Funkcje przynależności zazwyczaj są przedstawiane w postaci graficznej i mogą mieć dowolną postać, określoną zależnością (6-2). Często stosuje się trapezoidalną funkcję  $\mu(x)$ , której wykres przedstawiono na rys. 6-1. Rysunek ten ilustruje również podstawowe pojęcia związane z teorią zbiorów rozmytych.

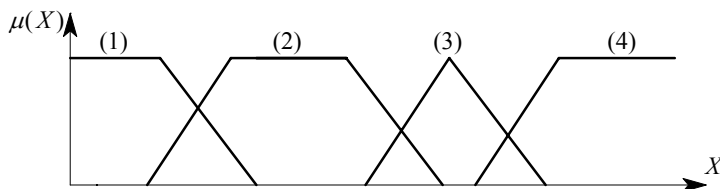
W wielu rozwiązaniach inżynierskich postać funkcji przynależności jest jeszcze bardziej uproszczona. Typowe funkcje przynależności są pokazane na rys. 6-1 i 6-2.

Przykład:

Przyjmijmy, że dane są dwa zbiory: *Cichy* oraz *Głośny*. Elementami zbioru *Cichy* są dźwięki uznawane za ciche, zaś głośne sygnały foniczne należą do zbioru *Głośny*. Niech dźwięk będzie cichy, gdy jego poziom głośności jest niższy od 60 dB, zaś głośny, gdy jego poziom przekracza 50 dB. Funkcje przynależności do zbioru *Cichy*  $\mu_C(x)$  oraz zbioru *Głośny*  $\mu_G(x)$  mogą być zdefiniowane następująco:



Rys. 6-1. Przykład trapezoidalnej funkcji przynależności



**Rys. 6-2.** Kształty funkcji przynależności najczęściej spotykane w praktyce: (1) funkcja typu Z, (2) funkcja typu PI, (3) funkcja typu Lambda, (4) funkcja typu S.

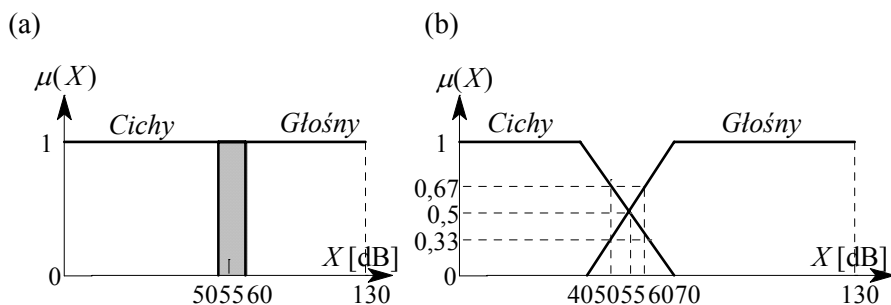
a) z punktu widzenia klasycznej teorii zbiorów (rys. 6-3a):

$$\mu_C(x) = \begin{cases} 1, & x \in [0,60] \text{ dB} \\ 0, & x \in (60,130] \text{ dB} \end{cases} \quad ; \quad \mu_G(x) = \begin{cases} 0, & x \in [0,50] \text{ dB} \\ 1, & x \in (50,130] \text{ dB} \end{cases}$$

b) z punktu widzenia teorii zbiorów rozmytych (rys. 6-3b):

$$\mu_C(x) = \begin{cases} 1, & x \in [0,40] \text{ dB} \\ -\frac{1}{30}x + \frac{7}{3}, & x \in (40,70] \text{ dB} \\ 0, & x \in (70,130] \text{ dB} \end{cases} \quad ;$$

$$\mu_G(x) = \begin{cases} 0, & x \in [0,40] \text{ dB} \\ \frac{1}{30}x - \frac{4}{3}, & x \in (40,70] \text{ dB} \\ 1, & x \in (70,130] \text{ dB} \end{cases}$$



**Rys. 6-3.** Funkcji przynależności dla zbiorów: (a) Kantorowskich, (b) rozmytych

Na gruncie klasycznej logiki dwuwartościowej w takiej sytuacji pojawiają się następujące problemy:

- dźwięk o poziomie np. 55 dB należy jednocześnie do obu zbiorów, czyli jest cichy i jednocześnie głośny. W takim przypadku informacja ta jest sprzeczna, zważywszy zwłaszcza na znaczenie słów *cichy* i *głośny*
- dla wartości 50 dB i 60 dB istnieje ostra i wyraźna granica między oboma zbiorami. Dźwięk o poziomie 59,99 dB należy jeszcze do zbioru *Cichy*, natomiast o poziomie 60,01 dB już do zbioru *Głośny*, mimo że są to dźwięki o niemal tej samej głośności. Staje się zatem oczywiste, że kryterium wyboru punktu granicznego (np. 60 dB) jest bardzo ważne. Warto zwrócić uwagę, że pojęcia *cichy* i *głośny* są subiektywne, a zatem nie można podać jednoznacznie tej wartości granicznej

Zastosowanie zbiorów rozmytych zmienia sposób widzenia problemu. Dźwięk o poziomie 55 dB jest postrzegany *częściowo* jako i cichy, i *częściowo* jako głośny. W obu przypadkach należy on więc *częściowo* do zbiorów *Cichy* oraz *Głośny*, w obu przypadkach z takim samym stopniem przynależności:  $\mu_C(55\text{dB}) = \mu_L(55\text{dB}) = 0,5$ . Z kolei ostre granice zbiorów *Cichy* i *Głośny* zostały „rozmyte”, dzięki czemu nie ma obecnie skokowych zmian wartości funkcji przynależności dla wartości granicznych (50 dB i 60 dB). W przedziale [40 dB, 70 dB] dźwięk jest zarówno cichy jak i głośny, z tą jednak różnicą, że w miarę zwiększania poziomu sygnału staje się on bardziej głośny i jednocześnie mniej cichy. Świadczą o tym stopnie przynależności do zbiorów *Cichy* i *Głośny* dla kolejnych wartości poziomu głośności dźwięku:

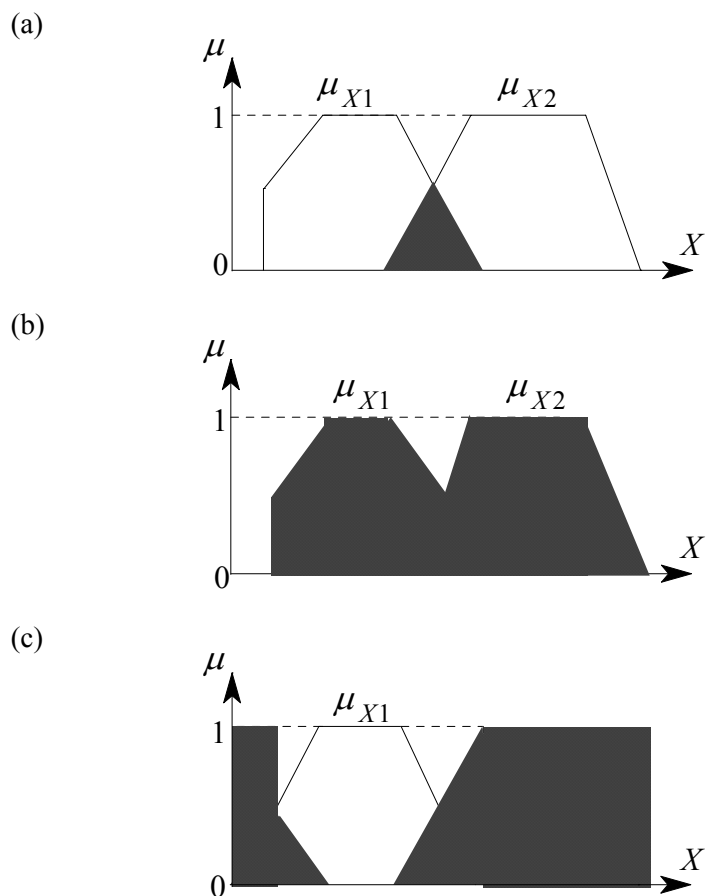
$\mu_C(40\text{dB}) = 1$	$\mu_G(40\text{dB}) = 0$
$\mu_C(50\text{dB}) = 0,67$	$\mu_G(50\text{dB}) = 0,33$
$\mu_C(55\text{dB}) = 0,5$	$\mu_G(55\text{dB}) = 0,5$
$\mu_C(60\text{dB}) = 0,33$	$\mu_G(60\text{dB}) = 0,67$
$\mu_C(70\text{dB}) = 0$	$\mu_G(70\text{dB}) = 1$

#### 6.1.1.2. Operacje na zbiorach rozmytych

Teoria zbiorów rozmytych wprowadza nowe definicje operacji na zbiorach. Niech  $\mu_{X1}(x)$  i  $\mu_{X2}(x)$  będą funkcjami przynależności do zbiorów  $X1$  i  $X2$ . Wówczas operacje na zbiorach rozmytych są zdefiniowane następująco (rys. 6-4):

- Iloczyn logiczny AND:  

$$\mu_{X1 \cap X2}(x) = \min[\mu_{X1}(x), \mu_{X2}(x)]$$



**Rys. 6-4.** Graficzna reprezentacja operacji na zbiorach rozmytych:  
(a) iloczynu, (b) sumy, (c) uzupełnienia

- Suma logiczna OR:  

$$\mu_{X1 \cup X2}(x) = \max[\mu_{X1}(x), \mu_{X2}(x)]$$
- Uzupełnienie logiczne NOT:  

$$\mu_{\overline{X1}}(x) = 1 - \mu_{X1}(x)$$

**6.1.2. Wnioskowanie rozmyte**

*6.1.2.1. Reguły rozmyte*

Praca systemu decyzyjnego opartego na logice rozmytej zależy od definicji reguł rozmytych, które są zawarte w bazie reguł. Reguły te mają postać IF...AND...THEN, np.:

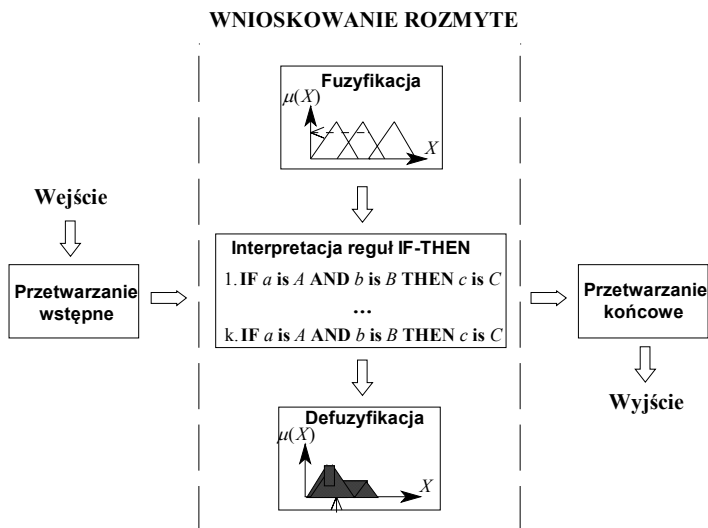
**IF a is A1 AND b is B1 THEN c is C1**

**IF a is A2 AND b is NOT B2 THEN c is C2**

gdzie: a, b, c są zmiennymi lingwistycznymi, natomiast A1, ..., C2 są podzbiórami rozmytymi (por. rys. 6-1). Istotną cechą odróżniającą reguły rozmyte od klasycznych reguł typu IF-THEN jest wykorzystanie zmiennych opisujących zbiory rozmyte, występowanie mechanizmu określającego stopień przynależności elementu do zbioru oraz wykorzystanie operacji na zbiorach rozmytych. Czynniki te powodują określone konsekwencje w procesie wnioskowania.

*6.1.2.2. Schemat wnioskowania rozmytego*

Na rys. 6-5 przedstawiono schemat przetwarzania danych z wykorzystaniem wnioskowania rozmytego. W systemie tym realizowane są następujące operacje:



Rys. 6-5. Schemat rozmytego przetwarzania danych

- a) Przetwarzanie wstępne, przetwarzanie końcowe  
Celem przetwarzania wstępnego (ang. *preprocessing*) jest przekształcenie danych doprowadzonych do wejścia systemu do formatu akceptowanego przez moduł wnioskowania. Analogicznie, przetwarzanie końcowe (ang. *postprocessing*) służy do konwersji danych wyjściowych z tego modułu do postaci zgodnej z wymogami układów zewnętrznych. Sam moduł wnioskowania oczekuje na wejściu ciągu liczb rzeczywistych i zwraca również ciąg liczb rzeczywistych, określanych w nomenklaturze związanej z logiką rozmytą jako *crisp values*.
- b) Fuzyfikacja  
Procedura fuzyfikacji (ang. *fuzzification*), określanej także mianem rozmywania, polega na transformacji wartości z dziedziny liczb rzeczywistych na wartości z dziedziny zbiorów rozmytych. W tym celu dokonuje się wyznaczenia wartości funkcji przynależności dla kolejnych zmiennych lingwistycznych i dla danej rzeczywistej wartości wejściowej. Przykładowe rozmycie (dla wybranych wartości poziomów dźwięku: 40, 50, 55, 60 i 70 dB) zostało pokazane w przykładzie opisanym w podrozdziale 6.1.1.
- c) Interpretacja reguł rozmytych  
Niniejszy etap składa się z dwóch podetapów. W pierwszej kolejności realizowany jest proces obliczania mocy reguł (ang. *rule evaluation*). W tym celu dla każdej zmiennej w przesłankach reguły wyznaczone są stopnie przynależności do odpowiedniego zbioru rozmytego. Ze względu na operację logiczną AND, moc reguły jest obliczana jako wartość minimalna ze wszystkich stopni przynależności (patrz rys. 6-6). Jeśli wartość mocy reguły jest zerowa, wówczas uznaje się, że nie nastąpiła aktywacja reguły. Jednocześnie wyznaczany jest zbiór rozmyty będący rezultatem uaktywnienia reguły. Zależy on od kształtu odpowiedniej funkcji przynależności oraz obliczonej mocy reguły. Zakresowany obszar na rys. 6-6 odpowiada takiemu zbiorowi. W następnym kroku następuje agregacja aktywnych reguł (ang. *rule aggregation*). Polega ona na sumowaniu rozmytych zbiorów wynikowych ze wszystkich reguł. Otrzymany w ten sposób zbiór rozmyty jest zbiorem wynikowym wnioskowania rozmytego.
- d) Defuzyfikacja  
Po zakończeniu procedury agregacji reguł, wynikiem wnioskowania rozmytego jest zbiór rozmyty. Zadaniem defuzyfikacji (ang. *defuzzification*), zwanej również wyostrzaniem, jest zatem przekształcenie odwrotne do rozmywania, czyli transformacja wartości z dziedziny zbiorów rozmytych do dziedziny liczb rzeczywistych. Przekształcenie tego można dokonać na wiele sposobów. Do głównych metod defuzyfikacji należą:



- Metoda największej wartości funkcji przynależności (ang. *Mean of Maximum*). Jako wartość wyjściową wybiera się argument  $x_0$ , będący punktem środkowym takiego przedziału wartości  $x$ , dla którego funkcja przynależności osiąga największą wartość, czyli:

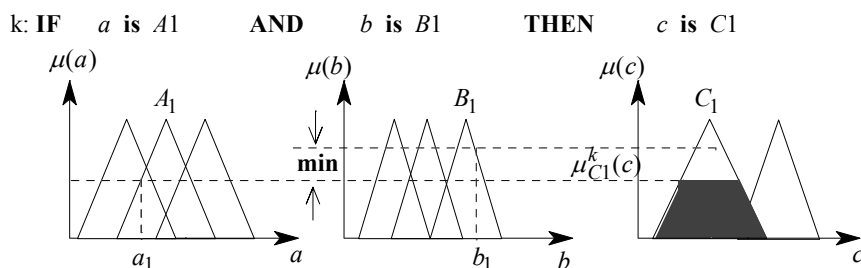
$$\mu(x_0) = \sup_{x \in X} \mu(x) \quad (6-4)$$

Jest to najprostszy i zarazem najmniej złożony obliczeniowo sposób defuzyfikacji. Polega na wyborze tej reguły, której moc była największa. Jego wadą jest brak uwzględnienia wszystkich pozostałych reguł.

- metoda centrowego środka ciężkości (ang. *Center Average*). Wartość wyjściowa  $x_0$  obliczana jest wg zależności (6-5):

$$x_0 = \frac{\sum_{i=1}^N c_i \cdot \mu_i(c_i)}{\sum_{i=1}^N \mu_i(c_i)} \quad (6-5)$$

gdzie:  $c_i$  jest punktem środkowym przedziału wartości  $x$ , dla których  $i$ -ta funkcja przynależności  $\mu_i(x)$  osiąga największą wartość, zaś  $N$  jest liczbą rozmytych zbiorów atrybutów dla zmiennej wyjściowej. Metoda ta jest bardziej złożona obliczeniowo od poprzedniej, bierze się w niej bowiem pod uwagę wszystkie reguły, dla których nastąpiła aktywacja podczas procesu wnioskowania. Z kolei do jej wad można zaliczyć brak uwzględnienia informacji o kształcie funkcji przynależności.



Rys. 6-6. Ilustracja wyznaczania siły aktywacji reguły:  $\mu_{C_1}^k = \min[\mu_{A_1}(a_1), \mu_{B_1}(b_1)]$

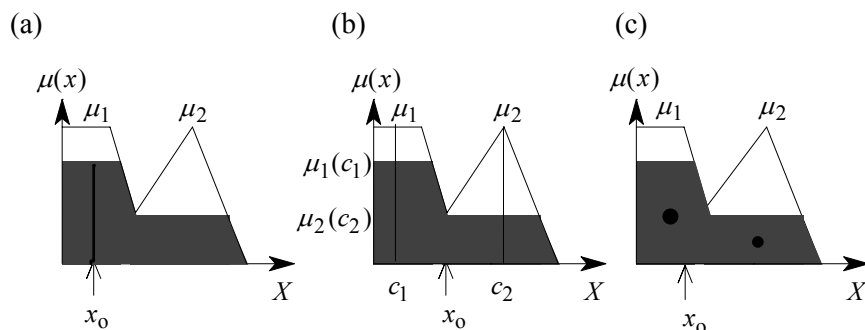
- Metoda wyznaczania środka ciężkości (ang. *Center of Gravity*). Wartość wyjściowa  $x_0$  jest wyznaczana na podstawie następującego wzoru:

$$x_0 = \frac{\int_X \mu(x) \cdot x dx}{\int_X \mu(x) dx} \quad (6-6)$$

Metoda ta jest najbardziej elastyczna, bierze bowiem pod uwagę zarówno wszystkie aktywne reguły, jak również kształty funkcji przynależności. W rezultacie jest najbardziej kosztowna obliczeniowo.

Graficzne interpretacje powyższych metod pokazano na rys. 6-7, na którym obszar zacieniony reprezentuje rozmyty zbiór zmiennej wyjściowej  $X$ .

Przykład:



**Rys. 6-7.** Graficzna interpretacja metod defuzyfikacji: a) metoda największej wartości funkcji przynależności, b) metoda centrowego środka ciężkości, c) metoda środka ciężkości

Niech funkcje przynależności do zbiorów rozmytych  $X_1$  i  $X_2$  zmiennej wyjściowej  $X$  przyjmują niezerowe wartości w sposób następujący:

$$\mu_{X1}(x) = \begin{cases} 1, & x \in [0,5] \\ -\frac{1}{2}(x-7), & x \in (5,10] \\ 0, & \text{dla pozost. } x \end{cases} ; \mu_{X2}(x) = \begin{cases} -\frac{1}{6}(x-4), & x \in [4,10] \\ -\frac{1}{2}(x-12), & x \in (10,12] \\ 0, & \text{dla pozost. } x \end{cases}$$

natomiast maksymalny stopień przynależności do zbioru rozmytego X (obszar zacieniony na rys. 6-8) wynosi 0,8; zaś wartości funkcji  $\mu$  w punktach  $x'$ ,  $x''$  i  $x'''$  będą następujące:  $\mu(x') = 0,8$  oraz  $\mu(x'') = \mu(x''') = 0,4$ .

Rzeczywista wartość wyjściowa  $x_0$  zależy od zastosowanej metody defuzyfikacji (rys. 6-8) i wynosi:

- obliczona metodą największej wartości funkcji przynależności (rys. 6-8a):

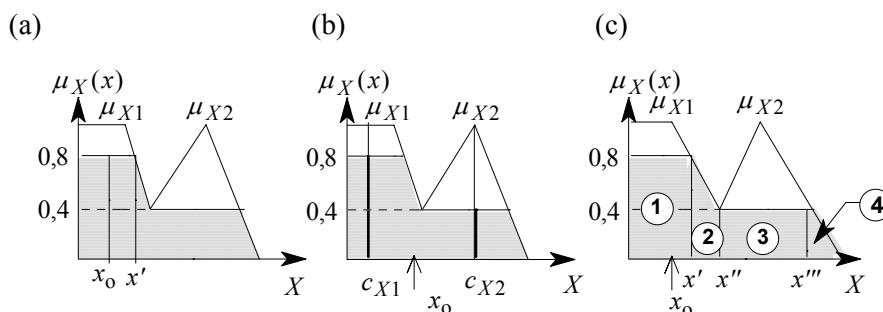
Ponieważ największa wartość funkcji przynależności występuje dla zbioru  $X1$  i wynosi ona 0,8; więc wartość  $x_0$  dana jest wzorem:

$$x_0 = \frac{1}{2}(x' - 0) = \frac{1}{2}x',$$

gdzie  $x'$  można wyznaczyć na podstawie definicji funkcji  $\mu_{X1}(x)$ :

$$\mu(x') = -\frac{1}{2}(x' - 7) \Rightarrow x' = -2 \cdot \mu(x') + 7 = 5,4$$

W rezultacie:  $x_0 = \frac{1}{2} \cdot 5,4 = 2,7$



**Rys. 6-8.** Wyznaczenie wartości wyjściowej przy pomocy: (a) metody największej wartości funkcji przynależności, (b) metody centrowego środka ciężkości, (c) metody środka ciężkości

- obliczona metodą centrowego środka ciężkości (rys. 6-8b):  
Zgodnie z zależnością (6-5), wartość wyjściowa  $x_0$  dana jest wzorem:

$$x_0 = \frac{c_{X1} \cdot \mu_{X1}(c_{X1}) + c_{X2} \cdot \mu_{X2}(c_{X2})}{\mu_{X1}(c_{X1}) + \mu_{X2}(c_{X2})}$$

Wartości centralne zbiorów  $X1$  i  $X2$  dane są wprost z definicji  $\mu_{X1}$  i  $\mu_{X2}$ :

$$c_{X1} = \frac{1}{2} \cdot (5 - 0) = 2,5 \quad \text{oraz} \quad c_{X2} = 10$$

$$\text{Zatem: } x_0 = \frac{2,5 \cdot 0,8 + 10 \cdot 0,4}{0,8 + 0,4} = 5$$

- obliczona metodą środka ciężkości (rys. 6-8c):

Dla niniejszego przypadku wzór (6-6) przyjmuje postać:

$$x_0 = \frac{\int_0^{12} \mu(x) x dx}{\int_0^{12} \mu(x) dx} = \frac{\int_0^{x'} \mu(x) x dx + \int_{x'}^{x''} \mu(x) x dx + \int_{x''}^{x'''} \mu(x) x dx + \int_{x'''}^{12} \mu(x) x dx}{\int_0^{x'} \mu(x) dx + \int_{x'}^{x''} \mu(x) dx + \int_{x''}^{x'''} \mu(x) dx + \int_{x'''}^{12} \mu(x) dx}$$

Korzystając z definicji  $\mu_{X1}$  i  $\mu_{X2}$  można obliczyć  $x'$ ,  $x''$  i  $x'''$ :

$$x' = 5,4 \quad \text{gdyż} \quad \mu(x') = -\frac{1}{2}(x' - 7) \Rightarrow x' = -2 \cdot \mu(x') + 7 = 5,4$$

$$x'' = 6,2 \quad \text{gdyż} \quad \mu(x'') = -\frac{1}{2}(x'' - 7) \Rightarrow x'' = -2 \cdot \mu(x'') + 7 = 6,2$$

$$x''' = 11,2 \quad \text{gdyż} \quad \mu(x''') = -\frac{1}{2}(x''' - 12) \Rightarrow x''' = -2 \cdot \mu(x''') + 12 = 11,2$$

Wartości wyrażeń całkowych dla poszczególnych obszarów wynoszą:

$$\textcircled{1} : \int_0^{5,4} 0,8 \cdot x dx = 11,664$$

$$\int_0^{5,4} 0,8 dx = 4,32$$

$$\textcircled{2} : \int_{5,4}^{6,2} -\frac{1}{2} \cdot (x - 7) \cdot x dx = 2,763$$

$$\int_{5,4}^{6,2} -\frac{1}{2} \cdot (x - 7) dx = 0,48$$

$$\textcircled{3} : \int_{6,2}^{11,2} 0,4 \cdot x dx = 17,4$$

$$\int_{6,2}^{11,2} 0,4 dx = 2$$

$$\textcircled{4} : \int_{11,2}^{12} -\frac{1}{2} \cdot (x-12) \cdot x dx = 1,835 \quad \int_{11,2}^{12} -\frac{1}{2} \cdot (x-12) dx = 0,16$$

Uwzględniając w wyrażeniu na  $x_0$  granice całkowania  $x'$ ,  $x''$  i  $x'''$  oraz wstawiając obliczone powyżej wartości całek oznaczonych, otrzymuje się następującą wartość wyjściową  $x_0$ :

$$x_0 = \frac{\int_0^{12} \mu(x) \cdot x dx}{\int_0^{12} \mu(x) dx} = \frac{11,664 + 2,763 + 17,4 + 1,835}{4,32 + 0,48 + 2 + 0,16} \Rightarrow \underline{\underline{x_0 = 4,837}}$$

### 6.1.2.3. Projektowanie systemów rozmytych

W paragrafie tym przedstawiono etapy projektowania systemu rozmytego. Są one następujące:

1. Określenie zadania oraz określenie sposobu jego realizacji,
2. Określenie zmiennych lingwistycznych i odpowiadających im atrybutów rozmytych,
3. Określenie funkcji przynależności,
4. Określenie bazy reguł rozmytych,
5. Wybór metody defuzyfikacji.

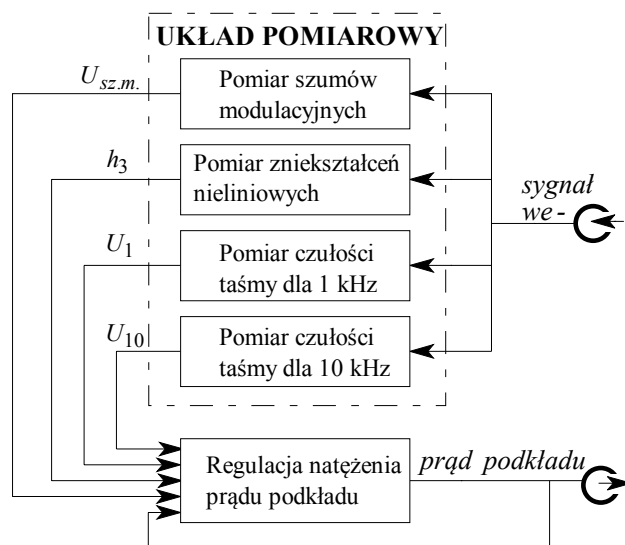
Pomimo prostej i naturalnej struktury systemów rozmytych, nie są znane metody umożliwiające optymalny dobór kształtu funkcji przynależności oraz bazy reguł. Z tej przyczyny bardzo często zadania wymienione powyżej w pkt. 3 oraz 4 są realizowane na drodze eksperymentalnej.

#### Przykład procesu wnioskowania rozmytego

Załóżmy, że naszym celem jest polepszenie dynamiki toru magnetofonowego na drodze pełniejszego wykorzystania możliwości taśmy, co można uzyskać poprzez odpowiednie sterowanie natężeniem prądu podkładu. Natężenie to ma wpływ na poziom zniekształceń nieliniowych, poziom szumów modulacyjnych oraz wartość siły elektromotorycznej indukowanej podczas odczytu w głowicy czytającej. Ten ostatni parametr zależy od częstotliwości zapisywanego sygnału i jest nazywany czułością taśmy.

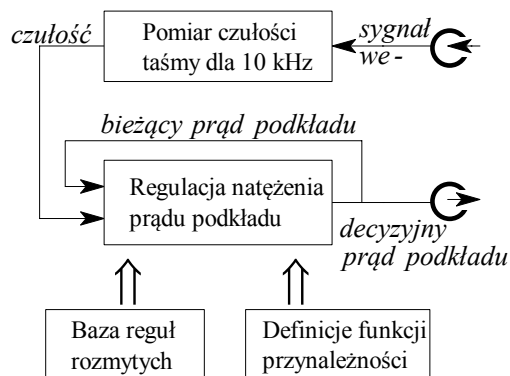
Schemat przykładowego systemu automatycznej kalibracji natężenia prądu podkładu pokazano na rys. 6-9. Sygnał z głowicy odczytu jest

podawany do układu pomiarowego, na którego wyjściu otrzymywane są wartości pomiaru poziomu szumów modulacyjnych ( $U_{sz.m.}$ ), amplitudy trzeciej harmonicznej ( $h_3$ ), której poziom reprezentuje zniekształcenia nieliniowe, oraz parametry czułości taśmy przy częstotliwościach 1 kHz ( $U_1$ ) i 10 kHz ( $U_{10}$ ). Układ regulacji prądu podkładu sterowany jest zarówno tymi sygnałami jak i bieżącą wartością natężenia prądu podkładu.



Rys. 6-9. Schemat układu kalibracji natężenia prądu podkładu

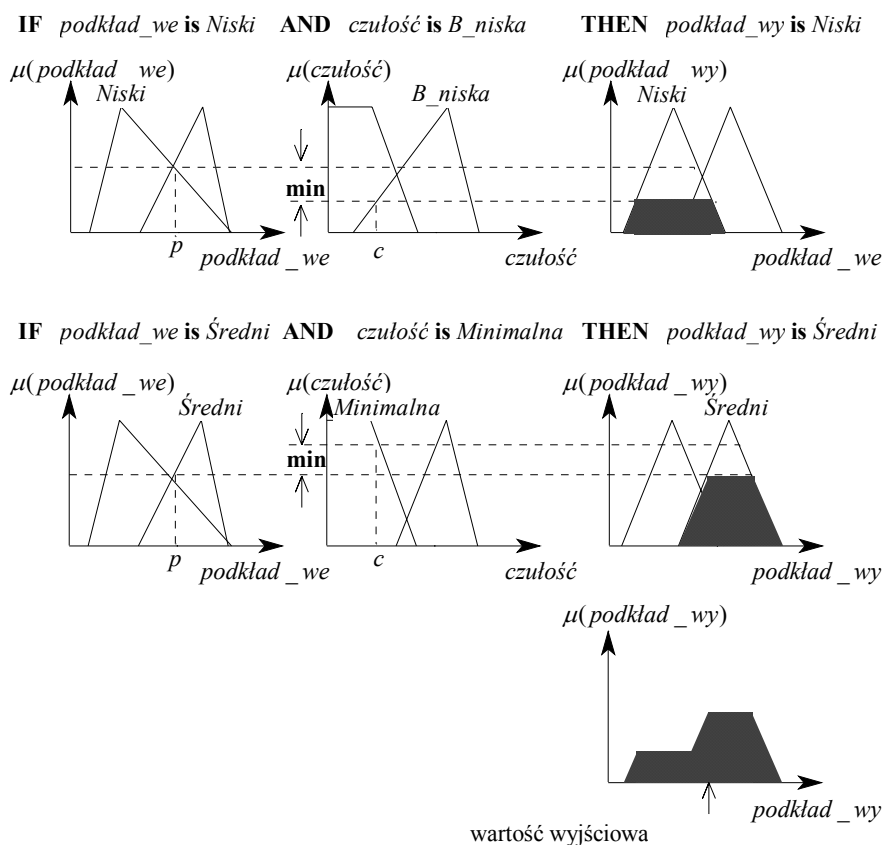
W uproszczonym systemie kalibracji (patrz rys. 6-10) przyjęto, że na wartość wyjściową natężenia prądu podkładu ma wpływ czułość taśmy przy częstotliwości 10 kHz oraz bieżąca wartość tego natężenia.



Rys. 6-10. Schemat uproszczonego systemu regulacji natężenia prądu podkładu

Sygnaly te steruj ukladem regulacji, zatem zostay im przypisane zmienne lingwistyczne: *czuoc* i *podklad\_we*.

Z kolei, dla odronienia wartoci decyzyjnej prdu podkladu od jego wartoci w ptli sprzezenia zwrotnego, sygna wyjciowy skojarzono ze zmienn *podklad\_wy*. Przebieg wnioskowania rozmytego w tym ukladzie zilustrowano na przykadzie pokazanym na rys. 6-11. Opisany system zosta skonstruowany i przebadany w KID PG.



Rys. 6-11. Ilustracja wnioskowania rozmytego

## 6.2 Sieci neuronowe

Dynamicznie rozwijająca się teoria sztucznych sieci neuronowych wywiera coraz większy wpływ na różnorodne, często odległe od siebie dyscypliny wiedzy. Automatyczna klasyfikacja, rozpoznawanie i predykcja sygnału, neuronowa symulacja podstawowych charakterystyk narządu słuchu, to tylko niektóre z zagadnień inżynierii dźwięku, w których metody sztucznej inteligencji, a zwłaszcza sieci neuronowych odgrywają coraz większą rolę. Niektóre spośród zastosowań sieci neuronowych, które były badane przez autora i współpracowników zostaną omówione w par. 6.4 i 6.6.

W niniejszym podrozdziale przedstawiono podstawowy model sztucznego neuronu, skrótowo podstawy teorii wybranych typów sieci neuronowych oraz omówiono najważniejsze struktury i algorytmy treningu tych sieci.

### 6.2.1. Model neuronu

Schemat modelu sztucznego neuronu zamieszczono na rys. 6-12. Struktura neuronu składa się z  $n+1$  ważonych wejść, jednostki przetwarzającej oraz jednego wyjścia. Odpowiednikami biologicznymi tego modelu są: dendryty wraz z kolbkami synaptycznymi, ciało komórki oraz akson. Jedno połączenie synaptyczne jest wyróżnione: wartość na jego wejściu jest stała i wynosi  $-1$ , natomiast wartość jego wagi zwana jest wartością progową.

