

Rozdział 5

Inne

5.1 Analiza sygnałów wielowymiarowych

Stan badanego systemu może być odzwierciedlany zmianami więcej niż jednego sygnału. Do analizy rynku konieczne może być uwzględnienie zmian wartości wielu spółek i jednocześnie innych wskaźników gospodarczych, na potrzeby badania czynności elektrycznej mózgu (EEG) czy serca (EKG) zapisujemy przebiegi potencjału z wielu elektrod umieszczonych w różnych miejscach na głowie czy klatce piersiowej. *Sygnałem N -wymiarowym* nazwiemy przypisanie każdej chwili czasu t wektora wartości $\vec{s} = \{s_1, s_2 \dots s_N\}$, opisującego wartości przyjmowane przez każdy z mierzonych sygnałów $s_1(t), s_2(t) \dots s_N(t)$ w danej chwili t .

Pierwszą miarą użyteczną w tym przypadku będzie wspomniana już korelacja, opisująca podobieństwo dwóch mierzonych jednocześnie sygnałów. Istnienie opóźnień czasowych między sygnałami wykaże wspomniana również funkcja korelacji wzajemnej, mierząca podobieństwo dwóch sygnałów w funkcji przesunięcia w czasie jednego z nich. Z kolei funkcja koherencji opisuje podobieństwo dwóch sygnałów w funkcji częstości.

Trudniejszym zadaniem jest próba jednoczesnego uwzględnienia zależności między wszystkimi dostępnymi sygnałami. Stosuje się tu wiele metod, zarówno wywodzących się ze statystyki (jak analiza składowych głównych czy analiza czynnikowa) jak i opracowywanych specjalnie na potrzeby analizy sygnałów, jak wielowymiarowy model AR czy analiza składowych niezależnych. Dla przykładu opiszemy w skrócie niektóre z nich.

5.1.1 Analiza Składowych Głównych (PCA)

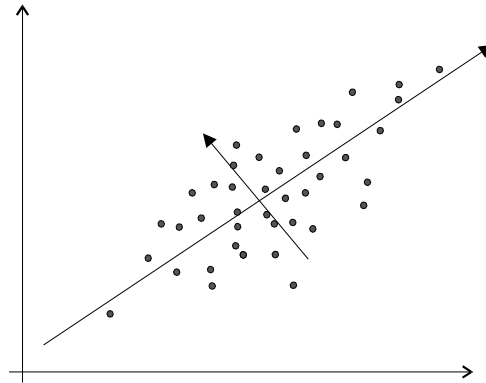
Analiza składowych głównych (*Principal Component Analysis*, PCA) znajduje bazę, w której macierz kowariancji c_{ij} sygnału wielowymiarowego \vec{s} przyjmuje postać diagonalną (baza Karhunen-Loève). Macierz transformacji B obliczamy zwykle znaną z algebry metodą:

1. Wartości własne macierzy kowariancji znajdujemy z równania $|C - \lambda I| = 0$:

$$\begin{vmatrix} c_{11} - \lambda & c_{12} & \dots & c_{1N} \\ c_{21} & c_{22} - \lambda & \dots & c_{2N} \\ & & \ddots & \\ c_{N1} & c_{N2} & \dots & c_{NN} - \lambda \end{vmatrix} = 0$$

2. wektory własne \vec{b}_i , odpowiadające wartościom własnym λ_i , spełniają $C\vec{b}_i = \lambda_i\vec{b}_i$.
3. normalizujemy \vec{b}_i . Wyznaczają one kierunki nowego układu współrzędnych, a złożona z nich macierz B odpowiada obrotowi diagonalizującemu macierz kowariancji C .

Celem jest zwykle redukcja wymiaru przez odrzucenie współrzędnych odpowiadających mniejszym wartościom λ_i lub też poszukiwanie nowych współrzędnych *per se* (rys. 5.1).



