

Grzegorz TYMA

Instytut Elektroniki
Politechniki Śląskiej

WYBRANE SIECI NEURONOWE I ICH ZASTOSOWANIA

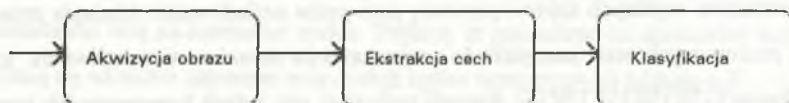
Streszczenie. Sieci neuronowe dzięki swoim cechom, takim jak równoległość przetwarzania, zdolność do uczenia i generalizacji oraz odporność na uszkodzenie cieszą się coraz większym zainteresowaniem naukowców. Dzięki staraniom wielu osób przestają one być przedmiotem rozważań tylko teoretycznych, nie są już tylko „sztuką dla sztuki”, a znajdują praktyczne zastosowania. Powstało wiele struktur sieci neuronowych. Część z nich naśladuje biologiczne pierwowzory. Istnieje jednak dylemat, czy takie naśladownictwo nie da podobnych wyników jak starania ludzi w naśladowaniu ptaków. Dlatego powstają także sieci neuronowe, które jak się obecnie wydaje, niewiele mają wspólnego z biologicznymi sieciami. Artykuł ten przedstawia reprezentantów tych dwóch rodzajów sieci neuronowych. Zawiera on także przykład zastosowania sieci do rozpoznawania cyfr.

SELECTED NEURAL SETS AND THEIR APPLICATIONS

Summary. Distributed computing, ability to learn by example and generalisation, fault tolerance are these features of neural networks which are of great interest of scientists in this domain. Because of work of many people artificial neural networks it is not only theory, it is also implementation. A lot of neural network's structures have been invented. Some of them try to follow biological prototypes. The question is if it is a correct way. The history of aircraft gave us example that sometimes artificial structures are better. This paper presents examples of neural networks which belongs to the different categories. An example of implementation in numeral recognition has been included.

Wprowadzenie

Sieci neuronowe bardzo często wykorzystywane są do zadań rozpoznawania obrazów. Schemat blokowy procesu rozpoznawania obrazów składa się z trzech elementów (rys.1.): rejestratora, ekstraktora cech oraz klasyfikatora [9][14][16][27][34][38].



Rys.1. Elementy systemu rozpoznawania obrazów
Fig.1. The model of pattern recognition

Rejestrator jest urządzeniem zamieniającym obraz wejściowy w sygnał, który następnie przetwarzany jest przez maszynę. Ekstraktor cech wybiera tę część informacji z sygnału wejściowego, która niezbędna jest do poprawnej klasyfikacji. Klasyfikator przypisuje określony zbiór cech do jednej z wcześniej zdefiniowanych kategorii. W ostatnim okresie powstało i rozwinęło się wiele technik klasyfikacji wzorów w różnych dziedzinach zastosowań. Wśród nich można wymienić teorię Bayesa, metody kwantyzacji wektorów, metody najbliższego sąsiedztwa. W ciągu ostatniej dekady metody statystyczne odgrywały dominującą rolę [7]. Ostatnio jednak coraz większym zainteresowaniem cieszą się techniki wykorzystujące sieci neuronowe [4][18][20][24][30][33].

Systemy z sieciami neuronowymi można podzielić na dwie grupy. Pierwsza z nich wykorzystuje klasyczne metody ekstrakcji cech, a sieci neuronowe używane są tylko jako klasyfikator. Druga grupa wykorzystuje sieci neuronowe zarówno do ekstrakcji cech, jak i do klasyfikacji.

Możemy wyróżnić cztery klasy systemów rozpoznawania opartych na sieciach neuronowych:

- sieci typu back-propagation, które funkcjonują bądź jako klasyfikatory, bądź jako kompletny system z wbudowaną niezależnością na przekształcenia izomorficzne obrazów [1][3][29][31];
- Neocognitron, sieć hierarchiczna (kolejne warstwy analizują coraz bardziej);
- ART, sieć neuronowa oparta na teorii rezonansu adaptacyjnego[15];
- sieci hybrydowe, ekstraktor i klasyfikator realizowane są przy użyciu sieci neuronowych, zwykle różnych typów [5][8][17][19][25][26].