Wartość sygnału wyjściowego  $o$  jest określona poprzez następującą relację:

$$o = f(\mathbf{w}^T \mathbf{x}) \quad (6-7)$$

gdzie:  $\mathbf{w}$  - wektor wag (ważonych połączeń synaptycznych),

$\mathbf{x}$  - wektor wartości sygnałów wejściowych

Ze uwagi na obecność wagi progowej, wektory  $\mathbf{w}$  i  $\mathbf{x}$  są rozszerzone odpowiednio o elementy  $w_{n+1}$  oraz  $-1$  i są zdefiniowane jako:

$$\mathbf{w} = [w_1, w_2, \dots, w_n, w_{n+1}]^T, \quad \mathbf{x} = [x_1, x_2, \dots, x_n, -1]^T \quad (6-8)$$

Funkcja  $f$  we wzorze (6-7) nazywana jest funkcją aktywacji neuronu. W Tab. 6-1 zostały zamieszczone przykłady różnych postaci funkcji aktywacji neuronu, zaś ich wykresy pokazano na rys. 6-13.

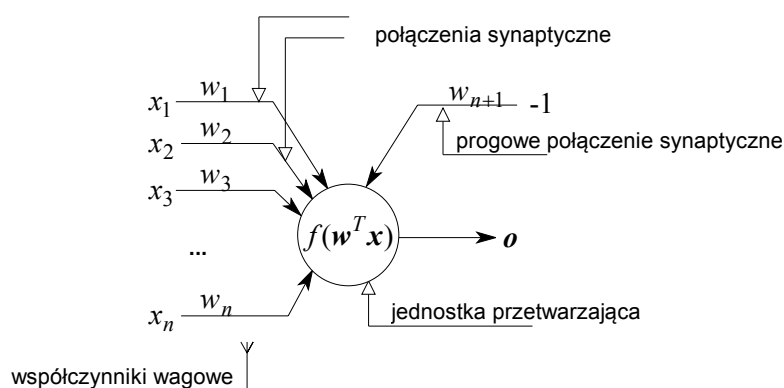


Dziedzina funkcji  $f$  jest reprezentowana przez zbiór wartości wyrażonych następująco:

$$\text{net} = \mathbf{w}^T \mathbf{x} \tag{6-9a}$$

lub bez rozszerzania wektorów  $\mathbf{w}$  i  $\mathbf{x}$  o  $(n+1)$  element następująco:

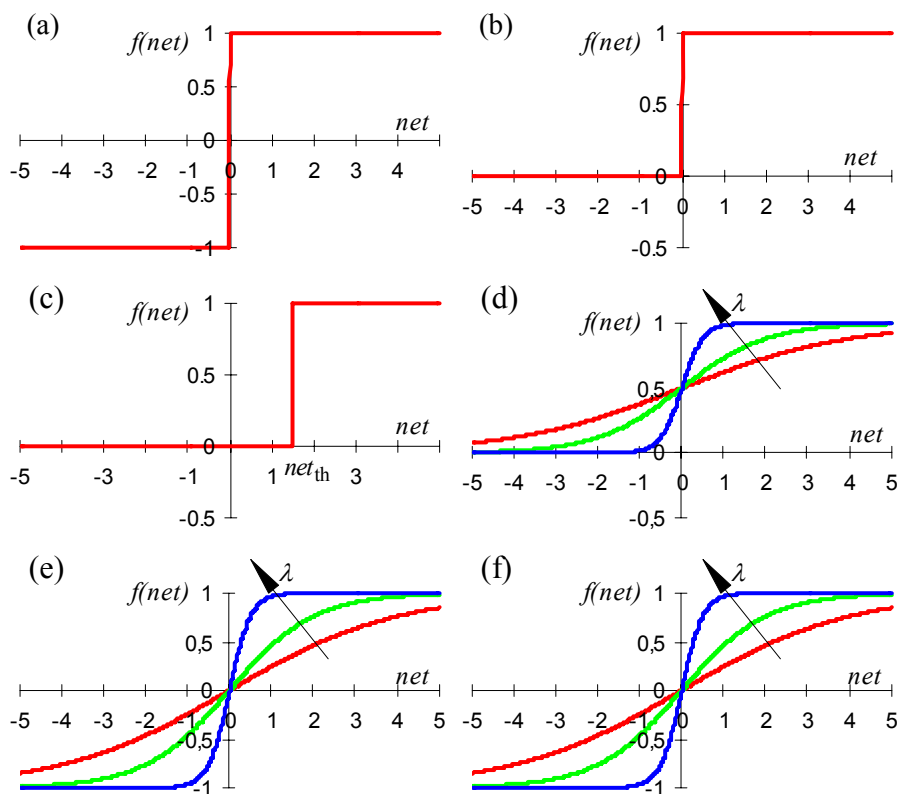
$$\text{net} = \mathbf{w}^T \mathbf{x} - w_{n+1} \tag{6-9b}$$



Rys. 6-12. Model sztucznego neuronu

Tab. 6-1. Przykłady niektórych postaci funkcji aktywacji neuronu. Oznaczenia:  
 $\text{net}$  - parametr wyznaczany przy pomocy zależności (6-9),  
 $\text{net}_{th}$  - wartość progowa,  
 $\lambda$  - współczynnik określający nachylenie funkcji  $f(\text{net})$ ,  
 $k$  - stały współczynnik

Funkcja nieciągła (skokowa)	Funkcja ciągła	
	liniowa	nieliniowa
funkcja progowa: $f(\text{net}) = \begin{cases} 1; & \text{net} > \text{net}_{th} \\ 0; & \text{net} < \text{net}_{th} \end{cases}$	$f(\text{net}) = k \cdot \text{net}$	funkcja unipolarna: $f(\text{net}) = \frac{1}{1 + \exp(-\lambda \cdot \text{net})}$
funkcja Heaviside'a: $f(\text{net}) = \begin{cases} 1; & \text{net} > 0 \\ 0; & \text{net} < 0 \end{cases}$		funkcja bipolarna: $f(\text{net}) = \frac{2}{1 + \exp(-\lambda \cdot \text{net})} - 1$
funkcja signum: $f(\text{net}) = \begin{cases} +1; & \text{net} > 0 \\ -1; & \text{net} < 0 \end{cases}$		funkcja tangens hiperboliczny: $f(\text{net}) = \frac{1 - \exp(-\lambda \cdot \text{net})}{1 + \exp(-\lambda \cdot \text{net})}$



**Rys. 6-13.** Wykresy wybranych funkcji aktywacji neuronu: (a) funkcja signum, (b) funkcja Heaviside'a, (c) funkcja progowa, (d) funkcja ciągła unipolarna, (e) funkcja ciągła bipolarna, (f) tangens hiperboliczny

Opisany tutaj model sztucznego neuronu jest obecnie najbardziej powszechnie stosowany. Jednak wykorzystywane są także inne architektury. Przykładowo, w par. 6.2.4 pokazano zastosowanie powyższego modelu w sieciach komórkowych po dokonaniu jego modyfikacji.

### 6.2.2. Dyskretny dychotomizator

Dyskretny dychotomizator jest przykładem najprostszego klasyfikatora, który dokonuje kwalifikacji wektora wejściowego do jednej z dwóch klas. Jego budowa może być oparta na strukturze neuronowej, w najprostszym przypadku składającej się z jednego neuronu o  $n+1$  wejściach, którego waga  $n+1$  połączenia jest wagą progową oraz jednego wyjścia. Jego zadaniem jest klasyfikacja  $n$ -wymiarowych obiektów. Jeśli wartość na

wyjściu dychotomizatora jest większa od 0, wówczas wektor na wejściu zaliczany jest do jednej klasy, jeśli zaś mniejsza od 0, to do drugiej. Interpretacja geometryczna jest następująca. Współrzędne  $n$ -wymiarowych wektorów w zbiorze uczącym są współrzędnymi punktów w  $n$ -wymiarowej przestrzeni wzorców. Jeśli punkty należące do dwóch różnych klas można odseparować od siebie za pomocą  $(n-1)$ -wymiarowej hiperpłaszczyzny, to wówczas klasy te są separowalne liniowo. Współrzędne wektora wag określają wówczas tę hiperpłaszczyznę i jednocześnie jej stronę dodatnią i ujemną. Jeśli brak wagi progowej, wówczas hiperpłaszczyzna separująca dwie klasy przechodzi przez początek układu współrzędnych.

Przykład:

Poniżej zamieszczono przykład dychotomizatora, którego zbiór uczący składa się z 2-wymiarowych wektorów i jest następujący:

Klasa 1:  $[1, 1]^T, [2, 2]^T, [0, 1]^T$  - oznaczenie: (●),

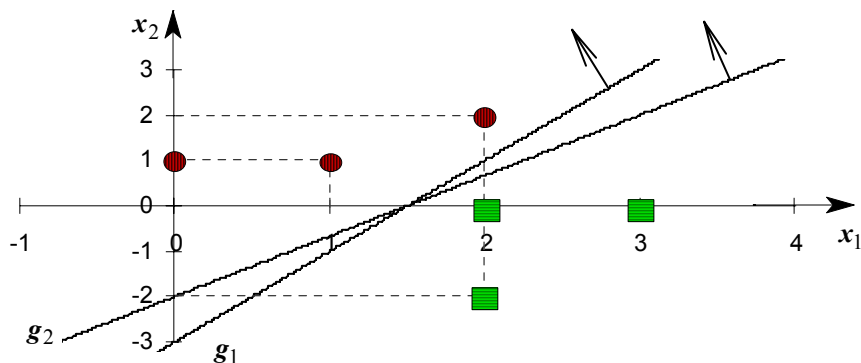
Klasa 2:  $[2, -2]^T, [2, 0]^T, [3, 0]^T$  - oznaczenie: (■)

Punkty określone przez współrzędne wektorów uczących zostały zaznaczone w przestrzeni obrazów jak na rys. 6-14. Jak wynika z rysunku, obie klasy są separowalne liniowo. Co więcej, pomiędzy tymi dwiema klasami można przeprowadzić nieskończenie wiele linii separujących. A zatem, teoretycznie może istnieć nieskończenie wiele wektorów wagowych, które są rozwiązaniem dla konkretnego problemu klasyfikacji. Na rys. 6-14 przedstawiono również przykładowe 2 linie separujące wraz ze strzałkami wskazującymi dodanie półpłaszczyzn:  $g_1(x) > 0$  oraz  $g_2(x) > 0$ . Wartość dodatnia na wyjściu neuronu wskazuje, że wektor wejściowy należy do klasy 1, natomiast wartość ujemna, że należy on do klasy 2.

Przykładowe linie separujące  $g_1$  i  $g_2$  z rys. 6-14 mają następujące równania:

$$g_1: -2x_1 + x_2 + 3 = 0$$

$$g_2: -4x_1 + 3x_2 + 6 = 0$$



Rys. 6-14. Przykład klas separalnych liniowo

Schematy odpowiadających im dychotomizatorów wraz z wartościami wag są przedstawione na rys. 6-15a i na rys. 6-15b.

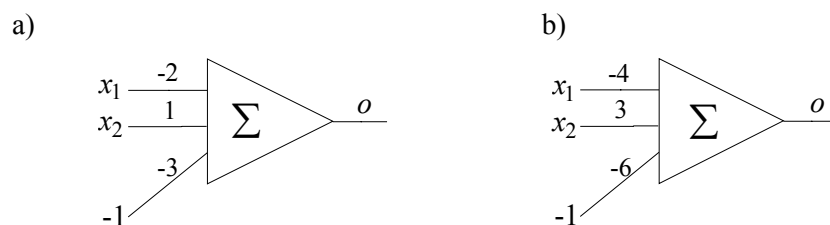
Poniżej przedstawiono krótką weryfikację poprawności działania powyższych klasyfikatorów. Kolumna lewa (i, iii) odpowiada dychotomizatorowi z rys. 6-15a, natomiast prawa (ii, iv), dychotomizatorowi z rys. 6-15b. W przypadku (i) oraz (ii) na wejście klasyfikatora podano wektor  $x = [1, 1]^T$  należący do klasy 1. Dodatnia wartość na wyjściu neuronu wskazuje, że oba dychotomizatory poprawnie sklasyfikowały wektor wejściowy. Z kolei w przypadku (iii) i (iv) wektor wejściowy  $x = [2, 0]^T$  należy do klasy 2. Ujemna liczba na wyjściu dychotomizatora oznacza, że również i tym razem wektory zostały poprawnie sklasyfikowane.

$$\begin{array}{ll} \text{(i)} & \text{(ii)} \\ o = [-2 \ 1 \ -3] \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = 2 > 0 & o = [-4 \ 3 \ -6] \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = 5 > 0 \\ \text{(iii)} & \text{(iv)} \\ o = [-2 \ 1 \ -3] \cdot \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = -1 < 0 & o = [-4 \ 3 \ -6] \cdot \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = -2 < 0 \end{array}$$

### 6.2.3. Typy sieci neuronowych ze względu na metodę treningu

#### 6.2.3.1. Podstawowe cechy procesu treningu

Istotną rolę w pracy systemów opartych na sieciach neuronowych pełni *faza treningu*, zwana również *fazą nauki*. Jest ona charakterystyczną cechą tych sieci, w trakcie której sieć uczy się poprawnie reagować na



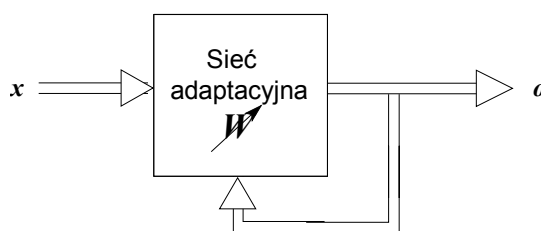
Rys. 6-15. Przykłady dychotomizatorów wraz z wartościami wektora wag

wzorce znajdujące się w zbiorze uczącym. Jednocześnie sieć nabywa zdolności generalizacyjnych, czyli zgodnego z oczekiwaniami konstruktora systemu reagowania na wektory wejściowe, które nie były zawarte w zbiorze uczącym. Można zatem stwierdzić, poprzez analogię do systemów biologicznych, że struktura neuronowa nabiera w procesie treningu doświadczenia, gdyż odpowiedzi sieci zależą w dużym stopniu od zawartości zbioru uczącego. Istnieją jeszcze inne analogie do uczenia się przez organizmy żywe. Proces nauki może być długi i nie zawsze kończy się powodzeniem. Ponadto, stosuje się dwie główne metody nauki: trening nadzorowany, zwany inaczej treningiem „z nauczycielem” oraz nie nadzorowany, określany również jako trening „bez nauczyciela”.

#### 6.2.3.2. Metoda treningu bez nadzoru

W procesie treningu bez nauczyciela na wejście sieci podawany jest wektor  $x$ . Na podstawie dotychczasowego treningu sieć neuronowa sama dokonuje klasyfikacji. W zależności od konkretnej konfiguracji neuronowej wektor wyjściowy  $o$  może reprezentować np. najbardziej typowy obiekt w danej klasie, stopień podobieństwa do uśrednionego wektora w danej klasie, rodzaj klasy, itd. Na rys. 6-16 przedstawiono schemat nie nadzorowanego procesu nauki.

Ponieważ ten rodzaj treningu jest stosowany stosunkowo rzadko, zwłaszcza w problemach związanych z inżynierią dźwięku, w niniejszym opracowaniu ograniczono się jedynie do powyższej krótkiej wzmianki o tego typu metodach nauki. Zainteresowany Czytelnik znajdzie więcej informacji



Rys. 6-16. Schemat procesu nauki bez nadzoru

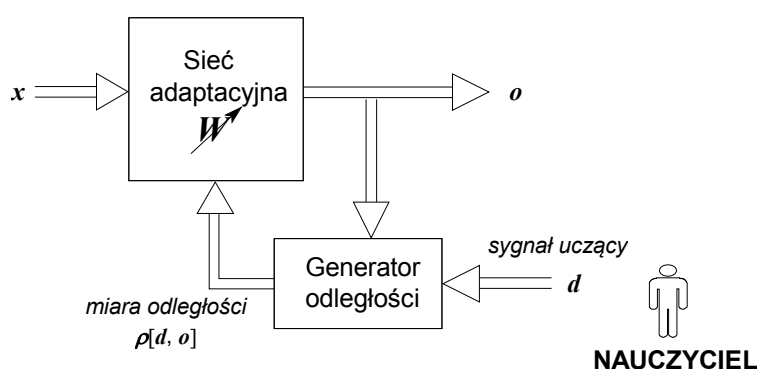
na ten temat w bogatej literaturze przedmiotu.

#### 6.2.3.3. Metoda treningu z nadzorem

W nadzorowanym procesie treningu, na wejście sieci neuronowej podawane są wektory  $x$  ze zbioru uczącego i jednocześnie znane są poprawne wektory  $d$ , jakie powinno otrzymać się na wyjściu sieci. Aktualny

wektor odpowiedzi  $\mathbf{o}$  służy do określenia miary odległości  $\rho$  pomiędzy wektorem wyjściowym  $\mathbf{o}$  a sygnałem uczącym  $\mathbf{d}$ . W zależności od wartości miary  $\rho$  dokonywana jest aktualizacja wag  $\mathbf{W}$  w taki sposób, aby dla danego wektora uczącego  $\mathbf{x}$  miara  $\rho[\mathbf{d}, \mathbf{o}]$  była mniejsza od poprzedniej. Na rys. 6-17 przedstawiono schemat nadzorowanego procesu treningu sieci.

Ten rodzaj treningu jest stosowany najbardziej powszechnie. Więcej informacji na ten temat można znaleźć w par. 6.2.5.



Rys. 6-17. Schemat nadzorowanego procesu treningu

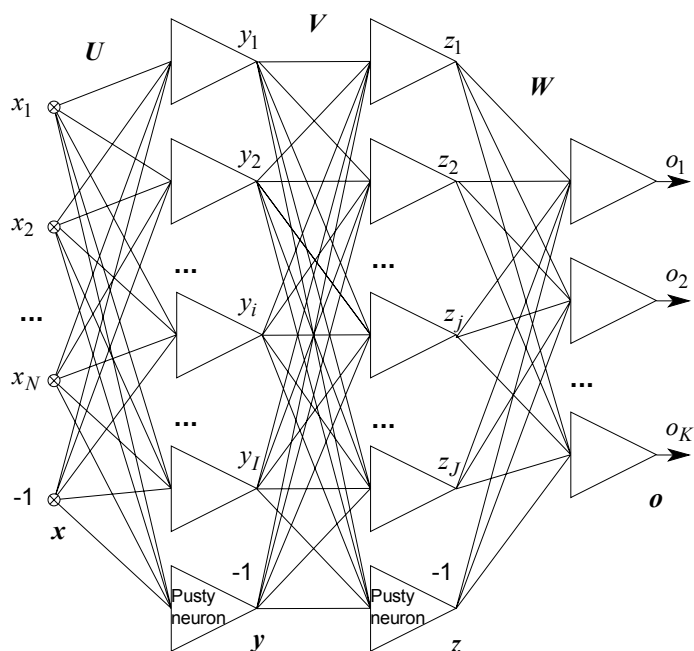
## 6.2.4. Typy sieci neuronowych ze względu na strukturę

### 6.2.4.1. Proste struktury jednokierunkowe

Sieci jednokierunkowe tworzy się w wyniku grupowania pojedynczych neuronów w warstwy, a następnie łączy się warstwy w ten sposób, że wyjście każdego neuronu z danej warstwy jest połączone z wejściem każdego neuronu z warstwy następnej. W sieci jednokierunkowej nie mogą zatem istnieć połączenia wewnątrz warstwy, jak również połączenia wsteczne, czyli połączenia z warstwą poprzednią.

Topologię sieci jednokierunkowej przedstawiono na rys. 6-18. Wektor wejściowy oznaczono jako  $\mathbf{x}$ , wektor wyjściowy z pierwszej i drugiej warstwy ukrytej oznaczono jako  $\mathbf{y}$  i  $\mathbf{z}$ , zaś wektor wyjściowy oznaczono jako  $\mathbf{y}$ . Macierze  $\mathbf{U}$ ,  $\mathbf{V}$ ,  $\mathbf{W}$  zawierają współczynniki wagowe dla poszczególnych połączeń synaptycznych.

Ponieważ sieci tego typu są najczęściej stosowane w praktyce, w podrozdziale 6.2.5 poświęcono więcej miejsca na omówienie tych struktur wraz z algorytmami treningu.



Rys. 6-18. Topologia sieci jednokierunkowej wielowarstwowej (trójwarstwowej)

6.2.4.2. Sieci ze sprzężeniem zwrotnym

Sieci ze sprzężeniem zwrotnym powstają wówczas, gdy wyjście przynajmniej jednego neuronu jest połączone pośrednio lub bezpośrednio z jego wejściem. Przykładem tego typu sieci może być sieć Hopfielda, którą zaprezentowano na rys. 6-19. Ogólnie, sieci ze sprzężeniem zwrotnym nie są powszechnie stosowane przy przetwarzaniu sygnałów fonicznych, dlatego też struktury neuronowe tego typu nie będą omawiane w tej pracy.

6.2.4.3. Sieci komórkowe

Topologia sieci komórkowych oparta jest na dowolnej regularnej strukturze geometrycznej. Rozważmy strukturę będącą płaską siatką prostokątną. Wówczas sieć utworzona na tej strukturze składa się z jednostek neuronowych zgrupowanych w I wierszach i J kolumnach, co pokazano na rys. 6-20.

Dowolna komórka  $c_{ij}$  w i-tym wierszu i w j-tej kolumnie połączona jest bezpośrednio tylko z neuronami leżącymi w promieniu sąsiedztwa,

zaznaczonymi na rys. 6-20 linią przerywaną. Oznaczając przez  $N_r(i,j)$  sąsiedztwo jednostki  $c_{ij}$  o promieniu  $r$ , przynależność dowolnego neuronu  $c_{i+k,j+l}$  do sąsiedztwa komórki  $c_{ij}$  można zapisać jako:

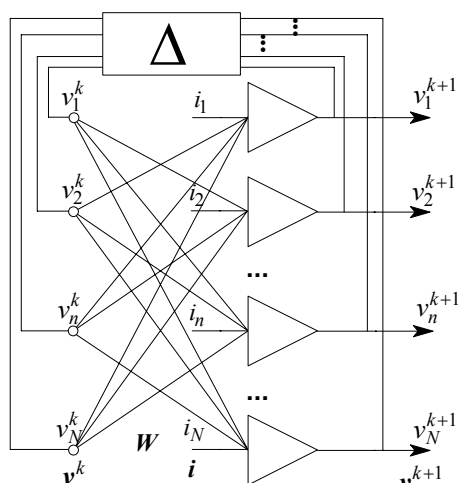
$$c_{i+k,j+l} \in N_r(i,j) \Leftrightarrow |k| \leq r \wedge |l| \leq r \quad (6-10)$$

Dla sieci komórkowej pewnej modyfikacji ulega model neuronu. Wektor wejściowy  $\mathbf{x}$ , jak również wektor wag  $\mathbf{w}$  nie mają już tego samego znaczenia jak we wzorze (6-8). Na wejście komórki  $c_{i,j}$  podawane są:

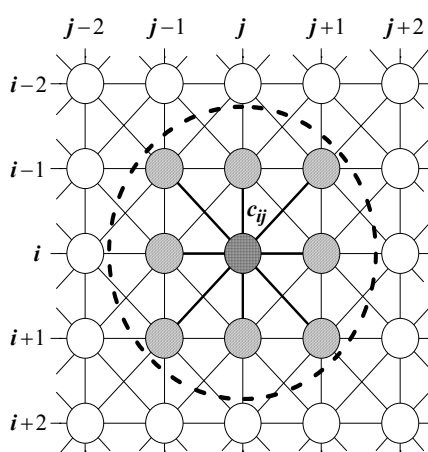
- wektor  $\mathbf{y}$ , którego elementami są sygnały wyjściowe komórek należących do sąsiedztwa  $N_r(i,j)$  oraz sygnał wyjściowy neuronu  $y_{i,j}$ ,
- wektor  $\mathbf{u}$ , którego elementami są sygnały wejściowe komórek należących do sąsiedztwa  $N_r(i,j)$  oraz sygnał wejściowy neuronu  $u_{i,j}$ ,
- stała wartość sygnału polaryzacji  $I$ , będąca odpowiednikiem pobudzenia progowego równego -1 dla standardowego neuronu

Z kolei wartości wag dla komórki są zgrupowane w wektorze *współczynników sprzężenia zwrotnego* oraz w wektorze *współczynników sterowania*, które są oznaczone symbolami  $\mathbf{A}$  i  $\mathbf{B}$ :

$$\mathbf{A} = \begin{bmatrix} A_{-1-1} & A_{-10} & A_{-11} \\ A_{0-1} & A_{00} & A_{01} \\ A_{1-1} & A_{10} & A_{11} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B_{-1-1} & B_{-10} & B_{-11} \\ B_{0-1} & B_{00} & B_{01} \\ B_{1-1} & B_{10} & B_{11} \end{bmatrix} \quad (6-11)$$



**Rys. 6-19.** Sieć Hopfielda - struktura sieci neuronowej ze sprzężeniem zwrotnym



**Rys. 6-20.** Struktura sieci komórkowej o promieniu sąsiedztwa równym 1



Trójka parametrów:  $\mathbf{A}$ ,  $\mathbf{B}$  oraz  $\mathbf{I}$  zwana jest szablonem  $\mathbf{T}$  (ang. *cloning template*). Projektowanie takiej sieci polega na wyznaczeniu szablonu  $\mathbf{T} = \{\mathbf{A}, \mathbf{B}, \mathbf{I}\}$  obowiązującego dla każdej komórki sieci.

Znając model komórki neuronowej, można napisać równanie stanu dla dowolnej jednostki w sieci z czasem dyskretnym:

$$x_{ij}(n+1) = \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} y_{i+k, j+l}(n) + \sum_{k=-r}^r \sum_{l=-r}^r B_{kl} u_{i+k, j+l} + \mathbf{I} \quad (6-12)$$

gdzie:  $x_{ij}(n+1)$  - stan komórki  $c_{ij}$  w chwili  $n+1$ ,  
 $y_{ij}$  - sygnał wyjściowy jednostki  $c_{ij}$ , zdefiniowany jako:

$$y_{ij}(x_{ij}) = 0,5 \cdot \left( |x_{ij} + 1| - |x_{ij} - 1| \right) \quad (6-13)$$

Sieci komórkowe oparte na płaskiej siatce prostokątnej, ze względu na ich topologię, są często stosowane do przetwarzania obrazów dwuwymiarowych.

### 6.2.5. Sieci jednokierunkowe

#### 6.2.5.1. Reguła delta i metoda wstecznej propagacji błędów

Rozważmy sieć neuronową przedstawioną na rys. 6-21. Składa się ona z dwóch warstw: ukrytej, zawierającej  $J$  neuronów oraz wyjściowej składającej się z  $K$  neuronów. Nie wzięto pod uwagę warstwy wejściowej, gdyż założono, że nie wpływa ona w żaden sposób na wartości wektora  $\mathbf{x}$ . Wektory z poszczególnych warstw: ukrytej i wyjściowej zostały oznaczone odpowiednio jako  $\mathbf{y}$  i  $\mathbf{o}$ . Ponadto, warstwa ukryta została rozszerzona o dodatkowy neuron o stałej wartości wyjściowej, reprezentujący pobudzenie progowe o wartości  $-1$  dla każdego neuronu z warstwy następnej (por. rys. 6-12). Analogicznie, wektor wejściowy  $\mathbf{x}$  jest rozszerzony o dodatkowy  $I$ -ty element o wartości  $-1$ . Wektory  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{o}$  mają zatem następującą postać (indeks górny  $T$  oznacza operację transpozycji wektora):

$$\mathbf{x} = [x_1 \quad \dots \quad x_{I-1} \quad -1]^T, \quad \mathbf{y} = [y_1 \quad \dots \quad y_{J-1} \quad -1]^T, \quad (6-14)$$

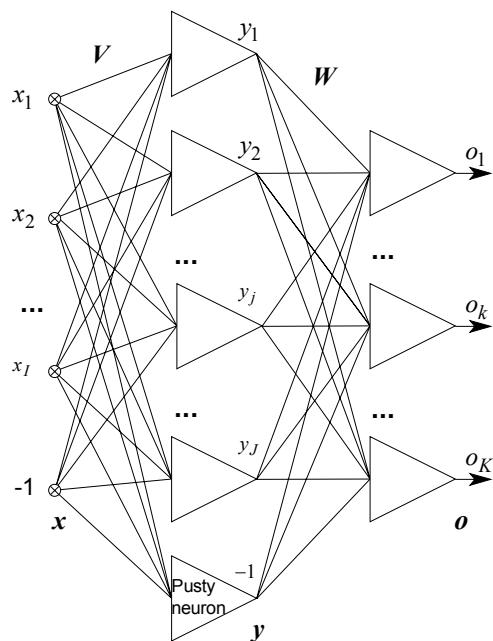
$$\mathbf{o} = [o_1 \quad \dots \quad o_K]^T$$

Przedstawiona sieć jest opisana dwiema macierzami współczynników wagowych:  $\mathbf{V}$  - macierz wag między warstwą wejściową i ukrytą oraz  $\mathbf{W}$  - macierz wag między warstwą ukrytą i wyjściową. Macierze te można zapisać jako:

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1J} \\ v_{21} & v_{22} & \dots & v_{2J} \\ \dots & \dots & \dots & \dots \\ v_{J1} & v_{J2} & \dots & v_{JJ} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1K} \\ w_{21} & w_{22} & \dots & w_{2K} \\ \dots & \dots & \dots & \dots \\ w_{J1} & w_{J2} & \dots & w_{JK} \end{bmatrix} \quad (6-15)$$

Zdefiniujmy dodatkowe dwa wektory pochodnych funkcji aktywacji neuronów dla warstwy ukrytej oraz warstwy wyjściowej:

$$\mathbf{f}'_y = [f'_1(\text{net}_1) \quad f'_2(\text{net}_2) \quad \dots \quad f'_J(\text{net}_J)]^T \quad (6-16a)$$



Rys. 6-21. Sieć jednokierunkowa z pojedynczą warstwą ukrytą

$$\mathbf{f}'_o = [f'_1(\text{net}_1) \quad f'_2(\text{net}_2) \quad \dots \quad f'_K(\text{net}_K)]^T \quad (6-16b)$$

oraz następujące operatory macierzowe:

$$\Phi[\mathbf{q}] = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ 0 & q_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & q_Q \end{bmatrix}, \quad \Gamma[\mathbf{q}] = \begin{bmatrix} f_1(q_1) & 0 & \dots & 0 \\ 0 & f_2(q_2) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & f_Q(q_Q) \end{bmatrix} \quad (6-17)$$

gdzie:  $f_1, f_2, f'_1, f'_2, \dots$  - funkcje aktywacji neuronu wraz z pochodnymi

Ponieważ metoda propagacji wstecznej błędu (ang. *error backpropagation*) należy do nadzorowanych metod treningu, określa się miarę odległości pomiędzy wskazaniem sieci  $\mathbf{o}$  a oczekiwaną odpowiedzią sieci  $\mathbf{d}$  (p. rys. 6-17), czyli tzw. funkcję błędu. Dla przypadku występowania  $P$  wektorów w zbiorze uczącym, funkcja błędu sieci w  $n$ -tym kroku treningu dana jest przez zależność:

$$E^n = \rho(\mathbf{d}^{(1)}, \mathbf{o}^{(1)}, \dots, \mathbf{d}^{(P)}, \mathbf{o}^{(P)}), \quad E^n = \rho(\mathbf{d}^{(1)}, \dots, \mathbf{d}^{(P)}, \mathbf{V}^n, \mathbf{W}^n) \quad (6-18)$$

W praktyce, najczęściej stosuje się funkcję średniokwadratową jako miarę błędu. Zatem wzór (6-18) przybiera postać:

$$E^n = \sum_{p=1}^P \sum_{k=1}^K \frac{1}{2} \cdot (d_k^{(p)} - o_k^{(p)})^2 = \frac{1}{2} \cdot \sum_{p=1}^P \sum_{k=1}^K (d_k^{(p)} - o_k^{(p)})^2 \quad (6-19a)$$

gdzie:  $P$ - liczba wektorów w zbiorze uczącym;  $K$  – liczba wyjść sieci

lub:

$$E^n = \frac{1}{2} \cdot \sum_{p=1}^P \|\mathbf{d}^{(p)} - \mathbf{o}^{(p)}\|^2 \quad (6-19b)$$

Wartość błędu liczona dla wszystkich wektorów w zbiorze uczącym (6-18, 6-19) zwana jest również błędem skumulowanym. Podczas fazy treningu, prezentowane są kolejno wektory uczące i wówczas funkcja błędu dla  $p$ -tego wektora przyjmuje postać:

$$E(p)^n = \rho(\mathbf{d}^{(p)}, \mathbf{o}^{(p)}) = \frac{1}{2} \cdot \|\mathbf{d}^{(p)} - \mathbf{o}^{(p)}\|^2 \quad (6-20)$$

W dalszej części paragrafu przyjęto, że rozważania dotyczą pojedynczego  $p$ -tego wektora ze zbioru uczącego, pominięto zatem stosowne indeksy. Jeśli odpowiedzi kolejnych warstw są następujące:

$$\mathbf{y} = \Gamma[\mathbf{V}\mathbf{x}], \quad \mathbf{o} = \Gamma[\mathbf{W}\mathbf{y}] = \Gamma[\mathbf{W} \cdot \Gamma[\mathbf{V}\mathbf{x}]] \quad (6-21)$$

to wektory sygnału błędu dla błędu określonego wzorem (6-20) są zdefiniowane jako:

$$\delta_{\mathbf{o}} = -\nabla E(\mathbf{o}) \quad \text{lub} \quad \delta_{ok} = -\frac{\partial E}{\partial \text{net}_k}, \quad \text{dla warstwy wyjściowej}, \quad (6-22a)$$

$$\delta_{\mathbf{y}} = -\nabla E(\mathbf{y}) \quad \text{lub} \quad \delta_{yj} = -\frac{\partial E}{\partial \text{net}_j}, \quad \text{dla warstwy ukrytej} \quad (6-22b)$$

W następstwie wykonania obliczeń, uzyskuje się następujące rozwiązania:

$$\delta_{\mathbf{o}} = \Phi[\mathbf{d} - \mathbf{o}] \cdot \mathbf{f}'_{\mathbf{o}} \quad \text{lub} \quad \delta_{ok} = (d_k - o_k) \cdot f'(\text{net}_k), \quad (6-23a)$$

$$\delta_{\mathbf{y}} = \mathbf{w}_j^T \cdot \delta_{\mathbf{o}} \cdot \mathbf{f}'_{\mathbf{y}} \quad \text{lub} \quad \delta_{yj} = f'_j(\text{net}_j) \cdot \sum_{k=1}^K (d_k - o_k) \cdot f'_k(\text{net}_k) \cdot w_{kj} \quad (6-23b)$$

Jak wynika ze wzorów (6-23), obecność pochodnej nakłada ograniczenia na postać funkcji aktywacji neuronu. Funkcja ta musi być ciągła i różniczkowalna. W Tab. 6-1 przedstawiono najczęściej stosowane funkcje, a ich poglądowe wykresy pokazano na rys. 6-13 d-f.

