

Alina MOMOT

Politechnika Śląska, Instytut Informatyki

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH W ROZPOZNAWANIU ZNAKÓW

Streszczenie. W artykule przedstawiono krótkie wprowadzenie w dziedzinę rozpoznawania znaków i omówiono najbardziej podstawowe pojęcia związane ze sztucznymi sieciami neuronowymi. Artykuł zawiera przykłady zastosowań sieci neuronowych w procesie rozpoznawania znaków, ze szczególnym uwzględnieniem etapów ekstrakcji cech i klasyfikacji.

Słowa kluczowe: rozpoznawanie znaków, sztuczne sieci neuronowe

APPLICATION OF ARTIFICIAL NEURAL NETWORK TO CHARACTER RECOGNITION

Summary. The article presents the short introduction to the domain of character recognition and describes the basic conceptions connected with artificial neural networks. The paper contains examples of applications of neural networks to a process of character recognition with special regard to stages of feature extraction and classification.

Keywords: character recognition, artificial neural networks

1. Wstęp

Początki automatycznego rozpoznawania znaków sięgają 1951 r., kiedy to M. Sheppard przedstawił wyniki swojej pracy nad maszyną zdolną „czytać” znaki. Była to zaprojektowana i skonstruowana przez niego maszyna, zwana GISMO – A Robot Reader-Writer. Wyznaczyła ona drogę rozwoju dziedziny określanej obecnie skrótem OCR (ang. Optical Character Recognition). W 1954 r. J. Rainbow podążając za przykładem Shepparda stworzył prototyp maszyny zdolnej rozpoznawać wielkie litery napisane na maszynie do pisania w

zadziwiającym ówczesnie czasie jednego znaku na minutę. Wkrótce potem kilka wielkich spółek handlowych zaczęło finansować projekty badawcze związane z systemami rozpoznawania znaków. Od 1967 r. takie spółki jak: IBM, Recognition Equipment, Inc., Farrington, Control Data i Optical Scanning Corporation posiadały już swoje własne systemy OCR. W tamtych czasach systemy OCR były wykorzystywane głównie przez agencje rządowe lub wielkie korporacje i traktowane były przez małe firmy lub osoby prywatne jako technologie egzotyczne i futurystyczne. Związane było to przede wszystkim z ceną tych systemów, sięgającą nierzadko miliona dolarów. Powstające wtedy systemy OCR tworzone były na podstawie różnych standardów, mających wspomagać automatyczną analizę dokumentów. Standardy te obejmowały między innymi [32]:

1. Character Set for Optical Character Recognition (OCR-A). ANSI X3.17-81
2. Character Set for Optical Character Recognition (OCR-B). ANSI X3.49-75
3. Paper Used in Optical Character Recognition Systems. ANSI X3.62-87
4. Optical Character Recognition (OCR) Inks. ANSI X3.86-80
5. Optical Character Recognition (OCR) Character Position. ANSI X3.93-81.

Rysunek 1 (zaczepnięty z [32]) przedstawia jako przykład znormalizowanych zestawów znaków zestawy OCR-A oraz OCR-B.

A B C D E F G H I J K L	A B C D E F G H I J K L
M N O P Q R S T U V W X	M N O P Q R S T U V W X
Y Z 1 2 3 4 5 6 7 8 9 0	Y Z 1 2 3 4 5 6 / 8 9 0

Rys. 1. Znormalizowane zestawy znaków: OCR-A (po lewej) i OCR-B (po prawej)
 Fig. 1. Standardized fonts: OCR-A (left) and OCR-B (right)

Początkowo algorytmy rozpoznawania znaków bazowały na prostym porównywaniu pojedynczych liter z ich odpowiednikami na liście wzorców. Jeśli odczytany obraz znaku był zgodny z obiektem wzorcowym, przechowywanym w pamięci w postaci macierzy pikseli, to zostawał mu przyporządkowany odpowiedni kod ASCII i dany znak uznawano za rozpoznany. Wystarczyły zatem niewielkie odstępstwa kształtu czcionki od zdefiniowanego wzorca, aby skutecznie uniemożliwić rozpoznanie znaku. Dopiero od 1975 r. technika porównywania ze wzorcem została zastąpiona techniką „analizy cech charakterystycznych” (ang. Feature Recognition), zwaną również metodą Omnifont [6]. Pierwszym urządzeniem wykorzystującym tę metodę był wdrożony w 1975 r. czytnik KRM (ang. Kurzweil Reading Machine), nazwany tak na cześć swojego twórcy – Raymonda Kurzweila.

Od lat osiemdziesiątych ubiegłego wieku można zauważyć zastosowanie sztucznych sieci neuronowych w dziedzinie rozpoznawania znaków. Początkowo wykorzystywane były one

wyłącznie jako klasyfikatory; na podstawie dostarczonej na wejście sieci informacji o matematyczno-geometrycznych cechach znaku podejmowały one decyzję o tym, jaki to znak. Zaletą sieci neuronowej, w porównaniu z klasyczną techniką analizy cech charakterystycznych, była możliwość uczenia się na przykładach i uogólniania wiedzy na temat kształtów poszczególnych znaków. Obecnie architektura sieci neuronowej do rozpoznawania pisma wzorowana jest często na strukturze połączeń występujących w tej części mózgu człowieka, która zajmuje się przetwarzaniem informacji wizualnej (Visual Cortex). Sieć nie tylko klasyfikuje cechy, ale też sama je wykrywa – specjalnie połączone warstwy neuronów są w stanie, w trakcie uczenia, wyodrębnić z pikseli obrazu zadanego znaku cechy geometryczne odróżniające go od innych znaków. Dodatkowo niektóre modele sieci neuronowych potrafią skutecznie rozpoznać całe sekwencje mocno zdeformowanych i zakłóconych znaków bez konieczności podziału sekwencji na pojedyncze znaki [6]. Jest to tak zwane rozpoznawanie bezsegmentacyjne (ang. Segmentation Free Method), w którym stosuje się techniki przetwarzania podobne do stosowanych przy rozpoznawaniu mowy.

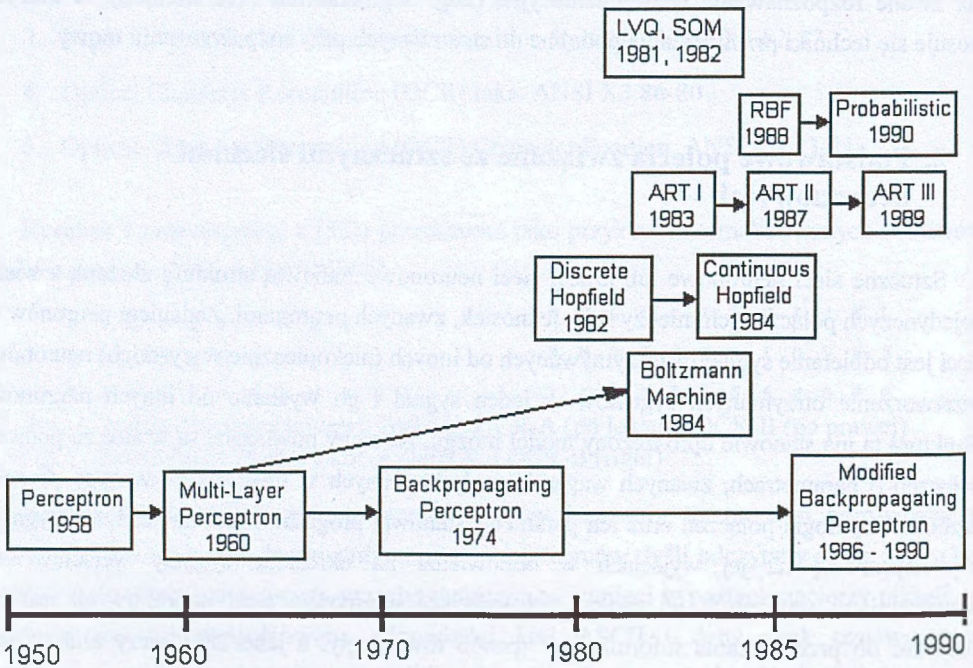
2. Podstawowe pojęcia związane ze sztucznymi sieciami neuronowymi

Sztuczne sieci neuronowe lub krócej sieci neuronowe stanowią strukturę złożoną z wielu pojedynczych połączonych między sobą jednostek, zwanych neuronami. Zadaniem neuronów w sieci jest odbieranie sygnałów otrzymywanych od innych (niekoniecznie wszystkich) neuronów, przetworzenie otrzymanych sygnałów w jeden sygnał i go wysłanie do innych neuronów. Struktura ta ma stanowić uproszczony model mózgu. Neurony powiązane są w sieć za pomocą połączeń o parametrach, zwanych wagami, modyfikowanych w trakcie tak zwanego procesu uczenia. Topologia połączeń oraz ich parametry stanowią program działania sieci, zaś sygnały pojawiające się na jej wyjściach w odpowiedzi na określone sygnały wejściowe są rozwiązaniami stawianych jej zadań. Najcenniejszą własnością sieci neuronowych jest jej zdolność do przetwarzania informacji w sposób równoległy, a jako zasadniczy atut można traktować proces uczenia zastępujący tradycyjne programowanie.

