

Urszula SZYFMAN, Magdalena TOPCZEWSKA

Politechnika Białostocka, Wydział Informatyki

ul. Wiejska 45A, 15-351 Białystok

E-mail: szyfman.urszula@gmail.com, m.topczewska@pb.edu.pl

Wyszukiwanie obiektów podobnych w podzbiorze wybranych znaków alfabetu migowego przy wykorzystaniu sieci konkurencyjnych

1 Wprowadzenie

Mózg ludzki, będący przedmiotem badań wielu dziedzin nauki, stał się inspiracją do stworzenia i rozwoju sztucznych sieci neuronowych. Jednym z wielu rodzajów sieci tego typu są sieci samoorganizujące się [4].

Samoorganizacja to proces polegający na automatycznym odwzorowaniu, w trakcie procesu uczenia, zbioru wektorów wejściowych w zbiór sygnałów wyjściowych, bez podania z góry wzorców wyjścia. Realizacja tego procesu może odbywać się poprzez zastosowanie jednej z kilku grup sieci. Najbardziej popularną stanowi *sieć Kohonena* nazwana *samoorganizującym odwzorowaniem* (ang. *SOM, Self-Organizing Map*) lub *samoorganizującym odwzorowaniem cech* (ang. *SOFM, Self-Organizing Feature Map*) [1]. Oprócz uczenia nienadzorowanego wykorzystuje się w tej sieci uczenie konkurencyjne – neurony rywalizują między sobą o przywilej uczenia się. Jako wynik procesu uczenia otrzymuje się odwzorowanie przestrzeni wielowymiarowej (liczba wymiarów większa niż 2) na przestrzeń jedno- lub dwuwymiarową przy zachowaniu odległości analizowanych obiektów.

Celem pracy jest wyszukanie obiektów podobnych przy wykorzystaniu sieci Kohonena w zbiorze obrazów przedstawiających znaki alfabetu migowego [5], a także obserwacja tego, w jaki sposób znaki alfabetu zostały rozmieszczone na siatce, czyli odwzorowania ich na siatce obiektów. Do badań wybranych zostało pięć znaków alfabetu migowego, zaś wektor cech został stworzony z wartości pikseli każdego obrazu.

2 Sieć samoorganizująca się

Niech będzie dany zbiór N obiektów – wektorów wejściowych, gdzie każdy obiekt x_i opisany jest za pomocą d -wymiarowego wektora atrybutów $(x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T, i=1, \dots, N; x_{ij} \in R^1 \text{ lub } x_{ij} \in \{0,1\})$.

Wektory wejściowe stanowią warstwę wejściową tworzonej sieci Kohonena, która jest siecią jednokierunkową, docelowo złożoną z dwu warstw. Warstwa wyjściowa zbudowana jest z neuronów, które tworzą dwuwymiarową siatkę lub jednowymiarowy łańcuch.

Każdy neuron połączony jest ze wszystkimi składowymi d -wymiarowego wektora wejściowego x_i , dlatego sieć SOM można zapisać jako zbiór n neuronów [2]:

$$w = \{w_1, w_2, \dots, w_n\}, \quad (1)$$

z których każdy posiada wektor wag:

$$\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{id}), i=1, 2, \dots, n. \quad (2)$$

Sposób połączenia między neuronami w warstwie wyjściowej pozwala na określenie jednej z kilku topologii sieci, wśród których wyróżnia się: siatkę prostokątną, siatkę heksagonalną, łańcuch otwarty lub zamknięty [4]. Określenie sposobu połączeń między neuronami jest następnie wykorzystywane w trakcie szukania neuronu zwycięzcy \mathbf{w}_c przy wykorzystaniu jednej z miar odległości. W artykule do obliczania odległości między wektorem wejściowym a neuronami wykorzystano metrykę euklidesową. Dodatkowo w pracy przeprowadzono proces normalizacji wektorów wejściowych, by ostatecznie otrzymać spójny podział przestrzeni danych.

W algorytmie uczenia sieci Kohonena, prócz neuronu zwycięzcy, procesowi uczenia podlegają również neurony znajdujące się w jego sąsiedztwie. Każdy neuron posiada zbiór sąsiadów oraz jest sąsiadem dla innych neuronów. Zbiór sąsiadów zależy od wybranej topologii. Prócz topologii na zbiór sąsiadów ma wpływ także rozmiar sąsiedztwa, który określany jest za pomocą parametru λ – promienia sąsiedztwa. Wszystkie neurony znajdujące się w zakresie promienia należą do sąsiedztwa rozpatrywanego neuronu. Wartość promienia sąsiedztwa jest określana przed rozpoczęciem procesu uczenia i maleje wraz z jego postępem, a co za tym idzie, zmniejszeniu ulega również zbiór sąsiadów [2,4].