Zgodnie z tzw. regułą delta (ang. *delta learnig rule*), aktualizacja macierzy wag  $\mathbf{V}^n$  i  $\mathbf{W}^n$  w kroku  $(n+1)$  następuje według zależności:

$$\begin{cases} \Delta \mathbf{V}^{n+1} = -\eta \cdot \nabla E(\mathbf{V}^n) \\ \Delta \mathbf{W}^{n+1} = -\eta \cdot \nabla E(\mathbf{W}^n) \end{cases} \quad (6-24)$$

gdzie:  $\eta$  - współczynnik szybkości treningu, przy czym najczęściej  $\eta \in (0,1]$

Po uwzględnieniu wzorów (6-22 i 6-23), nowe wartości macierzy wag  $\mathbf{V}^{n+1}$  i  $\mathbf{W}^{n+1}$  obliczane są następująco:

$$\begin{cases} \mathbf{V}^{n+1} = \mathbf{V}^n + \eta \delta_y \mathbf{x}^T \\ \mathbf{W}^{n+1} = \mathbf{W}^n + \eta \delta_o \mathbf{y}^T \end{cases} \quad (6-25)$$

Jak już wspomniano, metoda propagacji wstecznej błędu oparta jest na regule delta, zgodnie z którą następuje minimalizacja funkcji błędu. Jednakże podobnie jak i inne metody optymalizacji gradientowej, proces nauki może zatrzymać się w minimum lokalnym, nie osiągając żadanego minimum globalnego. Może się również zdarzyć, że przyrosty wartości macierzy wag (6-24) będą bardzo małe, co jest związane np. z tym, że proces minimalizacyjny funkcji kosztu utknie na płaskim obszarze hiperprzestrzeni błędu lub wartości funkcji aktywacji dla dużych wartości *net* osiągną obszar nasycenia (rys. 6-13 d-f). Dlatego metoda propagacji wstecznej błędu nie gwarantuje zbieżności procesu nauki. Aby polepszyć zbieżność wykorzystuje się różne metody opisane w literaturze, w praktyce zaś dość często stosuje się zabieg polegający na dodaniu tzw. członu momentu MT:

$$\text{MT}^{n+1} = \alpha \cdot \Delta \mathbf{w}^n \quad (6-26)$$

gdzie:  $\Delta \mathbf{w}^n$  - przyrost wektora wag w ostatnim, n-tym, kroku,  
 $\alpha$  - współczynnik momentu,  $\alpha \in (0, 1]$ .

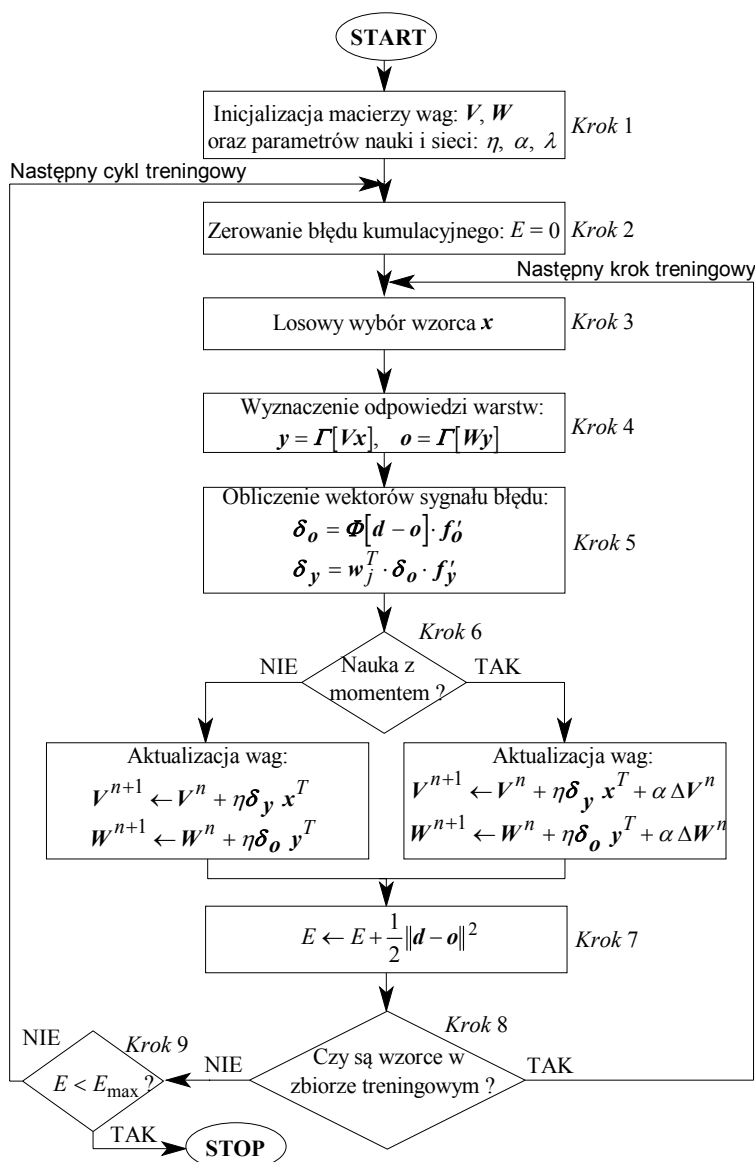
Wzory (6-25) dla wartości macierzy  $\mathbf{V}^{n+1}$  i  $\mathbf{W}^{n+1}$  przybierają zatem postać:

$$\begin{cases} \mathbf{V}^{n+1} = \mathbf{V}^n + \eta \delta_y \mathbf{x}^T + \alpha \cdot \Delta \mathbf{V}^n \\ \mathbf{W}^{n+1} = \mathbf{W}^n + \eta \delta_o \mathbf{y}^T + \alpha \cdot \Delta \mathbf{W}^n \end{cases} \quad (6-27)$$

Warto dodać, że nieznane są optymalne wartości parametrów nauki:  $\eta$  i  $\alpha$ . Dla każdego zestawu danych w zbiorze uczącym są one inne, a co więcej, raz dobrane na początku treningu nie muszą być jednakowo przydatne podczas całego procesu nauki. Rozwiązaniem jest uzmiennienie tych parametrów i wprowadzenie możliwości zmian ich wartości w zależności od postępów treningu.

6.2.5.2. *Algorytm wstecznej propagacji błędu*

Na rys. 6-22 pokazano diagram algorytmu propagacji wstecznej błędu EBPTA (ang. *Error Backpropagation Training Algorithm*), a następnie przedstawiono opis procedury treningu w oparciu o zależności z poprzedniego paragrafu.



Rys. 6-22. Schemat algorytmu wstecznej propagacji błędu

W poszczególnych krokach algorytmu zachodzą następujące procesy:

Krok 1:

Wartości macierzy wag  $\mathbf{V}$  i  $\mathbf{W}$  są inicjalizowane małymi losowymi wartościami z zakresu  $(-1, 1)$ . Jednocześnie ustawiane są parametry sieci (współczynnik  $\lambda$ ) oraz nauki:  $\eta$  i/lub  $\alpha$ . Najczęściej parametr  $\lambda$  jest przyjmowany, jako równy 1.

Krok 2:

Wartość błędu skumulowanego  $E$  jest ustawiana na 0. Celem treningu jest znalezienie takiego zestawu wartości wag, aby błąd skumulowany  $E$  dla wszystkich wektorów ze zbioru uczącego nie był większy od arbitralnie zadanej wartości  $E_{\max}$ , tj.:  $E \leq E_{\max}$ . Dlatego też, dla każdego nowego cyklu treningowego, wartość  $E$  jest zerowana.

Krok 3:

Wybór dowolnego wektora ze zbioru uczącego. Jest zalecane, aby wybór wektora  $\mathbf{x}$  następował w sposób losowy. Jednocześnie, ustawiana jest wartość wektora  $\mathbf{d}$ , reprezentującego oczekiwaną odpowiedź sieci na pobudzenie wektorem  $\mathbf{x}$ .

Krok 4:

Wyznaczane są odpowiedzi kolejnych warstw:  $\mathbf{y}$ ,  $\mathbf{o}$  dla pobudzenia sieci wektorem  $\mathbf{x}$  na podstawie wzoru (6-21)

Krok 5:

Zgodnie ze wzorem (6-23) obliczane są wektory sygnału błędu dla poszczególnych warstw:  $\delta_y$  i  $\delta_o$

Krok 6:

Sprawdzany jest warunek, czy trening przeprowadzany jest z uwzględnieniem momentu, czy nie. Jeśli tak, to nowe wartości macierzy wag  $\mathbf{V}^{n+1}$  i  $\mathbf{W}^{n+1}$  obliczane są według wzoru (6-27), jeśli zaś nie – to na podstawie wzoru (6-25)

Krok 7:

Obliczana jest wartość funkcji błędu sieci dla wektora  $\mathbf{x}$ . Następnie, wartość ta jest dodawana do błędu skumulowanego  $E$

Krok 8:

Jeśli wektor  $\mathbf{x}$  nie jest ostatnim wykorzystanym wzorcem w zbiorze uczącym w danym cyklu treningowym, to sterowanie algorytmem wraca do kroku 3, czyli wybierany jest następny wektor uczący z pominięciem

wektora  $x$ . Jeśli z kolei wektor  $x$  jest ostatnim wektorem uczącym w danym cyklu, to sterowanie przechodzi do następnego kroku

Krok 9:

Sprawdzany jest warunek, czy wartość błędu skumulowanego  $E$  nie jest większa od zadanej wartości progowej  $E_{\max}$ . Jeśli  $E \leq E_{\max}$ , to wówczas warunki zatrzymania procesu nauki są spełnione i algorytm zatrzymuje się. Jeśli jednak wartość błędu skumulowanego przewyższa  $E_{\max}$ , to następuje kolejny cykl treningowy. Sterowanie algorytmem wraca do *kroku 2* - wyzerowania wartości błędu skumulowanego  $E$ .

Przedstawiony algorytm jest oparty bezpośrednio na metodzie wstecznej propagacji błędu i gradientowej procedurze minimalizacji błędu. W literaturze jednak znane są liczne modyfikacje tego algorytmu, zmniejszające zarówno czas treningu, jak również eliminujące lub przynajmniej redukujące część wad tego podstawowego algorytmu. Pewną modyfikację tego algorytmu zaproponowano również w związku z eksperymentami dotyczącymi przetwarzania sygnałów fonicznych z wykorzystaniem jednokierunkowych sieci neuronowych (par. 6.4.1.2).

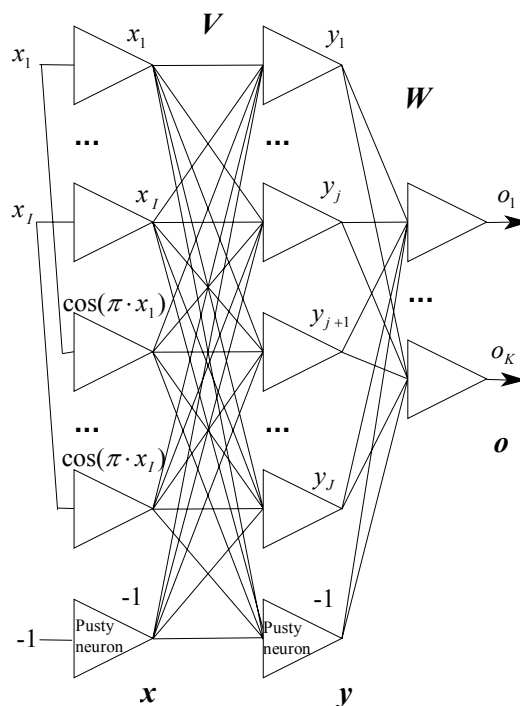
### 6.2.5.3. Sieci z połączeniami funkcjonalnymi

W pewnych przypadkach, przetwarzanie informacji przez sieć może być znacznie utrudnione, zwłaszcza gdy informacja wejściowa jest niewystarczająca do odtworzenia informacji wyjściowej. Może się tak zdarzyć, gdy rozmiar wektora wejściowego  $I$  jest dużo mniejszy niż rozmiar wektora wyjściowego  $K$ , tzn. gdy  $I \ll K$ . Wówczas zadanie dla sieci neuronowej staje się źle uwarunkowane, a zdolności generalizacyjne takiej sieci są zazwyczaj niewielkie. Ponadto, nawet gdy zadanie jest dobrze uwarunkowane, to ze względu na dużą złożoność sieci, taka sytuacja może prowadzić do niepoprawnego uogólniania (generalizacji), a także sam proces treningu może się znacznie wydłużyć.

Pewnym rozwiązaniem wspomnianych problemów może być rozszerzenie informacji wejściowej o dodatkowe połączenia funkcyjne (rys. 6-23). Sieć taka nosi nazwę sieci wyższego rzędu lub sieci z rozszerzeniami funkcyjnymi (ang. *functional link net*) i została zaproponowana przez Pao w 1989 r. Wspomnianymi rozszerzeniami funkcjonalnymi mogą być:

- funkcje wielomianowe,
- funkcje ortogonalne, np.:  $\sin(\pi \cdot x_i)$ ,  $\cos(\pi \cdot x_i)$ ,  $\sin(k \cdot \pi \cdot x_i)$ ,  $\cos(k \cdot \pi \cdot x_i)$ , gdzie  $k$  jest dowolną liczbą całkowitą,
- funkcje iloczynu wartości wejściowych, np.:  $\dots \cdot x_1 \cdot x_j \cdot \dots$ .





Rys. 6-23. Przykład sieci z kosinusowymi rozszerzeniami funkcjonalnymi

#### 6.2.5.4. Algorytm redukcji wag

Dla wszystkich sieci neuronowych wciąż nie rozwiązany problemem pozostaje dobór odpowiednich rozmiarów struktury neuronowej dla konkretnego zadania. Jeśli rozmiary sieci są za małe w stosunku do ilości danych uczących, wówczas pojemność takiej sieci zostaje wyczerpana (przepełniona), na skutek czego sieć nie nadaje się do przetwarzania danych. Z kolei kiedy struktura sieci jest zbyt duża, sieć przejawia tendencję do zapamiętywania danych (a nie ich generalizacji), co w rezultacie prowadzi do znacznego osłabienia zdolności podejmowania przez sieć trafnych decyzji. Dlatego istnieje potrzeba opracowania skutecznej metody pozwalającej na znajdowanie optymalnych lub suboptymalnych struktur sieci.

Sposoby rozwiązywania tego problemu można w zasadzie podzielić na dwie kategorie: powiększające oraz redukujące rozmiary sieci. Metody z pierwszej grupy dotyczą struktur neuronowych wyraźnie za małych. Rozmiary sieci są stale powiększane, aż osiągnie ona wystarczającą

pojemność. Z kolei algorytmy z drugiej wymienionej grupy dotyczą sieci o zbyt dużych rozmiarach. W trakcie działania algorytmu, redukcji ulegają wagi oraz niekiedy także neurony, jeśli okazuje się, że ich brak nie wpływa na jakość przetwarzania danych przez sieć neuronową. W dalszej części niniejszego paragrafu omówione zostanie jedynie ostatnie wymienione podejście ze względu na powszechność jego zastosowań.

Ogólnie, algorytmy redukcji wag (ang. *weight pruning algorithms*) ze względu na sposób działania można podzielić na:

- a) grupę metod, w której szacowana jest wrażliwość funkcji błędu na zmianę pojedynczych wag. Wagi związane z najmniejszymi zaobserwowanymi wrażliwościami są usuwane i proces treningu kontynuowany jest na zredukowanej strukturze.
- b) grupę metod, w której wprowadzona zostaje określona funkcja (tzw. funkcja kary) w miejsce nieefektywnej, czyli nadmiarowej struktury.

W obu przypadkach, w wyniku działania algorytmu redukcji wag może się zdarzyć, że usunięty zostanie cały neuron. Metody pochodzące z pierwszej grupy (*Optimal Brain Damage* - OBD, *Optimal Brain Surgeon* - OBS) są bardzo skuteczne, ale ich zasadniczą wadą jest znaczny koszt obliczeniowy. Są zatem nieefektywne z punktu widzenia szybkości przetwarzania danych. Z kolei, algorytmy korzystające z funkcji kary są proste i stosunkowo skuteczne w stosunku do ich złożoności obliczeniowej. Co więcej, są one również stosowane w metodach wydobywania reguł (ang. *rule discovery*), stosowanych do celu uzyskania szkieletu struktury neuronowej, co z kolei prowadzi do wyodrębnienia reguł, na których niejawnie oparte jest działanie sieci.

Biorąc pod uwagę zarówno efektywność, jak i złożoność obliczeniową poszczególnych algorytmów, przedstawiono poniżej podejście do redukcji wag z wykorzystaniem funkcji kary. W omawianym przykładzie modyfikacji ulega funkcja błędu i przyjmuje ona dla wagi  $w_{ij}$  nową postać:

$$E'(\mathbf{W}) = E(\mathbf{W}) + \frac{1}{2} \cdot \gamma \cdot \sum_{i,j} \frac{w_{ij}^2}{1 + w_{ij}^2} \quad (6-28)$$

gdzie  $\gamma$  jest dodatnią stałą. W rezultacie modyfikacja wartości wagi  $w_{ij}$  dla standardowej propagacji wstecznej błędu zachodzi na podstawie wzoru:

$$w'_{ij} = w_{ij} \cdot \left( 1 - \eta \gamma \cdot \frac{1}{[1 + w_{ij}^2]^2} \right) \quad (6-29)$$

## **6.3 Zbiory przybliżone**

Teoria zbiorów przybliżonych jest obecnie bardzo dobrze rozwinięta. Stąd w niniejszej pracy możliwe jest jedynie omówienie związanych z nią zagadnień podstawowych.

Przełamanie ograniczeń związanych z określeniem zbioru Kantorowskiego jest także możliwe poprzez odrzucenie wymogu istnienia ściśle określonych granic zbioru, który jest dokładnie zdefiniowany poprzez należące do niego elementy. Powstaje zatem możliwość zdefiniowania zbioru w oparciu o jego przybliżenie dolne i górne. Taki zbiór, skoro nie jest określony w pełni dosłownie, może zawierać elementy, które należą do niego wielokrotnie. Podobnie, jak w przypadku zbiorów rozmytych, przełamanie tradycyjnych aksjomatów zastosowane w przypadku zbiorów przybliżonych powoduje, że logika oparta na teorii zbiorów przybliżonych zyskuje zupełnie nowe właściwości, które czynią ją niezwykle przydatną do rozwiązywania wielu problemów wymagających inteligentnej analizy danych, poszukiwania ukrytych zależności pomiędzy danymi a nawet podejmowania trafnych decyzji w sytuacji istnienia niepełnych lub częściowo sprzecznych przesłanek. Autorem koncepcji zbiorów przybliżonych jest polski profesor Zdzisław Pawlak, który ogłosił podstawy tej teorii na początku lat osiemdziesiątych. Teoria ta szeroko upowszechniła się w świecie naukowym i stanowi obecnie jedną z najszybciej rozwijających się metod z dziedziny inteligentnych systemów decyzyjnych.

Wiedza w systemie opartym na zbiorach przybliżonych jest reprezentowana w postaci tablicy decyzyjnej. Na podstawie tablicy decyzyjnej uzyskuje się zestaw reguł o postaci IF-THEN, które z kolei mogą być regułami pewnymi albo niepewnymi. Decyzja podjęta przez system ekspercki, oparty na zbiorach przybliżonych, pochodzi zawsze od jednej, najbardziej pewnej reguły, którą można znaleźć dla bieżącego zestawu atrybutów.

W toku prowadzonych wraz ze współpracownikami prac badawczo-eksperymentalnych autor opracował kilka systemów eksperckich, których rdzeniem jest system decyzyjny oparty na metodzie zbiorów przybliżonych. Jeden z opracowanych systemów tego typu zostanie omówiony w par. 6.5.

### **6.3.1. Elementy teorii zbiorów przybliżonych**

#### *6.3.1.1. Tablica decyzyjna, koncept*

W systemach opartych na zbiorach przybliżonych istotną rolę odgrywa sposób reprezentacji wiedzy. Dostępna informacja przechowywana

jest w tablicy decyzyjnej w taki sposób, że: kolumny reprezentują *atrybuty*, wiersze *obiekty*, zaś elementami tablicy są *wartości atrybutów* dla poszczególnych obiektów i atrybutów. Taka tablica nazywana jest *tablicą decyzyjną* lub *systemem informacyjnym*.

Formalnie, tablica decyzyjna  $S$  to czwórka (ang. *quadruple*):

$$S = (U, Q, V, f) \quad (6-30)$$

gdzie:  $U$  - niepusty, skończony zbiór, którego elementy są *obiektami* lub *przykładami z tablicy  $S$* ,

$Q = C \cup D$  - zbiór, dla którego spełniony jest warunek  $C \cap D = \emptyset$ ,

$C$  - niepusty, skończony zbiór atrybutów dla  $S$ ,

$D$  - niepusty, skończony zbiór decyzji dla  $S$ ,

$V = \bigcup_{q \in Q} V_q$  - niepusty, skończony zbiór wartości atrybutów,

przy czym  $V_q$  jest zbiorem wartości atrybutu  $q$ ,

$f$  - *funkcja informacyjna* przypisująca wartość każdemu atrybutowi dla każdego obiektu z  $S$ , tj.:

$$f: U \times Q \rightarrow V, \forall x \in U \forall q \in Q f(x, q) \in V_q \quad (6-31)$$

Z tablicą decyzyjną związane jest ważne pojęcie *konceptu*. Jest to zbiór tych obiektów  $x \in U$ , dla których funkcja informacyjna  $f$  przyjmuje określoną wartość  $v_q \in V_q$  dla danego atrybutu (atrybutów): warunkowego lub decyzyjnego  $q \in Q$ , co można zapisać jako warunek:

$$X(q = v_q) = \{x \in U: f(x, q) = v_q\} \quad (6-32)$$

Przykład:

Niech przykładowa tablica decyzyjna ma postać jak Tab. 6-2.

Wówczas:

a) zbiór obiektów  $U = \{1, 2, 3, 4, 5, 6\}$

b) zbiór atrybutów  $C = \{c_1, c_2, c_3, c_4\}$

c) zbiór decyzji  $D = \{d\}$

d) poszczególne zbiory wartości kolejnych atrybutów mają postać:

$V_{c_1} = \{A, B\}$     $V_{c_2} = \{Yes, No\}$     $V_{c_3} = \{10, 50\}$     $V_{c_4} = \{-5, 7\}$

$V_d = \{Low, High\}$ ,

natomiast zbiór wartości atrybutów ma postać:

$$V = \{V_{c1}, V_{c2}, V_{c3}, V_{c4}, V_d\} = \{\{A, B\}, \{Yes, No\}, \{10, 50\}, \{-5, 7\}, \{Low, High\}\}$$

e) koncept  $X(d = Low) = \{1, 6\}$

Tablica decyzyjna o postaci jak w przypadku Tab. 6-2 umożliwia tworzenie reguł IF-THEN, np. obiekt nr 3 może być opisany następującą regułą:

$$\text{IF } c_1 = A \text{ AND } c_2 = Yes \text{ AND } c_3 = 10 \text{ AND } c_4 = 7 \text{ THEN } d = High$$

### 6.3.1.2. Relacja nierozróżnialności

Dla dowolnego zbioru atrybutów  $P \subseteq Q$  relacja nierozróżnialności (ang. *indiscernibility relation*) jest zdefiniowana następująco:

$$IND(P) = \{(x, y) \in U \times U : \forall_{a \in P} f(x, a) = f(y, a)\} \quad (6-33)$$

Jeśli  $(x, y) \in IND(P)$ , to obiekty  $x$  i  $y$  są *nierozróżnialne ze względu na P*.

Relacja nierozróżnialności jest relacją równoważności nad  $U$ , a więc dzieli  $U$  na klasy równoważności - zbiory obiektów nierozróżnialnych ze względu na zbiór  $P$ . Taki podział oznaczany jest symbolem  $U/IND(P)$  i odpowiada podzielonej populacji generalnej (ang. *Partitioned Universe*).

#### Przykład:

Niech tablica decyzyjna  $S$  ma postać jak Tab. 6-2. Dla zbioru  $P = \{c_1, c_2\} \subseteq Q$  podział na klasy równoważności daje w rezultacie:

$$U/IND(P) = \{\{1, 3, 4\}, \{2\}, \{5, 6\}\}$$

### 6.3.1.3 Aproksymacja zbiorów

Niech  $P \subseteq Q, X \subseteq U$ . Wówczas *aproksymacje zbiorów* są zdefiniowane następująco:

- *dolna aproksymacja X przez P* (ang. *P-lower approximation of X*):

$$\underline{P}X = \bigcup \{Y \in U/IND(P) : Y \subseteq X\}, \quad (6-34)$$

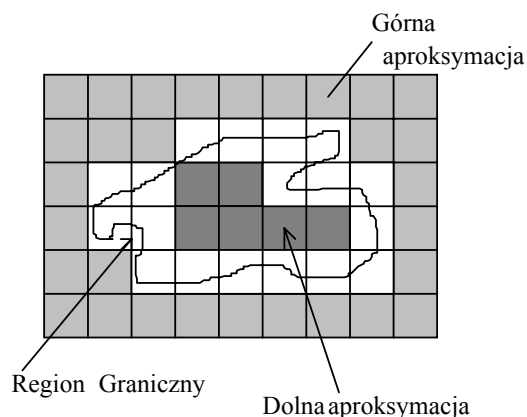
- *górna aproksymacja X przez P* (ang. *P-upper approximation of X*):

$$\overline{P}X = \bigcup \{Y \in U/IND(P) : Y \cap X \neq \emptyset\}. \quad (6-35)$$

Różnica aproksymacji górnej i dolnej zwana jest *regionem granicznym* (ang. *boundary region*):

$$BN_P(X) = \bar{P}X - \underline{P}X \quad (6-36)$$

Ilustrację graficzną powyższych pojęć przedstawiono na rys. 6-24. Obiekt  $x \in U$  jest *na pewno* zawarty w  $X$  wtedy i tylko wtedy, gdy  $x \in \underline{P}X$ , natomiast *może być* zawarty w  $X$  wtedy i tylko wtedy, gdy  $x \in \bar{P}X$ . Innymi słowy, obiekt  $x \in \underline{P}X$  jest opisany regułą pewną,  $BN_P(X)$  jest zbiorem reguł niepewnych, a obiekt  $x \in \bar{P}X$  może być opisany zarówno regułą pewną jak i niepewną.



Rys. 6-24. Interpretacja graficzna aproksymacji górnej, dolnej i regionu granicznego

*Przykład:*

Dla tablicy decyzyjnej  $S$ , takiej jak Tab. 6-2 niech zbiór  $X = \{1,3,4,5\} \subseteq U$  będzie aproksymowany przez zbiór  $P = \{c_1, c_2\} \subseteq Q$ . Podział na klasy równoważności daje w wyniku zbiór:  $U/IND(P) = \{\{1,3,4\}, \{2\}, \{5,6\}\}$ , a wówczas aproksymacje górne, dolne i region graniczny są określone następująco:

$$\bar{P}X = \{\{1,3,4\}, \{5,6\}\} = \{1,3,4,5,6\}$$

$$\underline{P}X = \{\{1,3,4\}\} = \{1,3,4\}; \quad BN_P(X) = \{5,6\}$$

#### 6.3.1.4. Zbiór przybliżony

W oparciu o pojęcia aproksymacji zbiorów oraz regionu granicznego można wprowadzić definicję zbioru przybliżonego. Niech  $X \subseteq U$  oraz

$P \subseteq Q$ . Wówczas zbiór  $X$  jest *zbiorem przybliżonym* (ang. *rough set*) względem zbioru  $P$ , jeśli  $\underline{P}X \neq \overline{P}X$ , co jest równoważne temu, że region graniczny nie jest zbiorem pustym:  $BN_P(X) \neq \emptyset$ . Zbiór  $X$ , dla którego  $\underline{P}X = \overline{P}X$  nazywany jest *zbiorem dokładnym* (ang. *exact, crisp set*).

Przykład:

W poprzednim przykładzie, zbiór  $X = \{1,3,4,5\} \subseteq U$  jest zbiorem przybliżonym przez zbiór  $P = \{c_1, c_2\} \subseteq Q$ , gdyż region graniczny nie jest zbiorem pustym:  $BN_P(X) = \{5,6\}$ .

6.3.1.5. Pozytywny region klasyfikacji

Niech  $P, R \subseteq Q$ . Zbiór  $POS_R(P)$  jest *pozytywnym regionem klasyfikacji* (ang. *positive region of classification*) zbioru  $R$  i jest on zdefiniowany następująco:

$$POS_R(P) = \bigcup_{X \in U / IND(P)} \underline{R}X \tag{6-37}$$

Zbiór  $POS_R(P)$  zawiera te obiekty ze zbioru  $U$ , które mogą być zakwalifikowane do jednej z klas równoważności  $IND(P)$  na podstawie atrybutów ze zbioru  $R$ .

Przykład:

Załóżmy, że tablica decyzyjna jest zdefiniowana jak Tab. 6-2. Dla zbiorów  $P$  i  $R$  zachodzi:

Tab. 6-2. Przykładowa tablica decyzyjna

$U$	$Q$				
	$C$				$D$
	$c_1$	$c_2$	$c_3$	$c_4$	$d$
1	<i>A</i>	<i>Yes</i>	10	-5	<i>Low</i>
2	<i>B</i>	<i>Yes</i>	10	-5	<i>High</i>
3	<i>A</i>	<i>Yes</i>	10	7	<i>High</i>
4	<i>A</i>	<i>Yes</i>	50	7	<i>High</i>
5	<i>B</i>	<i>No</i>	10	-5	<i>High</i>
6	<i>B</i>	<i>No</i>	10	-5	<i>Low</i>

$$P = D = \{d\} \subseteq Q,$$

$$R = C = \{c_1, c_2, c_3, c_4\} \subseteq Q,$$

pozytywny region klasyfikacji będzie określony następująco:

$$\text{POS}_R(P) = \text{POS}_C(D) = \bigcup_{X \in U / \text{IND}(D)} \underline{C}X$$

Ponieważ  $U / \text{IND}(D) = \{\{1,6\}, \{2,3,4,5\}\}$ , zatem:

$$\text{POS}_C(D) = \bigcup_{X \in U / \text{IND}(D)} \underline{C}X = \underline{C}\{1,6\} \cup \underline{C}\{2,3,4,5\}.$$

Biorąc pod uwagę, że  $U / \text{IND}(C) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5,6\}\}$ ,

to pozytywny region klasyfikacji  $\text{POS}_C(D)$  będzie określony następująco:

$$\text{POS}_C(D) = \underline{C}\{1,6\} \cup \underline{C}\{2,3,4,5\} = \{1\} \cup \{2\} \cup \{3\} \cup \{4\} = \{1,2,3,4\}$$

### 6.3.1.6. Redukcja atrybutów, redukty

Niech  $C, D \subseteq Q$  będą zbiorami atrybutów warunkowych i decyzyjnych. Wówczas:

- atribut  $a \in C$  jest *zbędny* w zbiorze  $C$  ze względu na  $D$ , jeśli  $\text{POS}_C(D) = \text{POS}_{(C-\{a\})}(D)$ . W przeciwnym wypadku, atribut  $a$  jest *niezbędny* w zbiorze  $C$  ze względu na zbiór  $D$ .
- Jeśli wszystkie atrybuty  $a \in C$  w zbiorze  $C$  są niezbędne ze względu na  $D$ , to zbiór  $C$  jest *niezależny od zbioru*  $D$ .
- Podzbiór  $C' \subseteq C$  jest *reduktem* zbioru  $C$  ze względu na zbiór  $D$  wtedy i tylko wtedy, gdy  $C'$  jest niezależny od  $D$  oraz  $\text{POS}_C(D) = \text{POS}_{C'}(D)$ .

#### Przykład:

Dla tablicy decyzyjnej Tab. 6-2 rozważmy następujące zbiory:

$$D = \{d\} \subseteq Q,$$

$$C = \{c_1, c_2, c_3, c_4\} \subseteq Q,$$

$$C' = C - \{c_1\} = \{c_2, c_3, c_4\} \subseteq Q,$$

$$C'' = C - \{c_3\} = \{c_1, c_2, c_4\} \subseteq Q.$$

Na podstawie zależności (6-37), pozytywny region klasyfikacji dla zbiorów:

$C$ ,  $C'$  oraz  $C''$  jest następujący:

$$\text{POS}_C(D) = \{1,2,3,4\},$$

$$\text{POS}_{C'}(D) = \{3,4\},$$

$$\text{POS}_{C''}(D) = \{1,2,3,4\}$$

Ponieważ  $\text{POS}_C(D) = \text{POS}_{C''}(D)$ , więc podzbiór  $C''$  może być reduktem zbioru  $C$ . Należy zatem sprawdzić, czy  $C''$  jest niezależny od zbioru  $D$ :

$$\text{POS}_{C''-\{c_1\}}(D) = \{3,4\} \neq \text{POS}_C(D),$$



$$\text{POS}_{C'-\{c_2\}}(D) = \{1,3,4\} \neq \text{POS}_C(D),$$

$$\text{POS}_{C'-\{c_4\}}(D) = \{2\} \neq \text{POS}_C(D)$$

Z powyższych obliczeń wynika, że atrybuty  $c_1$ ,  $c_2$  i  $c_4$  są niezbędne w zbiorze  $C$  ze względu na zbiór  $D$ , zaś atrybut  $c_3$  jest zbędny. Zbiór  $C''$  jest wobec tego niezależny od  $D$ , a ponieważ  $\text{POS}_C(D) = \text{POS}_{C''}(D)$ , więc jest również reduktom zbioru  $C$ .