Przyjmuje się, że dziedzina sztucznych sieci neuronowych zaistniała wraz z wydaniem pracy autorstwa W. C. McCullocha i W. Pittsa [24], w której po raz pierwszy dokonano matematycznego opisu komórki nerwowej i powiązано ten opis z problemem przetwarzania danych. Natomiast pierwszym szeroko znanym przykładem sieci jest Perceptron [25]. Był to układ częściowo elektromechaniczny, w którym zmienne wagi synaptyczne realizowano za pomocą potencjometrów obracanych przez odpowiednio sterowane silniki elektryczne, a częściowo elektroniczny (sumowanie pobudzeń). Zbudowany został w 1957 r. przez

F. Rosenblatta i Ch. Wightmana w Cornell Aeronautical Laboratory. Jego przeznaczeniem było rozpoznawanie znaków alfanumerycznych z procesem uczenia jako metodą programowania systemu [26]. Działanie Perceptronu nie było jednak zadowalające z punktu widzenia zasadniczego celu badań, gdyż układ nie radził sobie z bardziej złożonymi znakami i wykazywał znaczną wrażliwość na zmianę skali rozpoznawanych obiektów, ich położenie w polu widzenia (przesunięcia, obroty) oraz zmiany kształtu. Jednak podstawową zaletą Perceptronu, obok faktu, że był on pierwszą realnie działającą imitacją sieci nerwowej, była jego zdolność do zachowywania poprawnego działania nawet po uszkodzeniu pewnej części jego elementów [34].

Od czasu powstania Perceptronu, który był inspiracją dla wielu innych twórców modeli sieci neuronowych, można zaobserwować różne kierunki rozwoju struktur sieci neuronowych (patrz rys.2).



Rys. 2. Rozwój najpopularniejszych modeli sztucznych sieci neuronowych
Fig. 2. The evolution of the most popular artificial neural networks

Oryginalny Perceptron Rosenblatta był modelem sieci jednowarstwowej. Na jego podstawie w 1960 r. wprowadzono model sieci wielowarstwowej określanej skrótem MLP (ang. Multi-Layer Perceptron). Jednak początkowo wykorzystanie tych modeli było utrudnione, gdyż brak było odpowiednich algorytmów uczenia sieci wielowarstwowych. W 1974 r. został przedstawiony tak zwany algorytm wstecznej propagacji (ang. back-

propagation) dla trzywarstwowej sieci perceptronów. Natomiast w 1986 r. opublikowany został ogólny algorytm wstecznej propagacji dla wielowarstwowej sieci autorstwa D. E. Rumelharta i innych [28]. Modele sieci MLP należą do grupy sieci jednokierunkowych (ang. feed-forward networks), gdyż przepływ sygnału w tego typu sieciach przebiega zawsze w określonym kierunku: od warstwy wejściowej do warstwy wyjściowej.

W 1982 r. J. J. Hopfield [13] przedstawił strukturę sieci neuronowych ze sprzężeniami zwrotnymi (nazywanymi również sieciami rekurencyjnymi), w których rozwiązanie stawianych zadań polegało na poszukiwaniu przez sieć stanu równowagi w iteracyjnym procesie dynamicznym. W tego typu sieciach sygnał propagowany jest w obiegu zamkniętym, tzn. sygnały z warstwy wejściowej sieci podawane są z powrotem do warstwy wejściowej. Jednak pojęcie warstwy w sieciach rekurencyjnych ze względu na skomplikowaną topologię jest w znacznym stopniu umowne. Oprócz sieci Hopfielda z czasem dyskretnym znane są również sieci z tak zwanym czasem ciągłym, w których dynamika neuronu opisywana jest równaniem różniczkowym zamiast równaniem różnicowym, jak to ma miejsce w przypadku tak zwanego czasu dyskretnego.

Z sieciami Hopfielda kojarzone są zwykle tak zwane maszyny Boltzmanna, których koncepcja oparta jest na założeniu, że sygnał wyjściowy każdego neuronu może zmieniać się w sposób losowy z określonym prawdopodobieństwem. Prawdopodobieństwo to zależy od „energii” i „temperatury” sieci podobnie jak w termodynamicznych systemach fizycznych, w których gęstość prawdopodobieństwa energii systemu związana jest z temperaturą znanym wzorem Boltzmanna [34].

Jedną z bardziej znanych zasad budowy sieci neuronowych jest ART (ang. Adaptive Resonance Theory). Teoria ta oparta jest na pracach S. Grossenberga. Sieć ART jest siecią klasyfikującą bodźce lub rozpoznającą obrazy, w której uczenie zachodzi na zasadzie uczenia bez nadzoru (ang. unsupervised learning), która to zasada została po raz pierwszy wprowadzona przez D. Hebba [12]. Początkowo sieć (ART 1) służyła do analizy obrazów binarnych, jednak następną wersją (ART 2) umożliwiała już analizę obrazów analogowych [34].

Sieci o radialnych funkcjach bazowych RBF (ang. Radial Basis Function) zostały przedstawione przez D. S. Broomheada i D. Lowego [3] w 1988 r. Podstawowa idea tego rozwiązania przypomina metodę rozpoznawania obrazów za pomocą funkcji potencjalnych zastosowaną w strukturze trzywarstwowej sieci jednokierunkowej. Każdy z neuronów warstwy środkowej reprezentuje pojedynczą radialną funkcję bazową, natomiast neurony warstwy wyjściowej przetwarzają dostarczane sygnały w postaci sumy ważonej sygnałów wejściowych. Jako szczególny przypadek sieci RBF można traktować sieć probabilistyczną przedstawioną w 1990 r. przez J. Spechta [31], którą można opisać jako znormalizowaną sieć RBF (ang. normalized RBF network), charakteryzującą się tym, że centra neuronów warstwy

ukrytej odpowiadają kolejno wszystkim wektorom danych uczących oraz tym, że przetwarzanie sygnałów wejściowych w neuronach warstwy ukrytej dokonuje się za pomocą funkcji gęstości pewnego rozkładu zmiennej losowej (zwykle jest to rozkład gaussowski).

Zupełnie wyjątkowym modelem sieci neuronowych są sieci samoorganizujących się odwzorowań SOM (ang. Self-Organizing Map) zaprezentowanych w 1982 r. przez T. Kohonena [16]. Sieć SOM tworzy macierz neuronów pobudzanych przez sygnały wejściowe. Jest to sieć neuronowa, w której występuje czynnik konkurencji, a uczenie odbywa się bez nadzoru. Sieć SOM wywodzi się z algorytmu kwantyzacji wektorowej obecnego w sieci neuronowej typu LVQ (ang. Learning Vector Quantization), której idea pochodzi również od T. Kohonena i została opublikowana w 1972 r. [15].