Każdy neuron należący do sąsiedztwa posiada intensywność przynależności do niego, wyznaczaną na podstawie funkcji sąsiedztwa $G(\mathbf{w}_i, \mathbf{w}_c)$ i opartą na odległości neuronu zwycięzkiego od neuronu \mathbf{w}_i . Jedną z najpopularniejszych funkcji sąsiedztwa jest sąsiedztwo typu gaussowskiego:

$$G(\mathbf{w}_i, \mathbf{w}_c) = \exp\left(-\frac{d^2(\mathbf{w}_i, \mathbf{w}_c)}{2\lambda^2}\right), \quad (3)$$

gdzie: $d(\mathbf{w}_i, \mathbf{w}_c)$ – odległość między neuronem \mathbf{w}_i a neuronem zwycięzcą \mathbf{w}_c , λ – wartość promienia sąsiedztwa.

Drugim parametrem, którego wartość maleje wraz z upływem czasu uczenia, jest współczynnik uczenia η . Oba parametry mogą być zmniejszane na wiele sposobów: liniowo, hiperbolicznie lub wykładniczo. Dodatkowo współczynnik uczenia może być zmniejszony indywidualnie dla każdego neuronu. Zmniejszanie liniowe obu współczynników dokonywane jest w następujący sposób:

$$\eta(t) = \frac{\eta_0(T-t)}{T}, \quad \lambda(t) = \frac{\lambda_0(T-t)}{T}, \quad t=1, 2, \dots, T, \quad (4)$$

gdzie: η_0 – początkowa wartość współczynnika uczenia; λ_0 – początkowa wartość promienia sąsiedztwa; t – numer bieżącej epoki; T – maksymalna liczba epok.

Do najpopularniejszych sposobów uczenia rozpatrywanych sieci samoorganizujących zaliczane są dwa algorytmy: iteracyjny i wsadowy. W trakcie uczenia wartości wektora wag są modyfikowane w taki sposób, aby zminimalizować błąd kwantyzacji liczony jako średnia odległość wektorów wejściowych od ich neuronów wygrywających [2]:

$$E_q = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{w}_c\|^2, \quad (5)$$

gdzie: E_q – błąd kwantyzacji; \mathbf{x}_i – wektor wejściowy; \mathbf{w}_c – wektor wag neuronu zwycięzcy; N – liczba wektorów wejściowych.

Ze względu na fakt, że w dalszej części pracy prezentowane są wyniki tylko dla algorytmu iteracyjnego, tylko ten algorytm jest przedstawiony w artykule. Ponadto w celu zwiększenia przejrzystości opisu algorytmu pominięto indeksowanie aktualnego wektora wejściowego, oznaczając go jako \mathbf{x} .

Algorytm iteracyjny uczenia sieci samoorganizującej

1. Losowe wybranie wektora wejściowego $\mathbf{x}=(x_1, x_2, \dots, x_d)$.
2. Obliczenie odległości względem wybranej metryki między prezentowanym aktualnie wektorem wejściowym \mathbf{x} i każdym neuronem \mathbf{w}_i .
3. Wybranie neuronu zwycięzcy \mathbf{w}_c , czyli neuronu najbliższego do wektora wejściowego \mathbf{x} :

$$d(\mathbf{x}, \mathbf{w}_c) = \min_{1 \leq i \leq n} d(\mathbf{x}, \mathbf{w}_i) \quad (6)$$

gdzie: $d(\mathbf{x}, \mathbf{w}_i)$ – odległość między prezentowanym wektorem \mathbf{x} a neuronem \mathbf{w}_i , $i=1, \dots, n$;
 n – liczba neuronów.