W artykule przedstawiono system wykorzystujący sieć neuronową hybrydową do zadania rozpoznawania cyfr pisanych odręcznie. System ten składa się z sieci neuronowej trzeciego rzędu [37] do ekstrakcji cech i sieci Learning Vector Quantization [23], działającej jako klasyfikator. Zamieszczono także przykład sieci Radial Basis Functions [25].

Sieci neuronowe wyższych rzędów

Sieci neuronowe wyższych rzędów powstały jako próba naśladowania działania prawdziwych sieci neuronowych oraz narzędzie do rozpoznawania obrazów, niewrażliwe na transformacje obrazu [12][13][35][37][36]. Rozwój tych sieci jest jednak hamowany ich rozmiarami i kłopotami z realizacją sprzętową.

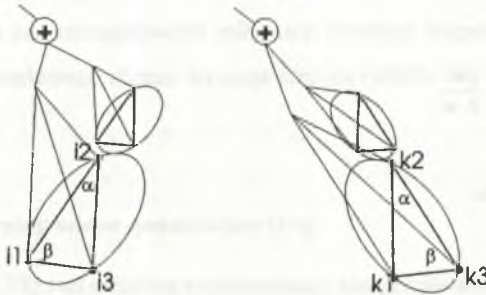
Argumentami funkcji aktywnej sieci wyższych rzędów są iloczyny sygnałów wejściowych. Sygnał wyjściowy o_k może być zapisany w postaci:

$$o_k = \left[\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \dots \sum_{l=0}^{n-1} W_{ij...lk} X_i X_j \dots X_l \right],$$

gdzie: $f[]$ funkcja aktywacji,

$w_{ij...lk}$ wagi połączeń,

x_i sygnał wejściowy.



Rys. 2. Sposób wyznaczania cech znaków z użyciem sieci neuronowej 3 rzędu
Fig. 2. Rules for constructing feature vector by third order network

Dla sieci neuronowej trzeciego rzędu sygnał wyjściowy opisany jest zależnością:

$$o_k = f \left[\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{l=0}^{n-1} W_{ijlk} X_i X_j X_l \right]$$

gdzie: $f[]$ funkcja aktywacji,

$w_{ij...lk}$ wagi połączeń,

x_i sygnał wejściowy.

Sieci neuronowe trzeciego rzędu mogą dawać na wyjściu sygnał niezależny od trzech przekształceń obrazu: skalowania, przesunięcia i obrotu. Rysunek 2 ilustruje, w jaki sposób może to być osiągnięte. Każdy zewzórów po szkieletowaniu jest pokrywany trójkątami, których wierzchołki leżą na kontuarze znaku. Trójkąty te podzielono na skończoną liczbę klas, które różnią się od siebie zakresem miar dwóch kątów wewnętrznych trójkąta α, β .

Dla każdego rozpoznawanego znaku obliczana jest znormalizowana liczba trójkątów w każdej klasie. Wektor zawierający wymienione liczby jest wektorem cech używanym przez klasyfikator.

Obliczanie długości wektora cech

W zależności od przyjętej szerokości przedziałów klasyfikowanych trójkątów otrzymujemy wektory cech o różnej długości. Spróbujmy określić tę długość w funkcji szerokości przedziału.

Oznaczmy kąty w trójkącie jako: α, β, δ . Muszą one spełniać zależność: $\alpha \leq \beta \leq \delta$. Wynika z tego, iż: $0 < \alpha \leq \frac{\pi}{3}, 0 < \beta < \frac{\pi}{2}$, gdyż: $\alpha + \beta + \delta = \pi$

Szerokość przedziału w została zdefiniowana tak, że:

$$W = \frac{\pi}{3 \cdot \omega}, Q = \frac{\pi}{2 \cdot \omega},$$

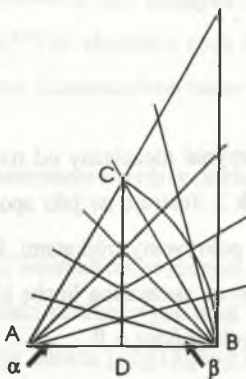
gdzie W i Q są liczbami naturalnymi.

Zatem:

$$k \cdot \omega \leq \alpha < (k+1)\omega, \text{ gdzie: } 0 \leq k \leq W-1$$

oraz

$$l \cdot \omega \leq \beta < (l+1)\omega, \text{ gdzie: } 0 \leq l \leq Q-1$$



Rys.3. Wyznaczanie liczby klas trójkątów
Fig.3. Calculating the number of classes

Na rys.3 przedstawiono możliwe klasy trójkątów dla danego ω .