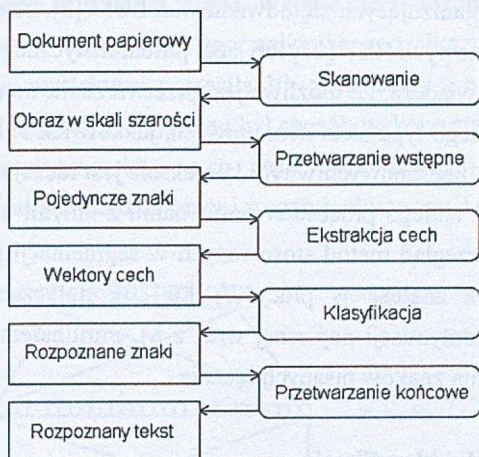
3. Przetwarzanie danych w systemach OCR

3.1. Etapy przetwarzania danych

W typowych systemach OCR przetwarzanie danych można podzielić na następujące etapy:

1. Skanowanie obrazu w poziomach szarości z rozdzielczością wahającą się zwykle od 300 do 1000 dpi (ang. dot per inch);
2. Przetwarzanie wstępne:
 - a. Poprawa jakości obrazu
 - b. Binaryzacja przy użyciu globalnych lub lokalnych metod adaptacyjnych
 - c. Segmentacja pozwalająca wyróżnić pojedyncze znaki
 - d. Konwersja do innej reprezentacji danych (np. szkieletyzacja lub wyodrębnienie konturu);
3. Ekstrakcja cech;
4. Klasyfikacja przy użyciu jednego lub kilku klasyfikatorów;
5. Przetwarzania końcowe obejmujące zwykle weryfikację kontekstową.

Etapy te w sposób symboliczny przedstawia rys. 3.



Rys. 3. Etapy przetwarzania danych w typowych systemach OCR
 Fig. 3. Stages of data processing in typical OCR systems

3.2. Przetwarzanie wstępne

Sztuczne sieci neuronowe można wielokrotnie spotkać we wstępnej fazie przetwarzania obrazów cyfrowych w roli prostych filtrów [7]. Przykładem tego jest między innymi praca D. Greenhila i E. R. Daviesa [9], dotycząca tłumienia szumu w obrazie za pomocą sieci jednokierunkowej, realizującej regresję (ang. regression feed-forward neural network), działającej w podobny sposób jak filtracja za pomocą masek. Na wejście sieci podawane są piksele obrazu pochodzące z kwadratowego okna o wymiarze 5, czyli sieć w warstwie wejściowej posiada 25 neuronów, natomiast warstwa wyjściowa składa się z pojedynczego neuronu. Innym przykładem jest praca D. De Riddera i innych [5], opisująca zastosowania modularnej jednokierunkowej sieci neuronowej (ang. modular feed-forward neural network) w celu uzyskania efektu filtracji filtrem Kuwahara (filtr wygładzający obraz przy jednoczesnym zachowaniu krawędzi). Jednak ich badania wykazują, że wykorzystanie prostej jednokierunkowej sieci neuronowej, w której algorytm uczenia opiera się na minimalizacji błędu średniokwadratowego, prowadzi jedynie do liniowej aproksymacji filtru Kuwahara. Jako przykład innych interesujących zastosowań sieci neuronowych do tłumienia szumu można wymienić uogólniony adaptacyjny filtr neuronowy GANF (ang. Generalized Adaptive Neural Filter) [39] lub sieć neuronowo-rozmytą, zaproponowaną przez F. Russo [29].

W literaturze spotkać można również wiele zastosowań różnego typu sieci neuronowych w dziedzinie segmentacji obrazów cyfrowych [7], np. jednokierunkowe sieci neuronowe typu

MLP [17], sieci samoorganizujących się odwzorowań SOM [8], sieci Hopfielda [27], sieci o radialnych funkcjach bazowych RBF [17] lub sieci probabilistyczne [17].

Opisane przykłady świadczą, że możliwe jest przetwarzanie wstępne obrazów cyfrowych przy wykorzystaniu różnego typu sieci neuronowych, jakkolwiek w komercyjnych systemach OCR zastosowanie sieci neuronowych w tym kontekście jest raczej nietypowe ze względu na większą czasochłonność takiego procesu w porównaniu z innymi metodami przy uzyskaniu podobnych wyników. Przegląd metod stosowanych w segmentacji pojedynczych znaków w systemach OCR można znaleźć w pracy Y. Lu [20], dotyczącej segmentacji znaków drukowanych oraz jej kontynuacji napisanej wraz z M. Shridhareem [21], opisującej metody segmentacji stosowane dla znaków pisanych ręcznie.

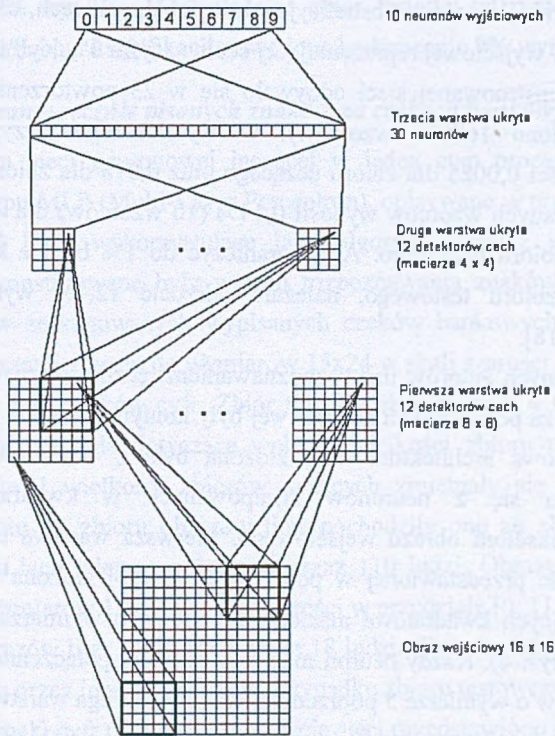
3.3. Ekstrakcja cech i klasyfikacja

W odróżnieniu od przetwarzania wstępnego sieci neuronowe są często wykorzystywane w systemach OCR w celu ekstrakcji cech i klasyfikacji dotyczących już obrazów wysegmentowanych pojedynczych znaków. Pomimo iż ekstrakcja cech na ogół jest etapem poprzedzającym klasyfikację, tzn. klasyfikatory jako dane wejściowe otrzymują wyliczone we wcześniejszym etapie wartości cech charakteryzujących wysegmentowany obraz, istnieją również systemy, w których etapy te zostały połączone. Dotyczy to zwłaszcza systemów neuronowych, w których klasyfikatory jako dane wejściowe otrzymują wartości odpowiadające pojedynczym pikselom obrazu (wtedy też zazwyczaj w etapach wstępnych nie przeprowadza się binaryzacji obrazu). Jako przykład tego typu podejścia może służyć sieć opisana przez Y. LeCuna i innych w pracy [18], przetwarzająca obrazy ręcznie pisanych kodów pocztowych (przykład ten będzie opisany szczegółowo w dalszej części niniejszej pracy). Często jednak ekstrakcja cech jest odrębnym etapem w procesie rozpoznawania znaków i w rozwiązaniach komercyjnych na tym etapie rzadko wykorzystuje się sieci neuronowe. Przegląd metod dla ekstrakcji cech stosowanych w systemach OCR można znaleźć w pracy Q. D. Triera i innych [35].

3.3.1. Rozpoznawanie ręcznie pisanych kodów pocztowych

Jako przykład sieci neuronowej łączącej w jeden etap proces ekstrakcji cech i klasyfikacji zostanie przedstawiona architektura jednokierunkowej sieci, wykorzystująca jako algorytm uczenia algorytm wstecznej propagacji, zaproponowana w pracy [18] przez Y. LeCuna i współpracowników z AT&T Bell Laboratories. Sieć ta była skonstruowana w celu rozpoznawania ręcznie pisanych cyfr – składników kodów pocztowych. Dane uczące wykorzystywane w tym eksperymencie pochodziły ze zbiorów urzędu pocztowego Stanów Zjednoczonych (ang. US postal service). Architektura sieci, przedstawiona na rysunku 4 (na

podstawie [18]), obejmowała 3 warstwy ukryte. Wejście sieci złożone było z 256 neuronów odpowiadających pikselom kwadratowego znormalizowanego obrazu cyfry o wymiarze 16. Liczba neuronów warstwy wyjściowej wynosiła 10, gdyż każdy z neuronów reprezentował pojedynczą cyfrę. Interesującą częścią systemu był sposób wykorzystania różnych warstw. W architekturze sieci wykorzystano ideę map cech pochodzącą od Kohonena wraz ze specyficznymi połączeniami i współdzielonymi wagami, aby uczynić proces uczenia bardziej efektywny.