### 6.3.1.7. Redukcja wartości atrybutów

Niech  $C, D \subseteq Q$  będą zbiorami atrybutów oraz niech będzie dana zależność  $C \Rightarrow D$ , gdzie  $C$  jest reduktom ze względu na zbiór  $D$ . Wówczas:

a) Wartość atrybutu  $q \in C$  jest *zbędna ze względu na zbiór*  $D$  dla obiektu  $x \in U$ , jeśli:

$$C(x) \subseteq D(x) \Rightarrow C_q(x) \subseteq D(x)$$

gdzie:  $C_q(x) = C - \{q\}$ . W przeciwnym razie, wartość atrybutu  $q$  jest *niezbędna ze względu na zbiór*  $D$  dla obiektu  $x$ .

b) Jeśli wartość każdego atrybutu  $q \in C$  jest *niezbędna ze względu na*  $D$  dla obiektu  $x$ , to zbiór  $C$  jest *niezależny od zbioru*  $D$  dla obiektu  $x$ .

c) Podzbiór  $C' \subseteq C$  jest *reduktom zbioru*  $C$  ze względu na zbiór  $D$  dla obiektu  $x$  wtedy i tylko wtedy, gdy  $C'$  jest niezależny od  $D$  oraz zachodzi:

$$C(x) \subseteq D(x) \Rightarrow C'(x) \subseteq D(x)$$

#### Przykład:

W poprzednim przykładzie reduktom zbioru atrybutów warunkowych  $C = \{c_1, c_2, c_3, c_4\}$  dla systemu informacyjnego z Tab. 6-2 był zbiór  $C'' = C - \{c_3\} = \{c_1, c_2, c_3, c_4\}$ , gdzie  $C'' \subseteq C$ . A zatem z tablicy decyzyjnej można usunąć kolumnę atrybutów  $c_3$ . Niech obecnie zbiór atrybutów warunkowych będzie następujący:  $C = C'' = \{c_1, c_2, c_4\}$ . Rozważmy kolejne podzbiory zbioru  $C$ :

$$C' = C - \{c_4\} = \{c_1, c_2\},$$

$$C'' = C - \{c_2\} = \{c_1, c_4\},$$

$$C''' = C - \{c_1\} = \{c_2, c_4\} \text{ dla obiektu nr 2 (} x = 2 \text{)}.$$

Podział na klasy równoważności  $U / \text{IND}(C')$  spowoduje podział zbioru wszystkich obiektów (reguł)  $U$  na podzbiory. W rezultacie reguły w takich podzbiorych mają te same przesłanki. Jeśli dla wszystkich reguł w takim podzbiorych  $R$  zbiór decyzji jest taki sam, to wówczas wartości atrybutów ze zbioru  $C - C' = \{c_4\}$  są nieistotne ze względu na zbiór decyzji, a zatem mogą być pominięte, czyli zachodzi relacja:

$$R = \left\{ X \in U / \text{IND}(C') : \bigvee_{x, y \in X} D(x) = D(y) \right\}$$

Podział na klasy równoważności dla zbiorów  $C, C''$  i  $C'''$  daje w rezultacie:

$$U / \text{IND}(C') = \{\{1,3,4\}, \{2\}, \{5,6\}\},$$

$$U / \text{IND}(C'') = \{\{1\}, \{2,5,6\}, \{3,4\}\},$$

$$U / \text{IND}(C''') = \{\{1,2\}, \{3,4\}, \{5,6\}\}.$$

Zwróćmy uwagę na podzbiory  $X$  zawierające interesujący nas obiekt ( $x = 2$ ):

- jeśli rozważamy  $C' = C - \{c_4\} = \{c_1, c_2\}$ , to zachodzi:

$$X = \{2\} \Rightarrow \forall_{x,y \in X} D(x) = D(y), \text{ tj. } D(2) = \text{High},$$

- jeśli rozważamy  $C'' = C - \{c_2\} = \{c_1, c_4\}$ , to spełniony jest warunek:

$$X = \{2,5,6\} \Rightarrow \exists_{x,y \in X} D(x) \neq D(y), \text{ tj. } D(2) = \text{High} \wedge D(6) = \text{Low}$$

- jeśli rozważamy  $C''' = C - \{c_1\} = \{c_2, c_4\}$ , to otrzymujemy:

$$X = \{1,2\} \Rightarrow \exists_{x,y \in X} D(x) \neq D(y), \text{ tj. } D(1) = \text{Low} \wedge D(2) = \text{High}$$

Zatem wartość atrybutu  $c_4$ , tj.  $v_{c_4} = -5$ ,  $v_{c_4} \in V_{c_4}$  jest zbędna dla obiektu nr 2, natomiast wartości atrybutów  $c_1$  i  $c_2$  są niezbędne. Co więcej, zbiór  $C'$  jest niezależny ze względu na  $D$  dla obiektu nr 2, gdyż wartości dla każdego atrybutu  $C'$  (tj.  $c_1$  i  $c_2$ ) są niezbędne. Ponieważ spełniona jest także relacja  $C' \subseteq C$ , więc podzbiór warunkowy  $C'$  jest reduktom zbioru  $C$  ze względu na zbiór decyzji  $D$  dotyczących obiektu  $x$ .

### 6.3.1.8. Miara przybliżona

*Miara przybliżona* (ang. *rough measure*)  $\mu(x)$  reguły (obiektu)  $x$  opisanej konceptem  $X$  jest dana zależnością:

$$\mu(x) = \frac{|X \cap Y|}{|Y|} \quad (6-38)$$

gdzie  $X$  jest konceptem, zaś  $Y$  jest zbiorem wszystkich reguł o tych samych atrybutach warunkowych, co obiekty opisane przez koncept  $X$ .

Miara przybliżona  $\mu(x)$  jest stosunkiem liczby wszystkich przykładów  $x$  dla konceptu  $X$  poprawnie opisanych przez regułę do liczby wszystkich obiektów opisanych przez tę regułę. Dla reguł pewnych miara  $\mu$  przyjmuje wartość 1, zaś dla reguł niepewnych przyjmuje wartości z zakresu  $(0, 1)$ . Stąd wniosek, że im większa miara przybliżona, tym reguła jest bardziej wiarygodna.

#### Przykład:

Niech  $X(d = \text{High})$  będzie konceptem dla tablicy decyzyjnej Tab. 6-2. Obliczmy miarę przybliżoną dla reguł niepewnych opisywanych przez ten koncept dla dwóch zbiorów atrybutów warunkowych:  $C$  oraz  $C' = \{c_1\}$ . Zbiór obiektów opisywanych przez koncept  $X(d = \text{High})$  jest następujący:

$$X(d = High) = \{2,3,4,5\}.$$

W celu określenia zbioru reguł niepewnych, wyznaczane są kolejne zbiory:

$$U / IND(C) = \{\{1\}, \{2\}, \{3,4\}, \{5,6\}\} \text{ dla } C = \{c_1, c_2, c_3, c_4\},$$

$$\underline{CX} = \{2,3,4\},$$

$$\overline{CX} = \{2,3,4,5,6\}$$

Zatem zbiór reguł niepewnych (tj. region graniczny) ma postać:

$$BN_C(X) = \overline{CX} - \underline{CX} = \{5,6\},$$

zaś reguły niepewne opisywane przez koncept  $X$  ( $d = High$ ) są następujące:

$$Z = BN_C(X) \cap X = \{5\}.$$

a) niech zbiór atrybutów warunkowych dla reguł niepewnych będzie określony następująco:  $C = \{c_1, c_2, c_3, c_4\}$ . Zbiór  $Y$  jest zbiorem tych reguł, które mają te same atrybuty warunkowe, co obiekty opisane przez koncept  $X$  ( $d = High$ ). Zatem:

$$Y = \{5,6\} \text{ dla } C(x) = \{c_1=B, c_2=No, c_3=10, c_4=-5\}.$$

Miara przybliżona  $\mu(x)$  dla  $x \in Z = \{5\}$  wyrażona jest zależnością:

$$\mu(x) = \frac{|\{2,3,4,5\} \cap \{5,6\}|}{|\{5,6\}|} = 0,5$$

b) niech teraz zbiór atrybutów warunkowych dla reguł niepewnych ma postać  $C' = \{c_1\}$ . Zbiór  $Y$  jest wówczas następujący:

$$Y = \{2,5,6\} \text{ dla } C'(x) = \{c_1=B\}.$$

Miara przybliżona  $\mu(x)$  dla  $x = \{5\}$  wynosi:

$$\mu(x) = \frac{|\{2,3,4,5\} \cap \{2,5,6\}|}{|\{2,5,6\}|} = 0,667$$

Ponieważ w drugim przypadku miara przybliżona jest większa, więc obiekt nr 5 reprezentuje wiedzę wyrażoną w postaci reguły:

**IF**  $c_1 = B$  **THEN**  $d = High$ ,

która pomimo swej prostoty jest bardziej wiarygodna niż oryginalna reguła:

**IF**  $c_1 = B$  **AND**  $c_2 = No$  **AND**  $c_3 = 10$  **AND**  $c_4 = -5$  **THEN**  $d = High$

### 6.3.2. System decyzyjny oparty na zbiorach przybliżonych

Jak wspomniano, informacja jest przechowywana w systemie opartym na metodzie zbiorów przybliżonych w postaci tablicy decyzyjnej, dzięki czemu można wyodrębnić atrybuty warunkowe (przesłanki) oraz atrybuty decyzyjne. Zapis taki reprezentuje wiedzę zgromadzoną w postaci reguł IF-THEN, zebraną przez eksperta (lub ekspertów). Nierzadko jednak decyzje ekspertów mogą się różnić, a co za tym idzie, tablica informacyjna staje się niespójna – wynikają z niej reguły niepewne. Istotną zaletą systemu decyzyjnego opartego na zbiorach przybliżonych jest możliwość

przetwarzania reguł sprzecznych. Z drugiej strony, systemy te operują na wartościach dyskretnych. Jeśli dane wejściowe są ciągłe, wówczas w fazie wstępnej należy przeprowadzić kwantyzację. W praktyce, od sposobu dokonywania kwantyzacji zależy w dużej mierze jakość działania takiego systemu.

Zwróćmy uwagę, że sama tablica decyzyjna nie umożliwia przetwarzania informacji, które nie zostały uwzględnione podczas tworzenia tej tablicy. Aby wartość atrybutu decyzyjnego lub atrybutów pojawiła się na wyjściu systemu decyzyjnego, określona reguła musi zostać uaktywniona. Oznacza to, że muszą być spełnione wszystkie jej przesłanki, tzn. wartości atrybutów wejściowych muszą się zgadzać z wartościami atrybutów warunkowych. W przypadku, gdy na wejściu systemu pojawia się wartość atrybutu, która nigdy wcześniej nie wystąpiła, system nie zareaguje. Dlatego też, obiekty (reguły) zebrane w tablicy decyzyjnej nazywane są także *faktami*.

Wykorzystanie własności zbiorów przybliżonych umożliwia rozszerzenie możliwości takiej tablicy, co może prowadzić do znacznego uproszczenia reguł. W konsekwencji, system decyzyjny nabiera zdolności generalizacyjnych, dzięki czemu może stanowić efektywne, inteligentne narzędzie obróbki danych.

Często celem systemu decyzyjnego opartego na zbiorach przybliżonych jest poszukiwanie ukrytych, a więc niejawnych reguł, które leżą u podstaw decyzji eksperta (lub ekspertów). Takie procesy wydobywania wiedzy ze zbioru danych określane są anglojęzycznymi terminami, które nie mają jeszcze ugruntowanych polskich odpowiedników. Są to „*knowledge discovery*” i „*data mining*”, co można przetłumaczyć jako „*odkrywanie wiedzy*” i „*wydobywanie reguł*”.

Nie istnieje uniwersalny sposób wydobywania wiedzy z danych. Istniejące metody w różny sposób wykorzystują własności zbiorów przybliżonych dyskutowane w podrozdziale 6.3.1. Poniżej przedstawiono jeden z takich algorytmów. Bliższe szczegóły jego działania zostały omówione w rozdziale 6.3.1. W krokach 2-4 analizowane są wszystkie podzbiory będące kombinacją atrybutów warunkowych. Dodatkowo w kroku 4 analiza ta przeprowadzana jest oddzielnie dla każdej klasy równoważności.

#### 1) *Przetwarzanie wstępne*

Procedura niniejsza umożliwia przygotowanie danych do ich przetwarzania w systemie. Pod tym pojęciem może się kryć wyznaczenie pewnych zbiorów, które są wykorzystywane wielokrotnie podczas pracy algorytmu. Tymi zbiorami są:

- zbiór wynikający z podziału na klasy równoważności względem zbioru atrybutów warunkowych  $U / IND(C)$  oraz decyzyjnych  $U / IND(D)$ ,

- zbiór reguł pewnych, tj. pozytywny region klasyfikacji  $POS_C(D)$ .  
 Jeśli dane wejściowe nie są dyskretne, to zwykle dokonuje się ich kwantyzacji.

- 2) *Redukcja atrybutów* (opisana wcześniej)
- 3) *Redukcja wartości atrybutów* (opisana wcześniej)
- 4) *Obliczenie miary przybliżonej dla reguł niepewnych* (opisane wcześniej)
- 5) *Wybór reguł o największej mierze przybliżonej*

Do dalszego wykorzystania w systemie decyzyjnym wybierane są reguły o największej wartości miary przybliżonej. Ponieważ miara przybliżona jest obliczana dla wszystkich podzbiorów kombinacji wartości atrybutu decyzyjnego (lub atrybutów), wybór jej wartości jest równoznaczny z redukcją wartości atrybutów warunkowych dla reguł niepewnych. Ze względu na to, że miara przybliżona reguły pewnej wynosi 1, zatem wszystkie reguły pewne mogą być bezpośrednio wykorzystane w systemie.

Przykład:

Niech za przykładową tablicę decyzyjną posłuży Tab. 6-2. Zgromadzona w niej dane (fakty) reprezentowane są bezpośrednio przez reguły:

1. **IF**  $c_1 = A$  **AND**  $c_2 = Yes$  **AND**  $c_3 = 10$  **AND**  $c_4 = -5$  **THEN**  $d = Low$
2. **IF**  $c_1 = B$  **AND**  $c_2 = Yes$  **AND**  $c_3 = 10$  **AND**  $c_4 = -5$  **THEN**  $d = High$
3. **IF**  $c_1 = A$  **AND**  $c_2 = Yes$  **AND**  $c_3 = 10$  **AND**  $c_4 = 7$  **THEN**  $d = High$
4. **IF**  $c_1 = A$  **AND**  $c_2 = Yes$  **AND**  $c_3 = 50$  **AND**  $c_4 = 7$  **THEN**  $d = High$
5. **IF**  $c_1 = B$  **AND**  $c_2 = No$  **AND**  $c_3 = 10$  **AND**  $c_4 = -5$  **THEN**  $d = High$
6. **IF**  $c_1 = B$  **AND**  $c_2 = No$  **AND**  $c_3 = 10$  **AND**  $c_4 = -5$  **THEN**  $d = Low$

Po przeprowadzeniu obliczeń z wykorzystaniem metodyki zbiorów przybliżonych, wiedza zawarta w tablicy decyzyjnej jest opisywana przez następujące reguły:

- reguły pewne

1. **IF**  $c_1 = A$  **AND**  $c_4 = -5$  **THEN**  $d = Low$
2. **IF**  $c_1 = B$  **AND**  $c_2 = Yes$  **THEN**  $d = High$
3. **IF**  $c_3 = 10$  **THEN**  $d = High$

- reguły niepewne

4. **IF**  $c_1 = B$  **THEN**  $d = High$        $\mu(x) = 0,667$
5. **IF**  $c_2 = No$  **THEN**  $d = Low$        $\mu(x) = 0,5$

Zachęca się Czytelnika do samodzielnego wydobycia powyższych reguł na podstawie Tab. 6-2 i opisanych powyżej procedur.

## 6.4 Usuwanie pasożytniczych impulsów

W roz. 4 wymieniono kilka metod cyfrowego przetwarzania sygnałów, które mogą być zastosowane do redukcji szumu impulsowego. Opisywane tam algorytmy są oparte na operacjach filtracji progowej, technikach ekstrapolacyjnych, metodzie predykcji liniowej, zastosowaniu transformacji falkowej lub filtracji adaptacyjnej. Jednocześnie niektóre zaawansowane metody analizy i przetwarzania danych są nadal rzadko stosowane w technice fonicznej. Chodzi tu o znane w informatyce algorytmy z dziedziny sztucznej inteligencji, np. takie, które omówiono w poprzednich paragrafach niniejszego rozdziału.

Wspólną cechą metod sztucznej inteligencji jest ich zdolność uczenia się na podstawie przykładów. W trakcie nauki (treningu) budowana jest baza wiedzy, umożliwiająca dokładne rozpoznawanie znanych sytuacji, parametrów, cech i współzależności. Wspecjalizowane algorytmy sztucznej inteligencji są w stanie generalizować nabytą wiedzę w taki sposób, że mogą dopasowywać do siebie niedokładne lub zaszumione dane, sekwencje przesunięte w czasie oraz sekwencje należące do tej samej klasy pomimo tego, że związki pomiędzy nimi są ukryte. Zatem zbiór metod wykorzystywanych do poprawy jakości sygnałów dźwiękowych należy rozszerzyć o algorytmy uczące się. Wniosek ten stał się inspiracją dla autora i jego współpracowników do poszukiwania takich inteligentnych metod przetwarzania sygnałów, które nadawałyby się do rozwiązywania problemów z dziedziny akustyki fonicznej, w szczególności metod które mogłyby być wykorzystane do rekonstruowania sygnału fonicznego.

Algorytmy uczące się zostały przez autora wykorzystane m. in. do wykrywania zakłóceń impulsowych i interpolacji przerw powstałych po usunięciu pasożytniczych impulsów. Ponieważ typowe algorytmy uczące się są wysoce nieliniowe, jedynie fragmenty sygnału uznane za zniekształcone powinny być przetwarzane przez te algorytmy, natomiast pozostała część materiału muzycznego powinna pozostać niezmienną. W przeciwnym przypadku trudno byłoby uniknąć silnych, nieliniowych zniekształceń, wpływających na brzmienie dźwięku. Zatem w omawianym podejściu, przetwarzanie przy użyciu algorytmu neuronowego musi być ograniczone jedynie do fragmentów sygnału zakłóconych przez niepożądane (pasożytnicze) impulsy.

W niniejszym podrozdziale omówiono algorytmy oparte na sieciach neuronowych, w zastosowaniu do wykrywania i usuwania trzasków. Pierwsza sieć neuronowa została wytrenowana do wykrywania trzasków i tworzenia tablicy uszkodzonych przedziałów. Kolejna sieć neuronowa została wytrenowana w taki sposób, że jest w stanie dokonywać pewnego rodzaju nieliniowej predykcji sygnału, umożliwiając odzyskiwanie próbek zniekształconych przez trzaski.

#### **6.4.1 Automatyczne wykrywanie pasożytniczych impulsów**

##### *6.4.1.1 Algorytmy progowe*

Celem procedury detekcyjnej jest zebranie informacji o trzaskach i innych zakłóceniach impulsowych, zniekształcających nagrania foniczne, czy transmisję tego typu sygnałów. Proces rekonstruowania sygnału zniekształconego przez zakłócenia impulsowe można podzielić na dwa etapy:

1. wykrywanie trzasków i ustalanie granic każdego kolejnego fragmentu, zawierającego utracone próbki,
2. odzyskiwanie uszkodzonych fragmentów przy zapewnieniu możliwie najlepszej aproksymacji charakterystyk oryginalnego sygnału

Można zauważyć, że w ogólności zakłócenia impulsowe posiadają następujące cechy:

- w dziedzinie czasu, ich energia jest skupiona w krótkim przedziale czasu,
- w dziedzinie widma, ich energia jest rozłożona w szerokim zakresie częstotliwości

Typowe algorytmy oparte na przetwarzaniu widma sygnału zakładają addytywny charakter zakłóceń impulsowych (trzasków). Wykorzystywana jest liniowość transformaty ortogonalnej, co umożliwia odjęcie części widma, związanej z zakłócającym impulsem, od całkowitej reprezentacji widmowej złożonego sygnału. Przykładowe algorytmy tego typu omówiono w par. 4.5.1. Metody tego typu posiadają dwa główne ograniczenia:

1. jedynie krótkie impulsy powodują takie zmiany widma sygnału, które są łatwo zauważalne przy porównaniu z sygnałem oryginalnym,
2. stany transjentowe, dźwięki perkusyjne oraz inne pożądane efekty impulsowe mogą zostać przez algorytm potraktowane jak zakłócenia

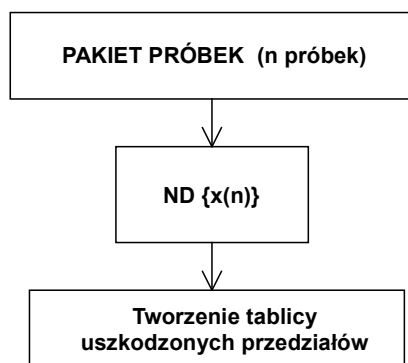
Wynika stąd potrzeba zastosowania algorytmów uczących się do przetwarzania sygnałów dźwiękowych, co umożliwia zbudowanie bazy wiedzy zarówno o zakłóceniach impulsowych, jak i o pożądanych własnościach sygnału. W ogólności, procedura wykrywająca powinna umożliwić dokładne wyznaczenie punktu początkowego i końcowego dowolnego przedziału czasowego, który zawiera próbki uszkodzone przez niepożądany impuls. Stany na wyjściu detektora można zdefiniować w ten sposób, że obecność pasożytniczego impulsu wewnątrz obserwowanego pakietu próbek powoduje pojawienie się wartości +1, natomiast sygnał nieuszkodzony jest reprezentowany przez wartość -1 na tym wyjściu.

Detektor generuje falę prostokątną, której zbocza są zsynchronizowane z początkowymi i końcowymi punktami pakietów uszkodzonych przez zniekształcenia impulsowe. Wspomniany detektor należy rozumieć jako system o wielu wejściach, ponieważ nie jest możliwe wykrycie trzasku na podstawie znajomości wartości jednej próbki lub niewystarczająco licznej grupy próbek. Zatem zasada działania detektora o wielu wejściach musi być oparta na przetwarzaniu grupy następujących po sobie próbek, pojawiających się kolejno na jego wejściach.

#### 6.4.1.2 Algorytmy uczące się

Omówiony wcześniej detektor może zostać skonstruowany na podstawie prostej struktury sieci neuronowej (perceptronu), wytrenowanej do rozpoznawania dwóch klas obiektów – uszkodzonych i nieuszkodzonych pakietów próbek. Wystarczające jest, aby warstwa wyjściowa takiej sieci wielowarstwowej składała się tylko z jednego neuronu, będącego elementem decyzyjnym. Wyjście sieci neuronowej jest zsynchronizowane ze środkową próbką rekonstruowanego pakietu. Pakiet  $n$  próbek przetwarzanych przez sieć neuronową powinien dawać pojedynczą wartość na wyjściu, składającą się z symbolu  $-1$  lub  $+1$ . Seria  $k$  takich pakietów przetworzonych przez sieć neuronową powoduje powstanie na wyjściu fali prostokątnej. Działający w ten sposób "neurodetektor" może być zatem zdefiniowany przez operację  $ND\{x(n)\}$ , zaimplementowaną do celu detekcji zakłóceń impulsowych, jak pokazano na rys. 6-25. Schemat jednokierunkowej, w pełni połączonej sieci neuronowej (perceptronu) z propagacją w przód pokazano na rys. 6-26.

Jako funkcję kształtującą neuronów przyjęto tangens hiperboliczny. Zatem stan na wyjściu  $j$ -tego neuronu jest opisany zależnością:



Rys. 6-25. Sieć neuronowa jako detektor uszkodzonych próbek



$$o = \tanh\left(\sum_i w_i x_i + b_j\right) \quad (6-39)$$

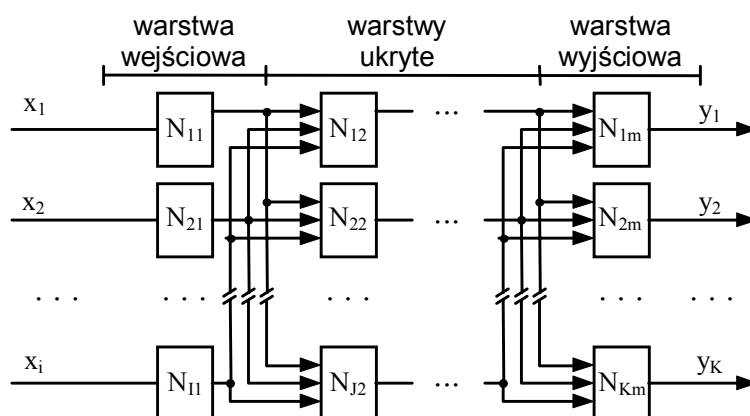
gdzie:  $x_i$  – sygnał wejściowy na  $i$ -tym połączeniu synaptycznym  
 $w_i$  – waga  $i$ -tego połączenia synaptycznego  
 $b_j$  – poziom wstępny  $j$ -tego neuronu.

Sieć jest opisana przez wektor stanu, zdefiniowany jako:

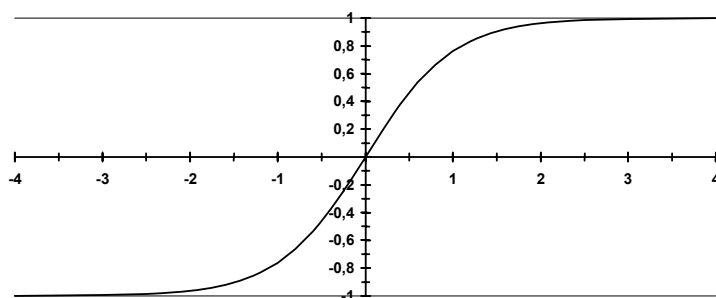
$$\mathbf{S} = [w_1, w_2, \dots, w_s, b_1, b_2, \dots, b_r] \quad (6-40)$$

gdzie:  $\mathbf{S}$  – wektor stanu;  $w_1, \dots, w_s$  – wagi synaptyczne  
 $b_1, \dots, b_r$  – poziomy wstępne,  $s$  – liczba synaps,  $r$  – liczba neuronów

(a)



(b)



**Rys. 6-26.** Sieć neuronowa zastosowana do detekcji uszkodzonych próbek  
 (a) struktura sieci; (b) funkcja kształtująca neuronu – tangens hiperboliczny

Procedura treningu sieci neuronowej wymaga dobrania przykładów z obu klas – fragmentów pożądanego sygnału i zakłócających impulsów. To założenie pozwala sformułować równanie, opisujące błąd średniokwadratowy (patrz par. 6.2.5.2), który jest minimalizowany podczas treningu:

$$E = \frac{1}{p_1} \sum_{i=1}^{p_1} (o_i - d_1)^2 + \frac{1}{p_2} \sum_{j=1}^{p_2} (o_j - d_2)^2 \quad (6-41)$$

gdzie:

- $p_1$  – liczba przykładów z klasy 1, reprezentującej sygnał bez zakłóceń
- $p_2$  – liczba przykładów z klasy 2, reprezentującej sygnał zakłócony
- $o_i$  – sygnał na wyjściu sieci dla  $i$ -tego przykładu z klasy 1
- $o_j$  – sygnał na wyjściu sieci dla  $j$ -tego przykładu z klasy 2
- $d_1$  – oczekiwany stan na wyjściu sieci dla sygnału należącego do klasy 1  
( $d_1 = -1$ )
- $d_2$  – oczekiwany stan na wyjściu sieci dla sygnału należącego do klasy 2  
( $d_2 = +1$ )

Ponieważ wszystkie przykłady reprezentujące obie klasy są wykorzystywane w procesie treningu sieci, wektor stanu jest znany dla każdego przykładu. Można zatem wprowadzić następujący zapis:

$$y_i = y_i(\mathbf{S}), \quad y_j = y_j(\mathbf{S}), \quad E = E(\mathbf{S}) \quad (6-42)$$

Procedura minimalizacji błędu, zdefiniowanego przez równanie (6-38), jest oparta na metodzie wstecznej propagacji błędu, co zapewnia minimalizację połowy normy błędu  $E$  względem wszystkich wag i poziomów. Wartości wag synaptycznych są korygowane w kolejnych iteracjach zgodnie z gradientową procedurą minimalizacji błędu. Gradient błędu jest zdefiniowany jako:

$$\nabla E = \left[ \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_s}, \frac{\partial E}{\partial b_1}, \frac{\partial E}{\partial b_2}, \dots, \frac{\partial E}{\partial b_r} \right] \quad (6-43)$$

W praktyce, zamiast pochodnych oblicza się następujące wartości przyrostów:

$$\frac{\Delta E}{\Delta w_i}; \frac{\Delta E}{\Delta b_j}, \quad i = 1, \dots, s; \quad j = 1, \dots, r \quad (6-44)$$

Po zainicjowaniu procedury (krok 0), wartości  $\Delta w_i$  i  $\Delta b_j$  są przyjmowane jako równe małej, empirycznie wybranej wartości  $\alpha_0$ :

$$\Delta w_1 = \Delta w_2 = \dots = \Delta w_s = \Delta b_1 = \Delta b_2 = \dots = \Delta b_r = \Delta b_0 = \alpha_0$$

W kolejnych krokach (krok 1, krok 2, ..., krok n)  $\Delta w_i$  i  $\Delta b_j$  przyjmują wartości  $\Delta_n$  takie, że:

$$\Delta_n = \alpha \cdot \Delta_{n-1}$$

gdzie  $\alpha < 1$  jest małą liczbą ustaloną arbitralnie.

Wielkości przyrostów stopniowo zmniejszają się, co jest spowodowane mnożeniem przez współczynnik  $\alpha$ . Zatem dokładność procedury minimalizacji błędu rośnie z każdym krokiem iteracji. W rezultacie uaktualnienia wag synaptycznych, sieć osiąga nowy stan, który spełnia warunek

$$E(\mathbf{S}_n) < E(\mathbf{S}_{n-1}) \quad (6-45)$$

gdzie  $\mathbf{S}_n$  jest wektorem stanu sieci w n-tym kroku.

Nowe wartości wektora stanu są obliczane na podstawie zależności:

$$\mathbf{S}_n = \mathbf{S}_{n-1} - \eta \cdot \nabla E \quad (6-46)$$

gdzie  $\eta$  jest współczynnikiem danym wzorem:

$$\eta = k \cdot \Delta_{n-1}, \quad (6-47)$$

zaś  $k$  jest największą liczbą całkowitą, która spełnia warunek:

$$E(\mathbf{S}_{n-1} - k \cdot \Delta_{n-1} \cdot \nabla E) < E(\mathbf{S}_{n-1} - (k-1) \cdot \Delta_{n-1} \cdot \nabla E) \quad (6-48)$$

Największa liczba całkowita  $k$  spełniająca powyższy warunek jest wykorzystywana do obliczenia końcowego wektora stanu  $S_n$  dla  $n$ -tego kroku iteracji, zgodnie z równaniami (6-41) i (6-42). Autor eksperymentalnie stwierdził, że wartości  $\alpha_0 = 10^{-4}$  i  $\alpha = 0,87$  pozwalają na optymalizację działania algorytmu propagacji wstecznej w omawianym zastosowaniu.

#### 6.4.2 Rekonstruowanie utraconych próbek

Po wykryciu przedziałów zawierających uszkodzone próbki i wyznaczeniu granic tych przedziałów, rozpoczyna się główny proces rekonstruowania uszkodzonego dźwięku, polegający na zastępowaniu uszkodzonych próbek estymatami próbek sygnału. W literaturze podawanych jest wiele metod ekstrapolacji i interpolacji utraconych próbek. Należą do nich:

- Interpolacja rzędu zerowego (technika ta nie zapewnia ciągłości sygnału na granicach interpolowanego fragmentu) lub rzędu pierwszego (nie jest zapewniona ciągłość pierwszej pochodnej) oraz wyższych rzędów
- Liniowa predykcja (LP), czyli estymacja przyszłych próbek na podstawie liniowej kombinacji  $k$  poprzednich próbek lub estymacja poprzednich próbek na podstawie liniowej kombinacji  $k$  kolejnych próbek
- Ekstrapolacja wielomianowa próbek utraconych w pobliżu przerw (operacja w dziedzinie częstotliwości)
- Inne techniki

Większość metod interpolacyjnych umożliwia odzyskanie brakujących próbek tylko wtedy, jeżeli z uszkodzonymi fragmentami sąsiadują przedziały zawierające próbki nieuszkodzone. Na przykład: algorytm ekstrapolacji oparty na metodzie predykcji liniowej wymaga, aby p sąsiednich próbek nie było uszkodzonych przez zniekształcenia impulsowe, gdzie  $p$  oznacza rząd predykcji. Idea bezpośredniej implementacji sieci neuronowej do przetwarzania próbek sygnału umożliwia natomiast zastosowanie pewnego rodzaju predykcji nieliniowej, która jest realizowana przez sieć neuronową.

##### 6.4.2.1 Neuropredyktor

Algorytm tego typu dokonuje przetwarzania kolejnych próbek w celu wygenerowania następnej próbki w przypadku predykcji w przód, lub próbki poprzedzającej w przypadku dokonywania predykcji utraconych próbek wstecz. Obliczona wartość próbki jest dołączana do istniejących próbek sygnału. Następnie wykonywany jest kolejny krok algorytmu, w którym procedura jest powtarzana. W tym przypadku wyjście sieci neuronowej jest

zsynchronizowane z próbkami znajdującymi się na zewnątrz odtworzonego pakietu. Schemat omówionego algorytmu przedstawiono na rys. 6-27.

Oznaczając funkcję neuropredyktora przez  $x_n = NP^+\{x, r\}$ , dokonującego nieliniowej predykcji próbek  $x_i$  na podstawie  $r$  poprzednich próbek  $x_{i-1}, x_{i-2}, \dots, x_{i-r}$ , gdzie  $r$  oznacza rząd predykcji równy liczbie wejść sieci neuronowej, można przewidzieć ciąg  $n$  następujących próbek na podstawie następującej procedury:

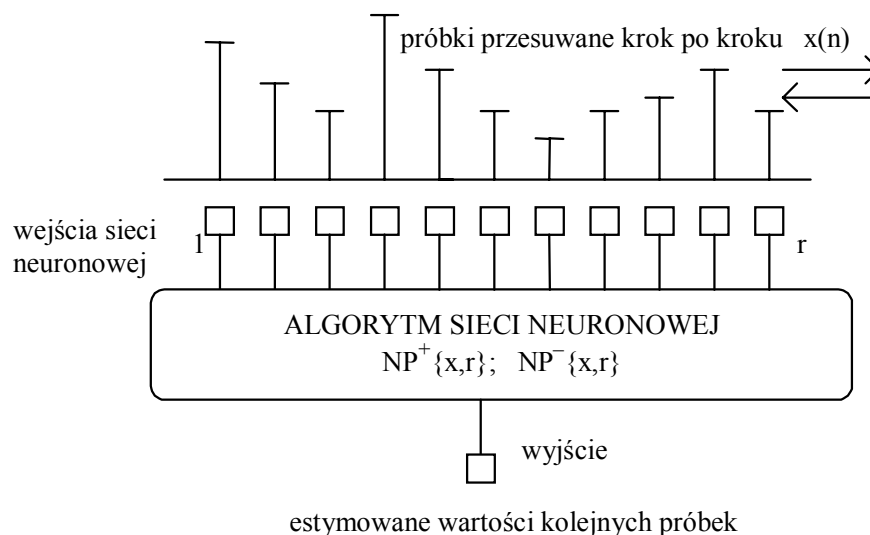
$$x_n = NP^+\{x, r\} \tag{6-49}$$

gdzie  $n = j, j + 1, j + 2, \dots, k$ ; zaś  $j, k$  oznaczają numery próbek sąsiadujących z uszkodzonym przedziałem.