Rysunek 5.1: Kierunki składowych PCA w dwóch wymiarach

5.1.2 Analiza składowych niezależnych (ICA)

Analiza składowych niezależnych (*Independent Components Analysis*, ICA) to jedno z określeń dla metod rozwiązywania problemu tzw. ślepej separacji źródeł (*blind source separation*, BSS). Przyjęty model zakłada, że mamy do czynienia z następującą sytuacją: dane którymi dysponujemy (\vec{x} – np. zapisy z kilku mikrofonów) są liniową mieszaniną kilku statystycznie niezależnych sygnałów (\vec{s} – np. głosy kilku mówiących jednocześnie osób, tzw. *cocktail party problem*)

$$\vec{x} = A\vec{s}$$

A zwiemy macierzą mieszającą, a rozwiązania szukamy w postaci macierzy separującej B , takiej, że wektor sygnałów

$$\vec{y} = B\vec{x}$$

jest możliwie bliski (nieznanym) sygnałom \vec{s} . Wymóg niezależności statystycznej elementów \vec{y} wymaga uwzględnienia statystyk rzędów wyższych niż 2, czyli korelacji (używanych w PCA). Przetwarzanie wstępne polega często na wyzerowaniu statystyk do rzędu 2, czy odjęciu średniej i obrocie diagonalizującym macierz kowariancji (zwykle PCA). Uzyskanie w prosty sposób dekorrelacji ułatwia działanie procedur realizujących dalsze wymagania niezależności. Realizowane są one zwykle z pomocą sztucznych sieci neuronowych o specjalnie dobieranych regułach uczenia.

Metodę w obecnej postaci wprowadzono do analizy sygnału w roku 1995 i większość aktualnych publikacji i algorytmów znaleźć można w Internecie.

5.1.3 Wielowymiarowy model AR

Model AR (równanie 2.35) opisuje wartość sygnału w chwili t jako kombinację liniową jego wartości w chwilach poprzednich (oraz szumu). W przypadku wielowymiarowym możemy włączyć do tego opisu wartości wszystkich sygnałów s_i , czyli wektora $\vec{s}(t)$. Wielowymiarowy model AR (MVAR, *multivariate autoregressive*) można wówczas opisać wzorem:

$$\vec{s}(t) = \sum_{i=1}^p A(i)\vec{s}(t-i) + \vec{\epsilon}(t), \quad (5.1)$$

gdzie $\vec{\epsilon}(t)$ będzie wektorem szumów, zaś $A(i)$ będą macierzami współczynników modelu. Przechodząc do przestrzeni częstości otrzymamy:

$$\vec{s}(\omega) = A^{-1}(\omega)\vec{\epsilon}(\omega) = H(\omega)\vec{\epsilon}(\omega), \quad (5.2)$$

gdzie $H(\omega)$ jest macierzą przejścia. MVAR jest modelem typu „czarna skrzynka”, gdzie na wejściu występują szumy, na wyjściu sygnały, a system jest opisany przez macierz przejścia. Zawiera on informacje o własnościach widmowych sygnałów i związkach między nimi.

Na podstawie macierzy $H(\omega)$ można obliczyć macierz gęstości widmowej zawierającą widma mocy dla pojedynczych kanałów jak również funkcje wzajemnej gęstości mocy pomiędzy kanałami. Stosując

tego typu podejście, w którym wszystkie sygnały generowane przez pewien proces są rozpatrywane jednocześnie, można policzyć z macierzy spektralnej nie tylko koherencje zwykłe pomiędzy dwoma kanałami, ale również koherencje wielorakie opisujące związek danego kanału z pozostałymi i koherencje cząstkowe opisujące bezpośrednie związki między dwoma kanałami po usunięciu wpływu pozostałych kanałów. W przypadku gdy pewien kanał 1 będzie wpływał na kanały 2 i 3, obliczając koherencję zwykłą znajdziemy związek między 2 oraz 3, chociaż nie są one ze sobą bezpośrednio powiązane, natomiast koherencja cząstkowa nie wykaże związku między nimi.