Rys. 4. Schemat architektury sieci wykorzystywanej przez Y. LeCuna i in.

Fig. 4. Schema of network architecture used by Y. LeCun et al.

Pierwsza warstwa ukryta była odpowiedzialna za wyodrębnienie lokalnych cech z obrazu wejściowego. Dla każdego kwadratu o wymiarze 5 stosowany był „detektor cech”, co pozwalało na wykrywanie cech niezależnie od ich położenia na obrazie wejściowym. Następne dwie warstwy miały na celu detekcję cech wyższego rzędu i ostatecznie warstwa wyjściowa miała rozpoznawać cyfry. Funkcjonalność sieci została osiągnięta dzięki ograniczeniu połączeń między warstwami i ich wagami. Połączenia do jedynie kilku neuronów z tego samego obszaru zmuszały neurony do wykorzystywania tylko lokalnych informacji. Te same detektory cech były potrzebne dla różnych lokalizacji na płaszczyźnie wejściowej. Zostało to osiągnięte przez współdzielenie wag, tzn. różne neurony, które

wykrywały te same cechy, w różnych położeniach miały identyczne, wspólne wagi, co w powiązaniu ze wspomnianą wcześniej redukcją liczby połączeń czyniło proces uczenia za pomocą algorytmu wstecznej propagacji bardziej efektywny.

Warto tu podkreślić fakt, że propagację sygnałów w architekturze sieci wykorzystywanej w tym eksperymencie można traktować jako proces hierarchicznej ekstrakcji cech. Każdy z neuronów „widzi jedynie okno” w poprzedniej warstwie neuronów i łączy cechy niższego poziomu widoczne w tym oknie w jedną cechę wyższego poziomu. Tak więc wyższe warstwy sieci dokonują ekstrakcji bardziej abstrakcyjnych i globalnych cech, czego ostatnim etapem jest poziom warstwy wyjściowej reprezentującej cechy: „cyfra 0”, „cyfra 1”, ..., „cyfra 9”.

Uczenie tak skonstruowanej sieci odbywało się w 23 powtórzeniach zbioru wzorców (łącznie przedstawiono 167693 wzorców). W wyniku tego otrzymano błąd średniokwadratowy wielkości 0,0025 dla zbioru uczącego oraz 0,018 dla zbioru testowego. Procent błędnie sklasyfikowanych wzorców wynosił 0,14% (10 wzorców) dla zbioru uczącego i 5% (102 wzorce) dla zbioru testowego. Aby ograniczyć do 1% błędne klasyfikacje wzorców pochodzących ze zbioru testowego, należało odrzucić 12,1% wyników (trudnych do zaklasyfikowania) [18].

Prace wymienionych autorów nad rozpoznawaniem ręcznie pisanych cyfr – składników kodów pocztowych za pomocą sieci neuronowej były kontynuowane, a wyniki opublikowane w artykule [19]. Nowa architektura sieci złożona była z 4 warstw ukrytych. Warstwa wejściowa składała się z neuronów zgrupowanych w kwadrat o wymiarze 28, odpowiadających pikselom obrazu wejściowego. Pierwsza warstwa ukryta skonstruowana była analogicznie do przedstawionej w poprzedniej pracy i złożona była z czterech grup neuronów obejmujących kwadratowe macierze neuronów o wymiarze 24 (por. z pierwszą warstwą ukrytą na rys. 4). Każdy neuron macierzy posiadał połączenia jedynie z wybranym kwadratem neuronów o wymiarze 5 poprzedniej warstwy. Druga warstwa ukryta obejmowała 4 grupy neuronów w macierzach o wymiarze 12. Każdy neuron macierzy posiadał połączenia jedynie z czterema odpowiadającymi mu neuronami warstwy poprzedniej. Trzecia warstwa ukryta złożona była z 12 macierzy neuronów o wymiarze 8. Neurony w poszczególnych macierzach miały połączenia jedynie z wybranymi neuronami zgrupowanymi w kwadrat wymiaru 5 poprzedniej warstwy (por. rys. 4). Czwarta warstwa ukryta i jej połączenia są analogiczne do warstwy trzeciej, tj. 12 macierzy neuronów o wymiarze 4, każdy neuron macierzy ma połączenia jedynie z wybranymi neuronami zgrupowanymi w kwadrat wymiaru 5 poprzedniej warstwy. Ostatnia warstwa (wyjściowa) złożona jest z 10 neuronów, odpowiadających poszczególnym cyfrom. Każdy neuron tej warstwy ma połączenia ze wszystkimi neuronami warstwy poprzedniej.

Uczenie tak skonstruowanej sieci odbywało się w 30 powtórzeniach zbioru wzorców, obejmującego 7291 ręcznie pisanych oraz 2549 drukowanych cyfr. W wyniku tego otrzymano błąd średniokwadratowy wielkości 0,017 dla zbioru uczącego oraz 0,024 dla zbioru testowego, obejmującego 2007 ręcznie pisanych oraz 700 drukowanych cyfr. Procent błędnie sklasyfikowanych wzorców wynosił 1,1% dla zbioru uczącego i 3,4% dla zbioru testowego. Wszystkie błędne klasyfikacje dotyczyły cyfr pisanych ręcznie.

Aby ograniczyć do 1% błędne klasyfikacje wzorców pochodzących ze zbioru testowego, należało odrzucić 5,7% wyników. Ograniczając zbiór testowy tylko do znaków pisanych ręcznie, maksymalny, 1% błąd klasyfikacji powodował odrzucenie 9% wyników [19].

3.3.2. Rozpoznawanie ręcznie pisanych znaków na czekach bankowych

Innym przykładem sieci neuronowej łączącej w jeden etap proces ekstrakcji cech i klasyfikacji są sieci typu MLP (Multi-Layer Perceptron), opisywane w pracach G. L. Martina i J. A. Pittmana [22] [23], wykorzystujące jako algorytm uczenia algorytm wstecznej propagacji. Sieci te konstruowane były w celu rozpoznawania znaków liter i cyfr. Cyfry pochodziły z obrazów zeskanowanych wypisanych czeków bankowych. Po skanowaniu i segmentacji obrazu normalizowano do wymiarów 15x24 w skali szarości w przedziale [0, 1]. Dane obejmowały 40 000 znaków cyfr. Zbiór testowy składał się z 4 000 cyfr, natomiast autorzy przeprowadzali badania dotyczące wpływu wielkości zbioru uczącego na jakość rozpoznawania znaków i wielkości zbiorów uczących zmieniały się od 100 do 32500 znaków. W odniesieniu do zbioru obrazów liter pochodziły one ze zbioru 8600 znaków wielkich liter alfabetu łacińskiego, napisanych przez 110 ludzi. Obrazy liter również były znormalizowane do wymiarów 15x24 w skali szarości w przedziale [0, 1]. Testowy zbiór liter składał się z 2368 obrazów liter napisanych przez 18 ludzi. Zbiór uczący zawierał od 500 do 6300 znaków pisanych przez inne osoby niż w przypadku zbioru testowego.

Znormalizowane znaki cyfr podawane na wejście sieci przedstawiono również niezależnie wybranej osobie, która przy rozpoznawaniu cyfr popełniła błąd w 3,4% przypadków [22]. Uczenie sieci przebiegało w oparciu o klasyczny algorytm wstecznej propagacji błędów z wykorzystaniem elementu momentum o wartości 0,9 i współczynnikiem uczenia o wartości 0,05. Współczynnik ten był zmniejszany podczas procesu uczenia sieci. Tabela 1 (zaczepnięta z [22]) zawiera wyniki w postaci procentów błędnie rozpoznanych znaków w zależności od wielkości odsetka odrzucanych rezultatów klasyfikacji. Zbiory uczące w przypadku wyników przedstawionych w tabeli 1 wynoszą odpowiednio 32500 znaków cyfr i 6300 znaków liter. Warto zwrócić uwagę na fakt, że procenty błędnych klasyfikacji liter nie odbiegają znacząco od analogicznych wyników uzyskanych w przypadku cyfr, pomimo iż liczba kategorii dla liter wynosi 26, natomiast dla cyfr tylko 10.