W podobny sposób wytwarzany jest ciąg próbek  $y_n$  w przypadku predykcji wstecz:

$$y_n = NP^-\{x, r\} \tag{6-50}$$

Procedura interpolacji przy użyciu sieci neuronowej może zapewnić wymaganą dokładność, pod warunkiem że w procesie treningu zostanie wykorzystana wystarczająca liczba przykładów wybranych spośród nieuszkodzonych fragmentów sygnału. Fragment wykorzystywany do treningu składa się z pewnej liczby próbek, równej liczbie wejść sieci, oraz



Rys. 6-27. Konfiguracja neuropredyktora

próbki o wartości, która ma być estymowana. W tym przypadku zebrane przykłady nie są dzielone na klasy, ponieważ ten rodzaj sieci neuronowej nie dokonuje żadnej klasyfikacji. Zatem liczbę klas można przyjąć jako równą jedności. Funkcja błędu  $E$ , minimalizowana podczas treningu, może zostać zdefiniowana jako:

$$E = \sum_{i=1}^p (o_i - d_i)^2 \quad (6-51)$$

gdzie:

- $p$  – liczba wykorzystanych przykładów
- $o_i$  – stan na wyjściu sieci neuronowej dla  $i$ -tego przykładu
- $d_i$  – wartość próbki pobranej z  $i$ -tego przykładu podczas treningu

Procedura minimalizacji błędu określonego zależnością 6-50 jest identyczna jak poprzednio. Pozwala ona na automatyczne utworzenie bazy wiedzy o sygnale wewnątrz sieci. Zatem można oczekiwać, że dokładność estymacji sygnału będzie coraz wyższa wraz każdym krokiem treningu.

Kolejnym problemem, który należy rozwiązać, jest właściwy dobór przykładów do treningu. Zadanie to można uprościć, biorąc pod uwagę właściwości predykcyjne sieci neuronowej. Ponieważ sieć ta służy jako nieliniowy predyktor, możliwe jest uzyskiwanie na jej wyjściu pewnej miary błędu predykcji. Sygnał błędu predykcji jest obserwowany podczas gdy pakiety niezniekształconych próbek sygnału są przetwarzane przez algorytm sieci neuronowej. Teoretycznie błąd predykcji dla sieci posiadającej pełną wiedzę o przetwarzanym sygnale powinien być zerowy. Zatem proces wyboru przykładów do treningu może być wspomagany przez obserwację wartości błędu. Wynika z tego, że w procesie treningu sieci neuronowej należy wykorzystać próbki niezniekształconego sygnału, powodujące powstawanie możliwie dużego błędu predykcji. Powtarzanie tej procedury powoduje zwykle zmniejszanie wartości błędu i jednocześnie zwiększanie dokładności estymacji sygnału.

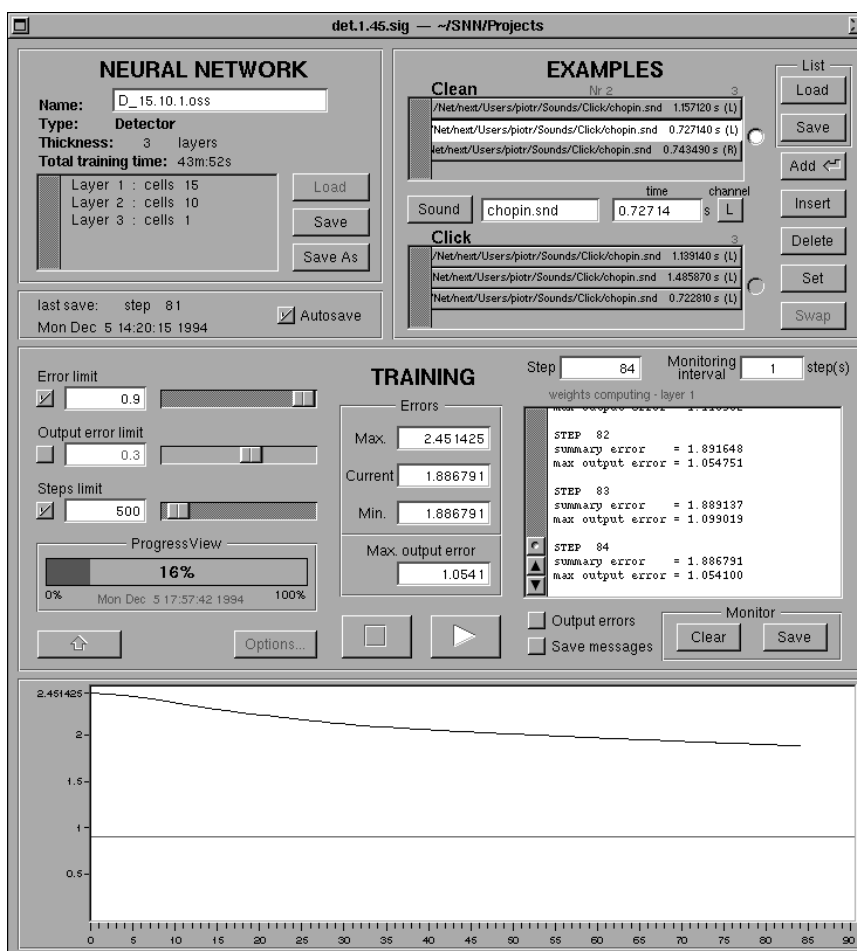
Jak wskazują wyniki uzyskane w KID PG, sieć neuronowa realizująca nieliniową predykcję wartości próbek może zostać wykorzystana do interpolacji uszkodzonych fragmentów sygnału.

#### 6.4.2.2 Implementacja programowa sieci neuronowych

Model komputerowy jednokierunkowej, w pełni połączonej sieci neuronowej został zaimplementowany w celu dokonania eksperymentów dotyczących usuwania zniekształceń impulsowych. W rezultacie

eksperymentów wykonanych przez autora okazało się, że funkcja tangens hiperboliczny (rys. 6-26) jest najbardziej efektywną funkcją kształtującą neuronów. Liczbę warstw wybierano opcjonalnie z przedziału od 3 do 5. W procesie terningu został wykorzystany zmodyfikowany algorytm wstecznej propagacji błędu, oparty na minimalizacji połowy normy błędu E względem wszystkich wag synaptycznych i poziomów neuronów, taki, jak opisany w poprzednim paragrafie.

W celu zilustrowania sposobu działania opracowanego systemu (program SigNet.app – system operacyjny UNIX/NeXTStep), na rys. 6-28 przedstawiono okno programu służące do sterowania treningiem sieci detekcyjnej i predykcyjnej. Panel *Training* (rys. 6-28) jest wykorzystywany do wczytywania i zachowywania konfiguracji sieci neuronowej. Sieć może



Rys. 6-28. Okno sterowania treningiem w programie SigNet.app

być zaprojektowana do rozpoznawania trzasków lub do rekonstruowania uszkodzonych próbek. W oknie *Examples* wylistowane są przykłady wybrane do treningu sieci. Każdy z przykładów jest przez użytkownika oznaczany jako *Clean* (nieuszkodzony) lub *Click* (trzask). Pola oznaczone jako *Error limit* (limit błędu), *Output error limit* (limit błędu na wyjściu) i *Step limit* (limit liczby kroków) umożliwiają zmianę parametrów związanych z treningiem sieci. Pole *Progress view* (postępy obliczeń) pokazuje stopień zaawansowania procesu treningu na podstawie analizy błędu na wyjściu sieci neuronowej. W środkowej części panelu znajdują się pola umożliwiające obserwowanie aktualnego stanu błędu. W prawym dolnym rogu okna wyświetlany jest rejestr działania programu, zawierający informacje o stopniu zaawansowania fazy treningu. W dolnej części okna prezentowany jest wykres wartości błędu w funkcji numeru kroku iteracji. Do sterowania procesem rekonstrukcji sygnału służy osobne okno programu.

### 6.4.3 Eksperymenty z wykorzystaniem sieci neuronowych

#### 6.4.3.1 Trening sieci neuronowych

W niniejszym paragrafie zostaną opisane eksperymenty związane z treningiem sieci neuronowych, których wyniki umożliwiły uzyskanie pożądaných właściwości detekcyjnych i predykcyjnych algorytmów neuronowych. Opis tych eksperymentów został zamieszczony w tej pracy z tego względu, że pozwoli Czytelnikowi zorientować się w sposobie prowadzenia treningu sieci neuronowych w ich zastosowaniach do przetwarzania sygnałów.

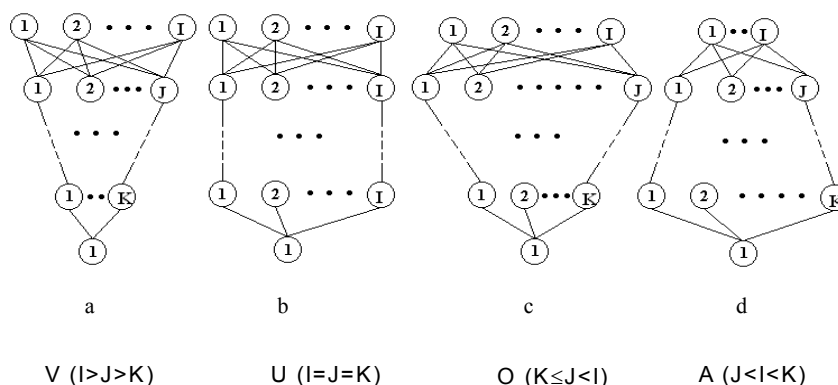
Wzorce użyte do treningu zostały uzyskane ze starych nagrań gramofonowych. W innych eksperymentach badano również sygnały zniekształcone przez pasożytnicze impulsy podczas ich transmisji w różnego rodzaju kanałach telekomunikacyjnych. Niezbędne było wykorzystanie zarówno przykładów dźwięków bez zniekształceń, jak i przykładów niepożądanych zakłóceń impulsowych. Fragmenty nieuszkodzonego dźwięku pozwalają na nauczanie sieci unikania błędnych klasyfikacji w przypadku przetwarzania pożądaných sygnałów muzycznych o charakterze impulsowym.

W jednym z eksperymentów wykorzystano nagranie ze starej płyty gramofonowej. Jakość dźwięku odczytywanego z płyty była ogólnie dobra, za wyjątkiem obecności silnych trzasków, powstałych w wyniku częstego jej



używania. Fragmenty użyte do treningu zostały spróbkowane z częstotliwością 44,1 kHz i z rozdzielczością 16 bitów (mono).

Pierwszym zadaniem było zbadanie szybkości i efektywności treningu różnych struktur sieci neuronowej. Zdefiniowano struktury sieci składające się z różnej liczby neuronów w warstwie wejściowej oraz różniące się liczbą warstw ukrytych. Badane struktury przedstawiono na rys. 6-29. Wszystkie struktury były trenowane przy użyciu tego samego zestawu przykładów dźwiękowych. Wykresy na rys. 6-30 i 6-31 przedstawiają postępy treningu dla tych struktur.

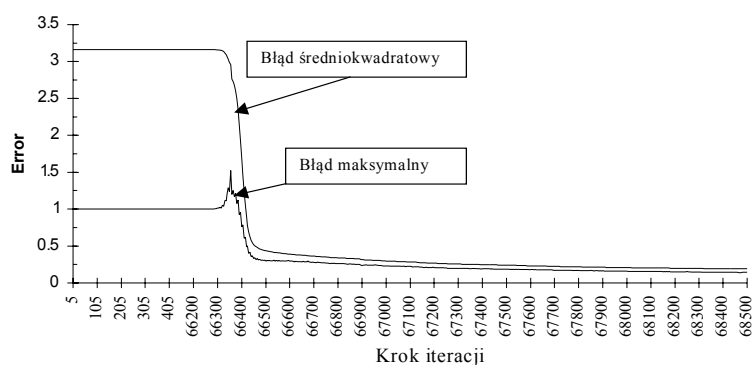


Rys. 6-29. Struktury badanych sieci neuronowych

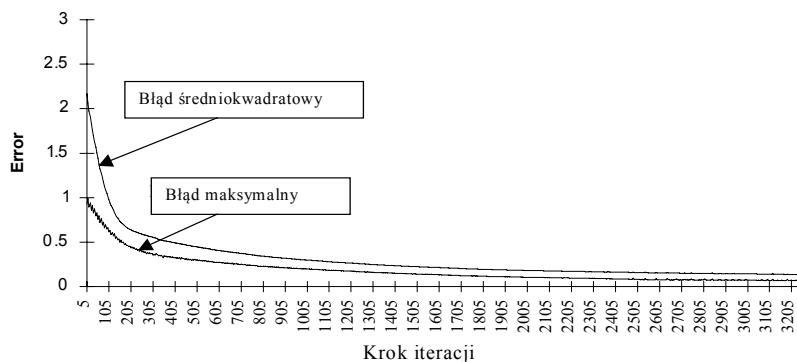
Struktura typu A często nie konwergowała w sposób właściwy. Podczas treningu błąd wyjściowy pozostawał duży lub nawet narastał. Jak wynika z wykresów na rys. 6-31, w procesie zmniejszania się wartości błędu wyjściowego można wyróżnić następujące fazy:

- *Faza 1:* Duża wartość błędu (bliska 1). Liczba iteracji niezbędnych do zakończenia tej fazy może być różna dla różnych struktur sieci (od 2 do około 66000 kroków iteracji).
- *Faza 2:* Szybkie zmniejszanie się błędu (do wartości ok. 0,5) Faza ta trwa dość krótko (200 – 300 kroków iteracji).
- *Faza 3:* Kolejne, wolniejsze zmniejszanie się błędu. W tej fazie wykres funkcji błędu może być dość złożony, ponieważ błąd może maleć w sposób nieregularny.
- *Faza 4:* Powolne zmniejszanie się błędu. Wykres funkcji błędu dąży asymptotycznie do pewnej wartości. W tej fazie mogą pojawić się lokalne fluktuacje, widoczne w postaci „impulsów” na niektórych wykresach funkcji błędu.

Podczas eksperymentów poddano treningowi kilka różnych struktur sieci neuronowych. Badane struktury zestawiono w Tab. 6-3, w której symbolicznie zapisano ich budowę. Np. d5v2 oznacza detektor o strukturze typu V, posiadający 5 neuronów w warstwie wejściowej oraz 2 warstwy ukryte. W tabelicy tej pokazano również liczbę neuronów we wszystkich warstwach dla badanych struktur. Zadanie polegało na wyborze tych struktur, które konwergują najbardziej efektywnie. Eksperymenty przeprowadzono w sposób systematyczny. Początkowo użyto 10 fragmentów zawierających trzaski i nieuszkodzone odcinki pięciosekundowych fragmentów muzycznych. Uzyskane wykresy funkcji błęd pokazano rys. 6-32 w formie pozwalającej na bezpośrednie porównanie różnych struktur sieci neuronowych, mających takie same liczby wejść i warstw ukrytych.



**Rys. 6-30.** Postępy treningu przykładowej sieci neuronowej:  
Struktura typu U- 21 neuronów



**Rys. 6-31.** Postępy treningu przykładowej sieci neuronowej:  
Struktura typu A- 27 neuronów

Tab. 6-3. Badane struktury sieci neuronowych

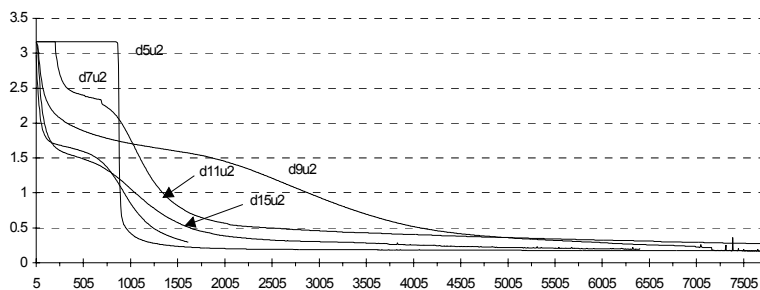
5 wejść	7 wejść	9 wejść	11 wejść	15 wejść	21 wejść
d5v2 5/4/2/1	d7v2 7/5/3/1	d9v2 9/7/5/1	d11v2 11/8/5/1	d15v2 15/11/7/1	d21v2 21/17/11/1
d5u2 5/5/5/1	d7u2 7/7/7/1	d9u2 9/9/9/1	d11u2 11/11/11/1	d15u2 15/15/15/1	
d5o2 5/7/5/1	d7o2 7/10/7/1	d9o2 9/12/9/1	d11o2 11/15/11/1	d15o2 15/18/15/1	
d5a2 5/6/7/1	d7a2 7/9/11/1	d9a2 9/11/13/1	d11a2 11/13/15/1	d15a2 15/17/19/1	
d5v3 5/4/3/2/1	d7v3 7/5/4/3/1	d9v3 9/7/5/3/1	d11v3 11/9/7/5/1		
d5u3 5/5/5/5/1	d7u3 7/7/7/7/1	d9u3 9/9/9/9/1	d11u3 11/11/11/11/1		
d5o3 5/9/7/5/1	d7o3 7/11/9/7/1	d9o3 9/15/12/9/1	d11o3 11/17/14/11/1		
d5a3 5/7/9/11/1	d7a3 7/9/11/13/1	d9a3 9/11/13/15/ 1	d11a3 11/13/15/17/1		

Na podstawie dokładnej analizy uzyskanych wyników można wywnioskować, że struktura typu V może być trenowana w sposób najbardziej efektywny. Jednocześnie struktura posiadająca dwie warstwy ukryte jest wystarczająco złożona, aby mogła wykrywać trzaski. Kompromis pomiędzy szybkością zmniejszania się błędu a czasem potrzebnym na wykonanie każdej iteracji prowadzi do wyboru struktury d15v2 jako jednej z najlepiej nadających się do wykrywania trzasków w materiale dźwiękowym.

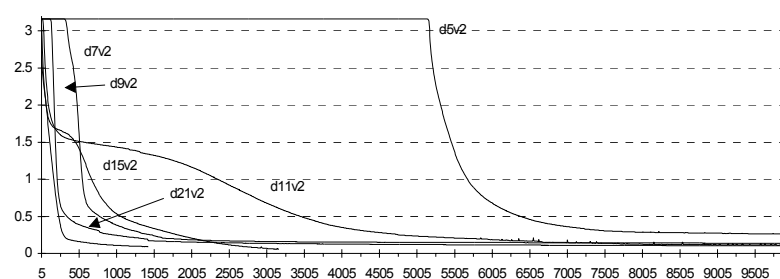
Dodatkowo sprawdzono w jaki sposób liczba fragmentów wykorzystanych podczas treningu wpływa na szybkość zmniejszania się błędu. Jak wynika z Tab. 6-4, w niektórych eksperymentach użyto 10, 20 i 40 fragmentów dźwiękowych. Dane pokazują, w jaki sposób zbyt mały lub zbyt duży zbiór fragmentów użytych do treningu może wydłużyć jego czas. Stwierdzono również, że zbyt mała liczba tych fragmentów może spowodować niemonotoniczne malenie błędu, natomiast w przypadku zbyt dużej ich liczby, sieć może nie być w stanie konwergować w możliwym do zaakceptowania czasie. Z rys. 6-33 wynika, że czas obliczeń przy użyciu pojedynczej stacji roboczej UNIX jest w ogólnym przypadku bardzo duży. Proces można przyspieszyć przy użyciu komputerów połączonych w sieć. Opracowane oprogramowanie w pełni umożliwia stosowanie takiej metody równoległego przetwarzania danych.

Po zakończeniu procesu treningu sieć neuronowa może być użyta do automatycznego wykrywania impulsów. W każdej minucie nagrania wykrywa się typowo kilka setek zniszczonych bloków. Proces wykrywania

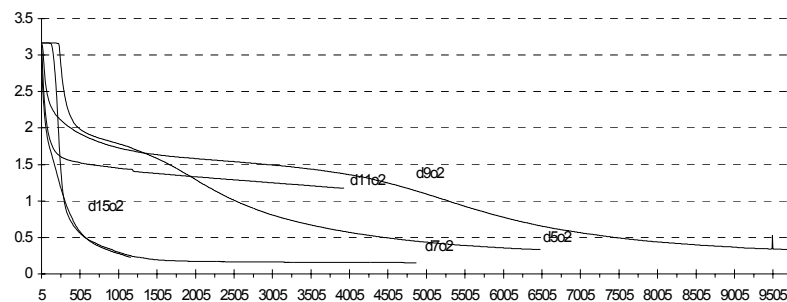
(a)



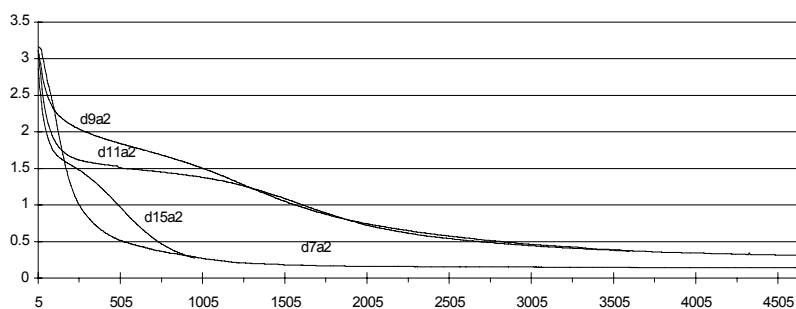
(b)



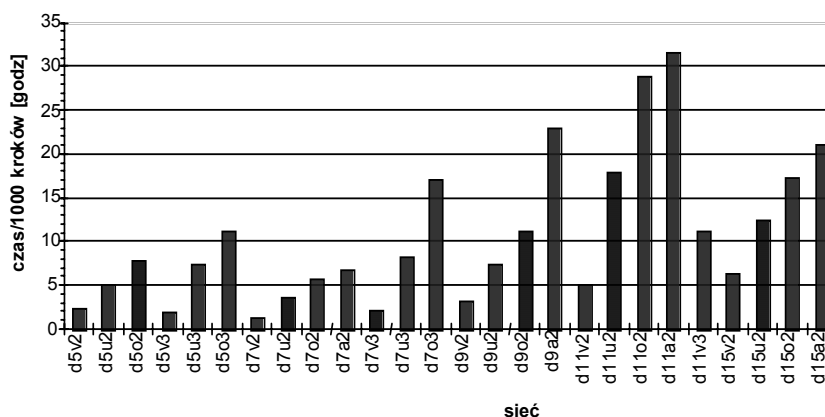
(c)



(d)



**Rys. 6-32.** Postępy treningu wybranych struktur neuronowych – na osiach rzędnych: wartości błęd średniokwadratowego; na osiach odciętych: liczba iteracji; (a) struktura typu V, (b) struktura typu U, (c) struktura typu O, (d) struktura typu A



Rys. 6-33. Porównanie czasu potrzebnego na wykonanie 1000 iteracji

impulsów jest wielokrotnie szybszy niż proces treningu. Jednakże konstruowanie tablicy uszkodzonych przedziałów nie jest realizowalne w czasie rzeczywistym na typowej stacji roboczej – może wymagać czasu kilkakrotnie dłuższego od czasu trwania oryginalnego nagrania.

Tab. 6-4. Wartości błędu dla różnej liczby wzorców użytych do treningu dla struktury V obejmującej 15 wejść, 2 warstwy ukryte

Liczba wzorców	Błąd średnio-kwadratowy	Maksymalny błąd wyjściowy	Czas obliczeń
10	0,25	0,24376	4 h 24 min
20	0,25	0,21217	3 h 47 min
40	0,9474	0,89450	10 h 00 min

#### 6.4.3.2 Odtwarzanie uszkodzonych przedziałów

Procedury podobne do opisanych wcześniej zostały przeprowadzone w przypadku drugiego algorytmu – predykcyjnej sieci neuronowej. Na podstawie badań podobnych do poprzednich, empirycznie określono stosowną konfigurację takiej sieci: 60.50.40.30.1 (60 neuronów w warstwie wejściowej, 3 warstwy ukryte zawierające odpowiednio 50, 40 i 30 neuronów oraz 1 neuron w warstwie wyjściowej).

Algorytm predykcyjny był trenowany przy użyciu wyłącznie wzorców czystego dźwięku (które nie uległy uszkodzeniu przez pasożytnicze impulsy). Koncepcja algorytmu była identyczna jak w przypadku neuropredyktora. Sieć nauczono przewidywać wartość kolejnej próbki na podstawie wartości 60 poprzednich próbek obecnych na jej wejściach. Algorytm zaprojektowano w taki sposób, aby każdy przykład był przetwarzany zarówno w kierunku kolejnych wejść sieci („w przód”), jak i w kierunku przeciwnym („wstecz”), tak aby sieć mogła nauczyć się przewidywać wartość brakującej próbki na podstawie wartości próbek sąsiednich. Teoretycznie sieć powinna przetworzyć cały nieuszkodzony materiał dźwiękowy, w celu utworzenia bazy wiedzy o wszystkich dostępnych sygnałach. Jednakże takie zadanie byłoby praktycznie niemożliwe do realizacji, a nawet niepożądane, ponieważ „przetrenowana” sieć może utracić zdolności generalizacyjne. W związku z tym program został opracowany w taki sposób, aby materiał wykorzystywany do treningu był automatycznie tworzony z próbek znajdujących się w sąsiedztwie wykrytego wcześniej, uszkodzonego przedziału. Pozwala to przyspieszyć proces treningu i zapewnia, że sieć nabywa wiedzę o fragmentach sygnału, mających podobny charakter do utraconych fragmentów, które mają być zrekonstruowane na podstawie predykcji. Oczywiście materiał wykorzystywany do treningu algorytmu predykcyjnego nie powinien zawierać pasożytniczych impulsów. Wymaga to w pewnych przypadkach automatycznego „sklejania” utraconych przedziałów. Procedura ta jest wykonywana zawsze, gdy na granicach utraconego przedziału występuje mniej niż 60 nieuszkodzonych próbek (w przypadku predykcyjnej sieci neuronowej o 60 wejściach). Zatem ciągi uszkodzonych przedziałów mogą być rekonstruowane poprzez łączenie ich w jeden dłuższy blok i przetwarzanie całego bloku.

Algorytm rekonstrukcji jest wykonywany w taki sposób, że zniekształcenia impulsowe są automatycznie wykrywane przez pierwszą sieć neuronową, tworzona jest tablica uszkodzonych przedziałów, wykonywana jest procedura sklejanie serii krótkich przedziałów, a następnie algorytm drugiej sieci neuronowej „przesuwa się” przez próbki w sąsiedztwie uszkodzonych przedziałów, wytwarzając strumień kolejnych i poprzednich próbek o wartościach uzyskanych w wyniku predykcji. Kolejnym krokiem jest dokonanie przemiksowania uzyskanych w ten sposób estymat.

#### *6.4.3.3 Łączenie zrekonstruowanych przedziałów*

Niezależnie od metod użytych do interpolacji sygnału, konieczne jest wykorzystanie technik przemiksowania, które mogą pomóc w poprawie subiektywnej jakości interpolowanego sygnału. W pierwszym podejściu

została zastosowana metoda liniowego sumowania ważonego, przebiegającego zgodnie z regułą:

$$z_n = \frac{(k-n)x_n + (n-j)y_n}{k-j} \quad (6-52)$$

gdzie:

$\langle j, k \rangle$  – przedział do rekonstrukcji

$x_n$  – ciąg próbek estymowanych „w przód”,  $n = j, j + 1, \dots, k$

$y_n$  – ciąg próbek estymowanych „wstecz”

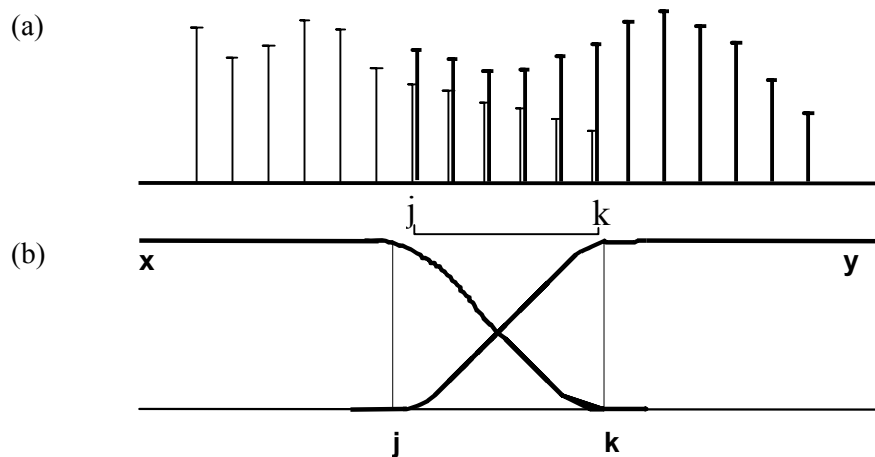
$z_n$  –wynikowy ciąg próbek

Liniowe sumowanie ważne ciągów estymowanych próbek nie zapewnia ciągłości pierwszej pochodnej sygnału. Możliwy jest do stosowania zatem również inny algorytm sumowania, zgodnie ze wzorem:

$$z_n = 0.5 \left\{ \left[ 1 + \cos\left(\frac{n-j}{k-j}\pi\right) \right] x_n + \left[ 1 - \cos\left(\frac{n-j}{k-j}\pi\right) \right] y_n \right\} \quad (6-53)$$

gdzie:  $n = j, j + 1, j+2, \dots, k$  (wykres tej funkcji przedstawiono na rys. 6-34).

Trzecia procedura przemiksowania ciągów próbek, którą można nazwać antysymetrycznym sklejeniem, jest wykonywana zgodnie z zależnością:



**Rys. 6-34.** Ilustracja procedury przemiksowania, łączącej próbki ekstrapolowane w przód i wstecz: (a) przemiksowane próbki, (b) nieliniowe sumowanie ważne na podstawie równania (6-53)

$$\begin{aligned}
 x_n &= 2x_{j-1} - x_{2j-n-2} \\
 y_n &= 2x_{k+1} - x_{2k-n+2} \\
 n &= j, j+1, \dots, k
 \end{aligned}
 \tag{6-54}$$

Nieformalne testy odsłuchowe wykazały, że procedura przemiksowania według równania (6-53) jest lepsza niż procedury przeprowadzane zgodnie z równaniami (6-52) i (6-54).

#### 6.4.3.4 Wyniki

Wykrywanie zakłóconych przedziałów zostało przeprowadzone przy użyciu przedstawionego wcześniej algorytmu sieci neuronowej. Jak widać z rys. 6-35a-c, prawidłowo wytrenowany detektor neuronowy jest w stanie odróżniać normalne dźwięki o charakterze impulsowym od pasożytniczych impulsów, podczas gdy procedura detekcji progowej, oparta na transformacji falkowej, wykrywa impulsy niezależnie od ich pochodzenia, a zatem nieprawidłowo uznaje niektóre normalne sygnały impulsowe za zakłócenia, które należy usunąć. Dla przykładu można rozpatrzeć środkowy fragment sygnału z rys. 6-35a, przedstawiający silny stan transjentowy. Ponieważ sieć neuronowa została wytrenowana przy użyciu wzorców muzycznych zawierających takie stany transjentowe, sieć nie klasyfikuje tego fragmentu jako pasożytniczego impulsu (rys. 6-35c), podczas gdy algorytm progowy (rys. 6-35b) reaguje na każdy widoczny sygnał impulsowy. Ponadto detektor neuronowy jest w stanie odnajdywać niektóre impulsy ukryte pod obwiednią sygnału, które nie są wykrywane przez algorytm progowy.