Macierz $H(\omega)$ jest niesymetryczna, a jej wyrazy pozadiagonalne mają sens przyczynowości Grangera, co oznacza, że uwzględnienie wcześniejszej informacji zawartej w jednym z sygnałów zmniejsza błąd predykcji drugiego sygnału. Opierając się na tej własności zdefiniowano Kierunkową Funkcję Przejścia (DTF, *directed transfer function*) jako znormalizowany element pozadiagonalny $H(\omega)$. DTF opisuje kierunek propagacji i skład widmowy rozchodzących się sygnałów.

Otrzymamy w ten sposób całościowy opis zmian wszystkich sygnałów jednocześnie. Co ciekawe, obliczona na tej podstawie funkcja charakteryzująca zależności między sygnałami s_i (funkcja przejścia) nie jest symetryczna, w przeciwieństwie do np. korelacji. Dzięki temu może służyć wnioskowaniu nie tylko o sile zależności między poszczególnymi sygnałami składowymi, ale też o kierunku przepływu informacji między nimi. W przybliżeniu odpowiada to informacji, w którym z sygnałów struktury odpowiadające danej częstości pojawiają się wcześniej.

5.2 Sztuczne sieci neuronowe (ANN)

Artificial Neural Networks, ANN powstały w wyniku dążenia do stworzenia systemów naśladujących działanie mózgu. Mózg człowieka składa się z ok. $10^{11} - 10^{12}$ neuronów, każdy z nich posiada ok. 10^4 połączeń z innymi neuronami. Ich działanie w największym uproszczeniu można opisać jak następuje:

Potencjał czynnościowy wygenerowany przez neuron propaguje się przez wypustki (akson i dendryty) do innych neuronów. Potencjały docierające do każdego z neuronów sumują się, z wagami zależnymi od siły połączeń. Połączenia między neuronami mogą mieć różną siłę, czyli potencjał czynnościowy generowany ze stałą wartością może mieć różny wkład w zależności od tego, przez jakie połączenie dotarł. Jeśli suma tych potencjałów (po odjęciu wkładu połączeń hamujących) przekroczy ustalony dla danego neuronu próg, to generowany jest potencjał czynnościowy o stałej wartości. Potencjał czynnościowy wygenerowany przez neuron propaguje się...

Ogólnym modelem matematycznym, naśladującym opisane procesy, będzie

sieć złożona z prostych jednostek obliczeniowych, połączonych kanałami komunikacyjnymi zdolnymi do przenoszenia wartości numerycznych. Jednostki działają wyłącznie w oparciu o swe lokalne dane i wejścia z kanałów komunikacyjnych (połączeń).

Jednak bliższe badania nad mózgiem człowieka wykazały nieporównanie wyższy niż w przytoczonym opisie stopień złożoności, związany m. in. z wpływem szeregu procesów biochemicznych. I tak np.