Tabela 1
 Procent błędnie rozpoznawanych znaków w przypadku maksymalnych zbiorów uczących

Odsetek znaków odrzucanych	Cyfry	Litery
0%	4%	5%
5%	3%	3%
10%	1%	2%
35%	0,001%	0,003%

W wymienionych pracach zostały również przedstawione wyniki badania wpływu liczby neuronów w warstwie ukrytej (50, 170 i 383 w przypadku cyfr oraz 170 i 365 w przypadku liter), a także wyniki wpływu topologii sieci (w sensie liczby i rodzaju połączeń między dwoma warstwami ukrytymi) na skuteczność generalizacji sieci w zależności od wielkości zbioru uczącego. Otrzymane wyniki świadczą o braku istotnego wpływu wymienionych czynników na zdolność sieci do realizacji zadań klasyfikacji.

3.3.3. Podejście strukturalne do klasyfikacji ręcznie pisanych znaków

Interesujące podejście do problemu rozpoznawania znaków zostało zaprezentowane w pracach S. Singha i A. Amina [1][30]. Autorzy przedstawili procedurę rozpoznawania znaków, składającą się z trzech etapów. W pierwszym etapie obraz znaku zostaje zeskanowany tworząc binarny obraz cyfrowy, który następnie poddawany jest ścienianiu przy wykorzystaniu algorytmu równoległego. Uzyskany w pierwszym etapie szkielet znaku jest „przeglądany” za pomocą kwadratowych okien o wymiarze 3 w celu utworzenia drzewa binarnego, które przechowuje informacje o „ścieżkach” w formie kodów łańcuchów Freemana, uzupełnionych o współrzędne początku ścieżki. Informacje te służą do wyodrębnienia elementów podstawowych takich jak linie proste i łuki (krzywe otwarte lub zamknięte) widoczne na rys.5(na podstawie [1]).

linie proste	—	/		\
łuki otwarte	U	C	∩	∪
łuk zamknięty	○			

Rys. 5. Podstawowe cechy wykorzystywane przez S. Singha i A. Amina
 Fig. 5. Primitive features used by S. Singh and A. Amin

Trzeci etap procedury związany jest z klasyfikacją znaków za pomocą sieci neuronowej. Jest to jednokierunkowa sieć typu MLP (Multi-Layer Perceptron). Na wejście sieci podawany jest wektor 7-elementowy, zawierający na każdej ze składowych informację o wykryciu

jednej z 9 cech podstawowych przedstawionych na rysunku 5 lub braku jakiegokolwiek z nich. Cechy te wykrywane są w oknach ustalonego wymiaru pokrywających cały obraz. Warstwa wyjściowa sieci składa się z 52 neuronów, odpowiadających 26 małym i 26 wielkim literom alfabetu łacińskiego. Sieć posiada jedną warstwę ukrytą złożoną ze 120 neuronów. Uczenie sieci przebiega w oparciu o klasyczny algorytm wstecznej propagacji błędów z wykorzystaniem elementu momentum o wartości 0,9 i współczynnikiem uczenia o wartości 0,77. W przeprowadzonym eksperymencie użyto 1115 obrazów reprezentujących 52 litery. Każdej z liter odpowiadało mniej więcej 21 wzorców. Skuteczność klasyfikacji sieci badana była za pomocą metody *K-fold cross validation* [33] dla parametru K równego 10. Proces uczenia sieci kończył się wraz z otrzymaniem minimalizowanego w metodzie wstecznej propagacji błędu średniokwadratowego poniżej wartości 0,0001, co następowało po liczbie iteracji nie przekraczającej 500. Otrzymany w wyniku badania skuteczności klasyfikacji procent prawidłowo rozpoznanych znaków wynosi 85%.

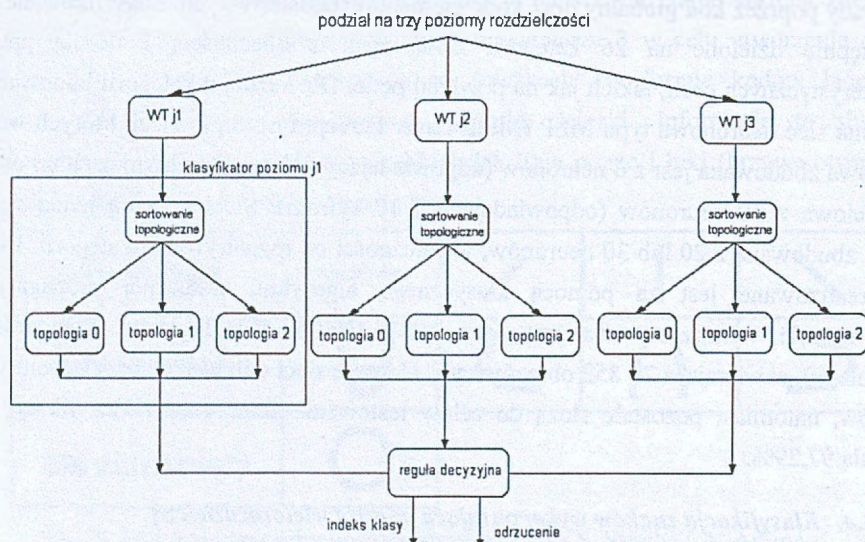
Inny przykład podejścia strukturalnego do klasyfikacji ręcznie pisanych cyfr zaproponowali J. Hu i H. Yan w pracy [14]. Przedstawiona procedura obejmuje wygładzenie obrazu cyfry, wyznaczenie jej szkieletu i jego dekompozycję na elementy podstawowe, z których każdy posiada unikalny kod. Topologiczna struktura szkieletu jest reprezentowana poprzez globalny kod, wykorzystywany do klasyfikacji znaków. Każdy znak jest zatem opisywany poprzez kod globalny oraz kody elementów podstawowych. Klasyfikowane znaki są wstępnie dzielone na 26 kategorii związanych z obecnością i ilością pewnych charakterystycznych cech, takich jak na przykład pętle. Dla każdej z kategorii budowana jest oddzielna sieć neuronowa typu MLP (Multi-Layer Perceptron). Są to sieci, których warstwa wejściowa zbudowana jest z 6 neuronów (odpowiadających 6 strukturalnym cechom obrazu), a wyjściowa z 10 neuronów (odpowiadających 10 cyfrom). Sieci te mają jedną warstwę ukrytą, zbudowaną z 20 lub 30 neuronów, w zależności od rozpatrywanej kategorii. Uczenie sieci realizowane jest za pomocą klasycznego algorytmu wstecznej propagacji. W eksperymencie wykorzystywana jest baza NIST (National Institute of Standards and Technology), zawierająca 20 852 obrazów cyfr. Uczenie sieci odbywa się przy użyciu 10 000 wzorców, natomiast pozostałe służą do celów testowania skuteczności klasyfikacji, która wyniosła 97,29%.

3.3.4. Klasyfikacja znaków wykorzystująca analizę wielorozdzielczą

Nowatorskie podejście do problemu rozpoznawania znaków zaprezentowali P. Wunsch i A. F. Laine w pracy [38]. Prezentują oni system rozpoznawania znaków bazujący na falkowych deskryptorach, które opisują kontury analizowanych znaków na wielu poziomach rozdzielczości jednocześnie. Klasyfikacja realizowana jest poprzez zbiór sieci neuronowych typu MLP (Multi-Layer Perceptron). Proces rozpoznawania składa się z 3 etapów. W

pierwszym wzorcom wejściowym jest wstępnie klasyfikowany ze względu na swój kształt topologiczny. Następnie na każdym poziomie rozdzielczości dokonywana jest niezależna klasyfikacja reprezentacji konturu wejściowego kształtu w przestrzeni wielorozdzielczej. Ostatni etap klasyfikacji wiąże wyniki otrzymane w poprzednim kroku w ostateczny rezultat.