Trójkąty, których kąty spełniają wymienione warunki, mają dwa wierzchołki w punktach A, B oraz wierzchołek mieszczący się w konturze BCD. Policzmy ile jest takich trójkątów.

W konturze ABC znajduje się N_{ABC} trójkątów:

$$N_{ABC} = 2 \cdot \sum_{k=0}^{\left(\frac{W}{2}-1\right)} (Q-k) = 2 \cdot \sum_{k=0}^{\left(\frac{Q}{3}-1\right)} (Q-k) = Q \cdot \frac{5Q+3}{9}$$

Zatem w konturze BCD mieści się N_1 trójkątów (czynnik 2 wynika z tego, iż rozróżniamy dwa trójkąty podobne, z których jeden obrocony jest o kąt π):

$$N_1 = (N_{ABC} - N_{ACD}) = 2Q \frac{Q+2}{3}$$

Dla $\omega = \pi/12$ otrzymujemy 32- elementowy wektor cech.

Istotnym problemem, jaki należy rozwiązać przy stosowaniu tego rodzaju sieci, jest problem kombinatorycznego wzrostu liczby połączeń wraz ze wzrostem rzędu sieci i rozmiarem analizowanego obszaru. Jednym ze sposobów zmniejszenia tej liczby połączeń jest ograniczenie obszaru okna, w którym jest analizowany obraz. Analogię tego rozwiązania można znaleźć w biologii. U wielu ssaków, także u człowieka, siatkówka oka nie jest jednorodna, dostarcza ona szczegółowych informacji o małych fragmentach analizowanej sceny. Identyfikacja kompletnego obrazu wymaga szeregu ruchów oka i koncentracji na wielu fragmentach sceny.

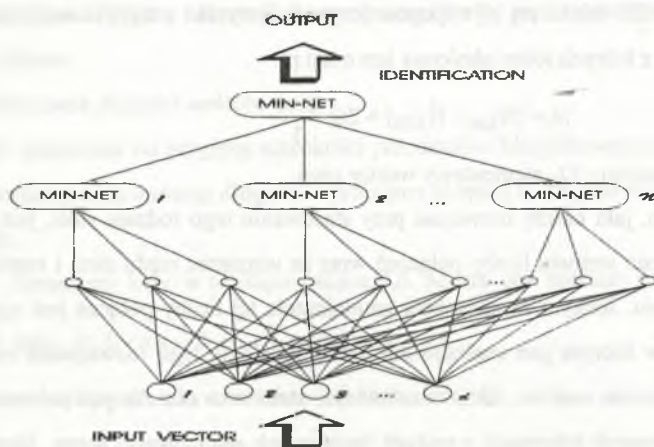
Sieć learning vector quantization (lvq)

Sieć LVQ jest siecią implementacją klasyfikatora najbliższego sąsiedztwa [21][22][23]. Każda klasa w przestrzeni cech reprezentowana jest przez wektory kodowe.

Zbiór wszystkich wektorów kodowych tworzy tzw. książkę kodową. Niech m_i będzie wektorem należącym do książki kodowej, który reprezentuje i -tą klasę w przestrzeni cech.

Oznaczmy przez m_c wektor kodowy m_i leżący najbliżej wektora wejściowego x :

$c = \arg \min_i \|x - m_i\|$. Wartości wektorów kodowych m_i , które minimalizują prawdopodobieństwo błędnej klasyfikacji, znajdowane są w iteracyjnym procesie uczenia.



Rys.4. Budowa sieci LVQ (d-wymiarowy wektor wejściowy i n klas)
Fig. 4. Structure of the LVQ network

Niech $x(t)$ i $m_i(t)$ będą odpowiednio wektorem uczącym i wektorem kodowym w dyskretnej dziedzinie czasu. Mając dane wartości startowe m_i możemy opisać przebieg uczenia LVQ1 następującymi równaniami:

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)], \text{ jeśli } x \text{ i } m_c \text{ należą do tej samej klasy,}$$

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)], \text{ jeśli } x \text{ i } m_c \text{ należą do różnych klas,}$$

$$m_i(t+1) = m_i(t), \text{ jeśli } i \neq c$$

$0 < \alpha(t) < 1$, $\alpha(t)$ mogą być stałe lub maleć monotonicznie w czasie.

Wykorzystywane są także modyfikacje tego algorytmu: procedury LVQ2 i LVQ3.