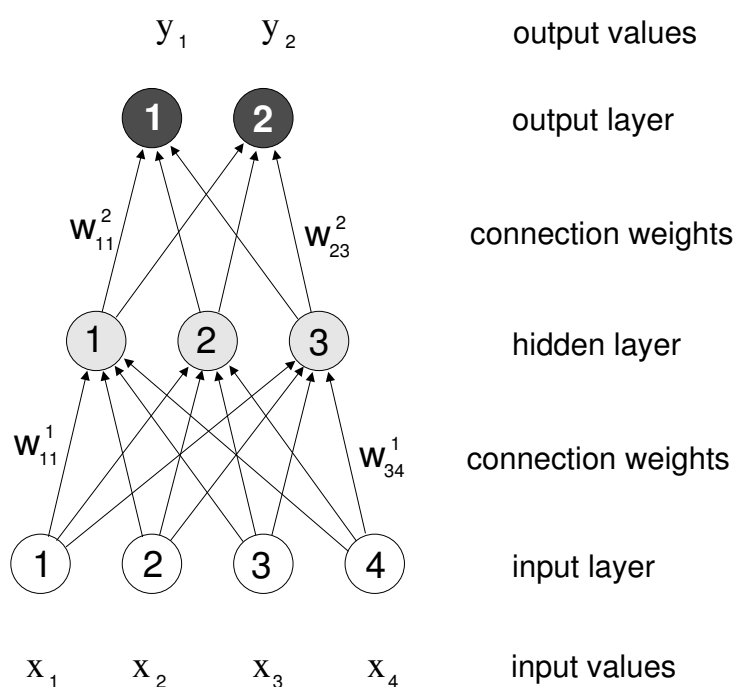
do przekazu impulsu z dendrytów do ciała neuronu konieczna jest obecność szeregu neurotransmiterów, czyli cząsteczek przenoszących impuls drogą chemiczną przez szczelinę synaptyczną między dendrytem a ciałem neuronu-adresata¹. Z kolei generacja potencjału czynnościowego, czyli jak się wydaje podstawowej jednostki przekazu informacji w układzie nerwowym, jest wynikiem złożonych procesów rządzących depolaryzacją błony neuronu. W stanie spoczynku wewnątrz neuronu wykazuje potencjał $-75\mu V$ względem otoczenia; jest on utrzymywany dzięki aktywności błony neuronu (pompa jonowo-sodowa), utrzymującej wyższe stężenie jonów Na^+ na zewnątrz oraz niższe stężenie Cl^- wewnątrz. Neurotransmitery uwolnione z synapsy neuronu wysyłającego impuls powodują otwarcie kanałów przepuszczających dodatnio naładowane jony sodu (Na^+) do wnętrza neuronu, co zmienia potencjał z $-75\mu V$ do $+55\mu V$. Ta depolaryzacja aktywuje kanały sodowe w sąsiednich częściach błony, co powoduje propagację pobudzenia.

¹wybiórce blokowanie neurotransmiterów przez niektóre leki psychotropowe czy narkotyki prowadzi m. in. do specyficznych zaburzeń działania mózgu

Przytoczony powyżej (z pominięciem roli jonów potasu) model Hodgkina-Huxleya (Nobel 1963) jest dziś uważany za bardzo uproszczony. Mimo to, jego stopień komplikacji tak dalece odbiega od sztucznych sieci neuronowych, że nie można ich traktować jako modelu działalności mózgu. Modelowanie procesów zachodzących w mózgu człowieka to osobna gałąź nauki. Za to właśnie dzięki drastycznym uproszczeniom sztuczne sieci neuronowe (ANN) zyskały efektywność obliczeniową, która w połączeniu z równoległością, tolerancją na błędy i atrakcyjną terminologią doprowadziła do rozpowszechnienia ANN w praktycznych zastosowaniach obliczeniowych.

Obszernym i aktualnym źródłem informacji o ANN są archiwa grupy dyskusyjnej `comp.ai.neural-nets` [12], w literaturze polskojęzycznej pierwszą klasyczną pozycją jest [14]. Poniżej przedstawimy w skrócie zdecydowanie najpopularniejszy typ sieci neuronowych.

5.2.1 Sieci warstwowe z propagacją wsteczną



Rysunek 5.2: Schemat sieci neuronowej z jedną warstwą ukrytą

Na rysunku 5.2 wagę połączenia przewodzącego pobudzenie od j -tego do i -tego neuronu w k -tej warstwie oznaczono w_{ij}^k . Suma pobudeń docierających w danej chwili do i -tego neuronu wyniesie $\sum_j w_{ij}^k n_j$. Przetwarzanie w neuronie polega na odjęciu od tej sumy charakteryzującego neuron progę μ_i i podzieleniu na wynik nieliniową funkcją Φ :

$$n_i = \Theta \left(\sum_j w_{ij}^k n_j - \mu_i \right) \quad (5.3)$$

W oryginalnym modelu Mc Culloha i Pittsa z roku 1943, $\Theta(\cdot)$ była funkcją progową:

$$\Theta(x) = \begin{cases} 0 & \text{dla } x < 0 \\ 1 & \text{dla } x \geq 0 \end{cases} .$$