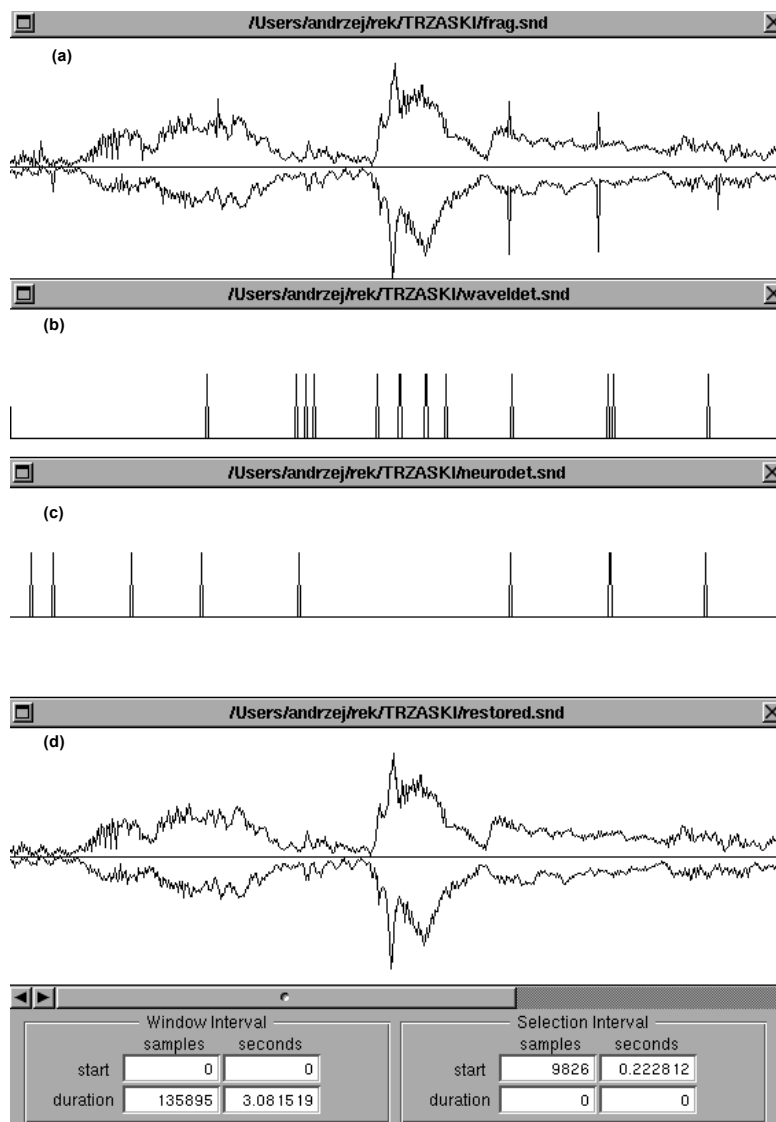
W celu dokonania oceny subiektywnej jakości dźwięku poddanego przetwarzaniu z wykorzystaniem różnych algorytmów, dokonano rekonstrukcji sygnału na podstawie następujących metod:

- *Test 1*: Interpolacja zerowego rzędu
- *Test 2*: Interpolacja drugiego rzędu
- *Test 3*: Interpolacja z wykorzystaniem liniowej predykcji „w przód” i „wstecz” 100-go rzędu, z wykorzystaniem nieliniowej procedury przemiksowania (równanie 6-53)
- *Test 4*: Interpolacja w oparciu o falkową procedurę progową i wypełnianie przerw kopiami sąsiednich fragmentów
- *Test 5*: Interpolacja oparta o algorytm predykcyjnej sieci neuronowej z nieliniowym przemiksowaniem estymet lewostronnych i prawostronnych (równanie 6-53)

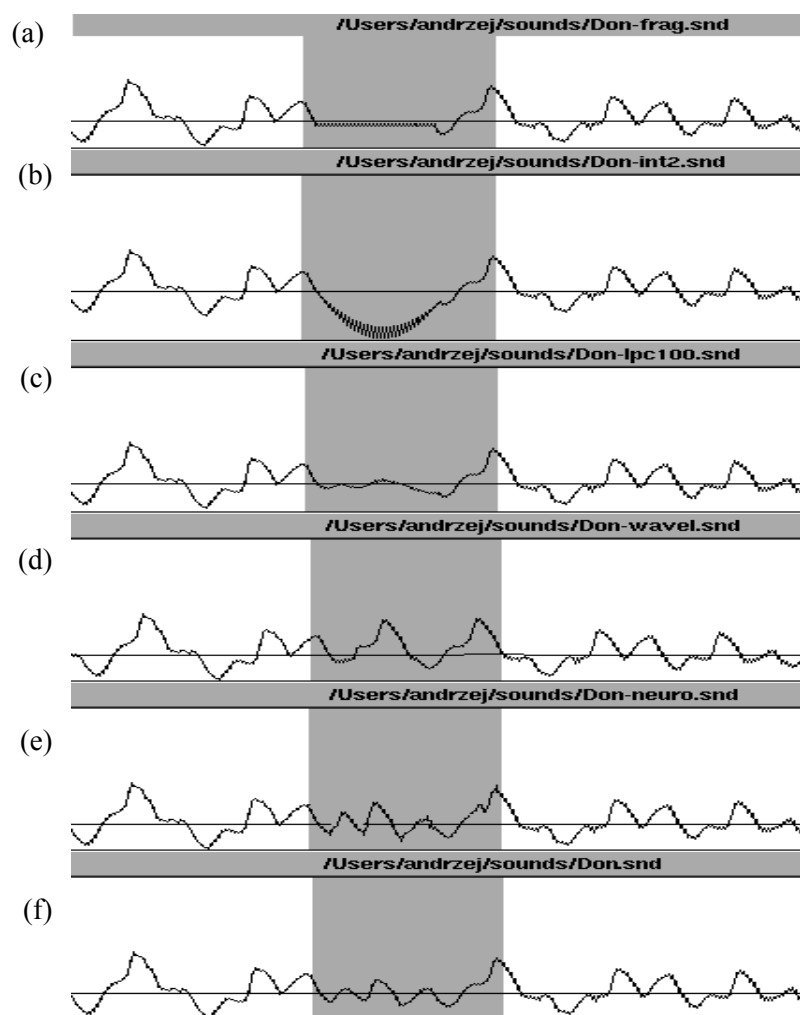
Wybrane wyniki przetwarzania sygnału za pomocą tych metod przedstawiono na rys. 6-35 d oraz 6-36. Wynik jak na rys. 6-35 d uzyskano przy pomocy algorytmu neuropredyktora z przemiksowaniem estymat



(równanie 6-53). W celu porównania efektywności różnych metod interpolacji, w sygnale pokazanym na rys. 6-36 f celowo utworzono przerwy (fragment z rys. 6-36 a). Następnie wartości próbek w tej przerwie były interpolowane przy użyciu różnych technik, jak pokazano na rys. 6-36 b–e.



Rys. 6-35. Wyniki detekcji i interpolacji impulsów: (a) sygnał zniekształcony przez trzaski, (b) trzaski wykryte przez falkową procedurę progową, (c) trzaski wykryte przez neuropredyktor, (d) sygnał zrekonstruowany przez algorytm neuropredyktora



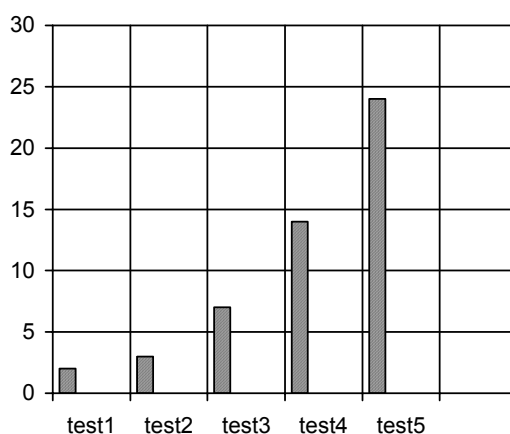
**Rys. 6-36.** Wyniki interpolacji brakujących próbek przy użyciu różnych metod: (a) celowo wprowadzone przerwy w sygnale (interpolowane przy pomocy interpolacji liniowej zerowego rzędu), (b) interpolacja drugiego rzędu, (c) interpolacja z liniową predykcją 100-go rzędu w przód i wstecz, z wykorzystaniem przemiksowania, (d) wykrywanie uszkodzonych przedziałów za pomocą procedury opartej na transformacji falkowej oraz wynik automatycznego wypełniania przerw kopiami sąsiednich fragmentów, (e) wynik interpolacji z zastosowaniem neuropredyktora, (f) sygnał oryginalny

W celu uzyskania subiektywnej oceny wyników przeprowadzono serię testów porównań parami przy pomocy pięciu słuchaczy, wybranych spośród pracowników KID PG. Wzorce dźwiękowe zostały przygotowane zgodnie z zasadami testów porównań parami. Pary porównywanych sygnałów utworzono na podstawie pięciu przetworzonych wersji, tego

samego 10-sekundowego fragmentu nagrania. W celu porównania pięciu fragmentów dźwiękowych należało utworzyć 10 niezależnych par. Całkowita liczba wyników testów jest równa  $10 \text{ par} \times 5 \text{ słuchaczy} = 50$ . W celu opracowania wyników testów napisano specjalny program komputerowy, pozwalający na sprawdzenie wiarygodności wyników testów subiektywnych przy użyciu testu  $\chi^2$  (testu Pearsona). Uzyskane wyniki udowadniają wyższość proponowanej metody predykcji neuronowej nad innymi metodami interpolacji. Wyniki testów subiektywnych pokazano na rys. 6-37.

Wyniki badań pokazują, że sieci neuronowe są właściwym narzędziem zarówno do wykrywania zakłóceń impulsowych, jak i do ich usuwania przy użyciu nieliniowej predykcji próbek w uszkodzonych przedziałach. Uczący się algorytm detekcji jest w stanie odróżnić niepożądane zakłócenia impulsowe od pożądaných dźwięków o charakterze perkusyjnym i od stanów transjentowych. Algorytm uczący się jest przydatny również do rekonstrukcji brakujących próbek podczas przetwarzania materiału dźwiękowego. Niestety, baza wiedzy o sygnale i zakłóceniach impulsowych, nabyta przez sieć neuronową w procesie treningu, nie nadaje się do uniwersalnego stosowania. Zatem proces tworzenia bazy wiedzy w sieci neuronowej musi być powtarzany, jeżeli charakter materiału dźwiękowego lub zakłóceń impulsowych ulega zmianie.

Problem długiego czasu trwania procesu treningu sieci neuronowych, który ogranicza możliwość zastosowania ich do celów praktycznych, nie dotyczy innych algorytmów, należących do dziedziny sztucznej inteligencji. Alternatywna metoda tworzenia bazy wiedzy o sygnale i zniekształceniach zostanie przedstawiona w kolejnym podrozdziale.



**Rys. 6-37.** Wyniki testów porównań parami -  
- liczba głosów oddanych przez ekspertów na poszczególne metody interpolacji

## 6.5 Usuwanie szumu impulsowego

### 6.5.1. Przetwarzanie sygnału przy użyciu metody zbiorów przybliżonych

Rozwiązania opisane w niniejszym paragrafie stanowią ilustrację sposobu tworzenia systemu eksperckiego do celu cyfrowego przetwarzania sygnałów fonicznych. W dziedzinie tej opracowano dotychczas niewiele praktycznie działających systemów, zaś wszystkie opracowane aplikacje mają charakter badawczo-eksperymentalny. Tym niemniej, z uwagi na dobre wyniki działania systemów eksperckich i ich pewne szczególne zalety, o których będzie mowa w tym rozdziale, można oczekiwać szerszego upowszechnienia się tego typu metod w zastosowaniach do cyfrowego przetwarzania danych i sygnałów fonicznych.

Często występujący w nagraniach gramofonowych i w transmisji sygnałów fonicznych szum impulsowy (ang. *hiss*) jest spowodowany obecnością dużej liczby pasożytniczych impulsów o małych szerokościach. W celu usunięcia tego typu szumu stosuje się zwykle modele predykcyjne lub algorytmy odejmowania widmowego (patrz roz. 4). Ponieważ gęstość takich impulsów jest bardzo duża, nie jest celowe wyodrębnianie pojedynczych impulsów, usuwanie ich, a następnie dokonywanie interpolacji brakujących próbek. Usuwanie tego typu szumu wymaga odmiennego podejścia, ponieważ opisany w poprzednim podrozdziale proces treningu sieci neuronowej jest długotrwały, a w przypadku wysokiej gęstości pasożytniczych impulsów na jednostkę czasu wymagane jest zebranie większej ilości informacji o zakłóceniach impulsowych. Z tego względu autor zdecydował się na opracowanie koncepcji systemu eksperckiego, opartego na metodzie zbiorów przybliżonych. Metoda ta stanowi system decyzyjny oparty na regułach, umożliwiający szybką budowę bazy wiedzy, która jest następnie używana w celu dokonania separacji pożądaných składowych widmowych od zakłóceń, z uwzględnieniem subiektywnej oceny jakości wynikowego sygnału fonicznego przez eksperta. Metoda zbiorów przybliżonych została wybrana jako główna część systemu eksperckiego z uwagi na jej wyjątkową efektywność generalizacji wiedzy i adaptacji. Każdy z algorytmów estymacji sygnałów w szumie musi radzić sobie z problemem niepewnej informacji. Typowe algorytmy tego typu, których wiele zostało opisanych w literaturze, są oparte na statystycznym opisie sygnału i szumu. Tymczasem baza wiedzy systemu eksperckiego opartego na regułach jest systematycznie powiększana o wiedzę nabytą podczas uczenia się na podstawie nowych przykładów. Procedura treningu algorytmu opartego na zbiorach przybliżonych jest nadzorowana przez

operatora (eksperta) w taki sposób, że tworzony jest model sygnału o jakości zgodnej z subiektywnymi odczuciami eksperta.

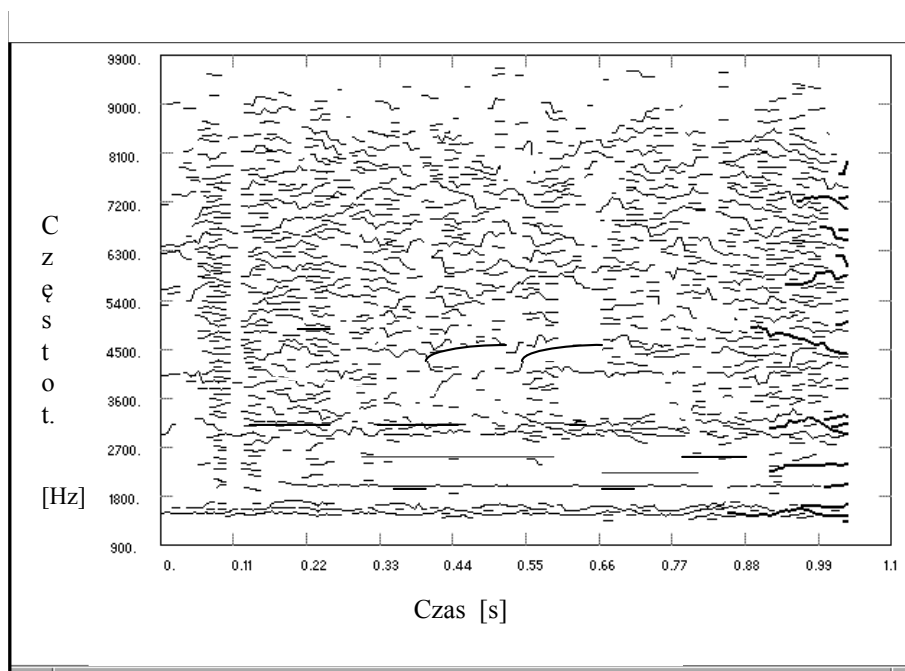
#### *6.5.1.1. Model analizy–syntezy*

W eksperymentach został wykorzystany algorytm analizy–resyntezy wzorowany na modelu McAulay’a–Quatieriego (patrz par. 2.3.3.4). Algorytm ten umożliwia dokonywanie syntezy sygnału na podstawie skończonej liczby składowych sinusoidalnych, reprezentowanych przez sumę określoną wzorem (2-71).

W algorytmie, który był implementowany w KID PG, sygnał wejściowy jest dzielony na pakiety o długości 256 próbek. Powstałe w ten sposób bloki próbek są następnie nakładkowane (długość zakładki wynosi 32 próbki). W kolejnych krokach bloki sygnału są poddawane okienkowaniu (wykorzystano okno Hamminga), po czym obliczana jest dyskretna transformata Fouriera (DFT). W wyniku obliczenia DFT uzyskuje się 256 liczb zespolonych, które są przechowywane w pamięci RAM, w formie dwóch tablic liczb zmiennoprzecinkowych, zawierających dane odnoszące się do widma amplitudowego i widma fazowego. Kolejnym zadaniem jest znalezienie maksimów obwiedni widma amplitudowego sygnału, które są związane ze składowymi sinusoidalnymi analizowanego fragmentu. Zastosowany model dopuszcza, aby w wynikowym widmie mogły być zawarte składowe nieharmoniczne. Algorytm wykrywania maksimów jest powtarzany dla kolejnych pakietów, a jeżeli maksima o tej samej częstotliwości i amplitudzie pojawiają się w kolejnych pakietach, są one dopasowywane do siebie. W związku z tym analiza McAulay’a–Quatieriego umożliwia uzyskanie charakterystycznych wykresów, na których są widoczne ścieżki łączące sąsiednie maksima widma. Wykres pokazany na rys. 6-38 przedstawia wynik analizy McAulay’a–Quatieriego fragmentu sygnału muzycznego, zniekształconego przez silny szum impulsowy. Jak można zauważyć, zakłócenia impulsowe są widoczne na wykresie w postaci krótkich odcinków, podczas gdy składowe sygnału są reprezentowane przez dłuższe i bardziej wyraźne linie, odpowiadające czasom rzędu wielu milisekund.

W rekonstruowanym dźwięku występują pewne zakłócenia (pochodzące od szumu), które nie są zgodne z modelem McAulay’a–Quatieriego, który odnosi się sygnału użytecznego. Zatem pierwszym problemem, jaki musi zostać rozwiązany w toku analizy, jest wykrycie pożądaných składowych, aby zakwalifikować je do wykorzystania w późniejszej resyntezie sygnału. Aby to osiągnąć, reprezentacja widmowa sygnału musi zostać odfiltrowana przy użyciu pewnych operacji progowych,

które zostaną opisane w dalszej części rozdziału. Następnie sygnał może zostać odtworzony przy pomocy syntezy addytywnej, opartej na odwrotnej dyskretnej transformacji Fouriera.



**Rys. 6-38.** Wykres na płaszczyźnie czas-częstotliwość, obrazujący wyniki analizy McAdulay'a-Quatieriego przykładowego sygnału muzycznego

### 6.5.1.2 Operacje progowe

Operacje progowe umożliwiają wydzielenie tych składowych zaszumionego sygnału dźwiękowego, które są zgodne z zastosowanym modelem sygnału. Zakłada się przy tym, że składowe poniżej określonego progu reprezentują szum. Poziomy prog jest funkcją częstotliwości. Poziomy progowe amplitudy widma dla wszystkich częstotliwości powinny zmieniać się wraz z sygnałem, ponieważ konieczne jest oddzielenie szumu od pożądaných składowych niezależnie od poziomu i kształtu widma sygnału. Istnieją dwa odmienne podejścia do wyznaczania cech sygnału: obiektywne i subiektywne. Proces budowania bazy wiedzy o parametrach sygnału powinien wykorzystywać oba podejścia równocześnie. Zatem parametry opisujące fragmenty sygnału muszą zostać wyznaczone obiektywnie na podstawie przetwarzania kolejnych pakietów próbek, jak również na podstawie subiektywnej oceny eksperta, obsługującego system na etapie

treningu. Oczywiście oceny subiektywne nie dotyczą pojedynczych pakietów próbek, lecz muszą być formułowane na podstawie pewnych charakterystycznych fragmentów nagrania, przy wykorzystaniu jego krótkich odcinków, złożonych z pewnej liczby pakietów traktowanych jako wzorcowe fragmenty zrekonstruowanego sygnału.

Aby uzyskać efekt redukcji szumu, opracowano dwie procedury automatycznego wyznaczania progów odcięcia szumu przy przetwarzaniu sygnału fonicznego. Pierwszy, inteligentny algorytm progowy umożliwia „zgrubną” regulację poziomu progu odcięcia, natomiast drugi, uzupełniający algorytm nieinteligentny, kończy proces ustalania progu, umożliwiając jego dodatkowe dopasowanie do aktualnego kształtu widma.

### 6.5.1.3 Główna procedura wyznaczania maksimów w widmie

W omawianym systemie rekonstrukcji dźwięku wiedza jest zebrana w postaci reguł. Reguły te mają następującą postać:

$\langle \text{ciąg atrybutów warunkowych} \rangle \Rightarrow \langle \text{opinia eksperta} \rangle$

System ekspercki określony w ten sposób wykorzystuje niżej określone atrybuty warunkowe. Znaczenie tych atrybutów zostanie opisane w dalszej części podrozdziału.

1. Ustawienia definiowane przez operatora dla całego fragmentu sygnału:
  - c – względny poziom odcięcia składowych widmowych (filtracja w dziedzinie widma amplitudowego)
  - h – histereza rewerberacyjna (filtracja związana ze stosunkami amplitud poszczególnych składowych widma w kolejnych pakietach)
  - $f_p$  – początkowa częstotliwość preemfazy, w [Hz]
  - p – nachylenie preemfazy, w [dB/okt.]
2. Parametry obliczane przez system:
  - $m_1, m_2, m_3$  – pierwsze trzy centralne momenty widma wzorców dźwiękowych

Widmo użyte do obliczenia tych parametrów jest uśredniane po fragmencie sygnału, którego czas trwania wynosi ok. 3 s. Parametr  $m_1$  jest obliczany na podstawie następującej definicji:

$$m_1 = \frac{\sum_{i=1}^n i \cdot A_i}{\sum_{i=1}^n A_i} \quad (6-55)$$

gdzie:

$m_1$  – środek ciężkości widma

$A_i$  – amplituda  $i$ -tego prążka uśrednianego widma

$n$  – liczba prążków widma

Podobnie,

$$m_2 = \frac{\sum_{i=1}^n (i - m_1)^2 \cdot A_i}{\sum_{i=1}^n A_i} \quad (6-56)$$

gdzie:  $m_2$  jest drugim momentem centralnym (wariancją) widma, odzwierciedlającą rozkład energii widma w stosunku do jego „środka ciężkości”, zaś:

$$m_3 = \frac{\sum_{i=1}^n (i - m_1)^3 \cdot A_i}{\sum_{i=1}^n A_i} \quad (6-57)$$

gdzie:  $m_3$  jest trzecim momentem centralnym, odzwierciedlającym stopień symetrii widma w stosunku do jego środka ciężkości (przy czym dla widma o kształcie symetrycznym  $m_3 = 0$ ).

$d_1, d_2, d_3$  – parametry opisujące charakter zanikania sygnału w całym fragmencie (zwykle zanikanie jest zdeterminowane zjawiskami rewerberacji).

Szybkość zanikania jest określana na podstawie zmian poziomu w niektórych podpasmach widma częstotliwości słyszalnych. Może być ona wyrażona za pomocą zależności:



$$d_r = \frac{1}{j} \cdot \sum_{i=1}^j (L_{r,i} - L_{r,i-1}) \quad (6-58)$$

gdzie:

- $L_{r,i}$  – zmiana poziomu sygnału w pakiecie  $i$ , w stosunku do pakietu  $i - 1$ , w podpaśmie  $r$ . Obliczenia prowadzono w trzech podpasmach: 0 – 1 kHz; 1 – 5,5 kHz; 5,5 – 11 kHz; stąd  $r \in \{1, 2, 3\}$
- $j$  – liczba pakietów próbek (zawierających po 256 próbek) w całym 3-sekundowym fragmencie;  $j = 3 \cdot f_s / 256 \cong 260$ .

Znaczenie poszczególnych parametrów wymaga dodatkowego wyjaśnienia. Wykorzystanie histerezy w charakterystyce czasowej progu widmowego pozwala uniknąć derewerberacji sygnału. Derewerberacja może wystąpić na skutek tendencji algorytmów progowych do obcinania pogłosu za każdym razem, gdy jego poziom spada poniżej aktualnie wyznaczonego poziomu progowego. Współczynnik histerezy pozwala uniknąć derewerberacji poprzez uzmiennienie wartości parametru  $h \in (0, 1)$ . Wartość tego parametru może być zmieniana przez operatora podczas odsłuchu fragmentów dźwiękowych. Ustalenie niezerowej wartości współczynnika uzależnia poziom progu odcięcia składowych widmowych od wartości tego poziomu w poprzednim pakiecie. Mechanizm ten pozwala na zmianę poziomu progu odcięcia w zależności od charakterystyki zaniku sygnału obserwowanego w kolejnych pakietach.

Potrzeba wprowadzenia preemfazy (wzmocnienia widma amplitudowego w zakresie wysokich częstotliwości) wynika z naturalnego charakteru zaniku energii sygnałów fonicznych w górnej części pasma częstotliwości, ponieważ jeżeli nie zostanie wprowadzona właściwa preemfaza, filtracja progowa działająca w dziedzinie widma będzie prowadziła do redukcji składowych o wysokich częstotliwościach. Zgodnie z zasadą preemfazy, składowe sygnału w zakresie wysokich częstotliwości mogą być wzmacniane powyżej częstotliwości początkowej  $f_p$  (z nachyleniem widma równym  $+p$  dB na oktawę). Parametry  $f_p$  i  $p$  są zmieniane przez operatora podczas eksperymentów.

Parametry  $c, h, f_p, p, m_1, m_2, m_3, d_1, d_2, d_3$  zostają skwantowane, aby były reprezentowane przez wartości dyskretne, nie zaś ciągle. Kwantyzacja atrybutów umożliwia uniknięcie wygenerowania zbyt wielu specyficznych reguł opartych na niewielkich zmianach wartości atrybutów, ponieważ w praktyce każdy atrybut może przyjmować wartości jedynie z kilku zakresów. Zatem reprezentacja za pomocą zakresów pozwala na zastąpienie reprezentacji w postaci liczb przez kilka liczb całkowitych, reprezentujących

poszczególne zakresy. Na przykład nachylenie preemfazy  $p$  może być przedstawione za pomocą następujących zakresów:

- 0 – 3 dB na oktawę
- 3 – 6 dB na oktawę
- 6 – 12 dB na oktawę.

Pozostałe atrybuty są również poddawane kwantyzacji w taki sposób, że poszczególne parametry są reprezentowane przez zbiór kilku wartości. Podczas testów ustalono, że odpowiednia jest rozdzielczość wartości parametrów ograniczona do 3 – 5 zakresów. Nieliniowa zależność pomiędzy wartościami całkowitymi, reprezentującymi zakresy, a rzeczywistymi wartościami parametrów nie jest przeszkodą dla systemu decyzyjnego opartego na zbiorach przybliżonych, pod warunkiem, że wybrane wartości parametrów są rozróżnialne i specyficzne z uwagi na ich wpływ na decyzję eksperta. Jak wspomniano wcześniej, skwantowane wartości parametrów, wykorzystywane przez system zbiorów przybliżonych jako atrybuty, stanowią przesłanki reguł, podczas gdy atrybuty decyzyjne reprezentują opinię eksperta, wskazującą na jedną z kategorii subiektywnych, odnoszących się do jakości dźwięku. Reguły są generowane automatycznie, przy wykorzystaniu kolejnych atrybutów, jeden po drugim. Algorytm generowania reguł jest przedstawiony na rys. 6-39.

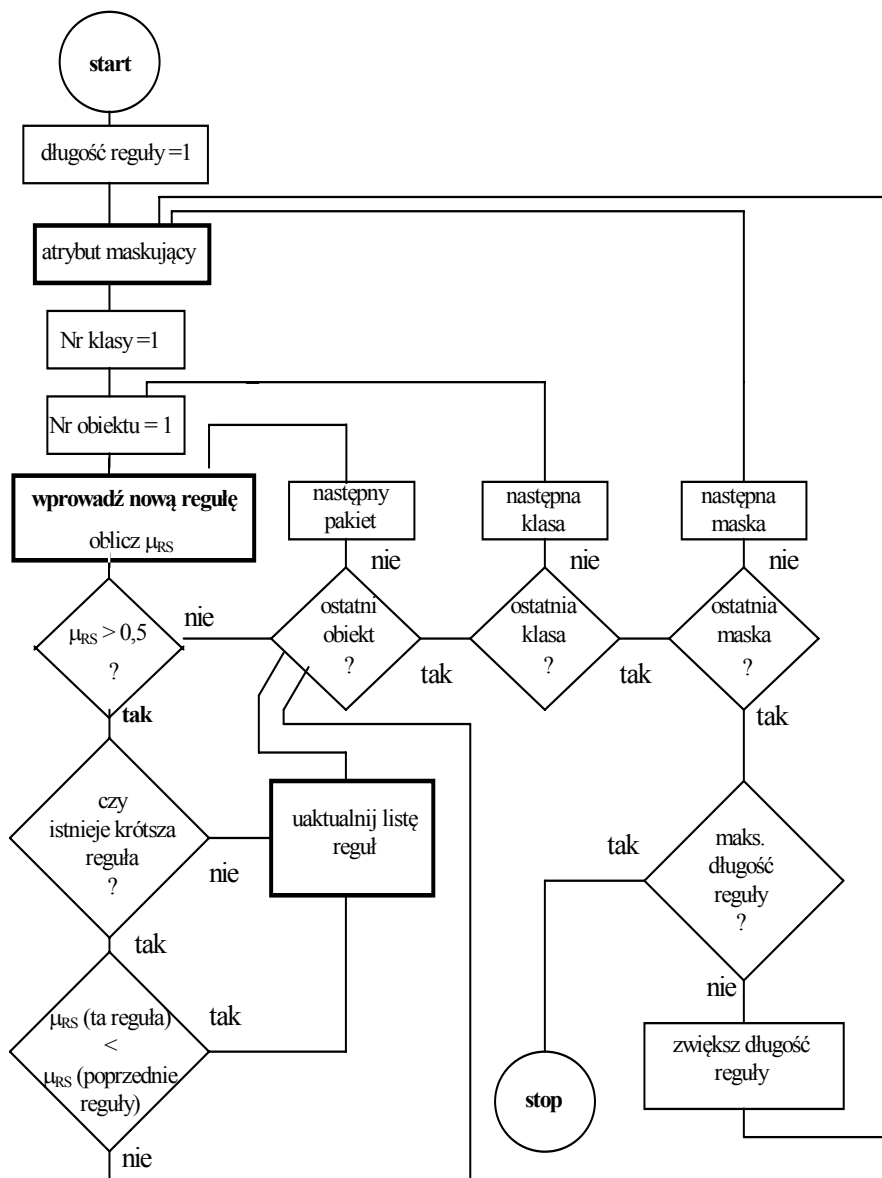
W praktyce skwantowane atrybuty są reprezentowane przez pięć liczb, wskazujących do którego z ponumerowanych zakresów należy aktualna wartość odpowiedniego parametru. Przykładowa reguła oparta na dotychczas zdefiniowanych parametrach może być zapisana następująco:

$$[c = 2] \ \& \ [h = 1] \ \& \ [f_p = 2] \ \& \ [p = 2] \ \& \ [m_1 = 1] \ \& \ [m_2 = 2] \ \& \ [m_3 = 0] \ \& \ [d_1 = 0] \ \& \ [d_2 = 1] \ \& \ [d_3 = 0] \ \Rightarrow \ [dec.=2]$$

gdzie  $[dec.=2]$  oznacza, że wzorzec dźwiękowy wstępnie przefiltrowany z progiem zdefiniowanym na podstawie przesłanek reguły został zakwalifikowany jako należący do kategorii 2. Lista kategorii subiektywnych zostanie omówiona w dalszej części podrozdziału.

Pełna lista reguł może zawierać reguły odpowiadające wszystkim przykładom podawanym przez eksperta. Gdy proces akwizycji reguł jest zakończony, należy zdefiniować koncept. Jest to normalny sposób postępowania z systemem opartym o zbiory przybliżone. W przypadku rekonstrukcji dźwięku, koncept jest związany z pozytywną opinią eksperta o jakości przetwarzanych wzorców dźwiękowych. Zatem tylko reguły związane z pozytywną oceną jakości wzorców są brane pod uwagę. Reguły wybrane w ten sposób są następnie przetwarzane. Przetwarzanie polega na zamianie parametrów  $c$ ,  $h$ ,  $f_p$ ,  $p$ , użytych poprzednio jako atrybuty warunkowe, na atrybuty decyzyjne. Po takiej transformacji bazy wiedzy powstają nowe reguły, np.:

$[m_1 = 1] \& [m_2 = 2] \& [m_3 = 0] \& [d_1 = 0] \& [d_2 = 1] \& [d_3 = 0] \Rightarrow [c = 1]$   
 $[m_1 = 1] \& [m_2 = 2] \& [m_3 = 0] \& [d_1 = 0] \& [d_2 = 1] \& [d_3 = 0] \Rightarrow [h = 2]$   
 $[m_1 = 1] \& [m_2 = 2] \& [m_3 = 0] \& [d_1 = 0] \& [d_2 = 1] \& [d_3 = 0] \Rightarrow [f_p = 2]$   
 $[m_1 = 1] \& [m_2 = 2] \& [m_3 = 0] \& [d_1 = 0] \& [d_2 = 1] \& [d_3 = 0] \Rightarrow [p = 0]$



Rys. 6-39. Uproszczony algorytm generowania reguł

Przy użyciu tych reguł mogą być podejmowane decyzje, pozwalające na automatyczną regulację ustawień systemu rekonstrukcji podczas przetwarzania całego materiału dźwiękowego (jest to w rzeczywistości cel systemu eksperckiego). Jak widać, atrybuty warunkowe reprezentują teraz skwantowane wartości obliczone na podstawie pojedynczych pakietów sygnału, natomiast atrybuty decyzyjne mogą być bezpośrednio użyte do regulacji parametrów procedury rekonstrukcji podczas przetwarzania tych pakietów. Wartości  $m_1$ ,  $m_2$ ,  $m_3$  mogą być teraz ustalone na podstawie pojedynczego pakietu, natomiast  $d_1$ ,  $d_2$ ,  $d_3$  wykorzystują dane zawarte w dwóch kolejnych pakietach.

Lista reguł jest zwykle o wiele dłuższa niż zaprezentowany przykład i zawiera praktycznie wszystkie możliwe kombinacje parametrów. Warto tu wspomnieć, że metoda zbiorów przybliżonych wykształciła wiele narzędzi, pozwalających na automatyczne zmniejszenie objętości zbioru reguł poprzez wybór reguł zawierających istotne atrybuty, czyli redukty (patrz par. 6.3.1.6). Dokładny opis tej problematyki wykracza poza zakres niniejszego rozdziału z uwagi na potrzebę ograniczenia jego objętości.

System rekonstrukcji dźwięku wykorzystujący bazę nabytej wiedzy jest w stanie przetworzyć cały sygnał dźwiękowy, pakiet po pakiecie, dokonując automatycznej regulacji parametrów wpływających na cechy sygnału. Przy wykorzystaniu tej możliwości, poziom odcięcia filtracji widmowej może być automatycznie regulowany, pozwalając na automatyczne kwalifikowanie składowych sygnału do resyntezy, jak pokazano na rys. 6-40.