W LVQ2 modyfikowane są dwa najbliższe wektory z książki kodowej m_i i m_j . Jeden z nich musi należeć do tej samej klasy co wektor uczący x , a drugi do innej klasy. Ponadto x musi leżeć w tzw. oknie, które znajduje się w obszarze granicznym między m_i i m_j . Załóżmy, że d_i i d_j są odpowiednio odległościami euklidesowymi x od m_i i m_j . Mówimy, że x znajduje się w oknie o szerokości w , jeśli:

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \text{ gdzie } s = \frac{1-w}{1+w}$$

Algorytm LVQ2 opisany jest wzorami:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)],$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)],$$

gdzie: m_i i m_j są najbliższe x oraz x i m_j należą do tej samej klasy, x i m_i do różnych klas.

Ponadto x musi leżeć wewnątrz okna.

Algorytm LVQ3 dany jest wzorami:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)],$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)],$$

gdzie: m_i i m_j są najbliższe x oraz x i m_j należą do tej samej klasy, x i m_i do różnych klas,

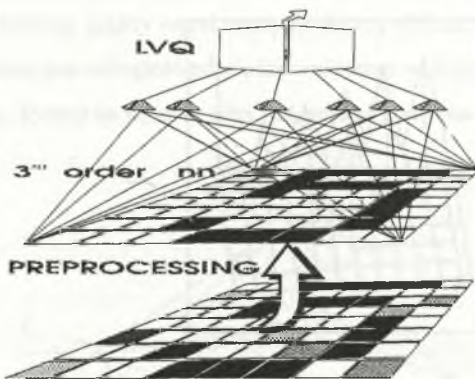
x musi leżeć wewnątrz okna oraz:

$$m_k(t+1) = m_k(t) + \epsilon \alpha(t)[x(t) - m_k(t)]$$

dla $k \in \{i, j\}$, jeśli x, m_i , i m_j należą do tej samej klasy.

Opis systemu

System do rozpoznawania cyfr składa się z następujących części: sprzętu do akwizycji dsanych, modułu przetwarzania wstępnego (filtracja, szkieletowanie), sieci neuronowej trzeciego rzędu (ekstraktor cech) oraz sieci LVQ (klasyfikator)(rys.5).



Rys.5. System do rozpoznawania znaków
Fig.5. The recognition system

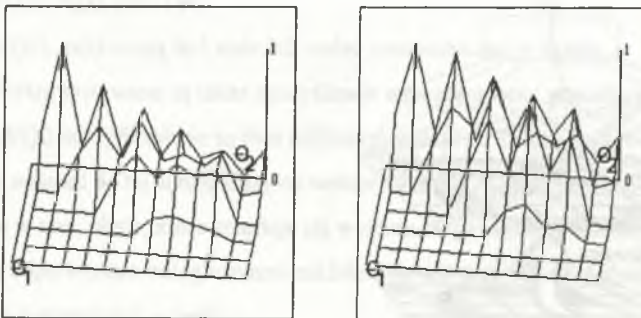
Każda z cyfr reprezentowana jest przez mapę bitową o rozmiarach 16x16 punktów. Każdy piksel może przyjmować dwie wartości: 0 i 1. Na rys.6 przedstawiono fragment zbioru z rozpoznawanymi cyframi.



Rys. 6. Fragment zbioru z odręcznie pisanymi cyframi
Fig. 6. A fragment of the set of handwritten numerals

Dla każdego wzoru obliczane są, przy użyciu sieci trzeciego rzędu, wektory cech (rys.7).

Dla przyjętej tolerancji $\omega = \pi/12$ długość wektora cech wynosi 32.



Rys.7. Wektory cech dla dwóch cyfr 8
Fig.7. Features vectors for two digits 8

Obliczone wektory cech poddawane są klasyfikacji przy użyciu sieci LVQ. Do uczenia sieci LVQ używane były algorytmy LVQ1, LVQ2 i LVQ3.

Książka kodowa sieci LVQ zawiera 40 wektorów kodowych, każdy o długości 32. W celu redukcji rozmiarów sieci trzeciego rzędu zastosowano okienkowanie obrazu na wejściu systemu. Przetestowano trzy rozmiary okien 16x16, 8x8 i 4x4 piksele.