Opisywany eksperyment dokonuje klasyfikacji na trzech poziomach rozdzielczości uzyskanych w wyniku zastosowania transformaty falkowej do obrazu wzorca (rys. 6, na podstawie [38]). Dla obrazów każdej z rozdzielczości dokonywane jest wstępne sortowanie obrazów ze względu na liczbę Eulera (liczba zwartych części obiektu minus liczba dziur w obiekcie). W wyniku takiego podziału uzyskuje się trzy grupy wzorców o liczbie Eulera równej 0 (24 znaki), 1 (9 znaków) oraz 2 (3 znaki). Dla każdej grupy tworzona jest niezależna sieć neuronowa, klasyfikująca wzorce danej kategorii. Sieci te mają liczbę neuronów wejściowych zależną od poziomu rozdzielczości obrazu, tzn. odpowiednio 72, 36 lub 18. Liczba neuronów warstwy wyjściowej zależy od klasyfikowanej grupy wzorców topologicznych, tzn. odpowiednio 24, 9 lub 3 (liczba znaków w grupie). Sieci te mają jedną warstwę ukrytą o liczbie neuronów bliskiej logarytmowi o podstawie 2 liczby wyjść sieci. Jeżeli klasyfikatory odpowiadające wszystkim trzem rozdzielczościom dają zgodny wynik, znak uznaje się za rozpoznany. W przeciwnym przypadku znak traktowany jest jako nierozpoznawalny.



Rys. 6. Schemat wielorozdzielczej metody rozpoznawania znaków
Fig. 6. Multiresolution scheme of character recognition

Skuteczność opisanej metody klasyfikacji badano na zbiorze 6480 wzorców pochodzących od 15 osób. Każda z tych osób napisała 12 wersji 26 wielkich liter alfabetu

łacińskiego oraz 10 cyfr. Wszystkie napisane wzorce zostały zeskanowane i znormalizowane do kwadratu o wymiarze 64 (4096 pikseli). Badanie skuteczności przeprowadzono na dwa sposoby: zbiór testowy zawierał wzorce znaków pisane przez te same osoby, od których pochodziły wzorce wykorzystywane w uczeniu sieci (pierwszy sposób); zbiór testowy zawierał wzorce znaków pisane przez inne osoby niż te, od których pochodziły wzorce wykorzystywane w uczeniu sieci (drugi sposób). Wyniki skuteczności klasyfikacji dla poszczególnych poziomów rozdzielczości zebrano w tabeli 2 (zaczepnięta z [38]).

Tabela 2

Skuteczność klasyfikacji dla poszczególnych poziomów rozdzielczości

Poziom	L	NN0	NN1	NN2	B ³ d	Procent
3	72	5	5	2	13	0,6%
4	36	4	5	1	6	0,28%
5	18	5	2	2	10	0,46%
3	72	5	2	2	38	1,75%
4	36	4	5	1	27	1,25%
5	18	2	5	2	35	1,62%

Trzy górne wiersze tabeli opisują wyniki dla pierwszego sposobu testowania, natomiast trzy dolne – drugiego. Pierwsza kolumna tabeli zawiera poszczególne poziomy rozdzielczości, druga – liczbę cech (neuronów warstwy wejściowej). Kolumny od trzeciej do piątej opisują odpowiednio liczby neuronów warstwy ukrytej dla każdego z przypadków sieci, prowadzące do najlepszego rezultatu procesu rozpoznawania. Kolumna szоста wyraża liczbę błędnych klasyfikacji, natomiast siódma – procent błędnych klasyfikacji.

Uwzględniając całą strukturę klasyfikatora, czyli łącznie wszystkie poziomy rozdzielczości, otrzymano zero błędnych klasyfikacji w przypadku pierwszego sposobu testowania skuteczności klasyfikacji, przy jednoczesnym odrzuceniu 29 wzorców, co stanowi 1,34%. Natomiast w przypadku drugiego sposobu testowania otrzymano 6 błędnych klasyfikacji (0,28%) przy jednoczesnym odrzuceniu 83 wzorców (3,84%). Zatem w pierwszym przypadku procent poprawnych klasyfikacji wynosi 98,66%, a w drugim – 95,88%.

Innym przykładem zastosowania analizy wielorozdzielczej do klasyfikacji znaków jest praca G. Y. Chena, T. D. Bui i A. Krzyzaka [4]. Autorzy wyznaczają kontury znaków i następnie je normalizują tak, aby uzyskać postać niezmienniczą ze względu na przesunięcia i skalowanie. W dalszym etapie dokonywana jest wielofalkowa transformacja (ang. multiwavelet orthonormal shell expansion) na uzyskanym konturze, a współczynniki przekształcenia traktowane jako cechy podawane są na wejścia sieci neuronowej typu MLP (Multi-Layer Perceptron), mającej rozpoznawać ręcznie pisane cyfry. Użycie transformacji wielofalkowej motywowane jest uzyskaniem lepszych rezultatów w przypadku łącznej transformacji w osiach OX i OY niż te, które zostały przedstawione w pracy P. Wunscha i

A. F. Lainego [38], gdzie zastosowano oddzielną ekstrakcję cech we współrzędnych x i y . W przypadku dekompozycji obrazu do poziomu trzeciego (poziom rozdzielczości) uzyskano 92,20% poprawnych klasyfikacji, co stanowi lepszy rezultat w porównaniu do 85,25% odpowiadających temu samemu poziomowi rozdzielczości tradycyjnej transformaty falkowej (przy zastosowaniu metody P. Wunscha i A. F. Lainego dla tej samej bazy wzorców).

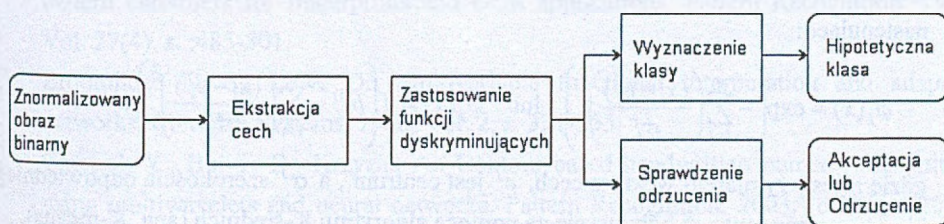
3.3.5. *Podejście statystyczne do klasyfikacji znaków*

Wśród licznych neuronowych klasyfikatorów opartych na wektorach cech wyróżnia się grupa wykorzystująca transformatę Karhunen-Loevego (KLT) do ekstrakcji cech obrazów będących przedmiotem rozpoznawania. Ponieważ transformata ta bazuje na zbiorze wektorów własnych macierzy kowariancji, tworzony jest zatem statystyczny opis zmienności występujących w badanych wzorcach. Dzięki użyciu współczynników transformaty KL jako reprezentacji cech obrazu znaku, zbiór wektorów własnych może być traktowany jako naturalny ekstraktor cech dla klasyfikatora neuronowego [10]. Wykorzystanie transformaty Karhunen-Loevego (nazywanej również transformatą Hotellinga) przez J. P. Grothera przedstawione w pracy [10] służy również do redukcji wymiaru wektora cech podawanego na wejście sieci neuronowej. Odbywa się to poprzez użycie jedynie największych co do wartości bezwzględnej wartości własnych, gdyż one to opisują wariacje przekształconego obrazu, zatem małe wartości odpowiadają niewielkim zmianom i wykazują niewielką użyteczność w opisie różnic między analizowanymi obrazami różnych znaków. Z uwagi na ograniczenie bazy i wykorzystanie jedynie części wektorów własnych tylko te obrazy, które są podobne do obrazów zbioru uczącego, mogą być odpowiednio reprezentowane przez niekompletną bazę. Ważne jest zatem przy stosowaniu tej metody, aby wektory własne wyznaczone były ze zbioru zawierającego odpowiednio wiele wzorców, stanowiących statystycznie reprezentatywną próbę.