Aktualnie najczęściej stosowaną formą nieliniowości jest funkcja logistyczna

$$\Theta(x) = \frac{1}{1 + e^{-\beta x}}$$

Tak więc działanie sieci z rys. 5.2 wygląda następująco:

- Sygnał wejściowy $\vec{x} = \{x_1, x_2, x_3, x_4\}$ podawany jest na neurony warstwy wejściowej.
- Aktywacja każdego z neuronów warstwy ukrytej obliczana jest na podstawie równania 5.3. Dla przykładu dla „środkowego” neuronu

$$n_2 = \frac{1}{1 + e^{-\beta(w_{21}^1 x_1 + w_{22}^1 x_2 + w_{23}^1 x_3 + w_{24}^1 x_4 - \mu_2)}}.$$

- Aktywacja neuronów warstwy wyjściowej, czyli odpowiedź sieci na sygnał \vec{x} , obliczana jest analogicznie jak w poprzednim kroku, tylko sygnały wejściowe x_i zastępują obliczone aktywacje neuronów warstwy ukrytej.

„Inteligencja” sieci zawarta jest w wagach w_{ij}^k połączeń między kolejnymi warstwami. *Clou* problemu polega na dobraniu tych wag tak, by realizowały interesujące nas odwzorowanie.

W klasycznym przypadku, nieznanie odwzorowanie mamy zadane z pomocą zestawu „przykładów” postaci (wektor wejściowy, prawidłowa klasyfikacja) — oznaczmy je (\vec{x}^i, \vec{r}^i) . Rozważmy dla przykładu konstrukcję sieci rozpoznającej pisane odręcznie litery. Zestaw przykładów stanowić będą mapy bitowe obrazów reprezentujących litery i ich prawidłowe klasyfikacje. Wagi dobierane są w procesie *uczenia sieci*. Cała procedura przebiega jak następuje:

1. Ustalamy *architekturę* sieci:

- rozmiar warstwy wejściowej jest zwykle zdeterminowany przez rozmiar wektora danych; np w przypadku zapisu pisanych odręcznie liter w postaci obrazów o rozmiarach 15x15 pixeli, rozmiar warstwy wejściowej wyniesie 225 neuronów,
- rozmiar i ilość warstw ukrytych są problemem otwartym; najczęściej używamy jednej warstwy ukrytej, której rozmiar dobieramy empirycznie,
- rozmiar warstwy wyjściowej zależy od sposobu kodowania informacji, którą chcemy na niej odczytywać; w przypadku rozpoznawania dużych liter alfabetu polskiego najlepiej wybrać 35 neuronów wyjściowych: literę „A” reprezentować będzie jedynka na pierwszym neuronie i zera na pozostałych, „B” — jedynka na drugim neuronie itd.

2. Inicjalizujemy wagi połączeń w_{ij}^k przypisując im małe liczby losowe — warto zapamiętać, że w związku z tym każde uruchomienie *tego samego algorytmu na tych samych danych* może dać w wyniku *inną* sieć.

3. Wybrany losowo ze zbioru uczącego wektor \vec{x}^l prezentujemy na wejściu sieci i obliczamy odpowiedź $\vec{n}^{\text{wyjść}}$.

4. Wartości odczytane z neuronów wyjściowych $n_i^{\text{wyjść}}$ porównujemy z „prawidłową odpowiedzią” r_i^l . Błąd średniokwadratowy obliczamy jako

$$\sum_i (n_i^{\text{wyjść}} - r_i^l)^2 \quad (5.4)$$

5. Modyfikujemy wagi połączeń kolejnych warstw proporcjonalnie do ich wkładu w dany wynik (propagacja wsteczna błędu)².

6. Punkty 3–5 powtarzamy na losowo wybieranych przykładach aż do uzyskania zamierzonego efektu.

Ostatni punkt wymaga podjęcia wysoce nietrywialnej decyzji, od której m.in. zależeć będzie zdolność sieci do *generalizacji* wiedzy podanej w postaci przykładów.