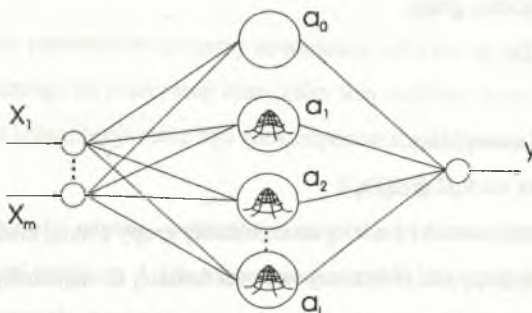
Tabela 1 prezentuje osiągnięte współczynniki poprawnej klasyfikacji dla różnych rozmiarów okien.

Tabela 1

Rozmiar okna [liczba okien]	Poprawne klasyfikacje
16 by 16 , [1]	98%
8 by 8 , [4]	93%
4 by 4 , [49]	86%
4 by 4 , [16]	77%

Sieć radial basis function (RBF)

Sieci RBF dokonują klasyfikacji korzystając z tzw. funkcji rdzeniowych, będących najczęściej funkcjami gaussowskimi [6][32]. Funkcje te umieszczane są w węzłowych punktach przestrzeni wejściowej. Ważone sumy wyjść funkcji rdzeniowych odwzorowują przestrzeń wejściową w zbiór zdefiniowanych uprzednio klas. Sieć RBF składa się z warstwy wejściowej, warstwy zawierającej węzły z gaussowskimi funkcjami aktywacji oraz liniowej warstwy wyjściowej. Każdy węzeł warstwy ukrytej oblicza wartość funkcji gaussowskiej, której argumentem jest odległość pomiędzy wektorem wejściowym a wartością średnią funkcji gaussowskiej. Sygnał na wyjściu sieci jest liniową kombinacją wartości wyjściowych warstwy ukrytej.



Rys. 8. Struktura sieci RBF

Fig.8. RBF network architecture

Sieć RBF realizuje funkcję f , która aproksymuje odwzorowanie:

$$f(\mathbf{x}_p) = y_p \quad \text{dla } p=1 \dots P,$$

gdzie: P jest liczbą par uczących (\mathbf{x}_p, y_p) .

Można to zapisać w postaci:

$$f(x) = \sum_{i=0}^I a_i \phi(\|x - c_i\|), \quad i=0 \dots I,$$

gdzie: $(I+1)$ jest liczbą węzłów w warstwie ukrytej, c_i jest wartością średnią dla i -tego węzła funkcją gaussowską.

Funkcja gaussowska $\phi(\cdot)$ jest zdefiniowana:

$$\phi(\|x - c_i\|) = \frac{1}{(2\pi)^{d/2} |S|^{1/2}} \exp\left[-\frac{1}{2}(x - c_i)^T S^{-1}(x - c_i)\right],$$

gdzie: d - wymiar wektora wejściowego,

S - macierz kowariancji,

$|S|$ - wyznacznik macierzy kowariancji.

Uczenie sieci RBF polega na:

- określeniu konfiguracji sieci,
- znalezieniu wartości średnich i macierzy kowariancji dla funkcji gaussowskich,
- określeniu wag połączeń pomiędzy warstwą ukrytą a wyjściową.

Bardzo ważnym etapem jest ustalenie liczby węzłów w warstwie ukrytej oraz znalezienie wartości średnich (położenie środków funkcji gaussowskich). Do wyznaczenia tych parametrów wykorzystano algorytm analizy skupień.

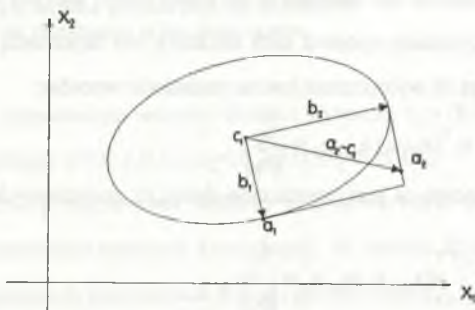
Algorytm analizy skupień [39]

1. Przypisz każdemu wektorowi uczącemu grupę.
2. Losowo ponumeruj grupy ($K=1..C$).
3. Wybierz pierwszą grupę ($K=1$).
4. Znajdź inną grupę należącą do tej samej klasy.
5. Połącz te dwie grupy i znajdź nową wartość średnią.
6. Oblicz odległość d_{op} od wyznaczonej wartości średniej do najbliższej grupy z innej klasy.
7. Oblicz odległość R (promień danej grupy) od obliczonej wartości średniej do najdalszej grupy, która podlegała połączeniu (z tej samej klasy).