J. P. Grother w swojej pracy [10] jako klasyfikator mający rozpoznawać ręcznie pisane cyfry zastosował jednokierunkową sieć typu MLP (Multi-Layer Perceptron) z jedną warstwą ukrytą, obejmującą 32 neurony lub w innym przypadku 48 neurony. Na wejście sieci podawane były współczynniki transformaty Karhunen-Loevego, natomiast warstwa wyjściowa zawierała 10 neuronów odpowiadających rozpoznawanym cyfrom. Uczenie sieci wykorzystywało metodę propagacji wstecznej. Skuteczność klasyfikacji badana była przy wykorzystaniu bazy wzorców NIST (National Institute of Standards and Technology) i obejmowała 102340 znaków uzyskanych od 944 osób. Każdy znak był reprezentowany przez kwadratowy obraz cyfrowy o wymiarze 32 pikseli. Autor w pracy badał wpływ zmiany wielkości wymiaru wektora cech (otrzymanego w wyniku zastosowania transformaty KL), przy ustalonej liczbie neuronów warstwy ukrytej oraz wpływ wielkości zbioru uczącego na skuteczność klasyfikacji. Uzyskał on między innymi następujące wyniki. Dla zbioru

testowego obejmującego 15000 znaków cyfr uzyskanych od 312 osób skuteczność klasyfikacji wyniosła 96,5% dla sieci o 48 neuronach warstwy wejściowej, 48 warstwy ukrytej i 10 warstwy wyjściowej uczonej na 76755 wzorcach. Dla sieci o 32 neuronach warstwy wejściowej, 32 warstwy ukrytej i 10 warstwy wyjściowej uczonej na 7400 wzorcach skuteczność wynosiła 93,7%, lecz gdy zwiększono do 48 liczbę neuronów warstwy ukrytej, skuteczność ta wzrosła do 94,5%. Warto przy tym zaznaczyć, że wzorce używane do testowania skuteczności klasyfikacji nie były używane do wyznaczania macierzy kowariancji i jej zbioru wektorów własnych.

Badania wykorzystujące transformatę Karhunen-Loevego jako ekstraktor cech w celu klasyfikacji obrazów cyfr były kontynuowane przez P. J. Grothera i innych, a ich wyniki przedstawione zostały w pracach [11] i [2]. Autorzy przedstawiają porównanie skuteczności różnego typu klasyfikatorów w zależności od wielkości wymiaru wektora cech (otrzymanego w wyniku zastosowania transformaty KL). Opisywane są zarówno klasyfikatory statystyczne szeroko znane w tradycyjnej literaturze dotyczącej rozpoznawania obrazów, takie jak między innymi klasyfikatory minimalno-odległościowe, jak również klasyfikatory neuronowe. System rozpoznawania zbudowany na potrzeby eksperymentów składał się z kilku modułów (rys. 7, na podstawie [11]). Znormalizowane obrazy kwadratowe o wymiarze 32 pikseli zostawały przekształcane za pomocą niepełnej transformaty Karhunen-Loevego. Dzięki temu następowała redukcja wymiaru wektorów cech podawanych na wejścia funkcji dyskryminujących. Następnie wartości tych funkcji podawane zostawały równoległe do modułu wyznaczającego hipotetyczną klasę i modułu stwierdzającego, czy wyznaczona klasa jest wiarygodna, czy też system nie potrafi właściwie jej zaklasyfikować. Wyznaczanie klasy obejmowało analizę wartości funkcji dyskryminujących odpowiadających podziałowi zbioru wektorów cech na rozłączne klastry, których liczba była nie mniejsza niż liczba rozpoznawanych klas (liczba ta była różna dla różnych klasyfikatorów wykorzystywanych w eksperymencie). Rozpoznana klasa reprezentowana była poprzez klaster, dla którego funkcja dyskryminująca osiągała największą wartość.



Rys. 7. Składniki systemu klasyfikującego

Fig. 7. Component of classification system

Stosowane w eksperymentach klasyfikatory to:

1. Klasyfikator EMD (Euclidean Minimum Distance), dla którego funkcje dyskryminujące odpowiadają zanegowanym kwadratam odległości euklidesowych wektora cech od wektora średnich odpowiednio dla każdego klastra.
2. Klasyfikator QMD (Quadratic Minimum Distance), dla którego funkcje dyskryminujące odpowiadają zanegowanym kwadratam odległości Mahalanobisa wektora cech od wektora średnich odpowiednio dla każdego klastra.
3. Klasyfikator NRML (Normal), parametryczny klasyfikator bayesowski zakładający, że wektory cech dla każdego klastra mają wielowymiarowy rozkład normalny.
4. Klasyfikator k-NN (k-Nearest Neighbor), klasyfikator k najbliższych sąsiadów.
5. Klasyfikator WSNN (Weighted Several Nearest Neighbors), zmodyfikowany klasyfikator k najbliższych sąsiadów.
6. Klasyfikator MLP (Multi-Layer Perceptron), klasyfikator neuronowy wykorzystujący sieć typu MLP, zawierającą jedną warstwę ukrytą; funkcja dyskryminująca dla klastra o indeksie i opisana jest wzorem:

$$D_i(x) = f\left(b_i^{(2)} + \sum_{j=1}^{N^{(2)}} w_{ij}^{(2)} f\left(b_j^{(1)} + \sum_{k=1}^{N^{(1)}} w_{jk}^{(1)} x_k\right)\right),$$

gdzie $f(x) = (1 - e^{-x})^{-1}$ jest sigmoidalną funkcją, b i w są wagami odpowiednich warstw sieci, zaś $N^{(1)}$ i $N^{(2)}$ liczbami neuronów odpowiednio warstwy ukrytej i wyjściowej.

7. Klasyfikatory RBF1 i RBF2 (Radial Basis Function), klasyfikatory neuronowe wykorzystujące sieć o radialnych funkcjach bazowych; funkcja dyskryminująca dla klastra o indeksie i opisana jest wzorem:

$$D_i(x) = f\left(b_i^{(2)} + \sum_{j=1}^{N^{(2)}} w_{ij}^{(2)} \varphi_j(x)\right),$$

gdzie $f(x) = (1 - e^{-x})^{-1}$ jest sigmoidalną funkcją, b i w są wagami warstwy wyjściowej sieci, $N^{(2)}$ liczbą neuronów warstwy wyjściowej, natomiast funkcje radialne określone są następująco:

$$\varphi_j(x) = \exp\left[-\sum_{k=1}^n \left(\frac{(x_k - c_k^j)}{\sigma_k^j}\right)^2\right] \quad \text{lub} \quad \varphi_j(x) = f\left(b_j^{(1)} - \sum_{k=1}^n \left(\frac{(x_k - c_k^j)}{\sigma_k^j}\right)^2\right),$$

gdzie n jest wymiarem wektora cech, c^j jest centrum, a σ^j szerokością odpowiedniego klastra; wybór centrum odbywa się za pomocą algorytmu K-średnich (ang. K-means).

8. Klasyfikator PNN (Probabilistic Neural Network), klasyfikator neuronowy wykorzystujący sieć probabilistyczną; funkcja dyskryminująca dla klasy o indeksie i opisana jest wzorem:

$$D_i(x) = \frac{\hat{p}_i}{M'} \sum_{j=1}^{M'} \exp\left(-\frac{1}{2\sigma^2}(x-x_j^{(i)})^T(x-x_j^{(i)})\right),$$

gdzie \hat{p}_i jest estymowanym prawdopodobieństwem wystąpienia klastra i , M' jest liczbą wektorów cech zbioru uczącego dla klastra i , a σ jest „parametrem wygładzającym”.

Skuteczność klasyfikacji badana była przy wykorzystaniu bazy wzorców NIST (National Institute of Standards and Technology) i obejmowała 7480 wzorców znaków cyfr pisanych przez 250 osób dla potrzeb zbioru uczącego oraz 23140 wzorców znaków cyfr pisanych przez innych 250 osób. Najlepsze wyniki zostały uzyskane dla sieci probabilistycznej (klasyfikator PNN), dla parametru σ równego 3, przy wektorze cech o wymiarze 40 (błąd klasyfikacji wyniósł 2,5%) i taki sam wynik uzyskano dla zmodyfikowanego klasyfikatora WSNN, a podobny dla klasyfikatorów k -NN (dla k równego 1 i 3 otrzymano błąd klasyfikacji wynoszący 2,6% przy 40-wymiarowym wektorze cech). Dla klasyfikatora MLP najlepszy wynik to 4,3% błąd klasyfikacji dla 64 neuronów warstwy ukrytej i 52-wymiarowego wektora cech. Najlepszym wynikiem klasyfikatora RBF1 jest 4,2% błąd klasyfikacji dla 48-wymiarowego wektora cech przy 6 centrach dla każdej z klas, natomiast dla RBF2 – 3,9% błąd klasyfikacji dla 44-wymiarowego wektora cech przy 6 centrach dla każdej z klas.