4. Wybranie zbioru neuronów należących do sąsiedztwa neuronu $N_c(t)$.
5. Aktualizacja wag neuronów:

- Faza pierwsza według zasady WTM (ang. *Winner Takes Most*) [3]: promień sąsiedztwa λ jest stosunkowo duży, co powoduje, że oprócz neuronu zwycięzcy \mathbf{w}_c również jego sąsiedzi aktualizują swoje wagi:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) * G(\mathbf{w}_i, \mathbf{w}_c) * [x_j - w_{ij}(t)], \quad (7)$$

gdzie: $w_{ij}(t+1)$ – wartość wagi j neuronu i po aktualizacji; $w_{ij}(t)$ – wartość wagi j neuronu i przed aktualizacją; $\eta(t)$ – wartość współczynnika uczenia w bieżącej epoce; $G(\mathbf{w}_i, \mathbf{w}_c)$ – wartość funkcji sąsiedztwa dla neuronu \mathbf{w}_i ; \mathbf{x} – aktualnie prezentowany wektor wejściowy, $i=1, \dots, n$; n – liczba neuronów, $j=1, \dots, d$, d – rozmiar wektora wejściowego, t – numer bieżącej epoki.

- Faza druga według zasady WTA (ang. *Winner Takes All*) [3]: promień sąsiedztwa λ maleje do 0, co oznacza, że neuron zwycięzca nie posiada sąsiedztwa – tylko wektor wag neuronu zwycięzcy \mathbf{w}_c ulega aktualizacji:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) * [x_j - w_{ij}(t)], \quad (8)$$

gdzie: $w_{ij}(t+1)$ – wartość wagi j neuronu i po aktualizacji; $w_{ij}(t)$ – wartość wagi j neuronu i przed aktualizacją; $\eta(t)$ – wartość współczynnika uczenia w bieżącej iteracji; \mathbf{x} – aktualnie prezentowany wektor wejściowy; $i=1, \dots, n$; n – liczba neuronów, $j=1, \dots, d$; d – rozmiar wektora wejściowego; t – numer bieżącej epoki.

Kroki od 1 do 5 powtarzane są dla wszystkich wektorów \mathbf{x} ze zbioru wejściowego.

Po prezentacji wszystkich danych wykonywane są kroki 6 i 7 algorytmu.

6. Zwiększenie licznika epok: $t=t+1$.
7. Aktualizacja wartości parametrów: współczynnika uczenia oraz promienia sąsiedztwa według wzorów ustalonych przed rozpoczęciem procesu uczenia.

Wszystkie kroki od 1 do 7 wykonywane są, dopóki wartość licznika epoki t nie przekroczy wartości maksymalnej liczby epok T , czyli dopóki $t \leq T$.

3 Eksperymenty

W celu przeprowadzenia eksperymentu został wybrany zbiór danych zawierający obrazy wybranych znaków alfabetu migowego [5]. Wśród wybranych ze względu na różnorodność kształtów pięciu znaków znalazły się litery: A, B, D, G, V. Każdy ze znaków przedstawiony był na jasnym i ciemnym tle. W oryginalnej wersji zbiory obrazów były dostępne w skali szarości oraz miały rozmiar 128x128 pikseli. Jeśliby wektory cech były definiowane jako odcienie poszczególnych pikseli, gdzie każdy

piksel to oddzielny atrybut, spowodowałyby to powstanie wektorów dla każdego obrazu o długości 16384. Ze względu na nieakceptowalny rozmiar wektora, obrazy zostały poddane transformacji (zmniejszeniu) do rozmiaru 30x30 pikseli. Wektor cech ostatecznie osiągnął długość 901 pikseli (900 atrybutów numerycznych z wartością pikseli oraz znak, jaki przedstawia obraz).

Na potrzeby pracy opracowane zostały trzy zbiory znaków:

- na jasnym tle (Jdata),
- na ciemnym tle (Cdata),
- na jasnym i ciemnym tle (JCdata).

Następnie każdy ze zbiorów został podzielony na część uczącą i testującą (tab. 1).