8. Jeśli $d_{op} > aR$ (gdzie a jest stałą z przedziału $\langle 1,3 \rangle$), połączenie dokonane w kroku 5 jest akceptowane i przechodzimy do kroku 4 (wartość K odpowiada nowo utworzonej grupie, $C=C-1$). Jeśli warunek $d_{op} > aR$ nie jest spełniony, połączenie nie jest dokonywane i przechodzimy do kroku 4 (wartości K i C nie są zmieniane). Powtarzaj algorytm od kroku 4 do 8 dla wszystkich grup, następnie zwiększ K o jeden.

9. Powtarzaj kroki od 4 do 8 aż spełniony będzie warunek $K=C$.

Efekt działania tego algorytmu jest znalezienie liczby funkcji gaussowskich oraz ich wartości średnich.



Rys.9. Wyznaczanie macierzy kowariancji

Fig.9. Covariance matrix estimation

Następnie należy obliczyć macierze kowariancji. Kontur stałej wartości dla wielowymiarowej funkcji gaussowskiej tworzy elipsoidę w przestrzeni wielowymiarowej. Jej kształt zależy od macierzy kowariancji. Należy tak dobrać parametry macierzy kowariancji, aby:

- osiągnąć jak największą generalizację,
- obszary wspólne dla funkcji gaussowskich reprezentujących różne klasy były jak najmniejsze.

Dobór parametrów macierzy kowariancji odbywa się przy wykorzystaniu wektora uczącego należącego do przeciwnej klasy, który leży najbliżej danej wartości średniej.

Macierz kowariancji może być zdekomponowana do postaci:

$$S = Q\Lambda Q^{-1},$$

gdzie wartości własne i wektory własne macierzy kowariancji są odpowiednio wartościami na przekątnej macierzy Λ i kolumnami macierzy Q . Do znalezienia wartości własnych i wektorów własnych macierzy kowariancji wykorzystano procedurę ortogonalizacji Gramma-

Shmidta. Wektory własne określają kierunki osi elipsoid, a pierwiastki wartości własnych ich długości.

Rozpatrzmy przypadek dwuwymiarowy.

Wyznaczanie parametrów macierzy kowariancji (rys.9)

1. Znajdujemy najbliższy do c_i (c_i - wartość średnia dla i -tej funkcji gaussowskiej) wektor uczący należący do innej klasy (oznaczymy go jako a_i).

2. Wyznaczamy pierwszą oś:

$$b_1 = a_i - c_i, e_1 = -\frac{b_1}{\|b_1\|}.$$

3. Szukamy pozostałych wektorów uczących nie należących do danej klasy, których rzuty na kierunek e_1 są mniejsze niż $\|b_1\|$. Wybieramy spośród nich ten, który ma najkrótszą normalną do c_1 (oznaczamy go jako a_2). Druga oś wyznaczana jest na podstawie wzorów:

$$b_2 = (a_2 - c_i) - e_1^T (a_2 - c_i) e_1, e_2 = \frac{b_2}{\|b_2\|}$$

Dla M -wymiarowego przypadku procedura ta jest powtarzana dotąd, aż znalezionych zostanie M osi. Zatem:

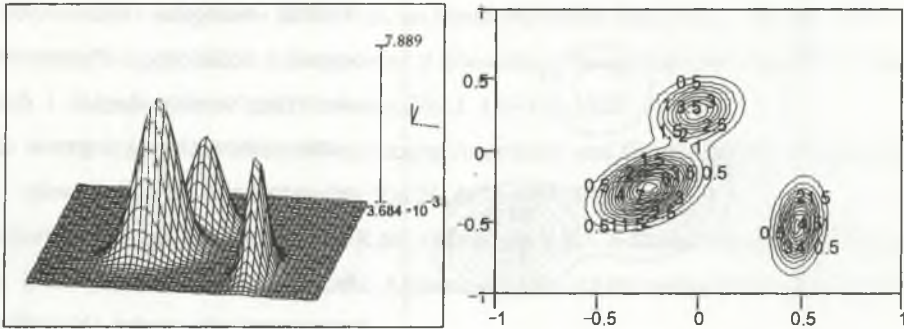
$$b_j = (a_j - c_i) - \sum_{k=1}^{j-1} e_k^T (a_j - c_i) e_k, e_j = \frac{b_j}{\|b_j\|},$$

ponadto:

$$\|e_k^T (a_j - c_i)\| < \|b_k\| \quad \text{dla } k=1, 2, \dots, j-1 \text{ (rzut mniejszy od } \|b_k\|),$$

$\|b_j\|$ nie może być zero; jeżeli zachodzi taki przypadek, to uwzględniany jest drugi najbliższy wektor uczący z klasy przeciwnej. Wagi a_i łączące warstwę ukrytą z wyjściową wyznaczane są za pomocą jednej z metod optymalizacyjnych.