Jak wynika z wcześniejszych rozważań, regulacja progu jest kontrolowana przez trzy główne atrybuty decyzyjne:

1. Poziom odcięcia filtracji widmowej  $c \in \{0, 1, 2, 3, 4\}$
2. Współczynnik  $k$  związany z preemfazą (wzmocnieniem zakresu dużych częstotliwości), zdefiniowany jako:

$$k_i = \begin{cases} 1, & i \leq \frac{m f_p}{f_s} \\ \frac{1}{(i \cdot f_s / m f_p)^{p/6}}, & i > \frac{m f_p}{f_s} \end{cases} \quad (6-59)$$

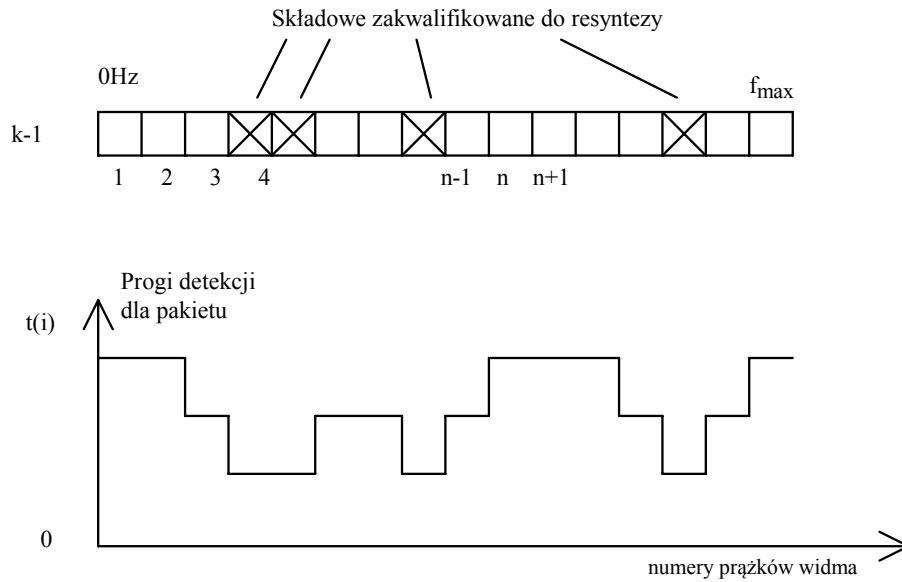
gdzie:

$f_s$  – częstotliwość próbkowania = 22,05 kHz

$m$  – liczba próbek w pojedynczym pakiecie = 256

$i$  –  $i$ -ty prążek widma związany z częstotliwością  $f_i = i \cdot f_s / m$

$f_p$  – początkowa częstotliwość preemfazy,  $f_p \in \{0, 1, 2\}$



Rys. 6-40. Próg filtracji w dziedzinie widmowej

- $p$  – nachylenie preemfazy [dB na oktawę],  $p \in \{0, 1, 2\}$
3. Współczynnik histerezy  $h \in \{0, 1, 2, 3\}$ ,  $0 \leq h < 1$ . Jeżeli współczynnik ten ma wartość niezerową, wiąże on poziom odcięcia składowych widmowych z poziomem w poprzednim pakiecie. Poziom progę odcięcia w poprzednim pakiecie  $f_d(i)$  jest zdefiniowany jako:

$$f_d(i) = \begin{cases} -h, & A'_i > t_p(i) \\ 0, & A'_i < t_p(i) \cap [A'_{i-1} > t_p(i) \cup A'_{i+1} > t_p(i)] \\ h, & A'_i < t_p(i) \cap A'_{i-1} < t_p(i) \cap A'_{i+1} < t_p(i) \end{cases} \quad (6-60)$$

gdzie:

$A'_i$  – amplituda  $i$ -tego prążka widma w poprzednim pakiecie

$t_p(i)$  – próg ustawiony w poprzednim pakiecie

Próg odcięcia  $t(i)$  jest obliczany na podstawie następującej zależności, wykorzystującej uprzednio zdefiniowane parametry:

$$t(i) = \bar{A}_i \cdot c \cdot k_i \cdot [1 + f_d(i)] \quad (6-61)$$

gdzie:

$$\bar{A}_i = \frac{1}{n_h} \cdot \sum_{i=1}^{n_h} \frac{A_i}{k_i} - \text{amplituda widmowa uśredniona dla całego pakietu}$$

próbek,

$n_h$  – indeks prążka widma na górnej granicy pasma częstotliwości,

$$n_h = m/2$$

Jak wynika z równania (6-61), aktualna wartość progu filtracji widmowej jest zdeterminowana przez średnią amplitudę widma i współczynniki regulowane przez system decyzyjny oparty na zbiorach przybliżonych. Współczynniki te (atrybuty decyzyjne powiązane ze zwycięskimi regułami) wpływają na niektóre cechy sygnału, tzn. na:

1. poziom – za pośrednictwem parametru  $c$ ,
  2. charakterystyki widmowe – za pośrednictwem parametru preemfazy  $k$ ,
  3. kształt obwiedni czasowej – za pośrednictwem parametru  $h$ .
- Bieżący zestaw składowych uzyskanych w wyniku filtracji progowej zależy natomiast od obliczonych wartości progu.

Sposób wyboru zwycięskich reguł spośród reguł zawartych w bazie jest zgodny z zasadą działania systemów decyzyjnych opartych na zbiorach przybliżonych. Proces ten omówiono w dalszej części niniejszego rozdziału.

#### 6.5.1.4 *Uzupełniająca procedura wyznaczania maksimów widma*

Opisana uprzednio procedura służy do zgrubnego określania progów filtracji. Istnieje jednak potrzeba bezpośredniego powiązania poziomu progowego z amplitudami poszczególnych prążków widmowych, ponieważ w przypadku praktycznych sygnałów dźwiękowych, stosunek sygnału do szumu jest zależny od częstotliwości. Opracowano zatem kolejną procedurę, pozwalającą na automatyczne wyszukiwanie maksimów w widmie, a ponadto zmieniającą poziom progowy jeżeli prawdopodobne jest, że zmiana progu pozwoli uniknąć odcięcia pożądanych składowych widmowych lub umożliwi dodatkową redukcję szumu bez naruszania tych składowych.

W celu znalezienia składowych nie wykrywanych przez główną procedurę, kolejna procedura jest zawsze uruchamiana automatycznie. Procedura składa się z analizy widmowej przy użyciu okna przesuwającego się po osi częstotliwości i wybierającego niektóre prążki widma (liczba prążków jest typowo równa 3). Następnie obliczane są: środek ciężkości energii okienkowanych składowych  $M_g$  oraz punkt środkowy ruchomego okna  $M_w$ , na podstawie następujących zależności:

$$M_g(k) = \frac{\sum_{n=k}^{k+l-1} n \cdot r_f \cdot E_n}{\sum_{n=k}^{k+l-1} E_n} \quad (6-62)$$

oraz:

$$M_w(k) = \frac{1}{l} \sum_{n=k}^{k+l-1} n \cdot r_f \quad (6-63)$$

gdzie:

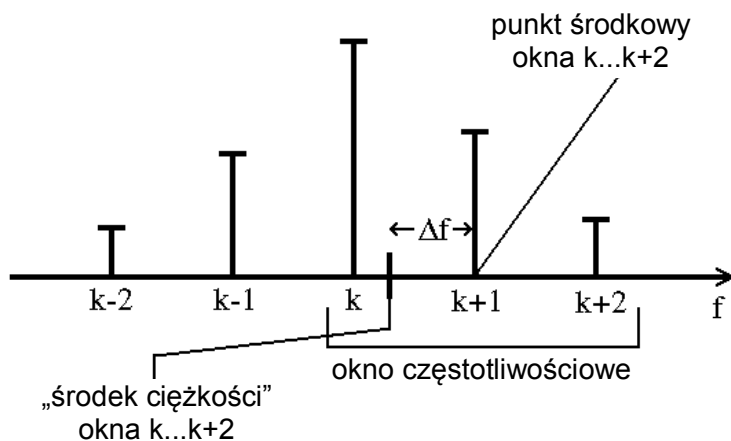
- $M_g(k)$  – środek ciężkości energii widmowej składowych widma zawartych w oknie,
- $M_w(k)$  – punkt środkowy ruchomego okna w  $k$ -tym kroku,
- $E_n$  – energia  $n$ -tego prążka widma,
- $r_f$  – rozdzielczość analizy widmowej,  
 $r_f = f_s / m = 22\,050 / 256 \cong 86$  Hz
- $l$  – długość okna częstotliwości,
- $k$  – numer kroku,
- $n$  – numer prążka widma

Obie wartości obliczane na podstawie równań (6-62) i (6-63) są wyrażone w Hz. Różnica między nimi wynosi:

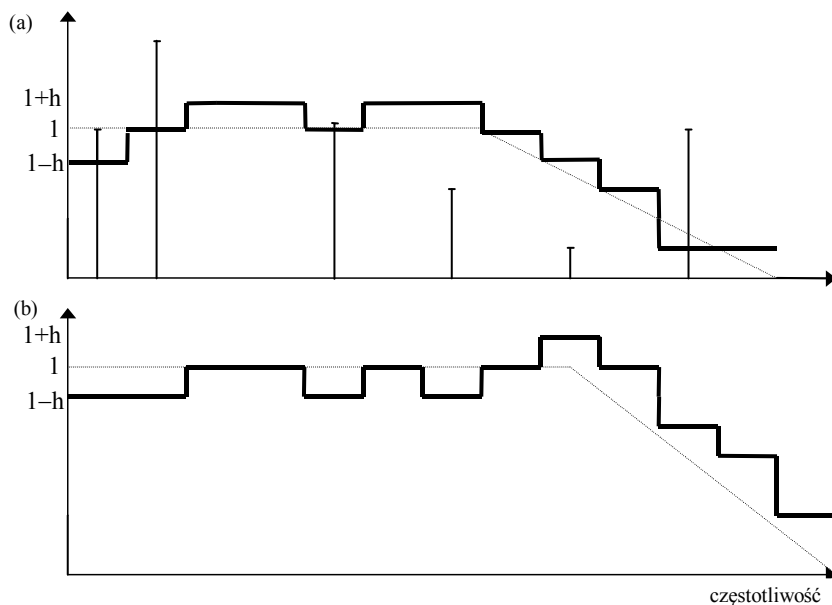
$$\Delta f = M_g - M_w$$

$i$  jest wykorzystywana do wyznaczania lokalnego maksimum reprezentacji widmowej. Równocześnie analizowana jest różnica pomiędzy kolejnymi wartościami  $\Delta f$  obliczanymi dla każdego kroku (w wyniku przesuwania okna). Składowa jest kwalifikowana do procedury resyntezy, jeżeli wartość  $\Delta f$  jest wyższa od pewnej wartości  $\alpha$ , która jest granicą o eksperymentalnie ustalonej wartości 50 Hz.

Procedura wyboru maksimum przy użyciu zdefiniowanych wcześniej parametrów została zilustrowana na rys. 6-41. Procedura ta działa na niższym poziomie niż opisana wcześniej procedura „zgrubna”, stanowi ona zatem jej uzupełnienie. Kształt przykładowych progów filtracji, uzyskanych w wyniku działania obu procedur, przedstawiono na rys. 6-42.



**Rys. 6-41.** Ilustracja procedury wyboru maksimum za pomocą przesuwającego się okna



**Rys. 6-42.** Próg odcięcia określany przez procedurę główną i przez procedurę uzupełniającą:  
 (a) i-ty pakiet; kształt progu wynika z obecności silnych prążków widmowych wykrytych w bieżącym pakiecie oraz obecnych w pakiecie  $i - 1$ , składowe pozostające poniżej progu zostały wykryte przez procedurę uzupełniającą;  
 (b) pakiet  $i + 1$ ; Próg wyznaczony przy użyciu parametrów preemfazy, zmodyfikowany z uwagi na silną składową o wysokiej częstotliwości w i-tym pakiecie



### 6.5.2. Procedury rekonstruowania dźwięku

Jak podano poprzednio, próg pozwalający zakwalifikować składowe widmowe do resyntezy może być uaktualniany dla każdego pakietu, na podstawie zbioru reguł. Zbiór ten powstaje w oparciu o przykłady dźwiękowe przetwarzane w fazie treningu. Zdefiniowano zatem trzy kategorie (klasy) decyzyjne:

1. Dźwięk zaszumiony (próg zbyt niski)
2. Dźwięk czysty (próg właściwy)
3. Dźwięk zniekształcony (próg zbyt wysoki)

Powyższe wyrażenia odnoszą się do subiektywnej oceny efektów filtracji progowej w dziedzinie widma. Jeżeli próg jest zbyt niski, to szum impulsowy (syczenie) jest wyraźnie słyszalny. Gdy próg jest zbyt wysoki, wiele pożądaných składowych jest usuwanych i wynikowy dźwięk jest czysty, lecz zostaje obciążony wyraźnie słyszalnymi zniekształceniami nieliniowymi. Przy odpowiednim ustawieniu progu szum jest usuwany bez utraty zbyt wielu pożądaných składowych.

Mechanizm wyznaczania progu jest oparty na algorytmie decyzyjnym reprezentowanym przez zbiór reguł, generowanych automatycznie w fazie treningu. Na podstawie wygenerowanych reguł i wyników działania procedury uzupełniającej, składowe widmowe położone powyżej progu są wykrywane i zachowywane, podczas gdy pozostałe składowe, leżące poniżej progu, są zerowane. Operacje te są dokonywane na całym widmie zespolonym. Przetwarzane są zatem składowe rzeczywiste i urojone, położone zarówno na dodatniej, jak i na ujemnej półosi częstotliwości. Problem pojawiających się nieciągłości fazy został zaniedbany, może on być przedmiotem dalszych badań. Po dokonaniu (w procesie resyntezy) odwrotnej transformacji Fouriera otrzymuje się bezpośrednio nowe wartości próbek, reprezentujących zrekonstruowany sygnał.

Algorytm uczący się, generujący reguły, składa się z:

1. Procedury wyboru krótkich fragmentów nagrania,
2. Fazy ustawiania wartości parametrów ( $c$ ,  $h$ ,  $f_p$ ,  $p$ ) przez operatora,
3. Subiektywnej oceny efektów działania filtracji progowej po resyntezie.

W powyższy sposób konstruowana jest baza wiedzy, na podstawie subiektywnych ocen eksperckich, które dotyczą poszczególnych przykładów dźwiękowych. W praktyce wystarczy wybrać przykłady odpowiadające najbardziej charakterystycznym fragmentom nagrania. Zwykle wybiera się do treningu ok. 5% całkowitego materiału muzycznego, a następnie ocenia się go na podstawie 3-sekundowych fragmentów. W ten sposób powstaje strumień przykładowych danych. Wynikowe reguły reprezentują klasy decyzyjne, oznaczone jako „dźwięk zaszumiony”, „dźwięk czysty”, „dźwięk zniekształcony”. Następnie reguły są przetwarzane zgodnie z konceptem „dźwięk czysty”. Definiowane są nowe reguły dla parametrów  $c$ ,  $h$ ,  $f_p$ ,  $p$  na

podstawie aktualnych wartości parametrów  $m_1, m_2, m_3, d_1, d_2, d_3$ , które są obliczane automatycznie gdy przetwarzany jest nowy materiał. W ten sposób budowana jest baza wiedzy, zawierająca wiedzę dotyczącą fragmentów użytych do treningu (na pewno) i wiedzę odnoszącą się do pozostałej części nagrania (prawdopodobnie). W praktyce zdolności generalizacyjne algorytmów inteligentnych, do których należy metoda zbiorów przybliżonych, pozwalają na podejmowanie trafnych decyzji również w stosunku do nowego materiału, nie używanego na etapie treningu.

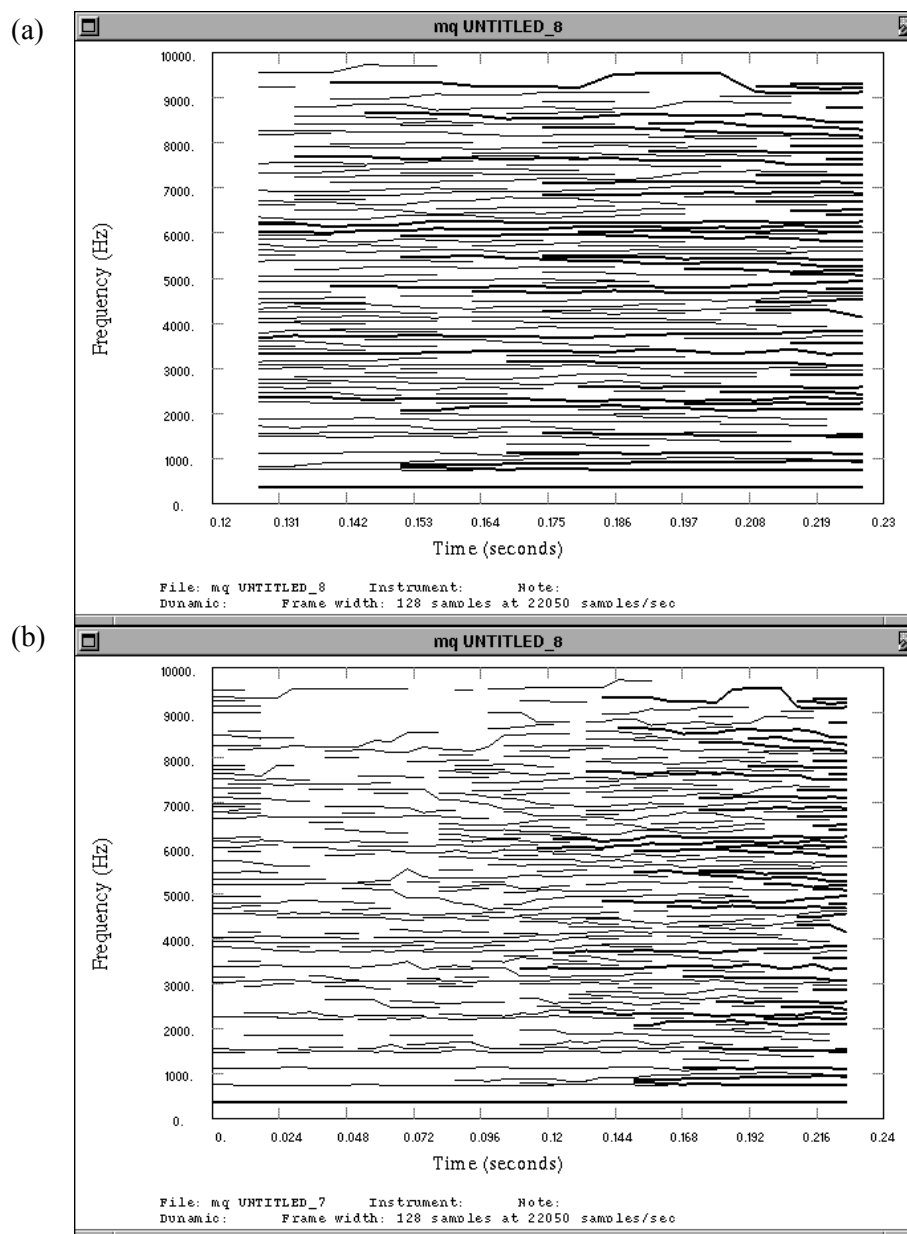
Jak wynika z par. 6.3.1.8, miara wiarygodności reguł, nazywana miarą przybliżoną jest oznaczana jako  $\mu_{RS}$ . Miara ta spełnia warunek  $\mu_{RS} = 1$  dla reguł „pewnych” oraz  $0 < \mu_{RS} < 1$  dla reguł „prawdopodobnych”. Zatem  $\mu_{RS}$  odzwierciedla stopień zaufania do reguły. Zwykle reguły o mierze  $\mu_{RS} < 0,5$  są usuwane, ponieważ stopień zaufania powinien przekraczać 50 %. Ponadto, zgodnie z zasadami metody zbiorów przybliżonych, nadmiarowe i zbędne reguły są usuwane automatycznie, natomiast najważniejsze reguły pozostają w bazie wiedzy. Jednak decyzja zawsze pochodzi od pojedynczej, najsilniejszej reguły. Zatem reguła taka jest w danej chwili wykorzystywana do uaktualniania wartości parametrów procedury rekonstrukcyjnej ( $c, h, f_p, p$ ), wpływając na próg filtracji, pod warunkiem, że spełnione są jednocześnie następujące warunki:

1. W bazie wiedzy istnieje reguła, która odpowiada bieżącej konfiguracji atrybutów warunkowych,
2. Miara przybliżona  $\mu_{RS}$  przewyższa 50 %,
3. Reguła ta jest najsilniejsza spośród wszystkich reguł opartych na tych samych atrybutach.

Jeżeli pierwsze dwa warunki nie są spełnione, to wartość progu nie jest uaktualniana (pozostaje taka sama, jak w poprzednim pakiecie).

### 6.5.3 Przykładowe eksperymenty i ich wyniki

W opisywanych eksperymentach została wykorzystana muzyka ze starych nagrań płytowych, zniekształcona przez silny szum słyszalny jako syczenie. Algorytm analizy–resyntezy wraz z „inteligentną” metodą sterowania filtracją progową został zastosowany m.in. do fragmentu muzycznego, odczytanego ze starej płyty gramofonowej wraz z bardzo silnym szumem impulsowym. Wyniki analizy materiału muzycznego przed i po jego przetworzeniu pokazano na rys. 6-43.



**Rys. 6-43.** Wyniki rekonstrukcji sygnału: (a) analiza McAulaya-Quatieriego sygnału oryginalnego, zakłóconego silnym szumem impulsowym, (b) wynik jednoczesnego działania obu procedur filtracji

Po przeprowadzeniu wstępnych testów ustalono eksperymentalnie następujące wartości atrybutów, mające zauważalny wpływ na jakość przetwarzanego dźwięku:

1. Atrybuty warunkowe:
 

atr.1 = $c_s$ ,	$s \in \langle 0, 1, 2, 3, 4 \rangle$ ,	$c \in \{0,2; 0,4; 0,6; 0,8; 1,0\}$
atr.2 = $h_s$ ,	$s \in \langle 0, 1, 2, 3 \rangle$ ,	$h \in \{0,2; 0,4; 0,6; 0,8\}$
atr.2 = $f_{ps}$ ,	$s \in \langle 0, 1, 2 \rangle$ ,	$f_p \in \{0,4; 1,0; 3,0\}$ [kHz]
atr.3 = $p_s$ ,	$s \in \langle 0, 1, 2 \rangle$ ,	$p \in \{3, 6, 12\}$ [dB na oktawę]

atr.4 ÷ atr.9  $\Rightarrow m_1, m_2, m_3, d_1, d_2, d_3$  są obliczane przez algorytm przy użyciu pakietów sygnału i automatycznie kwantowane
2. Atrybuty decyzyjne:
  - dec.1 = {dźwięk zaszumiony}
  - dec.2 = {dźwięk czysty }
  - dec.3 = {dźwięk zniekształcony}

Zbiór reguł otrzymano automatycznie przy pomocy procedury uczącej się, na podstawie 10-ciu charakterystycznych fragmentów materiału dźwiękowego o czasie trwania każdego ok. 3 sekundy, spróbkowanych z częstotliwością 22,05 kHz, co jest równoważne około 260-ciu pakietom zawierającym po 256 próbek. Podczas analizy wygenerowano 441 reguł o różnych wartościach miary przybliżonej  $\mu_{RS}$ , odzwierciedlających stopień zaufania do każdej z reguł.

Warto zaznaczyć, że baza wiedzy w systemie eksperckim, opartym na zbiorach przybliżonych może być łatwo analizowana przez eksperta. Reguły są czytelne, a ich znaczenie jest w pełni zrozumiałe, zwłaszcza dla operatora systemu eksperckiego, który nabył właściwego doświadczenia w zakresie interpretacji wpływu poszczególnych parametrów na wynik filtracji progowej.

Uzyskana baza reguł jest wykorzystywana do automatycznego wyznaczania progów we wszystkich pakietach próbek, odpowiadających całemu nagraniu. W trakcie przetwarzania typowego pakietu sygnału uruchamianych jest wiele reguł, z których niektóre są pewne (miara przybliżona równa 1), a inne niepewne (przybliżone). Zgodnie z zasadami metody zbiorów przybliżonych, decyzja zawsze pochodzi od najsilniejszej reguły, charakteryzującej się największą wartością miary przybliżonej. Zatem próg filtracji widmowej jest uaktualniany zgodnie z parametrami powiązаныmi ze zwycięską regułą. Wynik przetwarzania sygnału muzycznego z wykorzystaniem opisaney metody zilustrowano na rys. 6-43b.

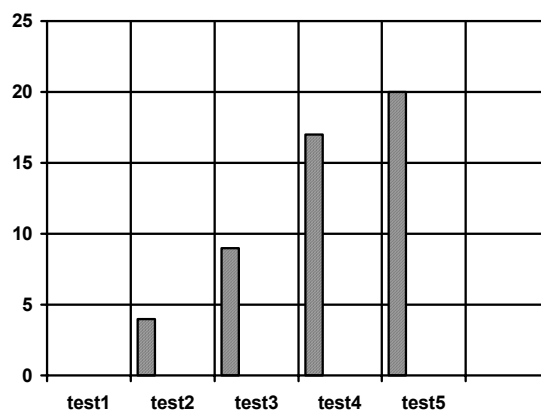
Przeprowadzono również subiektywne testy metodą porównywania parami. Pary sygnałów porównywano na podstawie pięciu przetworzonych

wersji tego samego dziesięcioszekundowego fragmentu muzyki. W teście wzięło udział pięciu ekspertów z dziedziny inżynierii dźwięku. Słuchacze oceniali pary sygnałów utworzone na podstawie fragmentów muzycznych i przetworzone przez opisany algorytm rekonstrukcji, wybierając lepszy element z każdej pary, zgodnie z ich subiektywnymi opiniami.

Porównywano następujące techniki usuwania szumu impulsowego:

- *Test 1*: Żadna (sygnał oryginalny zniekształcony przez szum impulsowy)
- *Test 2*: Stały poziom progu odcięcia, wybrany przez operatora (wyższy próg)
- *Test 3*: Stały poziom progu odcięcia, wybrany przez operatora (niższy próg)
- *Test 4*: Inteligentna filtracja progowa (tylko procedura „zgrubna”)
- *Test 5*: Złożona filtracja progowa (obie procedury jednocześnie)

Wyniki porównań parami przedstawiono na rys. 6-44. Podczas porównywania parami nigdy nie wybrano nieprzetworzonego dźwięku. Stały, wysoki próg wybrany przez operatora pozwala na dość efektywne stłumienie szumu impulsowego. Jednak jakość dźwięku staje się zła na skutek usunięcia większości składowych o dużych częstotliwościach i całkowitej derewerberacji sygnału. Nieco lepsze wyniki daje stały niski próg, jednak część szumu pozostaje słyszalna. Jakość dźwięku w testach 4 i 5 została najwyżej oceniona przez ekspertów. Statystyczna istotność różnic pomiędzy wynikami została zweryfikowana przy pomocy testu Pearsona  $\chi^2$ . Procedura uzupełniająca miała pozytywny wpływ na ostateczną ocenę przetworzonego fragmentu dźwiękowego.



Rys. 6-44. Wynik testów porównywania parami – liczba głosów oddanych przez ekspertów na poszczególne fragmenty dźwiękowe

Metoda zbiorów przybliżonych, pozwalająca na wyznaczenie pewnych i prawdopodobnych (niepewnych) reguł na podstawie fragmentów materiału dźwiękowego okazała się przydatna w redukcji szumu impulsowego. Metoda została wprowadzona z uwagi na element niepewności towarzyszący podejmowaniu decyzji w procesie usuwania szumu. Problem niepewności w zaprojektowanym systemie eksperckim pojawia się w każdym przypadku automatycznego ustawiania progu filtracji widmowej. Zdolność algorytmu opartego na zbiorach przybliżonych do uczenia się na podstawie przykładów, generalizowania wiedzy i adaptacji to najważniejsze cechy opisanego podejścia do problemu redukcji pasożytniczego szumu. Metoda zbiorów przybliżonych zastosowana do usuwania szumu impulsowego jest bardziej efektywna w porównaniu z innymi algorytmami, np. sieciami neuronowymi, ponieważ przy porównywalnych zdolnościach generalizacyjnych, czas treningu jest znacznie krótszy w przypadku zbiorów przybliżonych.

W innych eksperymentach, które będą opisane w następnym podrozdziale autor wraz ze współpracownikami zastosowali logikę rozmytą i sieci neuronowe do adaptacyjnej redukcji szumu w kanałach telekomunikacyjnych. Uzyskane w wyniku tych eksperymentów wyniki również potwierdzają przydatność metod inteligentnych do usuwania silnego szumu szerokopasmowego, którym mogą być obciążone sygnały foniczne.

## **6.6 Redukowanie szumu szerokopasmowego**

### **6.6.1 Inteligentne sterowanie algorytmem odejmowania widmowego**

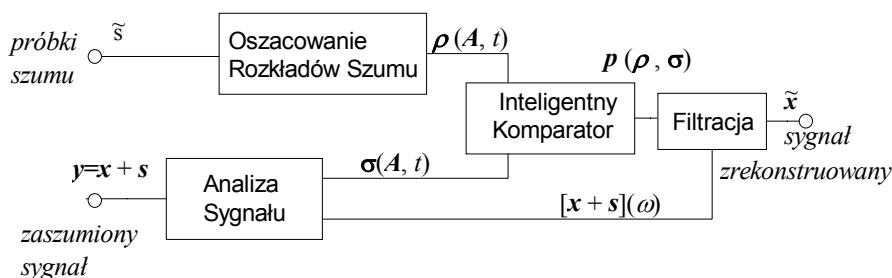
Przedstawiona w niniejszym podrozdziale metoda jest przeznaczona do redukowania tzw. szumu pustego kanału. Nadaje się szczególnie do zastosowania w sytuacjach, gdy układy automatycznej regulacji wzmocnienia wzmacniają silny szum tła akustycznego (np. odbiór mowy przez mikrofon w samochodzie, komunikacja z pilotami helikopterów, nurkami itp.). Metoda ta stanowi kolejny przykład implementacji systemu eksperckiego do celów redukcji szumu. W omawianym przypadku jest to szum szerokopasmowy, zaś obszarem praktycznych zastosowań opracowanej metody jest poprawa parametrów sygnałów fonicznych transmitowanych w torach telekomunikacyjnych.

Na rys. 6-45 przedstawiono ogólny schemat opracowanej metody redukcji szumu, która polega na odejmowaniu widmowym, sterowanym przez algorytm uczący się. Tradycyjne metody odejmowania widmowego

opisano w par. 4.5.2. Proponowany algorytm stanowi rozszerzenie tego typu metod, związane z wykorzystaniem algorytmów inteligentnych.

Na wejściu systemu są dwa sygnały:  $\tilde{s}$  - próbka szumu pustego kanału teletransmisyjnego oraz  $y=x+s$  - zaszumiony sygnał wejściowy. Zakłada się, że sygnały  $\tilde{s}$  i  $s$  są w pewnym stopniu skorelowane. Pobrany wzorec szumu  $\tilde{s}$  jest analizowany w kolejnych pasmach częstotliwości, odpowiadających pasmom krytycznym słuchu, w wyniku czego otrzymywany jest rozkład szumu w tych podpasmach, oznaczony jako  $\rho(A,t)$ . Jest to w istocie rozkład widmowej gęstości mocy szumu w poszczególnych pasmach krytycznych. Dla danego podpasma rozkład ten zależy od wartości amplitud widma mocy  $A$  w chwili czasu  $t$ . W podobny sposób przetwarzany jest sygnał zaszumiony, tzn. wyznaczone są rozkłady widmowej gęstości mocy sygnału zaszumionego  $\sigma(A,t)$  również w kolejnych pasmach krytycznych. Następnie, rozkłady  $\rho$  i  $\sigma$  są porównywane w module wnioskowania, pełniącym rolę inteligentnego komparatora. Jego zadaniem jest wykrywanie niejawnych powiązań pomiędzy tymi dwoma rozkładami, tj. określenie miary podobieństwa pomiędzy nimi. Na wyjściu komparatora uzyskiwany jest wektor współczynników określających stopień podobieństwa obu rozkładów, oznaczony jako  $p(\rho,\sigma)$ ; im wyższa wartość tych współczynników, tym większe prawdopodobieństwo, że poszczególne składowe sygnału  $y=x+s$  mają charakter szumowy. Dlatego też sygnał  $y$  jest następnie filtrowany z uwzględnieniem rozkładu  $p(\rho,\sigma)$ . Oznacza to, że pojedyncza składowa sygnału  $s+n$  jest pomniejszana zgodnie z rozkładem  $p$  lub że jest ona kwalifikowana do bezpośredniego wykorzystania w resyntezie (na podstawie rozkładu  $1-p$ ). W rezultacie na wyjściu układu pojawia się pożądany sygnał  $\tilde{x}$  (estymata sygnału niezaszumionego  $x$ ).

W opisywanej metodzie sygnał foniczny można wielokrotnie poddawać działaniu algorytmu redukcji szumu, co prowadzi do stopniowej



**Rys. 6-45.** Ogólny schemat blokowy metody redukcji szumu. Oznaczenia:  
 $\rho(A,t)$  - rozkład widmowej gęstości mocy szumu,  
 $\sigma(A,t)$  - rozkład widmowej gęstości mocy sygnału zaszumionego,  
 $p(\rho,\sigma)$  - rozkład określający podobieństwo sygnałów

poprawy stosunku sygnału do szumu.

### 6.6.2. Opis metody

Załóżmy, że sygnał użyteczny  $x(k)$  jest zakłócony przez szum addytywny  $s(k)$ . Wówczas zakłócony sygnał wynikowy  $y(k)$  może być opisany w dziedzinie czasu następująco:

$$y(k) = x(k) + s(k) \quad (6-34)$$

Po dokonaniu transformacji w dziedzinę częstotliwości przy pomocy algorytmu DFT (Dyskretnej Transformacji Fouriera), powyższa zależność przybiera postać:

$$Y(\omega) = X(\omega) + S(\omega) \quad (6-35)$$

W celu uzyskania wyrażenia na sygnał użyteczny  $X(\omega)$ , wzór (6-35) można zapisać jako :

$$X(\omega) = Y(\omega) - S(\omega) = \left(1 - \frac{S(\omega)}{Y(\omega)}\right) \cdot Y(\omega) \quad (6-36)$$

Jednakże w praktyce trudno jest uzyskać dokładny opis sygnału zakłócającego  $S(\omega)$  w formie jawnej. Dlatego istnieje potrzeba posługiwania się estymatą  $\tilde{S}(\omega)$  szumu  $S(\omega)$ . Wówczas sygnał wynikowy może być tylko estymatą  $\tilde{X}(\omega)$  oryginalnego sygnału użytecznego  $X(\omega)$ , tj.:

$$S(\omega) \rightarrow \tilde{S}(\omega) \Rightarrow X(\omega) \rightarrow \tilde{X}(\omega) \quad (6-37)$$

A zatem, oszacowany użyteczny sygnał  $\tilde{X}(\omega)$  może być wyrażony jako:

$$\tilde{X}(\omega) = \left(1 - \frac{\tilde{S}(\omega)}{Y(\omega)}\right) \cdot Y(\omega) = H(\omega) \cdot Y(\omega) \quad (6-38)$$

gdzie:  $H(\omega)$  - transmitancja układu redukcji szumu,

$Y(\omega)$  - widmo sygnału użytecznego x zakłóconego przez szum s



Otrzymany wzór jest zgodny z zależnością (4-93), co wynika z faktu, że analizowany układ jest w ogólności systemem, w którym dokonywane jest odejmowanie widmowe.

W wyniku zastosowania filtracji, kolejne prążki widmowe sygnału poddawanego rekonstrukcji mogą być pomniejszone określoną wartością albo zostają zachowane. Odejmowanie to jest dokonywane zgodnie z rozkładem  $p(\tilde{S}(\omega), Y(\omega))$ , który w pewien sposób odzwierciedla stopień podobieństwa pomiędzy sygnałami  $x$  i  $s$ , co można wyrazić jako:

$$H(\omega) = f(\tilde{S}(\omega), Y(\omega)) = \begin{cases} p_i & \text{zgodnie z rozkładem: } p(\tilde{S}(\omega), Y(\omega)) \\ 0 & \text{zgodnie z rozkładem: } 1 - p(\tilde{S}(\omega), Y(\omega)) \end{cases} \quad (6-39)$$

gdzie:  $p_i$  – jest odjemnikiem amplitudy  $i$ -tego prążka widmowego, otrzymywanym na podstawie rozkładu  $p(\tilde{S}(\omega), Y(\omega))$ .

Ponieważ sygnał  $\tilde{S}(\omega)$  jest tylko estymatą sygnału zakłócającego, dlatego trudno jest wyznaczyć rozkład  $p(\tilde{S}(\omega), Y(\omega))$  w postaci jawnej, zwłaszcza gdy szum  $S(\omega)$  jest niestacjonarny. Aby zatem oszacować rozkład  $p(\tilde{S}(\omega), Y(\omega))$  bez potrzeby przyjmowania założeń a priori odnoszących się do rozkładów sygnału i szumu, zastosowano wnioskowanie oparte na algorytmach uczących się.