Tab. 1. Podział zbiorów na część uczącą i testującą; d – wymiar wektora cech (w tym klasa); n_u – liczba obiektów w zbiorze uczącym, n_t – liczba obiektów w zbiorze testującym

Tab. 1. The division of data sets into learning and testing subsets; d – the dimension of the feature vector (with the class label); n_u – number of objects in the learning set, n_t – number of objects in the testing set

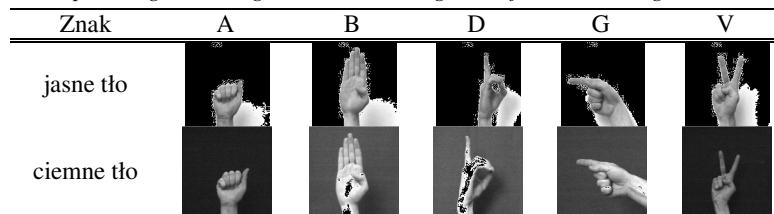
	zbiór uczący		zbiór testujący
	d	n_u	n_t
Jdata	901	61	57
Cdata	901	60	60
JCdata	901	118	114

W przypadku zbioru znaków na jasnym tle w zbiorze uczącym znalazło się 61, natomiast w zbiorze testującym 57 obiektów. W zbiorze znaków na ciemnym tle podział na zbiór uczący i testujący był dokładnie w stosunku 1:1 – w części testującej i uczącej znalazło się po 60 obrazów. Ze zbioru znaków na jasnym i ciemnym tle 118 obiektów zostało przydzielonych do zbioru uczącego, zaś 114 do zbioru testującego.

Obrazy wybranych do analizy znaków zostały przedstawione w tabeli 2.

Tab. 2. Przykładowe obrazy na jasnym i ciemnym tle dla wybranych znaków

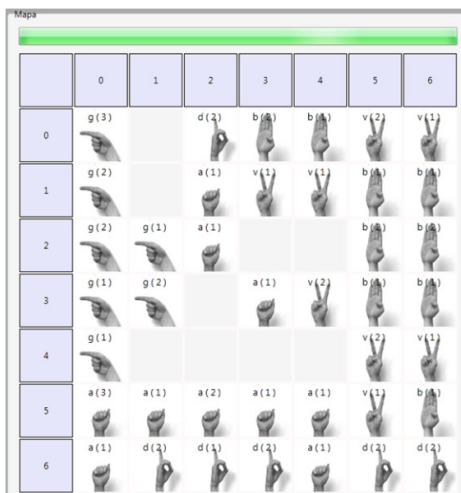
Tab. 2. Sample images on a light and dark background for selected signs



Po przeprowadzeniu szeregu badań i przetestowaniu różnych wartości parametrów najlepsze wyniki uzyskano dla następującego zestawu: liczba iteracji 2000; współczynnik uczenia 0.4; prostokątny rodzaj siatki; promień sąsiedztwa – 4; rozmiar siatki 7x7; gaussowska funkcja sąsiedztwa oraz iteracyjny algorytm uczenia bez uwzględnienia zmęczenia neuronów.

*Wyszukiwanie obiektów podobnych w podzbiorze wybranych znaków
alfabetu migowego przy wykorzystaniu sieci konkurencyjnych*

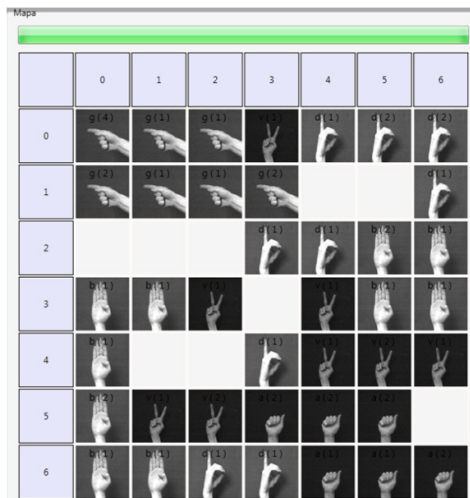
Pierwsza część eksperymentu dotyczyła zbioru znaków na jasnym tle. Celem było sprawdzenie, czy sieć prawidłowo wyszuka obiekty podobne i pogrupuje je w pewne klasy podobieństwa. Uzyskane wyniki prezentuje rysunek 1.



*Rys. 1. Wyniki nauczonej sieci w przypadku znaków przedstawionych na jasnym tle
Fig. 1. The results of the network in the light background signs case*