Przeanalizujmy proces sieci oraz jej działanie na przykładzie. Wygenerujmy dwuelementowe wektory wejściowe, które należą do trzech klas o rozkładach normalnych z zadanymi wartościami średnimi i wariancjami (rys.10).



Rys.10. Rozkład gęstości prawdopodobieństwa
Fig. 10. Distribution of the input vectors

Odpowiednio wartości średnie wynoszą: $c_1 = (0,0,5)$, $c_2 = (-0,5,-0,5)$, $c_3 = (0,8,-0,8)$ a wariacje $\sigma_1 = (0,2,0,2)$, $\sigma_2 = (0,2,0,1)$, $\sigma_3 = (0,1,0,3)$.

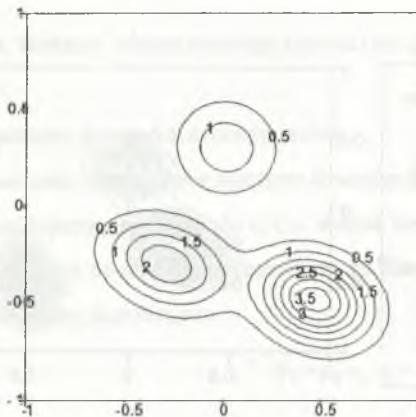
Tak przygotowane wektory wejściowe użyto w algorytmie analizy skupień i algorytmie wyznaczania macierzy kowariancji. W efekcie dobrano trzy węzły w warstwie ukrytej o następujących parametrach funkcji gaussowskich:

$$\begin{aligned} c_1 &= \begin{pmatrix} 0,024453 \\ 0,510798 \end{pmatrix} & S_1 &= \begin{pmatrix} 0,089 & -0,004 \\ -0,004 & 0,078 \end{pmatrix} \\ c_2 &= \begin{pmatrix} -0,481154 \\ -0,501910 \end{pmatrix} & S_2 &= \begin{pmatrix} 0,117 & -0,024 \\ -0,024 & 0,067 \end{pmatrix} \\ c_3 &= \begin{pmatrix} 0,7888640 \\ -0,80753 \end{pmatrix} & S_3 &= \begin{pmatrix} 0,096 & -0,016 \\ -0,016 & 0,062 \end{pmatrix} \end{aligned}$$

Mając wyznaczone parametry warstwy ukrytej, dobierzmy wagi połączeń do warstwy wyjściowej. Nadajemy poszczególnym klasom wartości $y_p : 1, 2, i 3$. Minimalizując błąd średniokwadratowy:

$$E = \sum_p [f(x_p) - y_p]^2$$

otrzymujemy następujące wartości wag: $a_1 = 0,699$, $a_2 = 1,296$, $a_3 = 1,824$.



Rys.11. Odzworowanie wektorów wejściowych w sygnał wyjściowy

Fig. 11. Mapping of the input vectors to output signal

Na rys. 11 przedstawiono odwzorowanie wektorów wejściowych w sygnał wyjściowy. Przedstawione algorytmy analizy skupień i doboru parametrów węzłów w warstwie ukrytej poprawnie skonfigurowały sieć RBF. Otrzymane rezultaty potwierdziły przydatność zaproponowanej sieci neuronowej do zadań klasyfikacyjnych.

LITERATURA

1. Alpaydin E.: *Optical Character Recognition Using Artificial Neural Networks*, I IEEE Conf.on ANN, 191-195, 1989
2. Arbib M.A.: *Brains, Machines, and Mathematics*. Springer-Verlag, New York Inc,1987
3. Babic R.: *Parallel Processing System with Fixed Connection as a New Approach to Handwritten Digit Recognition*. ICANN-1992, Brighton, 1163-1166,1992
4. Beale R., Finlay J.E.: *Neural Networks and Pattern Recognition in Human-Computer Interaction*. Ellis Horwood, New York, 1992
5. Blackwell K.T., Vogl T.P., Hyman S.D., Barbour G.S.,Alkon D.L.: *A New Approach to Hand-Written Character Recognition*. Pattern Recognition Vol.25,No.6, 655-666,1992
6. Botros S.M.,Atkeson C.G.: *Generalization Properties of Radial Basis Functions*. Advances in Neural Information Processing Systems 3, Kaufmann, San Mateo, 707-713, 1992
7. Chen C-H: *Statistical Pattern Recognition*. Hayden Book Comp.Inc. Rochelle Park,New Jersey, 1973
8. Chen C-H: *A decision-enhanced pattern classifier based on neural network approach*. Pattern Recognition Letters, Vol.13, 315-323,1992
9. Duda R.O.,Hart P.E.:*Pattern classification and scene analysis*. Willey,New York, 1973