### 6.6.3. Szacowanie rozkładu szumu

Pewną miarą zawartości szumu  $m$ -tym paśmie krytycznym może być stosunek  $\rho$  liczby prążków widma mocy, które reprezentują szum pustego kanału do liczby wszystkich prążków widma mocy w tym paśmie. W chwili  $t_j$  stosunek, o której mowa może być obliczony zgodnie ze wzorem:

$$\rho(A, t) = \frac{n(A, t)}{X} \Big|_{t = t_j} \quad (6-40)$$

gdzie:  $n(A, t)$  - liczba składowych szumowych o mocy  $A$ ,  
 $X$  – całkowita liczba prążków widmowych w danym paśmie krytycznym

Jednakże w przypadku sygnałów cyfrowych wartości mocy oraz czasu są dyskretne. Zatem w  $k$ -tej chwili odpowiednikiem powyższego wzoru są następujące wyrażenia:

- dla szumu niestacjonarnego:

$$\rho(\Delta_i, t_j) = \frac{n(\Delta_i, t_j)}{X} \Big|_{j=k}; \quad i = 1, \dots, I \quad (6-41)$$

- dla szumu stacjonarnego:

$$\rho(\Delta_i) = \frac{\bar{n}(\Delta_i)}{X}; \quad i = 1, \dots, I \quad (6-42)$$

gdzie:  $\Delta_i$  - przedział wartości mocy dla prążków z zakresu:  
 $[(i-1) \cdot \Delta, (i+1) \cdot \Delta)$ ,  
 $\Delta$  - jest stałą szerokością przedziału elementarnego,  
 $n(\Delta_i, t_j)$  - liczba szumowych prążków widmowych, o mocach należących do  $i$ -tego przedziału  $\Delta_i$ ,  
 $\bar{n}(\Delta_i)$  - średnia liczba szumowych prążków widmowych, o mocach należących do  $i$ -tego przedziału  $\Delta_i$ ,  
 $I$  - liczba przedziałów  $\Delta_i$ , na które jest podzielony cały zakres mocy. Dla maksymalnej wartości mocy prążków MAX w  $m$ -tym paśmie krytycznym można zapisać:

$$\lceil \text{MAX} \rceil = \Delta \cdot I \quad (6-43)$$

gdzie:  $\lceil \cdot \rceil$  - operacja *entier*, tj. operacja zaokrąglania w górę do najbliższej liczby całkowitej.

W praktycznych eksperymentach przyjmowano wartość  $\Delta$  w zakresie od  $\Delta=2$  do  $\Delta=16$ .

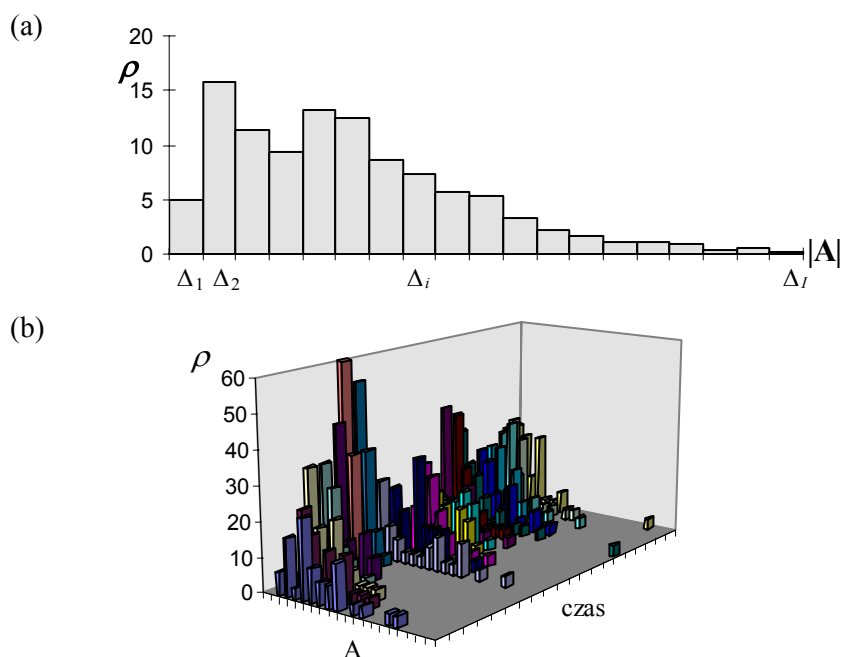
Rozkłady widmowej gęstości mocy szumu wyrażone wzorem (6-72), zilustrowane na rys. 6-46 spełniają zależność:

$$\sum_{i=1}^I \rho(\Delta_i, t_j) \Big|_{j=k} \leq 1; \quad j, k = 1, 2, \dots \quad (6-44)$$

Oszacowanie rozkładów bieżącego sygnału przeprowadzane jest w analogiczny sposób. A zatem dla dyskretnej postaci sygnału zaszumionego, wykorzystywana jest poniższa zależność:

$$\sigma(\Delta_i, t_j) = \frac{n(\Delta_i, t_j)}{X} \Big|_{j=k}; \quad i = 1, 2, \dots, I; \quad j, k = 1, 2, \dots \quad (6-45)$$

gdzie:  $n(\Delta_i, t_j)$  - liczba prążków widmowych sygnału zaszumionego, o mocach należących do  $i$ -tego przedziału  $\Delta_i$ .



Rys. 6-46. Rozkład widmowej gęstości mocy szumu  $\rho$  w pojedynczym paśmie krytycznym: (a) w określonym przedziale czasu, (b) względem czasu

#### 6.6.4 Wykorzystanie algorytmów inteligentnych

Jak wspomniano wcześniej, w celu wyznaczenia bieżącego rozkładu  $p(\tilde{S}(\omega), Y(\omega))$  zastosowano algorytmy inteligentne. Oszacowanie wartości

tego rozkładu następuje w inteligentnym komparatorze (z rys. 6-45), na wejścia którego podawane są sygnały  $\rho$  i  $\sigma$ , zaś otrzymywany jest rozkład  $p(\rho, \sigma)$ . Poniżej omówiono w skrócie sposób implementacji inteligentnego komparatora opartego na wnioskowaniu rozmytym oraz na przetwarzaniu neuronalnym.

Rozważmy  $m$ -te pasmo krytyczne oraz  $i$ -ty przedział widmowej gęstości mocy  $\Delta_i$  w tym podpaśmie. Przyjmijmy następujące oznaczenia:

- a)  $\rho$  - widmowa gęstość mocy szumu szacowana na podst. wzoru (6-72),
- b)  $\sigma$  - widmowa gęstość mocy sygnału zaszumionego wyznaczona na podstawie wzoru (6-75),
- c)  $p$  - miara określająca stopień podobieństwa pomiędzy sygnałami na obu wejściach inteligentnego komparatora

#### 6.6.4.1 Zastosowanie wnioskowania rozmytego

W każdej chwili czasu inteligentny komparator otrzymuje na wejściu dwie wartości, zaś na wyjściu generuje jedną wartość. Wartości te można uważać za zmienne lingwistyczne. A zatem, wejściowymi zmiennymi lingwistycznymi są:  $\rho$  i  $\sigma$ , natomiast wyjściową zmienną lingwistyczną jest  $p$ . Zbiory rozmyte atrybutów poszczególnych zmiennych przyjmują we wszystkich przypadkach nazwy (*Low*, *Medium*, *High*) i są określone przez funkcje przynależności:  $\mu(\rho)$ ,  $\mu(\text{current})$ ,  $\mu(p)$ . Kształty tych funkcji przynależności przedstawiono na rys. 6-47.

Tab. 6-5. Zestaw reguł dla modułu wnioskującego opartego na logice rozmytej

- |    |           |        |           |               |            |         |           |               |             |     |           |               |
|----|-----------|--------|-----------|---------------|------------|---------|-----------|---------------|-------------|-----|-----------|---------------|
| 1. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Low</i>    | <b>AND</b> | current | <b>is</b> | <i>Low</i>    | <b>THEN</b> | $p$ | <b>is</b> | <i>High</i>   |
| 2. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Low</i>    | <b>AND</b> | current | <b>is</b> | <i>Medium</i> | <b>THEN</b> | $p$ | <b>is</b> | <i>Medium</i> |
| 3. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Low</i>    | <b>AND</b> | current | <b>is</b> | <i>High</i>   | <b>THEN</b> | $p$ | <b>is</b> | <i>Low</i>    |
| 4. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Medium</i> | <b>AND</b> | current | <b>is</b> | <i>Low</i>    | <b>THEN</b> | $p$ | <b>is</b> | <i>Medium</i> |
| 5. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Medium</i> | <b>AND</b> | current | <b>is</b> | <i>Medium</i> | <b>THEN</b> | $p$ | <b>is</b> | <i>High</i>   |
| 6. | <b>IF</b> | $\rho$ | <b>is</b> | <i>Medium</i> | <b>AND</b> | current | <b>is</b> | <i>High</i>   | <b>THEN</b> | $p$ | <b>is</b> | <i>High</i>   |
| 7. | <b>IF</b> | $\rho$ | <b>is</b> | <i>High</i>   | <b>AND</b> | current | <b>is</b> | <i>Low</i>    | <b>THEN</b> | $p$ | <b>is</b> | <i>Low</i>    |
| 8. | <b>IF</b> | $\rho$ | <b>is</b> | <i>High</i>   | <b>AND</b> | current | <b>is</b> | <i>Medium</i> | <b>THEN</b> | $p$ | <b>is</b> | <i>Medium</i> |
| 9. | <b>IF</b> | $\rho$ | <b>is</b> | <i>High</i>   | <b>AND</b> | current | <b>is</b> | <i>High</i>   | <b>THEN</b> | $p$ | <b>is</b> | <i>High</i>   |

Działanie komparatora oparte jest na zbiorze reguł heurystycznych, który pokazano w Tab. 6-5.

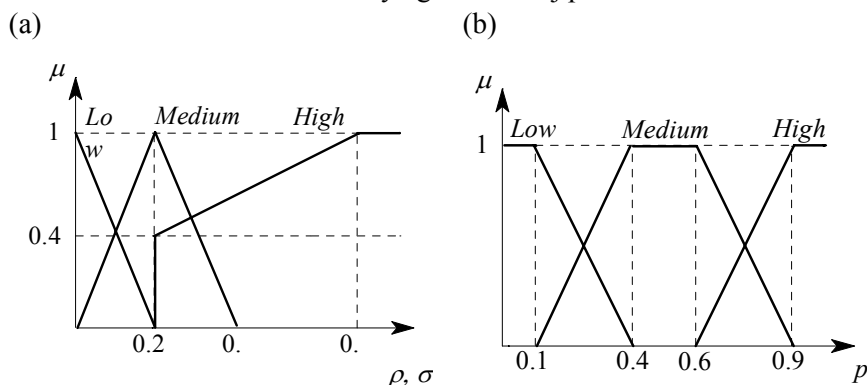
W celu odwzorowania wyjściowego zbioru rozmytego  $P$  w ściśle określoną wartość rzeczywistą  $p_i$  na drodze defuzyfikacji, zastosowano metodę środka ciężkości – por. zależność (6-6):

$$p_i(\tilde{S}(\omega), Y(\omega)) = \frac{\int \mu_p(p, \tilde{S}(\omega), Y(\omega)) \cdot p \, dp}{\int \mu_p(p, \tilde{S}(\omega), Y(\omega)) \, dp} \quad (6-46)$$

czyli:

$$p_o(\tilde{S}(\omega), Y(\omega)) = \frac{\int \mu_p(p, \rho, \sigma) \cdot p \, dp}{\int \mu_p(p, \rho, \sigma) \cdot p \, dp} \quad (6-47)$$

gdzie:  $p_i$  - wartość rzeczywista wyjściowej zmiennej lingwistycznej  $p$ ,  
 $\tilde{S}(\omega), Y(\omega)$  - estymata szumu i widmo sygnału zaszumionego,  
 parametry omówione w p. 6.6.2,  
 $P$  - dziedzina zbioru rozmytego zmiennej  $p$

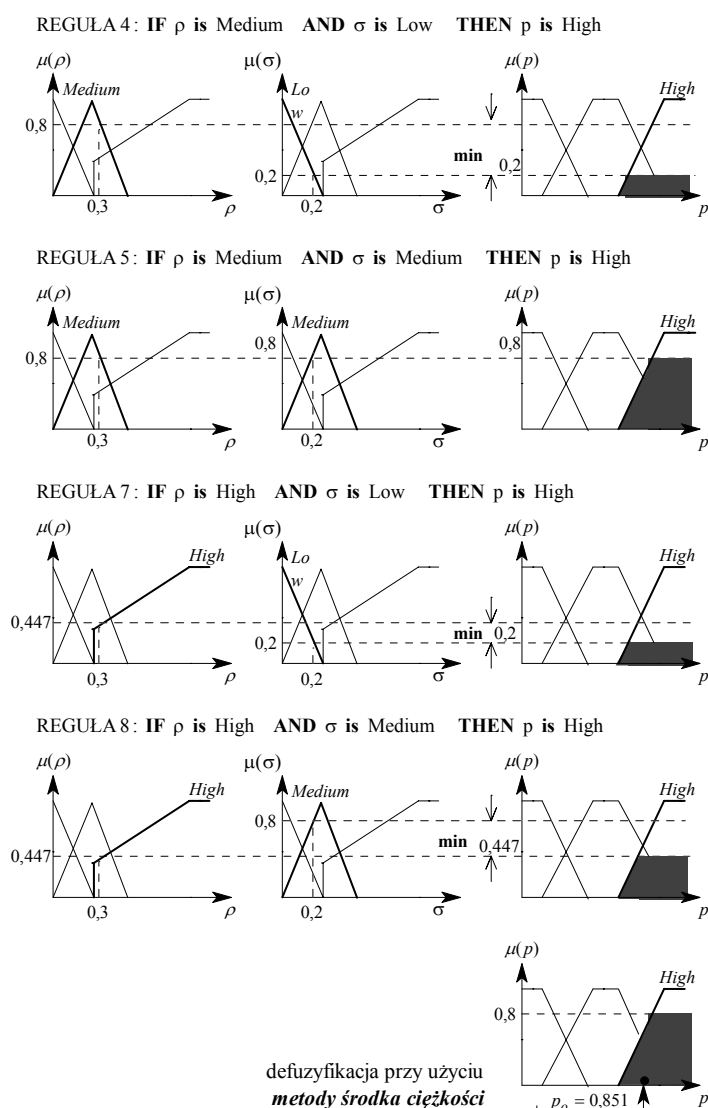


**Rys. 6-47.** Dobrane eksperymentalnie kształty funkcji przynależności dla: (a) zmiennych wejściowych  $\rho$  i  $\sigma$ , (b) zmiennej wyjściowej  $p$ ,  $\mu_p(\cdot)$  - funkcja przynależności zmiennej wyjściowej  $p$ ,

Przykład:

Niech system wnioskowania rozmytego będzie oparty na bazie reguł z Tab. 6-5, zaś rozmyte zbiory atrybutów dla poszczególnych zmiennych lingwistycznych będą określone przez funkcje przynależności z rys. 6-47. Ponadto przyjmijmy, że metodą defuzyfikacji jest metoda środka ciężkości (6-77). Załóżmy, że dla  $m$ -tego pasma krytycznego oraz  $i$ -tego przedziału  $\Delta_i$  w tym podpaśmie zmienne wejściowe komparatora przyjmują następujące

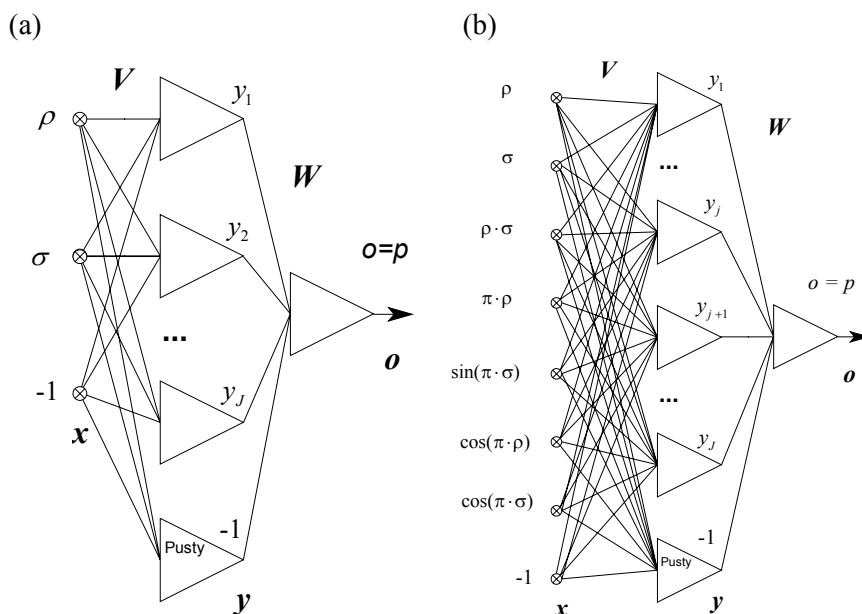
wartości:  $\rho = 0,3$  i  $\sigma = 0,2$ . Oparty na tych danych przykład wnioskowania rozmytego zilustrowano na rys. 6-48, na którym uwzględniono tylko te reguły, które zostały uaktywnione. Ponieważ wartości  $\rho$  i  $\sigma$  są do siebie zbliżone, należy więc oczekiwać, że większość prążków widmowych w przedziale  $\Delta_i$  reprezentuje szum. Wynik porównania rozmytego wychodzi na przeciw tym oczekiwaniom, gdyż stopień podobieństwa został określony przez system na 0,851. Oznacza to, że wszystkie prążki, których moc mieści się w przedziale  $\Delta_i$ , zostaną pomniejszone o wartość 0,851.



Rys. 6-48. Ilustracja procesu wnioskowania rozmytego

6.6.4.2 Zastosowanie sieci neuronowych

Moduł komparatora może być alternatywnie oparty na sieci neuronowej. Jej zadaniem jest nieliniowa aproksymacja rozkładu  $p(\tilde{S}(\omega), Y(\omega))$ , który jest pewną nieliniową funkcją wartości wejściowych  $\rho$ ,  $\sigma$ . Stąd, sieć powinna mieć co najmniej: dwa neurony wejściowe reprezentujące te zmienne oraz jeden neuron wyjściowy reprezentujący stopień podobieństwa sygnałów wejściowych  $\rho$ , w określony sposób wyrażający liczbowo możliwość usunięcia prążków widma o mocach należących do przedziału  $\Delta_i$  m-tego pasma krytycznego. Odpowiednią strukturą jest w tym przypadku jednokierunkowa sieć wielowarstwowa (por. par. 6.2.5.1), która może być rozszerzona o dodatkowe połączenia funkcjonalne. W dokonanej implementacji, w obu przypadkach wykorzystano sieć z pojedynczą warstwą ukrytą, dla której liczbę neuronów dobierano w zakresie od 10 do 20. Struktury tego typu sieci pokazano na rys. 6-49.



Rys. 6-49. Struktura sieci neuronowych stosowanych w module wnioskowania: (a) klasyczna sieć jednokierunkowa, (b) sieć z rozszerzeniami funkcjonalnymi

Korzystając z operatorów macierzowych  $\Gamma$  zdefiniowanych we wzorze (6-17), aproksymację  $p(\rho, \sigma)$  można uzyskać na podstawie zależności:

$$p_o = p(\tilde{S}(\omega), Y(\omega)) = p(\rho, \sigma) = \Gamma[W \cdot \Gamma[Vx]] \quad (6-48)$$

gdzie:  $V$  i  $W$  - macierze wag dla kolejnych warstw sieci,  
 $x$  - wektor wejściowy, którego postać w przypadku klasycznej sieci jednokierunkowej oraz sieci z rozszerzeniami funkcjonalnymi dana jest następującą zależnością:

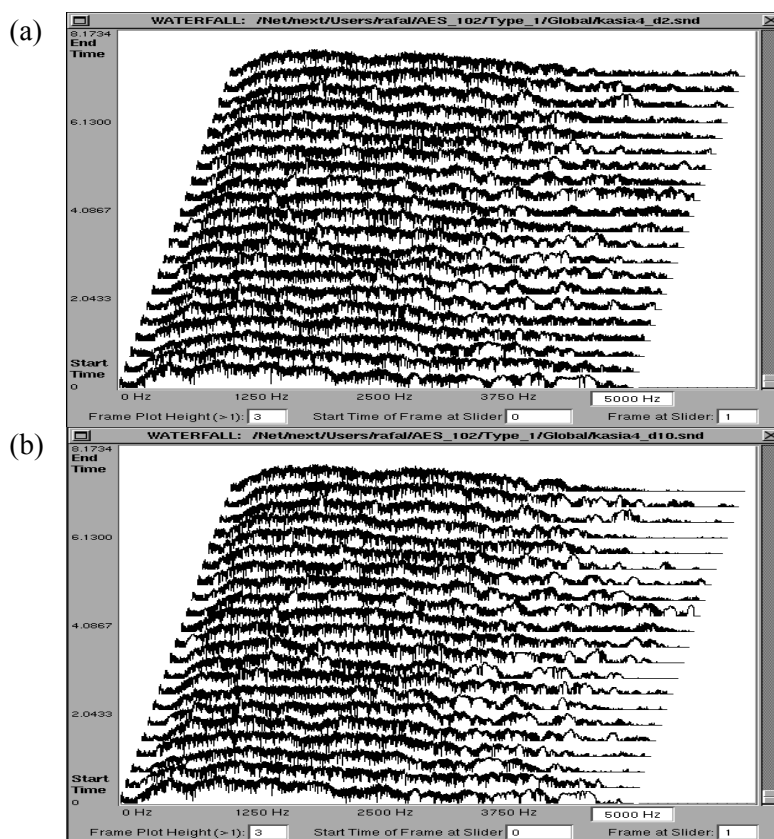
$$x^T = \begin{cases} [\rho, \sigma \cdot t]^T \\ [\rho, \sigma \cdot t, \rho \cdot \sigma, \sin(\rho \cdot \pi), \sin(\sigma \cdot t \cdot \pi), \cos(\rho \cdot \pi), \cos(\omega \cdot \pi)]^T \end{cases} \quad (6-49)$$

Charakter problemu wymusza nadzorowany tryb nauki. Dane podawane są przez eksperta, który dla kolejnych kombinacji par wartości wejściowych  $\rho$  i  $\sigma$  otrzymanych na podstawie analizy próbek dźwiękowych określa pożądaną wartość wyjściową  $p$ . Ze względu na zastosowany gradientowy algorytm treningu - wsteczną propagację błędu, wymagane jest użycie jednostek neuronowych o ciągłej funkcji aktywacji.

#### 6.6.4.3 Przykładowy proces rekonstrukcji zaszumionego sygnału fonicznego

Na rys. 6-50 przedstawiono przykład rekonstruowania sygnału fonicznego przy pomocy opracowanego algorytmu. W omawianym przykładzie głos męski jest obarczony szumem stacjonarnym o subiektywnie wysokim poziomie. Mowa została nagrana przy użyciu mikrofonu, zaś szum wygenerowano przy pomocy komputerowego generatora szumu. Rozkład czasowy widmowej gęstości mocy sygnału zaszumionego przedstawiono na rys. 6-50 a, natomiast odnośną analizę sygnału reprezentującego subiektywnie najlepszy wynik rekonstrukcji pokazano na rys. 6-50 b. W tym przypadku, inteligentny komparator został zaimplementowany w oparciu o logikę rozmytą. Analiza widmowa sygnałów szumu i sygnału użytecznego była oparta na transformacji Fouriera. Przyjęto długość ramki równą 1024 próbki, zaś długość zakładki równą 512 próbek. Częstotliwość próbkowania wynosiła 44,1kHz, zaś jako funkcję okienkową wybrano funkcję Hamminga. Przy pomocy opisanej metody rekonstruowano również sygnały obarczone szumem niestacjonarnym. W tym przypadku wzorzec szumu był pobierany z najbliższego pasma ciszy, występującego przed rekonstruowanym fragmentem sygnału. W tym celu w algorytmie uwzględniono procedurę wykrywania sygnału mowy opartą na badaniu cechy dźwięczności mowy (autokorelacyjna analiza okresowości sygnału). Dzięki odróżnieniu pasm





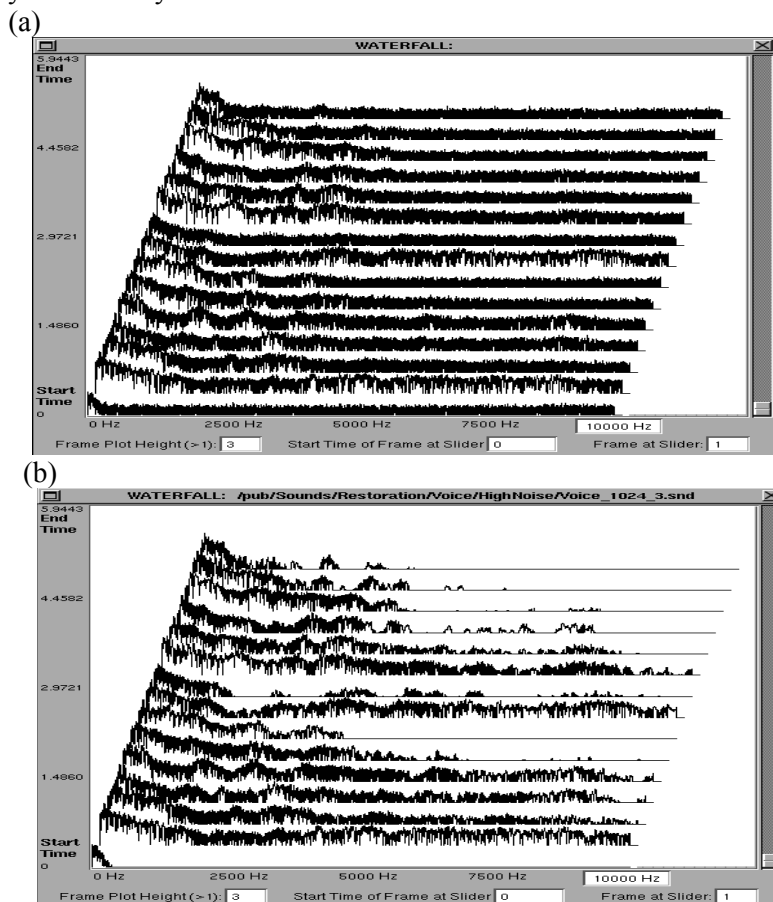
**Rys. 6-50.** Wynik rekonstrukcji fragmentu nagrania. Analizy widmowe (na osi x- częstotliwość, na osi y – poziom mocy widma, na osi z – czas):  
 (a) oryginalne nagranie obarczone szumem o wysokim poziomie,  
 (b) najlepszy subiektywnie wynik rekonstrukcji dla  $\Delta = 2$ , po 3-ch iteracjach

mowy od pasm ciszy, możliwe było pobieranie na bieżąco wzorców  $\rho$ , które mogły zmieniać się wraz ze zmianami szumu zakłócającego przekaz sygnału użytecznego.

W jednym z eksperymentów poddano odszumianiu sygnał mowy przesyłany za pośrednictwem nadajnika pracującego na falach krótkich i odbiornika stacji Gdynia Radio. Nadajnik był umieszczony na statku oceanicznym, znajdującym się w pobliżu Miami, w USA, zaś sygnał był nagrywany na wyjściu przewodowej linii telefonicznej w Gdańsku. Szum był w tym przypadku w wysokim stopniu niestacjonarny, a ponadto odznaczał się on bardzo wysokim poziomem, co spowodowało w konsekwencji, że odbierana mowa była niemal całkowicie niezrozumiała.

Wyniki rekonstruowania sygnału mowy z zastosowaniem modułu wnioskującego opartego na logice rozmytej zilustrowano na rys. 6-51.

Zrekonstruowana mowa została w znacznym pozbawiona szumu, dzięki czemu stała się bardziej zrozumiała. W procesie optymalizacji działania systemu wykorzystano subiektywne preferencje eksperta, który poprzez dobór funkcji przynależności i reguł rozmytych mógł wpływać na końcowe efekty działania systemu.



**Rys. 6-51.** Analizy widmowe próbek sygnału mowy obciążonej silnym szumem o charakterze niestacjonarnym: (a) przed rekonstrukcją; (b) wynik rekonstrukcji po 4-tej iteracji dla  $\Delta = 10$

Obecnie w KID PG trwają badania nad znacznie bardziej zaawansowaną metodą redukowania szumu, której zasada polega na regulowaniu położenia masek w algorytmie perceptualnego kodowania sygnałów fonicznych. Punktem wyjścia do tego typu obliczeń są zależności (5-25), (5-27), (5-33), (5-35) i (5-38). Położenie masek perceptualnych jest kontrolowane przez inteligentny algorytm decyzyjny, który na podstawie analizy wzorców szumu pustego kanału podejmuje decyzję o podniesieniu progów maskowania w taki sposób, aby uczynić niesłyszalnym szum

występujący w czasie transmisji sygnału użytecznego. W tym celu przyjmuje się, że moc chwilowa sygnału  $S(i)$  [dB] w  $i$ -tym podpaśmie  $[\Omega_{l,i}, \Omega_{u,i}]$  może być obliczona na podstawie wyrażenia:

$$S(i) = \sum_{\omega=\Omega_{u,i}}^{\Omega_{l,i}} X^2(e^{j\omega}) = \sum_{\omega=\Omega_{l,i}}^{\Omega_{u,i}} X_D^2(e^{j\omega}) + \sum_{\omega=\Omega_{l,i}}^{\Omega_{u,i}} X_U^2(e^{j\omega}) = S_D(i) + S_U(i) \quad (6-50)$$

gdzie:  $S_D(i)$ ,  $S_U(i)$  reprezentują moce chwilowe prążków widma  $X(e^{j\omega})$  - odpowiednio dla szumu i sygnału użytecznego w  $i$ -tym paśmie krytycznym.

Przy wykorzystaniu zależności (6-50), formuła (5-33) może być przekształcona do postaci:

$$E(i) = S_U(i) \otimes B(i, j) + S_D(i) \otimes B(i, j) = E_U(i) + E_D(i), \quad (6-51)$$

w której:  $E_U(i)$ ,  $E_D(i)$  - oznaczają pobudzenia błony podstawnej powodowane odpowiednio przez składowe użyteczne i zakłócające,  $i, j$  - oznaczają numery pasm krytycznych, dla których oblicza się sumaryczne wychylenie błony podstawnej.

Występujący w sygnale źródłowym szum można uczynić niesłyszalnym, jeżeli wzmocni się składowe użyteczne w taki sposób, że zamaskują ten szum. Jeśli składowe użyteczne w danym paśmie krytycznym ulegną  $\alpha$ -krotnemu wzmocnieniu, to otrzyma się nową wartość pobudzenia błony podstawnej  $E^\alpha(i)$  i nową wartość progu maskowania  $T^\alpha(i)$  (patrz wzory 6-51 i 5-33):

$$E^\alpha(i) = E_U^\alpha(i) + E_D(i) = \alpha^2 \cdot E_U(i) + E_D(i) \quad (6-52)$$

$$T^\alpha(i) = 10^{\log_{10} E^\alpha(i) - O^\alpha(i)/10} \quad (6-53)$$

Z kolei odległość  $O^\alpha(i)$  (patrz wzór 5-27) zależy od zmian SFM (patrz wzór 5-28). Stosując odpowiednie przekształcenia powyższych zależności można wykazać, że  $SFM^\alpha$  jest związane z SFM następującym równaniem:

$$SFM^\alpha = SFM + 10 \log_{10} \left( \alpha^{4 \frac{|U|}{N}} \cdot \left( 1 + (\alpha^2 - 1) \cdot \frac{\sum_{i=1}^{|U|} S_i^U}{\sum_{k=1}^{N/2} S_k} \right)^{-1} \right) \quad (6-54)$$

gdzie:  $S_i^U$ ,  $S_k$  - reprezentują moce odpowiednio  $i$ -tej składowej użytecznej widma i  $k$ -tej składowej widma,  $U$  - jest liczbą składowych użytecznych

Na podstawie powyższych zależności, po odpowiednich przekształceniach można otrzymać następujące wyrażenie:

$$\alpha = \sqrt{\frac{1}{E_U(i)} \left[ 10^{\log_{10} T^\alpha(i) + O^\alpha(i)/10} - E_D(i) \right]}, \quad (6-55)$$

które jest uwikłaną funkcją  $T^\alpha(i)$ , czyli progów maskowania, który ma taką wartość, że pozwala na zamaskowanie szumu występującego w sygnale źródłowym, w zakresie częstotliwości odpowiadającym danemu pasmu krytycznemu.

Jak wynika z rozważań, które były prowadzone w roz. 5, pomnożenie składowych widma w danym paśmie krytycznym przez obliczoną wartość  $\alpha$ , spowoduje w wyniku podniesienie progów maskowania w całym zakresie widma, we wszystkich pasmach krytycznych. Przedstawione powyżej postępowanie pozwala odpowiedzieć na pytanie: ile razy należy wzmocnić wszystkie składowe użyteczne, aby uzyskać odpowiedni próg maskowania  $T^\alpha(i)$  w danym paśmie krytycznym. Podobne rozumowanie przeprowadzane jest dla wszystkich pasm krytycznych i w rezultacie uzyskuje się tyle współczynników  $\alpha$ , ile wykorzystano pasm krytycznych. Ze względów praktycznych obliczanie wartości  $\alpha$  ( $\alpha \geq 1$ ), przez którą należy pomnożyć moce chwilowe prążków użytecznych przeprowadza się przy wykorzystaniu metod numerycznych.

Zwiększenie poziomu prążków użytecznych i wyznaczenie nowych progów maskowania  $T^\alpha(i)$  pozwala na zakodowanie perceptualne sygnału z pominięciem występującego w nim szumu. Decyzję, czy dany prążek reprezentuje szum, czy sygnał użyteczny podejmuje się w sposób identyczny, w stosunku do opisanego w par. 6.6.4.1. A zatem: zmiennymi wejściowymi systemu decyzyjnego są opisane w tamtym paragrafie zmienne:  $\rho$  i  $\sigma$ , natomiast zmienną wyjściową jest parametr decyzyjny  $p$ , który przyjmuje wartość "1", gdy składowa widmowa reprezentuje sygnał użyteczny lub wartość "0", gdy składowa ta reprezentuje szum.

W chwili oddania niniejszej pracy do druku przedstawiona metoda znajduje się w stadium badań. Wiadomo już jednak, że można uzyskać przy jej pomocy bardzo dobre rezultaty redukcji szumu bez zauważalnego wpływu na naturalność brzmienia dźwięku.