²Wzór na propagację wsteczną błędu otrzymujemy wypisując *explicite* wzór na wartości otrzymane na neuronach wyjściowych w zależności od wag i wartości wejściowych, i różniczkując go po wagach połączeń między kolejnymi warstwami.

Generalizacja

Głównym celem opisanej powyżej przykładowej konstrukcji nie było rozpoznawanie liter ze zbioru przygotowanego do uczenia sieci (to możnaby osiągnąć znacznie prostszymi metodami), lecz prawidłowa klasyfikacja *nowych*, nie widzianych przez sieć przypadków. Uzyskanie takiej własności wymaga prawidłowej generalizacji wiedzy „widzianej” w procesie uczenia. Doskonała klasyfikacja przypadków prezentowanych sieci w procesie uczenia nie implikuje wcale poprawnego zachowania sieci na nowych elementach — czasami dla uzyskania prawidłowej generalizacji konieczna jest wręcz pewna tolerancja w stosunku do wyników uzyskiwanych na zbiorze uczącym, co ilustruje rys. 5.3. Uzyskaniu poprawnej generalizacji służy m. in. dzielenie zbioru przykładów przeznaczonych do uczenia sieci na zbiór uczący i zbiór kontrolny, nie prezentowany sieci w procesie uczenia (punkty 3–5), i służący wyłącznie do weryfikacji zdolności generalizacji.

wykres :-)

Rysunek 5.3: Analogia tłumacząca kompromis między wynikami sieci na zbiorze uczącym a generalizacją. N punktów losowanych z równania $y_i = ax_i + b + \epsilon_i$, gdzie ϵ — szumowe odchylenia od prostej, można idealnie przybliżyć wielomianem stopnia N . Jednak przebieg wielomianu poza przedstawionymi punktami będzie odbiegał od prostej znacznie bardziej niż w przypadku, gdy kosztem pewnych niedokładności odtworzenia danych punktów zdecydujemy się na aproksymację funkcją liniową.

Opisany powyżej schemat to zaledwie szkielet metodologii stosowanej w rozwiązywaniu konkretnych problemów. Nawet w swej podstawowej postaci zawiera dwie arbitrarne wybierane stałe (współczynniki uczenia i bezwładności) od których zależy jego działanie. Metody uczenia stosowane w praktyce oferują ogromną ilość modyfikacji polepszających zbieżność i generalizację.

5.3 Algorytmy Genetyczne

Algorytmy genetyczne (*Genetic Algorithms, GA*) lub populacyjne to stochastyczna metoda przeglądu przestrzeni rozwiązań, oparta na mechanizmach rządzących ewolucją organizmów żywych. Praktyczne zastosowanie w analizie sygnałów jest na razie dość ograniczone ze względu na wysoki koszt obliczeniowy. Pomimo tego, ze względu na potencjalnie ogromne możliwości metody i szereg spektakularnych zastosowań w dziedzinach pokrewnych, omówimy poniżej w skrócie klasyczny AG.

Za początek badań w tej dziedzinie można uznać klasyczną monografię Hollanda [9]. Więcej informacji o teorii i zastosowaniach można znaleźć np. w książce Goldberga [7].

5.3.1 Elementarny Algorytm Genetyczny

Pierwszym problemem, który musimy rozwiązać przed zastosowaniem AG, jest wybór kodowania elementów przestrzeni rozwiązań — zwykle w postaci ciągu bitów, reprezentujących liczby (lub litery) opisujące dopuszczalne rozwiązanie.

Najbardziej charakterystyczną cechą AG jest fakt, że dobór rozwiązań przebiega właściwie bez związku ze znaczeniem kodowanych ciągów. Jedyнным łączem jest tu funkcja przystosowania, odpowiednik funkcji kosztu w sieciach neuronowych (z przeciwnym znakiem). Dla działania algorytmu konieczne jest obliczanie „jakości” każdego z zaproponowanych rozwiązań.