Analizując wyniki przedstawione na rysunku 1, można zaobserwować, że znak G został umieszczony w lewym górnym rogu siatki, natomiast znak D w prawym dolnym rogu. Znaki te mają zdecydowanie różne kształty, co wpłynęło na ułożenie neuronów odpowiadających za te znaki z daleka od siebie. Kształt litery V jest zbliżony do kształtu litery B, dlatego neurony odpowiadające za oba znaki umieszczone są obok siebie. Następne neurony, znajdujące się obok odpowiadających literze V, odpowiedzialne są za rozpoznawanie znaków z „jednym palcem”, czyli litery D. Na siatce nie można zauważyć wyraźnych granic pomiędzy rozpoznawanymi grupami. Wystąpił też problem z rozpoznawaniem poszczególnych znaków, takich jak B i V – neurony odpowiadające za rozpoznanie są przemieszane na siatce. Można jednak stwierdzić, że neurony odpowiadające za znaki podobne (B i V) można połączyć w jedną grupę. Neurony rozpoznające znak A są rozproszone po całej siatce. Wynik ten może być spowodowany faktem, że zaciśnięta pięść stanowi podstawę każdego ze znaków. Podsumowując, na siatce da się wyróżnić grupy znaków podobnych, wśród których wyszczególnić można znaki B i V oraz A i G.

Druga część eksperymentu polegała także na zbadaniu wzajemnego rozmieszczenia klastrów podobnych znaków na siatce neuronów. Tym razem do analizy wybrane zostały znaki przedstawione na ciemnym tle. Parametry sieci były takie same, jak w przypadku znaków na jasnym tle. Wyniki badania zostały przedstawione na rysunku 2.



Rys. 2. Wyniki nauczanej sieci w przypadku znaków przedstawionych na ciemnym tle
 Fig. 2. The results of the network in the dark background signs case

Analizując uzyskane wyniki, można stwierdzić, że organizacja neuronów odpowiedzialnych za poszczególne znaki w zdecydowanej większości jest chaotyczna. Dzieje się tak ze względu na niejednorodność kolor tła, który w tym przypadku wpłynął na rozmieszczenie neuronów na siatce. W przypadku znaków przedstawionych na ciemnym tle sieć nie miała problemów z rozpoznaniem znaku A – wszystkie neurony odpowiedzialne za rozpoznanie tego znaku zostały umieszczone obok siebie. Trudności sprawia wyszukanie grup obiektów podobnych, gdyż nie ma jednoznacznych granic pomiędzy grupami. Można jednak wyszczególnić podobieństwo między takimi znakami, jak: D, B i V.

W trzeciej części eksperymentu połączono obydwa zbiory znaków, by sprawdzić jak zostanie przedstawiona organizacja poszczególnych neuronów w sieci. Parametry, dla których uzyskano najlepsze wyniki, były następujące: liczba iteracji 2000; współczynnik uczenia 0.4; prostokątny rodzaj siatki; promień sąsiedztwa – 8; rozmiar siatki 11x11; gaussowska funkcja sąsiedztwa oraz iteracyjny algorytm uczenia bez uwzględnienia zmęczenia neuronów. Uzyskane wyniki zaprezentowane zostały na rysunku 3.

Analizując uzyskane wyniki, można zauważyć, że sieć rozdzieliła znaki przedstawione na jasnym od znaków na ciemnym tle. Liczba neuronów odpowiedzialna za rozpoznanie znaków na jasnym tle wyniosła 32, podczas gdy liczba ta dla znaków na ciemnym tle była równa 50. W obydwu grupach sieć nie miała problemów z wyszukaniem obrazów podobnych do siebie i odwzorowaniem ich na sąsiednie neurony, jednak w przypadku obrazów na ciemnym tle lepiej zostały wyszukane grupy podobieństwa – granice między tymi grupami są znacznie wyraźniejsze.

*Wyszukiwanie obiektów podobnych w podzbiorze wybranych znaków
alfabetu migowego przy wykorzystaniu sieci konkurencyjnych*

W przypadku grupy znaków przedstawionych na jasnym tle znalezione zostały dwie grupy podobieństwa: znaki A i G oraz B i V, zaś dla obrazów na ciemnym tle: A i D oraz B i V.



Rys. 3. Wyniki nauczonej sieci w przypadku znaków przedstawionych na jasnym i ciemnym tle

Fig. 3. The results of the network in the light and dark background signs case

Obserwując organizację siatki, można stwierdzić pewną zależność odwzorowania obiektów wejściowych na sąsiednie neurony – neurony odpowiedzialne za rozpoznanie znaku G na jasnym tle zostały umieszczone obok neuronów rozpoznających ten sam znak na ciemnym tle.

Ostatnim etapem eksperymentu była próba klasyfikacji znaków znajdujących się w zbiorach testujących. Wyniki zostały umieszczone w tabeli 3. Analizowana była jakość klasyfikacji, a jako informację dodatkową zamieszczono również liczbę martwych neuronów w sieciach.