Dalsze modyfikacje podejścia statystycznego opartego na transformacie Karhunen-Loevego dla przedstawionych klasyfikatorów neuronowych i analizę uzyskanych wyników można znaleźć w pracy C. L. Wilsona, P. J. Grothera i C. S. Barnesesa [37].

LITERATURA

1. Amin A., Singh S.: Optical character recognition: Neural network analysis of hand-printed characters. Proceedings of SSPR'98, Lecture Notes in Computer Science-Springer-Verlag, s. 492-499.
2. Blue L. J., Candela G. T., Grother P. J., Chellappa R., Wilson C. L.: Evaluation of pattern classifiers for fingerprints and OCR applications. Pattern Recognition, 1994, Vol. 27(4), s. :485-501.
3. Broomhead D. S., Lowe D.: Multivariable functional interpolation and adaptive networks. Complex Systems, 1988, Vol. 2, s. 321-355.
4. Chen G. Y., Bui T. D., Krzyzak A.: Contour-based handwritten numeral recognition using multiwavelets and neural networks. Pattern Recognition, 2003, Vol. 36, s. 1597-1604.

5. De Ridder D., Duin R. P. W., Verbeek P. W. et al.: The applicability of neural networks to non-linear image processing. *Pattern Analysis & Applications*, 1999, Vol. 2(2), s. 111-128.
6. Dołęga C. G.: Pecet uczy się czytać. *Chip*, 1998, No 11, s. 228-233.
7. Egmont-Petersen M., De Ridder D., Handels H.: Image processing with neural networks – a review. *Pattern Recognition*, 2002, Vol. 35(10), s. 2279-2301.
8. Ghosh A., Pal S. K.: Neural network, self-organization and object extraction. *Pattern Recognition Letters*, 1992, Vol. 13(5), s. 387-397.
9. Greenhil D., Davies E. R.: Relative effectiveness of neural networks for image noise suppression. *Proceedings of the Pattern Recognition in Practice IV*, Vlieland, 1994, s. 367-378.
10. Grother J. P.: Karhunen Loeve Feature Extraction For Neural Handwritten Character Recognition. *Applications of Artificial Neural Networks III*, 1992, Vol. 1709, s. 155-166.
11. Grother P. J., Candela G. T.: Comparison of Handprinted Digit Classifiers. Technical Report NISTIR 5209, NIST, 1993.
12. Hebb D.: *The Organization of Behavior*. Wiley, New York 1949.
13. Hopfield J. J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Science USA*, 1982, Vol. 81, s. 2554-2558.
14. Hu J., Yan H.: Structural primitive extraction and coding for handwritten numeral recognition. *Pattern Recognition*, 1998, Vol. 31(5), s. 493-509.
15. Kohonen T.: Correlation matrix memories. *IEEE Transactions on Computers*, 1972, Vol. 21, s. 353-359.
16. Kohonen T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982, Vol. 43, s. 59-69.
17. Le D. X., Thomas G. R., Wechsler H.: Classification of binary document images into textual or nontextual data blocks using neural network models. *Machine Vision & Applications*, 1995, Vol. 25(7), s. 289-304.
18. LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D.: Backpropagation applied to hand written zip code recognition. *Neural Computation* 1989, Vol. 1(4), s. 541-551.
19. LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D.: Handwritten Digit Recognition With a Back-propagation Network. *Neural Information Processing Systems* 1989, Vol. 2 (pod red. D.S. Touretzky), s. 396-404.

20. Lu Y.: Machine printed character segmentation – an overview. *Pattern Recognition*, 1995, Vol. 28(1), s. 67-80.
21. Lu Y., Shridhar M.: Character segmentation in handwritten words – an overview. *Pattern Recognition*, 1996, Vol. 29(1), s. 77-96.
22. Martin G. L., Pitman J. A.: Recognizing hand-printed letters and digits. *Neural Information Processing Systems 1989*, Vol. 2 (pod red. D.S. Touretzky), s. 405-414.
23. Martin G. L., Pitman J. A.: Recognizing hand-printed letters and digits using backpropagation learning. *Neural Computation*, 1991, Vol. 3, s. 258-267.
24. McCulloch W. C., Pitts W.: A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943, Vol. 5, s. 115-133.
25. Rosenblatt, F.: *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. *Psychological Review*, 1958, Vol. 65, s. 386-408
26. Rosenblatt F.: *Principles of Neurodynamics*. Spartan Books, New York 1962.
27. Rout S., Srivastava S. P., Majumdar J.: Multi-modal image segmentation using a modified Hopfield neural network. *Pattern Recognition*, 1998, Vol. 31(6), s. 743-750.
28. Rumelhart D. E., Hinton G. E., McClelland J. L.: Learning internal representations by error propagation. *Parallel Distributed Processing Vol. 1* (pod red. Rumelhart i McClelland), Cambridge 1986, MIT Press, s. 318-362
29. Russo F.: Hybrid neuro-fuzzy filter for impulse noise removal. *Pattern Recognition*, 1999, Vol. 32(11), s. 1843-1855.
30. Singh S., Amin A.: Neural Network Recognition of Hand-printed Characters. *Neural Computing and Application*, 1999, Vol. 8, s. 67-76.
31. Specht J.: Probabilistic neural network. *Neural Network*, 1990, Vol. 3, s. 45-74.
32. Srihari S. N., Lam S. W.: Character recognition. Technical Report, 1995, CEDAR-TR-95-1.
33. Stone M.: Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of Royal Statistical Society*, 1974, V. 36(1), s. 111-147.
34. Tadeusiewicz R.: *Sieci Neuronowe*. Kraków 1992.
35. Trier Q. D., Jain A. K., Taxt T.: Feature extraction methods for character recognition – a survey. *Pattern Recognition*, 1996, Vol. 29(4), s. 641-662.
36. Widrow B., Lehr M. A.: 30 Years of adaptive neural networks: perceptron, Madaline, and backpropagation, *Proc. IEEE* 1990, Vol. 78(9), 1415.
37. Wilson C. L., Grother P. J., Barnes C. S.: Binary decision clustering for neural-network-based optical character recognition. *Pattern Recognition*, 1996, Vol. 29(3), s. 425-437.

38. Wunsch P., Laine A. F.: Wavelet descriptors for multiresolution recognition of handprinted characters. *Pattern Recognition*, 1995, Vol. 28(8), s. 1237-1249.
39. Zhnag Z. Z., Ansari N.: Structure and properties of generalized adaptive neural filters for signal enhancement. *IEEE Trans. Neural Networks*, 1996, Vol. 7(4), s. 857-868.

Recenzent: Prof. dr hab.inż. Leszek Rutkowski

Wpłynęło do Redakcji 1 października 2003 r.

Abstract

The article presents the short review of the domain of character recognition since its beginning in 1951 when M. Sheppard invented GISMO – A Robot Reader-Writer. Early standards such as OCR-A and OCR-B, which were developed to help guide automatic document processing, were mentioned. The paper also describes the basic ideas behind artificial neural networks and quotes the main types of neural networks.

The paper contains examples of applications of neural networks to a process of character recognition with special regard to stages of the feature extraction and the classification. Two examples were shown, in which the feature extraction and the classification are joined in one common stage – the works of Y. LeCun et al which were regarded to a hand-written zip code recognition as well as the works of G. L. Martin and J. A. Pittman which were regarded to samples of hand-printed digits scanned off of bank checks. Three groups of examples were also shown which represent different approaches to the extraction of features, which were used as the input data for neural networks. These approaches were: structural, multiresolution analysis based and statistical ones.

Adres

Alina MOMOT: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-101 Gliwice, Polska, alamomot@polsl.gliwice.pl .