Po ustaleniu sposobu kodowania i liczenia wartości funkcji przystosowania, optymalizacja polega na powtarzaniu następujących kroków:

- reprodukcja
- krzyżowanie
- mutacja

reprodukcja w każdej pętli (pokoleniu), ciągi kodowe o wyższych wartościach funkcji przystosowania mają większą szansę na przejście do następnego pokolenia. Można to zrealizować z pomocą losowania ciągów z prawdopodobieństwem proporcjonalnym do względnej wartości funkcji przystosowania. Wylosowane ciągi przechodzą do następnego pokolenia, w którym odpowiednikiem rozmnażania jest krzyżowanie

krzyżowanie krzyżowane są losowo dobrane pary ciągów kodowych. Krzyżowanie polega na zamianie części ciągów poczynając od losowo wybranego punktu. Np. ciągi

0011010011101011 i

1010011000000111,

skrzyżowane poniżej szóstej pozycji, dadzą:

0011011000000111 i

1010010011101011

mutacja polega na zamianie bitu na losowo wybranej pozycji ciągu; np. pierwszy ciąg z powyższego przykładu, zmutowany na trzeciej pozycji, przyjąłby postać 0001010011101011.

Dla przykładu przyjmijmy, że poszukujemy parametrów funkcji Gabora, wyjaśniającej największy procent energii danego sygnału (iteracja³ algorytmu MP, rozdział 4.2). Jak w każdym zastosowaniu AG, pierwszym krokiem będzie wybór kodowania w przestrzeni rozwiązań. W tym przypadku przestrzeń rozwiązań to zbiór M czwórek parametrów opisujących znormalizowane funkcje Gabora — pozycja, częstość, skala i faza. Jeśli każdą z tych liczb zakodujemy jako 2-bajtową liczbę całkowitą⁴, to rozwiązanie najprościej zakodować jako $4 * 2 * 8 = 64$ -bitowy ciąg. Pierwszych 16 bitów możemy interpretować jako zapis pozycji, kolejne 32 — częstości itd.⁵

Skoro celem poszukiwania jest wyjaśnienie największej części energii sygnału, to za „przystosowanie” ciągu kodowego przyjmijmy wartość energii tłumaczonej przez opisywaną przez niego funkcję.

³Znacznie ciekawsze jest wykorzystanie GA do znajdowania zestawów M funkcji wyjaśniających największy procent energii sygnału. Tu ograniczymy się do przypadku łatwiejszego w opisie.

⁴W praktyce stosujemy liczby zmiennoprzecinkowe.

⁵Jeśli na 16 bitach zapiszemy liczbę z przedziału $(0, \text{MAXINT})$, gdzie $\text{MAXINT} = 2^{16} - 1 = 65535$, to dla sygnału długości N punktów i wylosowanej czwórki liczb (u, ω, s, ϕ) można rozpatrywać funkcje Gabora postaci

$$g_{(u, \omega, s, \phi)}(t) = K(u, \omega, s, \phi) e^{-\pi \left(\frac{t * \text{MAXINT} - u}{N * s} \right)^2} \sin \left(2\pi \frac{\omega}{\text{MAXINT}} \left(t - \frac{u * N}{\text{MAXINT}} \right) + 2\pi \frac{\phi}{\text{MAXINT}} \right)$$

gdzie $K(u, \omega, s, \phi)$ — stała normalizująca energię do 1 (por. równanie (4.8)).

Tak więc w pierwszym kroku losujemy populację przypadkowych ciągów 128 bitów, które interpretujemy jak zapis parametrów funkcji Gabora. Dla każdej z tak opisanych funkcji liczymy funkcję przystosowania. Wybieramy z nich najlepsze (np. metodą losowania proporcjonalnego), po czym z uzyskanego w ten sposób następnego pokolenia losowo dobieramy pary do krzyżowania. Następnie (oczywiście również losowo) wybieramy „osobniki” do mutacji. Gdy spełnimy kryteria zatrzymania algorytmu (np. względny brak poprawy w kolejnych pokoleniach), z ostatniego pokolenia wybieramy ciąg o największej wartości funkcji przystosowania.

...