Tab. 3. Wyniki klasyfikacji; Q – miara jakości klasyfikacji na zbiorze testującym; n_{dn} – liczba martwych neuronów

Tab. 3. Classification results; Q – the accuracy on the testing set; n_{dn} – the number of dead neurons

	Q [%]	n_{dn}
Jdata	64.91	9
Cdata	46.66	9
JCdata	55.26	39

W tworzeniu siatki neuronów we wszystkich przypadkach wybrany został algorytm bez uwzględnienia zmęczenia neuronów – ze uwagi na to w każdej sieci znalazły się

martwe neurony. W sieciach, które budowane były oddzielnie dla znaków na jasnym i ciemnym tle, liczba martwych neuronów wyniosła po 9, zaś w przypadku sieci dla wszystkich znaków liczba martwych neuronów była równa 39.

Najlepszy wynik jakości klasyfikacji uzyskano w przypadku zbioru znaków przedstawionych na jasnym tle – 64.91%. Jakość klasyfikacji dla zbioru znaków na ciemnym tle wyniosła zaledwie 46.66%, natomiast dla wszystkich znaków 55.26%. Tak niskie wartości jakości klasyfikacji spowodowane są dwiema przyczynami. Pierwsza z nich, to podział zbioru na część uczącą i testującą w stosunku prawie 1:1, natomiast drugą przyczyną jest różny odcień szarości w przypadku zbioru znaków na ciemnym tle.

4 Podsumowanie

Do wyszukiwania obiektów podobnych w badaniach zastosowano sieć Kohonena z uwzględnieniem różnych wartości parametrów oraz możliwością wyboru jednego z trzech algorytmów uczenia: iteracyjnego, wsadowego i uwzględniającego zmęczenie neuronów.

Eksperymenty przeprowadzono na zbiorze danych (obrazach) w postaci znaków alfabetu migowego. Dobrą organizację siatki w kontekście wyróżnienia grup podobieństwa uzyskano dla znaków na jasnym tle, a także dla zbioru łączonego. Natomiast dla sieci, która była uczona na obrazach z ciemnym tłem, siatka była zorganizowana chaotycznie.

W kolejnych badaniach planowane są eksperymenty na zbiorach danych, w których zmienne zostaną utworzone na podstawie cech graficznych poszczególnych obrazów. Powinno to zapewnić znaczną redukcję wymiaru przestrzeni cech, a co za tym idzie, przyspieszenie działania algorytmów. Jest to szczególnie ważne w przypadku znaków przedstawionych na ciemnym tle, gdyż kolor tła różni się pomiędzy obrazami.

Literatura

1. Kohonen T.: *Self-Organizing Maps. Springer Series in Information Sciences*, Vol. 30, Springer, Berlin, Heidelberg, New York 1995, 1997, 2001, 3rd edition
2. Ossowski S.: *Sieci neuronowe do przetwarzania informacji*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2006
3. Ossowski S.: *Sieci neuronowe w ujęciu algorytmicznym*. Wydawnictwo Naukowo-Techniczne, Warszawa 1996
4. Skubalska-Rafajłowicz E.: Samoorganizujące sieci neuronowe. *Biocybernetyka i Inżynieria Biomedyczna*, t. 6, M. Nałęcz, red., 2000
5. Triesch J., von der Malsburg C.: Classification of hand postures against complex backgrounds using elastic. *Image and Vision Computing*, nr 20, 2002

Streszczenie

W pracy opisano zastosowanie sieci konkurencyjnej (sieci Kohonena i ich modyfikacji) do wyszukiwania obiektów podobnych. Dokonano klasyfikacji obiektów na przykładzie rzeczywistego zbioru danych obrazowych wybranych znaków alfabetu migowego.

Słowa kluczowe: sieć samoucząca, sieć konkurencyjna, obiekty podobne, klasyfikacja obiektów

Searching for similar objects among selected sign language alphabet images using competitive networks

Summary

The paper describes the usage of the competing networks (Kohonen network and its modifications) to search for similar objects. A dataset containing selected images of the sign language alphabet has been tested for classification.

Keywords: self-organizing network, competitive network, similar objects, object classification

Artykuł został częściowo sfinansowany w ramach pracy S/WI/2/2013 realizowanej na Wydziale Informatyki Politechniki Białostockiej.