10. Fukushima K., Wake Nobuaki W.: *Handwritten Alphanumeric Character Recognition by the Neocognitron*. IEEE Tran. on Neural Networks, Vol.2, No.3, 355-365, 1991
11. Fukushima K.: *Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition*. Neural Networks, Vol.1, 119-130, 1989
12. Giles C.L., Maxwell T.: *Learning Invariance, and Generalization in High-Order Neural Networks*. Applied Optics, Vol.26, 4972-4978, 1987
13. Goggin S.D.D., Johnson K.M., Gustafson K.E.: *A Second-Order Translation and Scale Invariant Neural Networks*. Advances in Neural Information Processing Systems 3, Kaufmann, San Mateo 313-319
14. Gonzales R.C., Woods R.E.: *Digital Image Processing*. Addison-Wesley Publishing Company, Inc, 1992
15. Grossberg S.: *Nonlinear Neural Networks: Principles, Mechanisms, and Architecture*. Neural Networks, Vol.1, 17-61, 1988
16. Haralic R.M., Shapiro L.G.: *Computer and Robot Vision*. Vol.1., Addison-Wesley Publishing Company, Inc, 1992
17. Hinton G.E., Williams C. K., Revow M.D.: *Combining Two Methods of Recognizing Hand-Printed Digits*. ICANN-1992, Brighton, 53-60, 1992
18. Hubbard W. : *Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning*. IEEE Communications Mag., November 1989, 41-46, 1989
19. Iwata A., Ino Y., Hotta K., Suzumara N. : *Hand-written Japanese Kanji character recognition by a structured self-growing neural network „CombbNET-II”*. ICANN-1992, Brighton, 1189-1192
20. Keeler J.D., Rumelhart D.E., Leow W-K.: *Integrated Segmentation and Recognition of Hand-Printed Numerals*. Advances in Neural Information Processing Systems 3, Kaufmann, San Mateo, 557-563, 1992
21. Kohonen T., Kangas J., Laaksonen J., Tarkkola K.: *LVQ PAK The Learning Vector Quantization Program Package*. Finland, 1992
22. Kohen T.: *Self-Organization and Associative Memory*. Springer-Verlag, Berlin-Heidelberg, 1989
23. Kohen T.: *Improved version of learning vector quantization*. IJCNN-1990, San Diego, 1545-550, 1990

24. Krogh A., Hertz J. Palmer.: *Introduction to the theory of neural computation*. Addison-Weseley Publishing Company, 1991
25. Le Cun Y., Jackel L.D., Boser Denker J.S., Graf H.P., Guyon I., Henderson D., Howard R.E., Lee Y. *Handwritten Digit Recognition Using K Nearest-Neighbor, Radial-Basis Function, and Backpropagation Neural Networks*. *Neural Vcomputation* 3, 440-449, 1991
26. Li R.Y., Xu M.: *Character recognition using a fast neural net classifier*. *Pattern Recognition Letters*, Vol.13, 369-374, 1992
27. Lim J.S.: *Two-dimensional signal and image processing*. Prentice-Hall International Editions, 19989

Recenzent: Dr hab. Tomasz Kacprzak

Wpłynęło do Redakcji 28.09.1994 r.

Abstract

Interest in neural networks is rapidly growing and several neural network models has been proposed for various difficult problems, especially classification problems. This paper presents three different kinds of artificial neural networks i.e. a third order neural network, Learning Vecctor Quantization network and Radial Basis Function network. An example of implementation in numeral recognition has been included. The third order neural network has been used as a feature extractor, the LVQ network has been utilised as a classifier. The RBF network has been presented as well. Author proposed network configuration algorithms which find a number of hidden nodes and thier parameters for the RBF network. The performance of the RBF network has been presented by a classification task. Experiments confirmed usefulness of the artificial neural networks. Obtained results were as good as could be expected.